Editor: Dana Simian

Proceedings of The International Conference on Applied Informatics

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

May 11 - 13, 2023, Sibiu, Romania

Editor DANA SIMIAN

IMAGINATION, CREATIVITY, DESIGN, DEVELOPMENT

Proceedings of the International Conference on Applied Informatics
ICDD

May 11th – 13th, 2023
Sibiu, Romania

Lucian Blaga University of Sibiu

Lucian Blaga University of Sibiu, 2023

Editor Dana Simian

All papers in this volume were peer review by two independent reviewers

ISSN-L 2069 – 864X

Associated Editor Laura Florentina Stoica

Proceedings of the International Conference on Applied Informatics, ICDD
May 11th – 13th, 2023, Sibiu, Romania

Preface

This volume contains refereed papers presented within the International Conference on Applied Informatics, "Imagination, Creativity, Design, Development" - ICDD 2023, which was held between May 11th – 13th, at the Faculty of Sciences, Lucian Blaga University of Sibiu, Romania.

The conference is mainly addressed to young researchers from all over the world. The conference gives the participants the opportunity to discuss and present their research on informatics and related fields (like computational algebra, numerical calculus, bioinformatics, etc.). The conference welcomes submissions of original papers on all aspects of informatics and related fields ranging from new concepts and theoretical developments to advanced technologies and innovative applications. Specific topics of the conference included but are not restricted to: Algorithms and Data Structures, Graph Theory and Applications, Formal Languages and Compilers, Cryptography, Modelling and Simulation, Computer Programming, Computer Vision, Computer Graphics, Game Design, Data Mining, Distributed Computing, Artificial Intelligence, Service Oriented Applications, Networking, Grid Computing, Mobile Operating Systems, Scientific Computing, Software Engineering, Bioinformatics, Robotics, Computer Architecture, Evolutionary Computing, Multimedia Systems, Internet Communication and Technologies, Web Applications.

The conference has brought together participants from 4 countries (Bulgaria, Germany, Romania, Turcia).

We thank all the participants for their interesting talks and discussions. We also thank the members of the scientific committee for their help in reviewing the submitted papers and for their contributions to the scientific success of the conference.

November 2023

Dana Simian
Conference Chair

Motto:

“There are no limits, only your imagination”

Scientific committee

Kiril Alexiev - Bulgarian Academy of Sciences, Bulgaria
Vsevolod Arnaut - Moldova State University, Republic of Moldova
Galina Atanasova - Angel Kanchev University of Rousse
Alina Barbulescu - Ovidius University of Constanta, Romania
Arndt Balzer - University of Applied Sciences, Würzburg-Schweinfurt, Germany
Lasse Berntzen - University of South-Eastern Norway
Peter Braun - University of Applied Sciences, Würzburg-Schweinfurt, Germany
Amelia Bucur - Lucian Blaga University of Sibiu, Romania
Stelian Ciurea - Lucian Blaga University of Sibiu, Romania
Nicolae Constantinescu - Lucian Blaga University of Sibiu, Romania
Daniela Danciulescu - University of Craiova, Romania
Lyubomyr Demkiv - Lviv National Polytechnic University and Robotics Lead at SoftServe, Ukraine
Oleksandr Dorokhov - Kharkiv National University of Economics, Ukraine
Dmytro Dosyn - Institute of Computer Science and Information Technologies, Lviv Polytechnic National University
George Eleftherakis - The University of Sheffield International Faculty, City College Thessaloniki, Greece
Michael Emmerich - Leiden Institute of Advanced Computer Science, Leiden University
Calin Enachescu -University of Medicine, Pharmacy, Science and Technology of Targu Mures, Romania
Ralf Fabian - Lucian Blaga University of Sibiu, Romania
Tobias Fertig - University of Applied Sciences Würzburg-Schweinfurt
Stefka Fidanova - Bulgarian Academy of Sciences, Bulgaria
Ulrich Fiedler - Bern University of Applied Science, Switzerland
Adrian Florea - Lucian Blaga University of Sibiu, Romania
Teresa Gonçalves - University of Evora, Portugal
Andrina Granić - University of Split, Croatia
Katalina Grigorova - University of Ruse, Bulgaria
Piroska Haller - University of Medicine, Pharmacy, Science and Technology of Targu Mures, Romania
Daniel Hunyadi - Lucian Blaga University of Sibiu, Romania
Saleema JS - Chris University, Bangalore, India
Milena Lazarova - Technical University of Sofia, Bulgaria
Lixin Liang - Tsinghua University, Beijing, China
Suzana Loskovska - "Ss. Cyril and Methodius" University in Skopje, Republic of Macedonia

Contents

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Multi Source Music Player

Bozhidar Atanasov, Kameliya Shoylekova

Abstract

The paper discusses the creation of a system that creates playlists of music from multiple different sources eliminating the inconvenience of not having a certain song in a playlist and allowing for the easy sharing of said playlists

1 Introduction

At the time of writing, several sophisticated algorithms have been developed with the capacity to predict and provide songs tailored to the listener's taste. However, these algorithms exhibit a range of disparities and present multiple drawbacks. These include, but are not limited to:

- Bias Towards Prominent Tracks: some algorithms are primarily inclined towards the availability of renowned tracks, thereby impeding the upload and dissemination of music from emerging artists.
- Focus on Indie Artists: conversely, certain platforms chiefly concentrate on independent or "indie" artists, with limited emphasis placed on widely recognized and popular musicians.
- Compromised Sound Quality: a discrepancy in audio quality is observable across platforms, with some services offering inferior sound quality of identical tracks compared to their counterparts.
- Subpar Functionalities: certain platforms exhibit a dearth of features, compromising the overall user experience.
- Limited Sharing Capabilities: some services either provide restricted features for sharing playlists or lack such functionalities altogether.

Presently, there seems to be an absence of systems that permit users to amalgamate music providers to formulate their optimum music listening experience. The proposed system endeavours to address this gap in the interim, with aspirations to incorporate numerous essential and quality of life features in the long term.

The Multi-Source Music Player stands out due to its unique combination of multi-source accessibility, user-centric design, customizable playlists, open-source development, inclusive user authentication, privacy controls, and a clear roadmap for future enhancements, positioning it as a versatile and adaptable solution in the digital music landscape.

Feature	Multi-Source Music Player	Spotify	YouTube Music	SoundCloud
Multi-Source Accessibility	Yes	No	No	No
Customizable Playlists	Yes	Yes	Yes	Yes
Open-Source Development	Yes	No	No	No
Inclusive User Authentication	Yes	Yes	Yes	Yes
User-Centric Design	Yes	Yes	Yes	Yes
Privacy and Control over Playlists	Yes	Yes	Yes	Yes
Direct Link Song Addition	Yes	No	No	No
Supported Services	YouTube, SoundCloud, etc.	-	-	-
Future Expansion Plans	Yes	-	-	-

2 Exhibition

The Multi-Source Music Player can be accessed via the following web address: music-m-player.vercel.app. Upon accessing the link, users are presented with an initial page. Given the system's requirement for account ownership to maintain playlists, users are mandated to either sign up or log in if they have previously engaged with the service (Fig. 1).

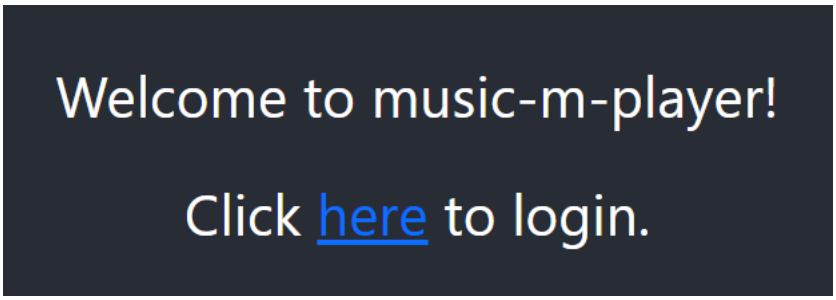


Fig. 1: Initial page of the web application

After link selection, the succeeding page is displayed, indicating various login methods facilitated by Firebase, including authentication via email and password, or through a Google account. Utilizing a Google account for this purpose simultaneously creates an account with the service (Fig. 2).

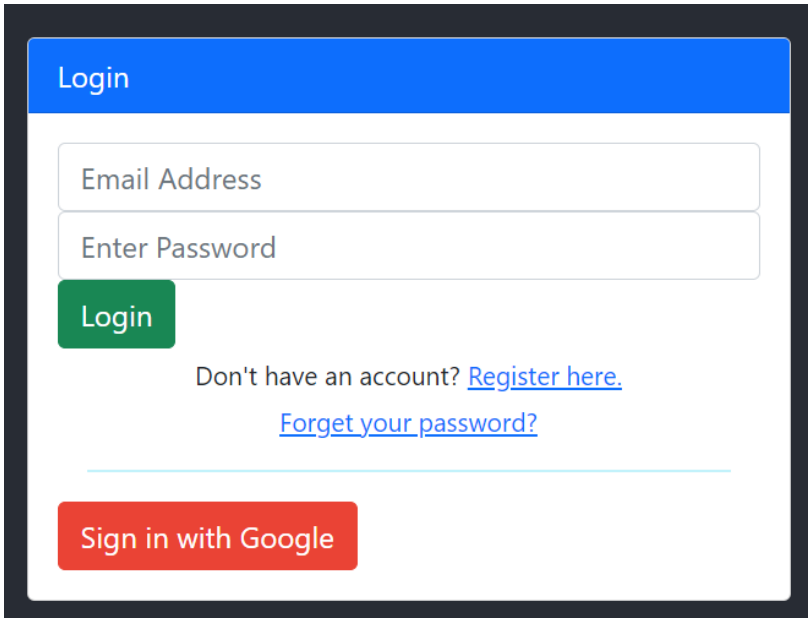


Fig. 2: Page showcasing user login options

For first-time users opting against Google account utilization, the registration page is available, enabling sign-up via email and password (Fig. 3).

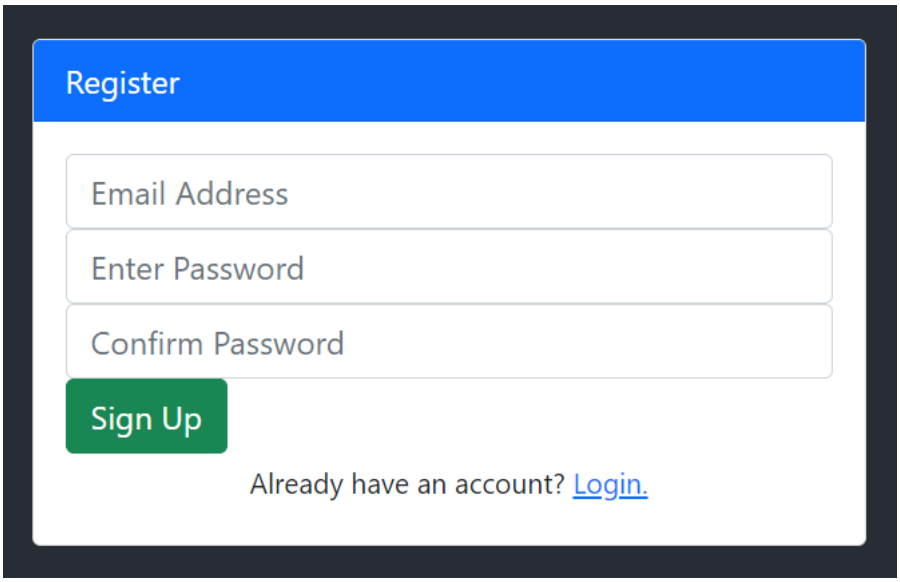


Fig. 3: Page facilitating user sign-up

Subsequent to account creation, users are redirected to the homepage, granting access to the service (Fig. 4).

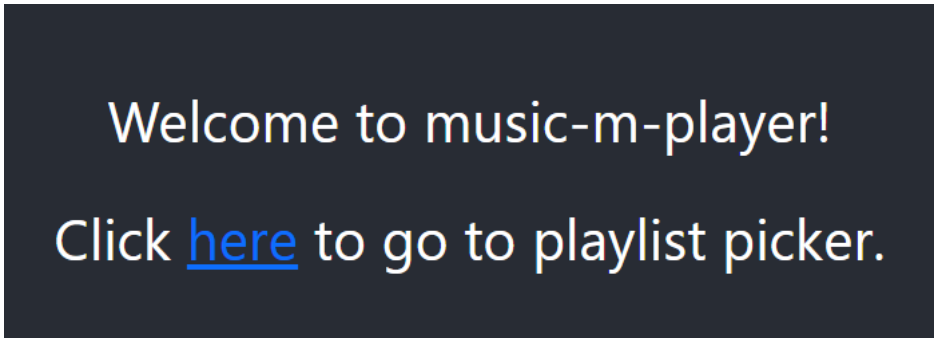


Fig. 4: Homepage post-user login

Navigating further leads to the playlist picker, initially empty but can be populated via the "Create Playlist" feature. Profile options are located in the top right dropdown, labeled "Profile" (Fig. 5).

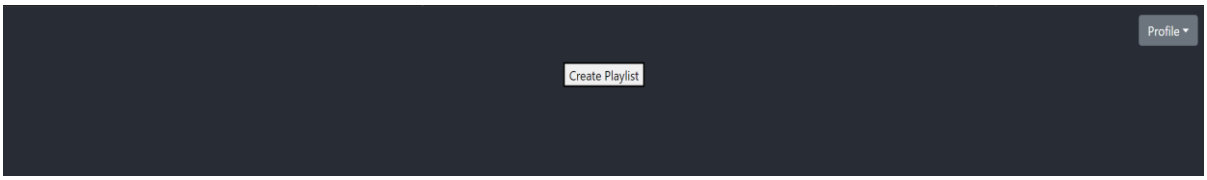


Fig. 5: Empty playlist picker page

Figure 6 illustrates a user in the process of playlist creation, showcasing the options available for each playlist. These include viewing, deleting, and toggling visibility, with editing privileges reserved exclusively for the logged-in creator (Fig. 6).

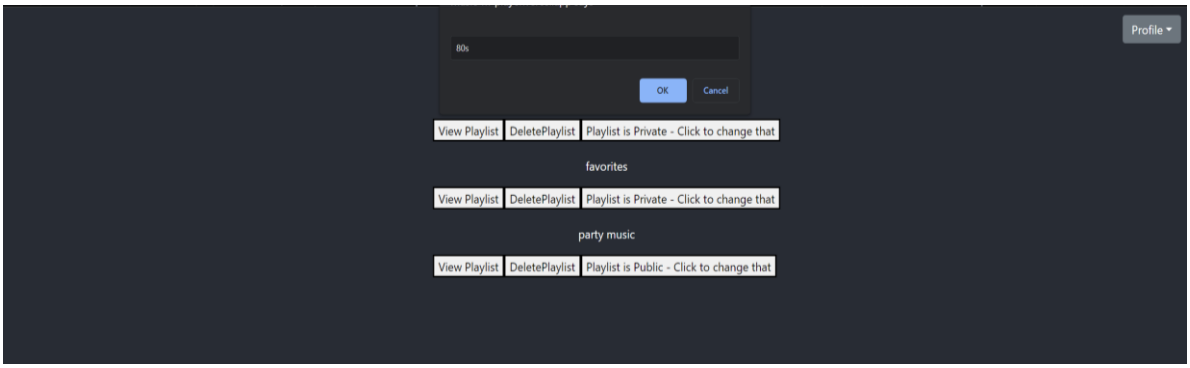


Fig. 6: Playlist page depicting various options and a user creating a new playlist

Figure 7 represents an initially empty playlist screen, with the ability to add new songs via a designated button (Fig. 7).

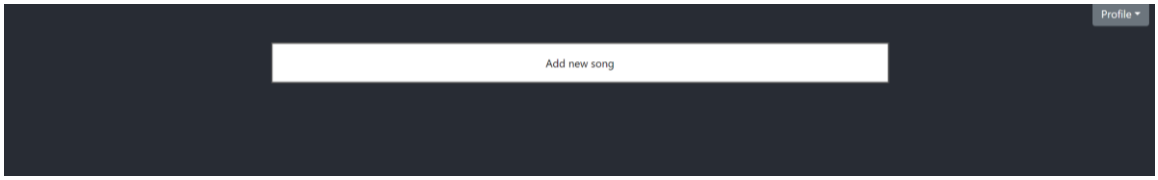


Fig. 7: Page illustrating an empty playlist

Adding new songs necessitates user-provided direct song links, as the system lacks affiliations with music labels or services and consequently, a search function. Upon URL input, the system attempts to auto-retrieve the song title, with the provision for user modification prior to addition to the playlist (Fig. 8, Fig. 9).

Fig. 8: Page depicting the process of adding a new URL

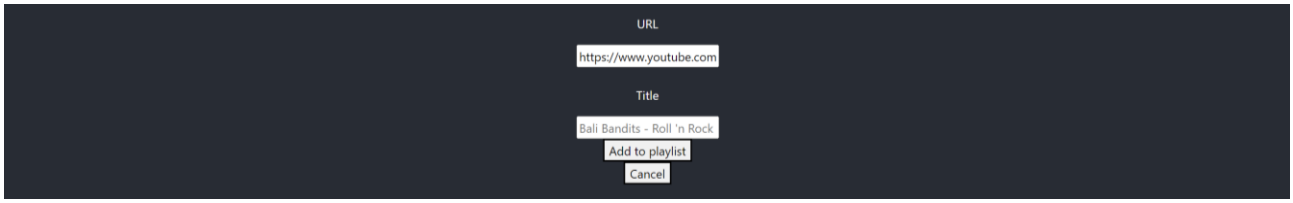


Fig. 9: Page displaying URL addition post-title retrieval

Once a song is added, it becomes accessible within the playlist for playback, with options for updating or deleting (Fig. 10).

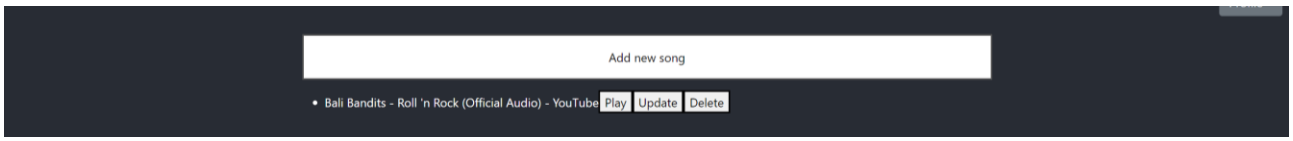


Fig. 10: Playlist page post-song addition

Playback occurs on the corresponding page, featuring functionality such as seeking, volume control, song navigation, and internal player display for the current service provider (Fig. 11).

Fig. 11: Playlist page showcasing the player during song playback

The system facilitates an intuitive means of playlist organization through a drag-and-drop mechanism, allowing users to reorder playlists as desired (Fig. 12).

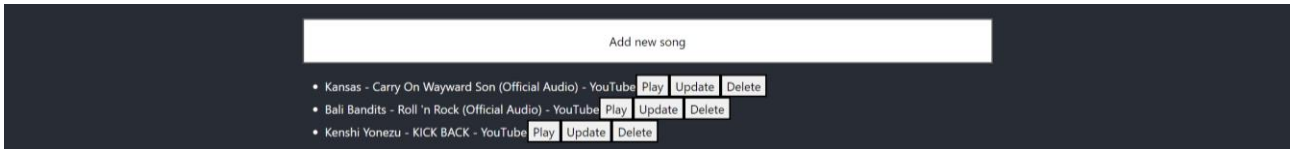


Fig. 12: Playlist page post-user rearrangement

Finally, this is the architecture of the application (Fig. 13).

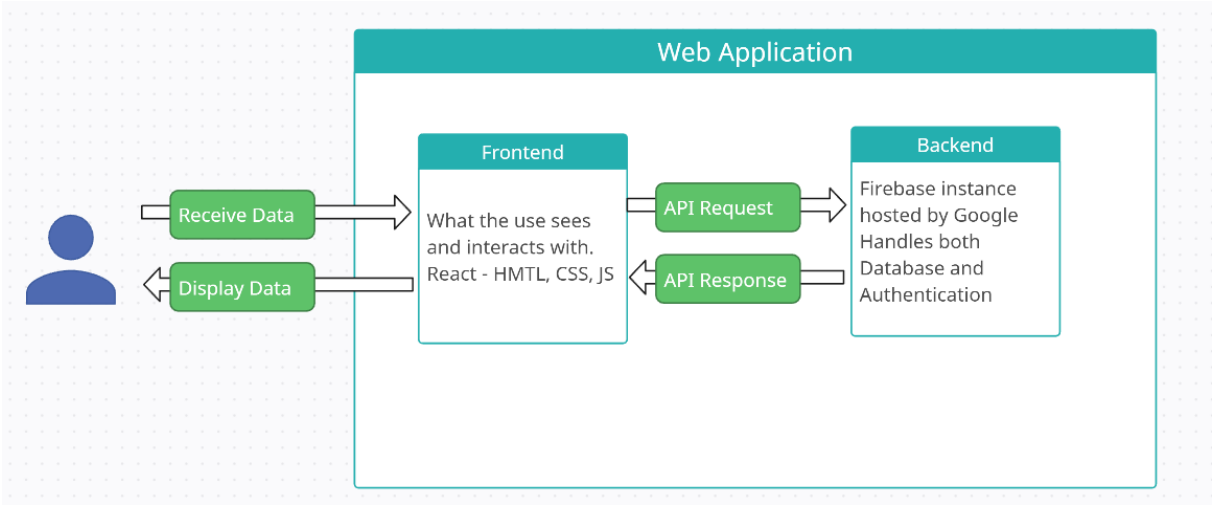


Fig. 13: Architecture of the application

Technologies employed: node, npm, react, react-player, typescript, firebase, and vercel.

3 Closing Remarks

This project was initiated in response to a perceived gap in the availability of platforms that facilitate the easy sharing and creation of playlists from multiple sources. It is anticipated that other individuals will find this project beneficial and will contribute to its growth on GitHub.

Future objectives for this project include addressing the absence of features commonly found in average music players, such as shuffle play. Currently, the supported services from which users can source songs include YouTube and Soundcloud, with major services like Spotify not yet incorporated. Remedying this, along with addressing stability concerns and enhancing additional features, form the pivotal focus for the project's future development.

References

[1] *Conditional Rendering*, https://reactjs.org/docs/conditional-rendering.html
[2] *ReactPlayer*, https://www.npmjs.com/package/react-player
[3] *Firebase*, https://firebase.google.com/docs/firestore/query-data/get-data
[4] *Gnuns*, https://github.com/gnuns/allorigins
[5] *Notorious Preying Mantis*, https://www.npmjs.com/package/react-movable
[6] *Remix-run*, https://github.com/remix-run/react-router
[7] *Stackoverflow - How to create a protected route with react-router-dom?*, https://stackoverflow.com/questions/66289122/how-to-create-a-protected-route-with-react-router-dom
[8] *Stackoverflow,* https://stackoverflow.com

Bozhidar ATANASOV
"Angel Kanchev" Univesity of Ruse
Natural Sciences and Education
ul. "Studentska" 8, POB 7017,
Ruse
BULGARIA
E-mail: s216271@stud.uni-ruse.bg

Kameliya SHOYLEKOVA
"Angel Kanchev" Univesity of Ruse
Natural Sciences and Education
ul. "Studentska" 8, POB 7017,
Ruse
BULGARIA
E-mail: kshoylekova@uni-ruse.bg

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

The Bezt Shaorma: A Fun and Engaging Approach to Learning About Physical Forces

Ștefănel-Alexandru Banu, Sorin-Ionuț Conea

Abstract

"The Bezt Shaorma" is a novel mobile game developed using Unity and programmed with C#, designed to improve players' reaction times, and promote a deeper understanding of physical forces. This initiative presents the development, implementation, and assessment of "The Bezt Shaorma" , which was published on the Google Play Store. The game's core objective involves players gathering raw food ingredients to create a virtual shawarma. To achieve this, players must navigate through an interactive environment that challenges their reaction time and problem-solving skills, while concurrently exposing them to the principles of classical mechanics. The game incorporates various tasks that require players to manipulate different physical forces, including gravity, friction, and tension, to achieve the desired outcomes. "The Bezt Shaorma" aims to provide an engaging and educational experience for players, fostering a deeper comprehension of physics concepts, improving their competitive thinking and enhancing their cognitive abilities. To evaluate the efficacy of the game in achieving these goals, a pre- and post-test experimental design was conducted, wherein participants' reaction times and understanding of physical forces were measured. The results indicate that "The Bezt Shaorma" effectively promotes improvements in reaction time and understanding of physical forces among players, demonstrating its potential as an innovative educational tool especially to develop children's motrical function.

1 Introduction

Video games are the future teaching tool of the people as it can currently enhance reaction time, understanding of physics, problem-solving skills, developing hand-eye coordination, faster and more accurate decision-making and much more.[4] Nowadays video games have evolved to offer players sensorial experiences beyond sight and sound.
In [5], the authors emphasize the significance of considering situational factors in understanding human behavior. This perspective has proven useful for recognizing the advantages of video games, as they can offer valuable experiences and promote positive outcomes when engaged in an appropriate context. Some mobile games have also resorted to similar features to improve the gaming experience. This indicates a move towards a more immersive and engaging future of gaming experiences.

When we engage in gaming our focus is not on the physical equipment used. The equipment creates a tactile element to the experience by establishing a physical link between the player and the game. Our concern for gamepads decreases as we become more immersed in the game, and our connection to the overall gaming experience grows deeper. This is the reason behind our capability of understanding real-world physics without even knowing it. The subconscious learns how the

movement of the in-game character is, having as proof that every individual that plays a game for more than 1 week will know how to move in that specific game without even looking at what he does with his hands. The same goes for the physics and forces inside a game.

Research has shown that certain types of video games can improve hand-eye coordination, fine motor skills, reaction time, and other physical abilities. For example, action video games that require quick reflexes and precise movements have been found to enhance visual attention, spatial skills, and cognitive flexibility, which are important components of motor development.

Moreover, studies have found that playing video games can promote physical activity and help children develop a positive attitude towards exercise, which can have long-term health benefits. However, it's important to note that not all video games are created equal in terms of their potential to promote physical development, and that parents should monitor their children's screen time and encourage a balanced lifestyle that includes physical activity and other healthy behaviors.

In[6] Green CS, Bavelier D. found that playing action video games improved visual selective attention, which is the ability to focus on important visual information while ignoring distractions. This skill is important for many activities that require motor skills, such as driving or playing sports. Another study[7] used brain imaging to show that playing video games that require complex visuomotor transformations (such as first-person shooter games) can lead to changes in the brain's cortical networks. These changes can improve the ability to perform complex motor tasks.

In [8] authors found that playing action video games can improve the speed of processing visual information, which can lead to faster reaction times and better motor performance.

This article provides[9] an overview of research on the effects of video game play, including evidence that playing video games can improve hand-eye coordination, reaction time, and spatial skills.

2 Physical forces and movement in Unity

Physical forces and movement in Unity are a major factor of both 2D and 3D games. From old 2D games such as Mario, Pong and Pacman to newer 3D games such as Grand Theft Auto V, Cyberpunk 2077 and Marvel’s Spider-Man Remastered, the forces and movements always cojoined into making development of games possible.

2.1 Gravity system in Unity

Gravitational forces in unity are applied directly from the editor, without the need of any code. We can see and change the gravity of both 2D and 3D environments by going to Edit > Project Settings and clicking on “Physics” for 3D or “Physics 2D” for the 2D environment.

In Unity, gravity is a physics setting that can be applied to a Rigidbody component. Rigidbody is a component that allows a GameObject to be affected by physics in the game world. When a Rigidbody is affected by gravity, it is pulled downwards until it hits a collider. The direction and strength of this pull are defined by the gravity setting.

The gravity setting can be applied to the entire game world or specific Rigidbodies. The default gravity setting for a new project is -9.81 m/s^2 on the Y-axis, which is equivalent to Earth's gravity. However, this can be modified by changing the gravity setting in the Physics Settings or by adjusting the Rigidbody component's gravityScale property to a different value.

While gravity has a preset of -9.81, it can also be modified thought code lines such as 2 power-ups in the game can do. “Reverse Gravity” reverses the gravity of all rigidbodies in the game by multiplying it with -1 and “No Gravity” sets the gravity to 0, just like space, having only the exact movement that the in-game arrow shows.

2.2 RigidBody

The most used component from Unity that applies the Physics from the "Project Settings" is the RigidBody or RigidBody2D in case of a 2D game. Adding a RigidBody component to an object will put its motion under the control of Unity's physics engine. Even without adding any code, a RigidBody object will be pulled downward by gravity and will react to collisions with incoming objects if the right Collider component is also present. In "The Bezt Shaorma" the RigidBody2D is applied on the shawarma and colliders are applied to each collectible that spawns. Because of this, players can get the collectibles from the game by touching their colliders with the shawarma.

```
public void OnTriggerEnter2D(Collider2D col)
{
      if(col.tag == "Collectibles")
      {
            col.GetComponent<Rigidbody2D>().Sleep();
            Collectible = col.name.Substring(0, col.name.Length - 13);
            //Subtract the collectible name from the instantiated prefab ("Fire_Spawn[CLONE]")
            col.enabled = false;
      }
…
}
```

2.3 Player Movement

Player movement is one of the most complex component of a game because it can be written in such many ways. A developer cannot learn a specific player movement script since each game has its own unique way of approaching. Each individual game developer must choose what works best with the game he has in mind, by reason of the components applied to the player character. Ultimately, the key to implementing effective player movement is to experiment with different approaches and test them in the context of the game being developed. By considering the unique needs of the game and the available tools and components, a developer can create a player movement system that is intuitive, responsive, and enjoyable to play.[10]

As for the presented game, the PlayerMovement consists of 3 important parts. "Input.GetMouseButtonDown(0)" which is the moment that the player touches the screen, "Input.GetMouseButton(0)" which stands for when the player still has the finger on the screen and "Input.GetMouseButtonUp(0)" which is the equivalent to the moment of launching the shawarma, lifting the finger off the screen. The main component that is always present inside the 3 important parts of the script is RigidBody2D. Each part manipulates this component in order to pause the movement, set a direction and a force and finally launch the shawarma through commands such as RigidBody2D.pause, RigidBody2D.simulated=true and RigidBody2D.AddForce(force, mode).

```
if(Input.GetMouseButtonUp(0) && results_bool == false) // launch phase
        {
           endPoint = camera.ScreenToWorldPoint(Input.mousePosition);
           endPoint.z = 15;
           rb.velocity = new Vector2(0, 0);
           rb.simulated = true;
              force = new Vector2(Mathf.Clamp(startPoint.x - endPoint.x, minPower.x, maxPower.x),
              Mathf.Clamp(startPoint.y - endPoint.y, minPower.y, maxPower.y));
           rb.AddForce(force * power, ForceMode2D.Impulse);
           trajectory.lr.positionCount = 0;
        }
```

3 About the game

The mechanics of "The Bezt Shaorma" are what makes the game unique and engaging. It presents a challenge to players who must balance launching a shawarma for collectibles while taking as little damage as they can. Players must carefully examine the timing and direction of each launch, which promotes quick decision-making and a deeper understanding of physical forces. The game's UI has an arrow that gets bigger as the launch force increases, allowing players to dictate the shawarma's direction and force. The arrow does not specifically show the gravitational pull affecting the shawarma's trajectory, so players must additionally take this into consideration. This means players must develop an understanding of the underlying physics to succeed in the game.

Fig. 1: Game's actual direction of movement

As players become more engaged with the game, they begin to understand the unshown forces at work and adjust their strategy accordingly. This leads to a more immersive and satisfying experience, as players develop a deeper understanding of the challenges offered by the game's mechanics. Successfully getting over these challenges of the game's mechanics, players take the collectibles which replenishes their health points and earns them a spot on the scoreboard, adding more motivation to master the game's principles.

```
if (Input.GetMouseButton(0) && results_bool == false)
{
      Vector3.currentPoint = camera. ScreenToWorldPoint(Input.mousePosition);
      currentPoint.z = 15;
      trajectory.RenderLine(camera. ScreenToWorldPoint(new Vector3(Screen.width / 2f, Screen.height / 2f,
camera.nearClipPlane))+(startPoint - currentPoint), camera.ScreenToWorldPoint(new Vector3(Screen.width
/ 2f, Screen.height / 2f, camera.nearClipPlane )));
}
```

Ultimately, "The Bezt Shaorma" provides a distinct gameplay experience that pushes players to improve their ability to make quick decisions and their comprehension of physical forces. The mechanics of the game present a challenging yet rewarding gameplay, encouraging players to keep practicing and enhance their abilities. Everyone can enjoy "The Bezt Shaorma" , whether they play for fun or as a more intense and competitive activity.

4 Creating “The Bezt Shaorma”

The development of the game began as a class assignment but quickly expanded into a fun game with a straightforward idea. Also, the faculty’s primary objective of the project was to create an interface and which I decided to do with sliders for volume control, tutorial screen and buttons that displayed text as the menu. Instead, it gradually evolved into a mobile game with new, improved buttons and a new menu layout.

Since studies show that warmer colors increase stimulation and are better for eye protection [11][12], the game's palette was chosen to mimic a function that certain devices offer. Taking this further, other studies showed that each color has an impact on people’s mood. The colors vary between a yellow and a fiery rose. Yellow stands for an energizing mood that provokes feelings of happiness and optimism, while the fiery rose generates a stimulating effect, increasing a person’s heart rate and energy levels. In color therapy, red helps patients release negative emotions and lingering anger [11]. Red also stands for anger and danger. This is the main reason behind choosing a fiery rose instead of red. Another reason behind choosing these specific colors is that red, orange and yellow are also seen as sauces often present in fast foods. In the end the two main colors are joined as a gradient applied on all the buttons and text.

4.1 Game Effects and textures

In the developing process, we used Aseprite to design the elements of the game. Aseprite is a powerful tool for creating pixel art, which helped us to create visually engaging characters, backgrounds, and objects that all share a consistent visual style. With Aseprite's pixel-perfect design tools, we were able to add intricate details and create the main character and the collectibles that brought our game to life.

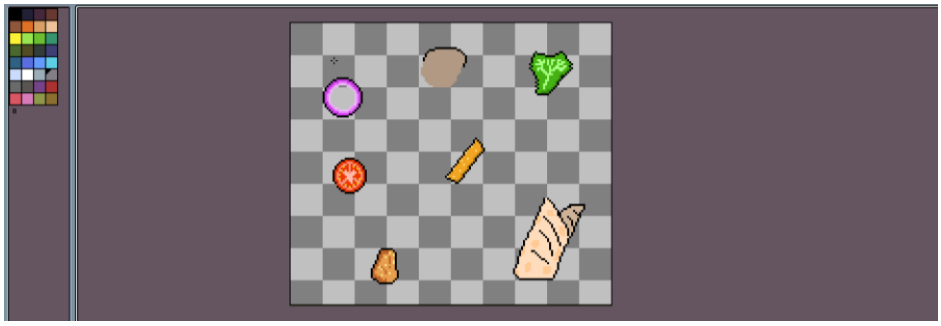


Fig. 2: Shawarma and collectibles in Aseprite

One key feature of Aseprite is its ability to create spritesheets. Spritesheets are collections of images that can be used for animation in games. Aseprite makes it easy to create and manage these animations for characters and objects in the game. This feature allows users to create pixel art animations, which enhances the overall gameplay experience.

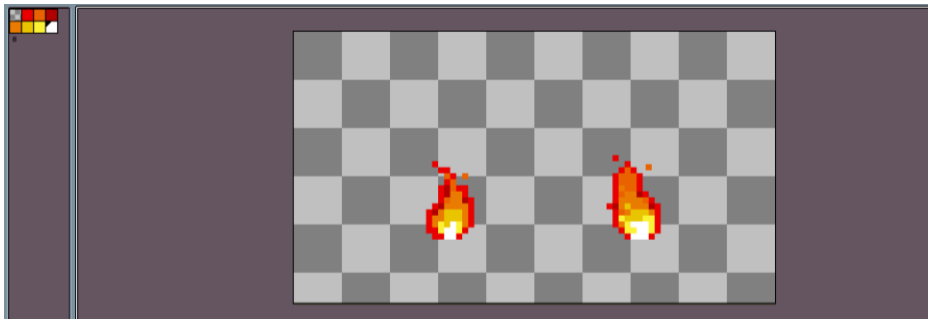


Fig. 3: Fire animation layers in Aseprite

The design of the user interface is a critical aspect of the game development process because it plays a key role in the player's navigation. In our game, we used Adobe Photoshop to design the buttons and the power-ups. Photoshop is a widely used tool in the game industry, and not only this program is used for business purposes and digital marketing. We found that Photoshop's layer styles were particularly useful in creating visually appealing button designs and power-up textures.

Fig. 4: Power-ups textures in Photoshop

4.2 Errors and impediments

While creating the game was easy, but time consuming, when it was time to build and deploy it, I realized the challenge was. Bumping from one problem to another, the errors kept on appearing. The first problem encountered was the incompatibility of Unity and the components installed for building the apk/aab. After trying way too many approaches including separately installing SDK, NDK, JDK and gradle, that are compatible with the unity version 2021.3.17f1, then replacing the ones pre-installed by unity, the game ended up only being exported from unity as a gradle file and loaded to Android Studio, where another problem was encountered. Android Studio was failing to build the game because of some duplicate dependencies. After solving the problem by going to File > Project Structure > Dependencies, selecting the unityLibrary module and then set mediation-sdk and unityads-adapter to testImplementation [Fig 5], the first apk was finally built and placed on a phone for testing purposes.

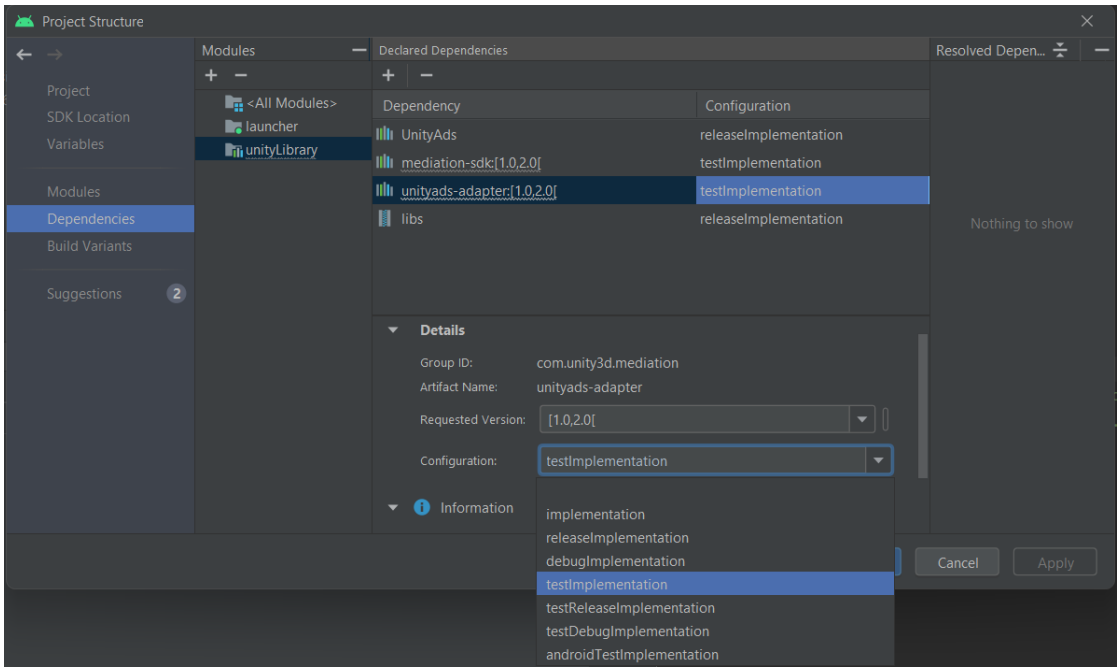


Fig. 5: Dependencies in Project Structure

Everything went smoothly going back and forth between testing and polishing the game, until it was time to finish the app bundle and another problem slipped in. After a lot of documentation and tutorials, it seemed impossible to adapt to Android Studio for creating the actual app bundle, so after getting back to Unity, copying the project and migrating it to a newer version with preinstalled components for Android, the previous errors were not encountered. The only problem now was "This project uses AndroidX dependencies, but the 'android.useAndroidX' proprety is not enabled. Set this proprety to true in the gradle.proprieties file and retry. See the console for details." which was easy to solve by just following the steps, finding the gradle.proprieties and adding the line quoted by the error, enabling the game to be built directly from Unity.

After a lot of effort, it finally looked like it was working. The last thing to do was to sign the app with a key. A signing key is a virtual key which can be created inside unity and it is required for an app bundle in order to be published on Google Play, because this is the main identification of the game and it will never change during its lifetime. However, after creating this signing key and building the release version app bundle from unity with a successful result, when trying to upload the game to Developer Play Console another error occurred. "You have uploaded an APK or Android App Bundle signed with a certificate that is not yet valid.". Being the first time developing games for mobiles, using the internet for research was an essential step. After repeatedly searching for about 10 days (about 1 and a half weeks), 4-5 hours a day, through every single corner of the internet, I never came across this exact same problem. After losing every hope, being one step away from dropping the project, it just snapped. During those 10 days, created keystores were also deleted about one hour after their creation and a new one was created, or they were deleted the next day before retrying for a fresh start. It turned out that the whole error was happening due to the time in the laptop that was responsible for creating the keystore. Although the laptop display hour was looking normal, it was 2 hours off its real timezone setting. The keystore was created 1 hour and 50 minutes before, resulting in a 10-minute wait to prove this was the problem. After 10 days that went by so fast, those 10 minutes felt like 5 hours. The actual quoted error text now made sense and it was finally working.

4.3 Scoreboard

Near the end of the creation progress, the game was missing a goal. As there were no levels and no other sense of progress, the implementation of a score logic and a scoreboard were a must have. Since there are 8 game modes from which players can choose, we created 9 different leaderboards. One for each game mode and a global one which is the highest score over all game modes. From the multitude of ways that the scoreboard could have been handled, using PlayFab was the best choice. PlayFab is a complete backend platform for live games with managed game services, real-time analytics, and LiveOps, made by Microsoft using Azure system. This choice was the best fit because it gives developers the ability to embed the leaderboard inside the game and design it as they wish. Due to the fact that the main goal of the game is to accumulate as much score as you can, the scoreboard had to be uniquely designed, this being the reason behind choosing a platform that lets individuals have overall control instead of the basic Google Play’s leaderboards.

5 Conclusion and future work

In the near future, the game will be optimized as there are some improvements to be done and an exploit of the game which involves that you can leave the game play by itself as the collectibles spawn around the player at an equal rate of probability. This problem will be solved by modifying the spawn, dividing it in 4 zones, then distributing spawn chance on these zones as following: top will have a 40% chance of spawning a collectible, bottom 10% and the difference of 50% will be equally split between left and right. This will not only get rid of the exploit, but it will also improve the gameplay by forcing the player to seek collectibles above them and by adding more use to the "reverse gravity" power-up, being the only way that the player can let the game play by itself. Because this is a power-up that lasts

until the next milestone, it will not be an exploit anymore. Other improvements to the game would be adding a more explanatory tutorial and more game modes, each having a more detailed description of what is good and what is not good to collect in that mode.

Furtherer, the game might also have a VR version, because of the VR ability to make people more engaged and immersed in the game. This version will have to be different because people will visualize a 3D world and the surroundings of the shawarma, enabling them to press the controller's button to start launching the shawarma and then look around them in order to see where they would like to launch it. This version will act as having a sling and the action will be in an oven. The player will be a 3d character having to launch the shawarma around the oven while being careful not to step on fire.

Due to the choice of colors and the unique game mechanics of “The Bezt Shaorma” , players are allowed to explore and have a feel on the physics of our world. Doing so, they are rewarded with a place on the scoreboards which also gives them the pleasure, satisfaction and fulfillment, an understanding of physical forces, the ability to make quick decisions and a positive thinking regarding that they can achieve anything they want by persisting.

6 Evaluation and experiments

Since the game was made in Unity, we will try to also modify some drawings such as drag power which has a finger as its artwork, to be pc compatible and we will also build a version for the windows.

The mimimum requirements for the game to run on Android are: 2GB RAM, 2Ghz processor, 256 mb video memory, Android 5.1, 30 MB Storage space.

References

[1] Positive effects of video games - https://gamequitters.com/positive-effects-of-video-games/ Last accessed March 2023
[2] Sensory Design in Games: Beyond Visual-BasedExperiences - https://www.researchgate.net/publication/342835393_Sensory_Design_in_Games_Beyond_Visual-Based_Experiences Last accessed March 2023
[3] KAT WALK C 2 VR TREADMILL - https://www.kat-vr.com/ Last accessed March 2023
[4] Matthew W.G. Dye, C. Shawn Green, Daphne Bavelier, Increasing Speed of Processing With Action Video Games - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2871325/
[5] I. Granic, A. Lobel, and R. C. M. E. Engels. The Benefits of Playing Video Games. Radboud University Nijmegen, Netherlands, 2014.
[6] Green CS, Bavelier D. Action video game modifies visual selective attention. Nature. 2003 May 29;423(6939):534-7. doi: 10.1038/nature01647. PMID: 12774121.
[7] Granek JA, Gorbet DJ, Sergio LE. Extensive video-game experience alters cortical networks for complex visuomotor transformations. Cortex. 2010 Sep;46(8):1165-77. doi: 10.1016/j.cortex.2010.06.010. Epub 2010 Jul 3. PMID: 20655069.
[8] Dye MW, Green CS, Bavelier D. Increasing speed of processing with action video games. Curr Dir Psychol Sci. 2009 Feb;18(2):321-6. doi: 10.1111/j.1467-8721.2009.01660.x. PMID: 20161490; PMCID: PMC2834727.
[9] Prot S, Anderson CA, Gentile DA, Brown SC. The positive and negative effects of video game play. Media Psychol. 2014 Feb 1;17(2):121-146. doi: 10.1080/15213269.2013.878443. PMID: 24653579; PMCID: PMC3956624.
[10] A. Trammell, G. Mittelmeier, and J. T. Velissaratou. Gaming and the Arts of Storytelling. In GAME Journal, Issue 8, pages 1-16. GAME - Games as Art, Media, Entertainment, 2021.
[11] How Color Can Affect Your Mood: All About Colors and Emotions - https://www.masterclass.com/articles/how-color-can-affect-your-mood Last accessed March 2023

[12] The Effects of Color on the Moods of College Students - https://www.researchgate.net/publication/315032059_The_Effects_of_Color_on_the_Moods_of_College_Students Last accessed March 2023

Ştefănel-Alexandru BANU
Vasile Alecsandri University of Bacău
Faculty of Science
Calea Mărăşeşti, nr. 157, Bacău, 600115
ROMANIA
E-mail: stefanelbanu@yahoo.com

Sorin Ionuț CONEA
Vasile Alecsandri University of Bacău
Department of Mathematics& Informatics
Calea Mărăşeşti, nr. 157, Bacău, 600115
ROMANIA
E-mail: stefanelbanu@yahoo.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Question-Answering System for Coffee Machines

Marius Benkert, Jochen Schmidt, Lennard Rose

Abstract

Our study presents a comprehensive Question Answering (QA) system for coffee machine related questions. The system covers a wide range of topics such as maintenance, usage, and troubleshooting of coffee machines, offering quick and accurate answers to the users through its intuitive interface and natural language processing capabilities. The system provides a seamless experience for coffee machine owners and users to access the information they need, ensuring smooth operation of their machines and allowing them to enjoy their favorite beverages with ease. To evaluate the performance of these models, we fine-tuned a range of BERT-based Transformers on a manually created dataset of 653 question-answer pairs. In conclusion, our findings demonstrate the feasibility of using NLP Question Answering models to deliver technical answers about coffee machines, and highlight the importance of fine-tuning these models on task-specific data.

1 Introduction

Coffee machines have significantly evolved in recent decades. This development has been driven by advances in technology and the demand for greater convenience and automation leading to today's modern, fully automatic machines. By using a variety of brewing methods, these machines provide a more efficient and consistent brewing process and a greater variety of coffee preparation options.

However, the increasing complexity of the machines and the introduction of new and advanced features made it difficult for users to fully understand and utilize all of the machines' capabilities. Therefore, the integration of new software and technology into everyday devices raised the need for more information to help users navigate the complex appliances, which resulted in user manuals becoming increasingly complicated.

Higher complexity also makes searching through the technical documents of these machines to find an answer to a specific question a more tedious task. Different manufacturers have different layouts and styles in writing up the instructions for their products. Even in products from the same manufacturer, it is not uncommon to find different structures within their documentations. This further makes it a more challenging task to find the specific answer to a question.

As a result, researching and developing techniques that can quickly and accurately return results for the imposed question within a specific manual could be beneficial. Such techniques could save both tedious work and time.

Recent advancements in Natural Language Processing (NLP) and Natural Language Understanding (NLU) driven by the use of transformer-based Large Language Models (LLMs) have enabled the development of various natural language based applications. A use case for such a LLM would be to extract answers to questions from a defined context, also known as extractive question-answering [10, 13].

Our approach addresses the difficulty of finding answers to specific questions within technical documents with an application that provides answers to user inquiries regarding coffee machines. This application is an end-to-end question answering system for coffee machine manuals. To the best of our knowledge, this is the first of its kind in the particular domain.

To begin, we collected, stored and preprocessed the coffee machine manuals. In a second step, the created corpus was used to build a custom question-answering dataset. Different transformer-based models have been tested and compared in their ability to answer questions with respect to our domain.

Additionally, we provide a web service where users can ask questions regarding their coffee machine.

The following chapters contain a detailed description of the implementation, as well as the reasons for the choice of the methods used. Section 2 provides an overview of the design choices made to create a scalable, resilient application. Section 3 describes the methods we used to collect and preprocess the data needed in training the model and to serve as a basis for the answers. Section 4.1 outlines the training and evaluation of the model. The main outcomes of the research and also suggestions for future work are in section 5.

2 Q&A System Architecture

2.1 Design

We present a data-centric approach to building a question answering system for coffee machines [2]. The logical procedure for the app can be split in two sub-processes.

In the first sub-process, the necessary data is collected and preprocessed to build a corpus. Further, the data is transformed to a form that can be utilized quickly and easily. This involves downloading manual documents and storing them. The second step involves preprocessing the data, which includes extracting the text from the PDFs, segmenting and separating the text by language to form a multilingual corpus. This corpus is further split up to paragraphs which are embedded for better processing. This procedure is explained in detail in section 3.

The second sub-process represents the actual functionality of the app. A webservice allows users to ask questions about coffee machines. For this, suitable context for the answer has to be extracted from the database. Based on these context, the used BERT-model generates answers to be presented to the user. The components and procedure of the service is further described in section 4

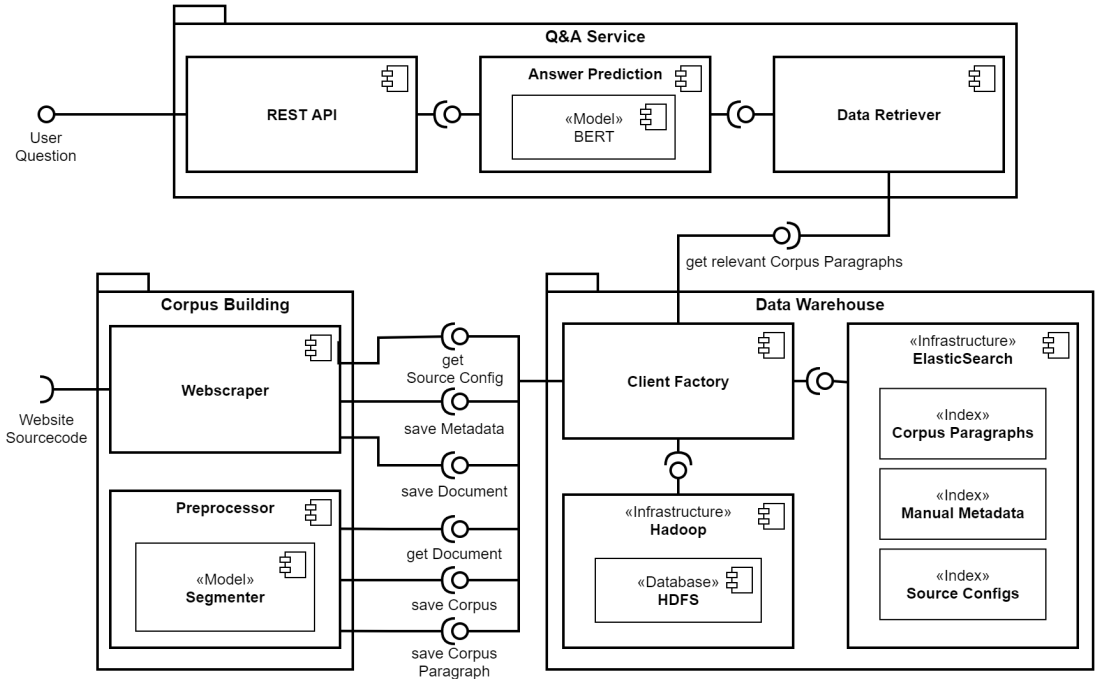


Figure 1: Applications Component Diagram

As seen in figure 1, the sub-processes are modeled in the applications architecture.

The process of data collection and processing is represented by the logical package *Corpus Building*. This contains the web scraper for collecting the documents and the preprocessor.
The question answering pipeline is modeled in the *Q&A Service* package. The structure follows the logical order from receiving the question request - processing the question - retrieving the data - processing the data - sending the answer response.
The data warehouse acts as the center point between both sub-processes. On the one hand, it stores all outputs from the *Corpus Building* package. On the other hand, it makes this stored data accessible easily and rapidly for the *Q&A service*. More about the data warehouse in the next section 2.2.

2.2 Data Warehouse

We chose a data warehouse as a centralized repository for storing and managing the large amounts of different data and to provide a single source of truth for the whole application. The data warehouse is optimized for consistent data storage, high availability as well as fast query and access.

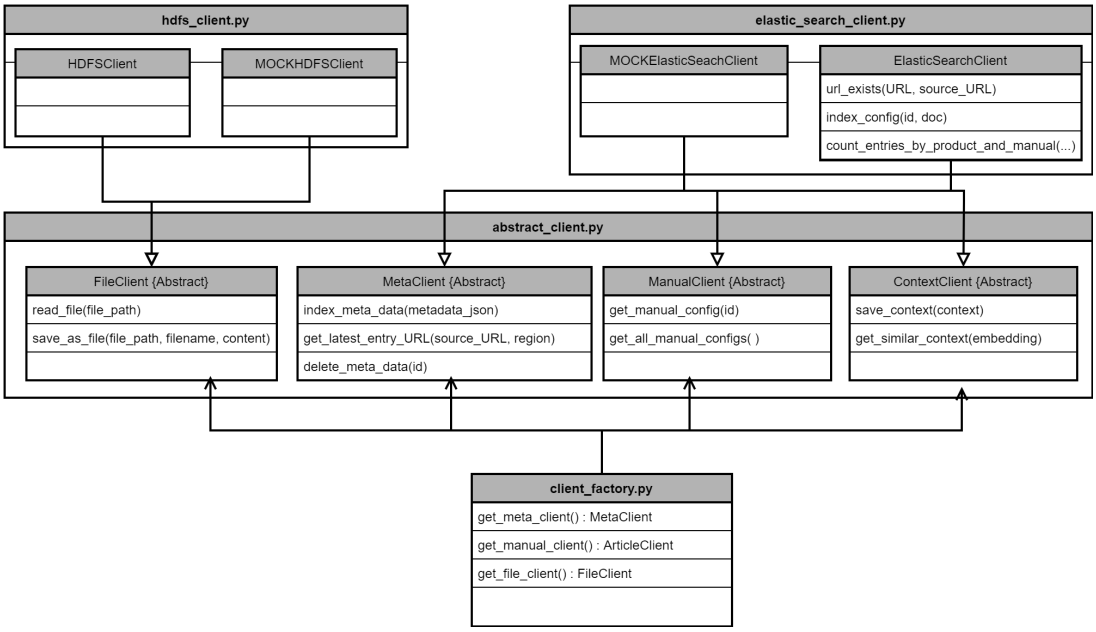


Figure 2: Data Warehouse Class Diagram

The abstract factory was the central module for building the data warehouse. The pattern allowed the encapsulation of client creation logic and to decouple the implementation of the database access to switch between different database and search engine implementations without affecting the overall system. Further, the data warehouse has to handle multiple types of data sources and structures, the abstract factory pattern simplifies this by encapsulating the logic for each kind of data. With changing requirements, this makes it easier to maintain and modify the system.
We defined four different abstract clients to handle our data types:

- FileClient: stores the PDF documents as well as the corpus documents
- MetaClient: stores meta data of other documents and data for easy query
- ManualClient: stores configurations needed to scrape documents from manufacturers websites
- ContextClient: stores the paragraphs that are used as a context for the question answering model

As the main data storage we chose Apache Hadoop. Hadoop provides a scalable and flexible solution for cost effective storage of large amounts of data storage [8]. Hadoop's distributed file system, HDFS, al-

lows data to be stored across multiple nodes, providing high availability and fault tolerance. Additionally, this distributed file system makes Hadoop exceptionally scalable.

To rapidly search and access data we used the popular search engine Elasticsearch [3]. It has, like Hadoop, a distributed architecture which enables Elasticsearch a near real-time search capability across large amounts of data [6]. We utilized this to be able to search for documents in the HDFS via the meta data stored in Elasticsearch. Further, we used the inbuilt similarity search (see section 4.2 for more information) to facilitate the retrieval of context paragraphs for the Question Answering Service.

3 Data Collection & Processing

As mentioned in Architecture - Design (subsection 2.1), the application can be separated in two parts - collection and service. This subsection explains the steps necessary to provide the data required for the question answering service to operate.

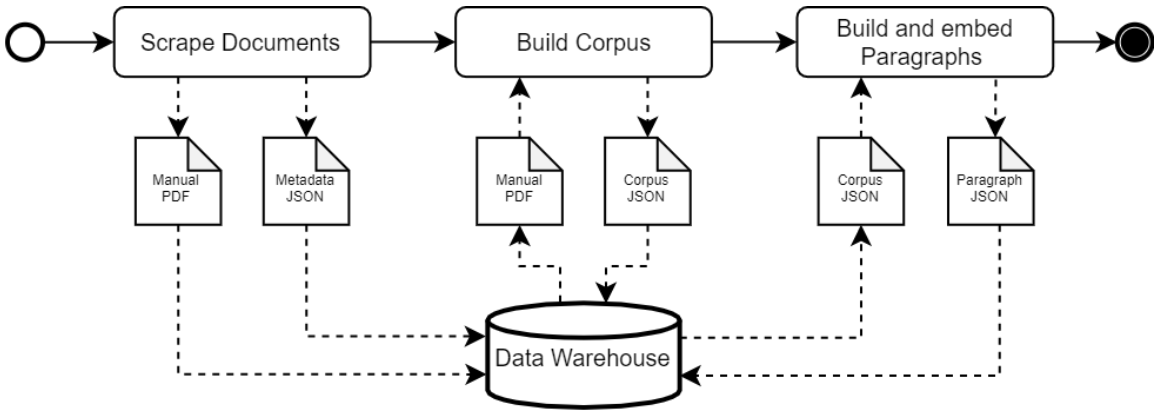


Figure 3: Data Processing Procedure

Figure 3 outlines the process of providing data. Starting with the collection process described in section Webscraping. Section Preprocessing explains the subsequent steps to make the data ready for use.

3.1 Webscraping

In order to apply natural language processing (NLP) techniques, a data basis is required. For that, we utilize web scraping. Web scraping is a method of automatically extracting data from websites and is widely used in the field of NLP for data collection. This automated and easily configurable method allowed us to gather large amounts of data from coffee machine manufacturer websites in a fast and efficient manner. Our web scraper can be easily configured to scrape data from additional sources, making it a flexible tool for gathering data for our question answering system.

The web scraper was developed from the ground up using Python programming language due to its comprehensive collection of libraries suitable for web interactions and scraping processes. The key packages employed in this implementation are *BeautifulSoup* [17] and *Selenium* [14]. *BeautifulSoup* is utilized to parse the web page and produce an accessible object from which relevant information can be extracted. However, some websites are not easily scrapable due to their use of JavaScript to dynamically load and render elements. To address this issue, we used *Selenium*'s WebDriver, which serves as an interface to a local web browser, enabling the execution of JavaScript scripts on the pages. This, in turn, enables *BeautifulSoup* to extract even dynamically loaded content.

Our web scraper was created to be flexible and easily configurable. It is configured by a JSON file for each source, enabling the simple addition of new sources. The scraper accesses the manufacturer's website and performs a breadth-first search for relevant products (see Figure 4), after which it accesses the detail pages to extract the meta data and manuals. We differentiate between the PDF files and the

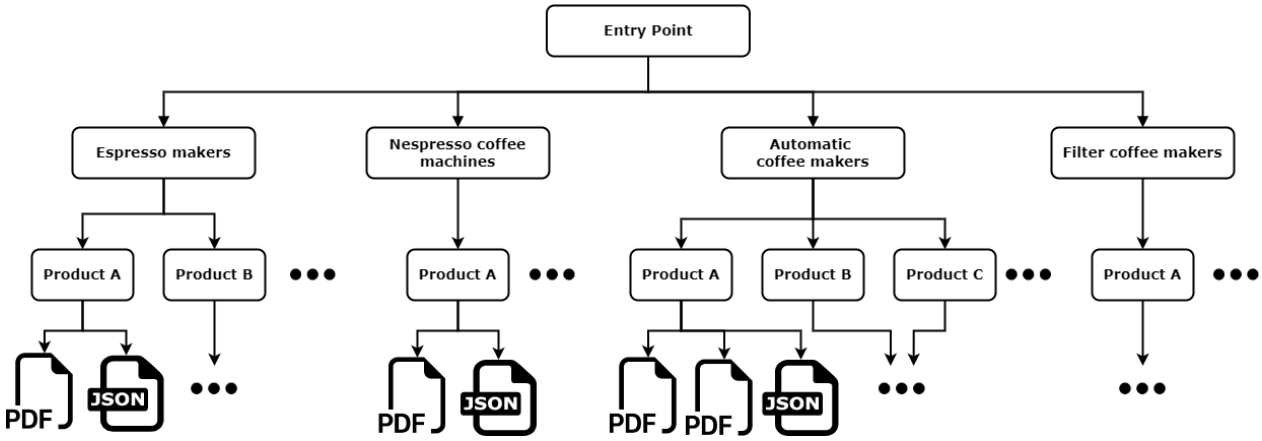


Figure 4: Example Hierarchy View of a Website

meta data, such as the product name, manual name, and product type. The meta data is saved in our data warehouse using Elasticsearch, while the manuals are stored in the Hadoop distributed file system (HDFS).

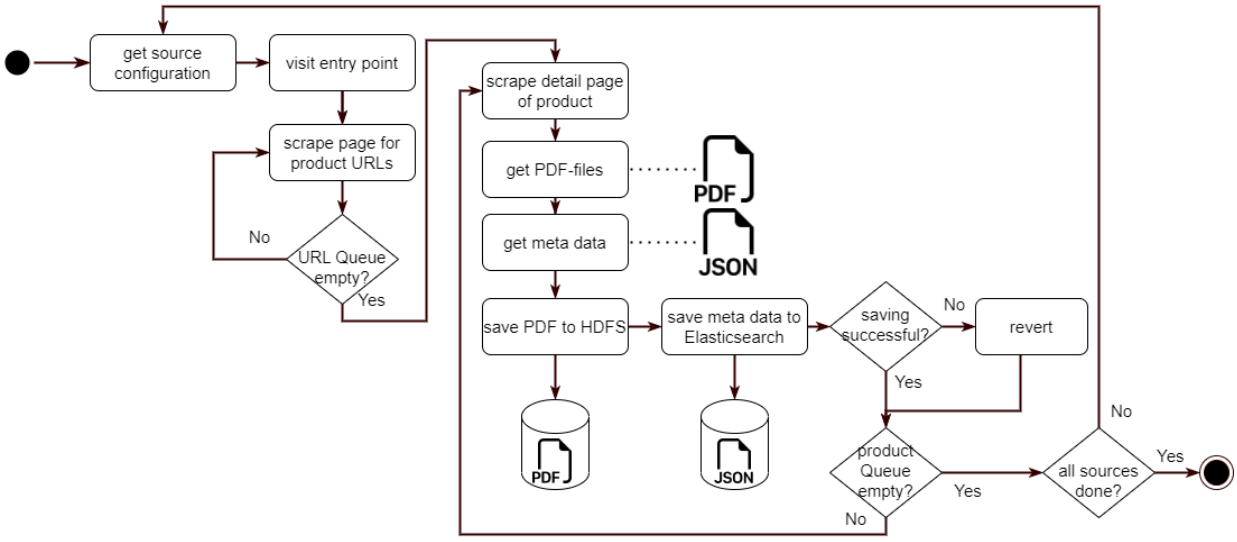


Figure 5: Flow Chart of the Web Scraper

By automatically extracting data from manufacturer websites, we were able to gather large amounts of information that would otherwise have been difficult and time-consuming to obtain. Nevertheless, in the case of a collection of available offline PDFs, it is possible to carry out the subsequent preprocessing steps with prior entry of the metadata. The data collected are the basis for the fine-tuning of our NLP model and later for providing the relevant context to the question of the user. In our data pipeline, web scraping is the initial and crucial step.

3.2 Preprocessing

With the crawled manuals being in PDF format, they first had to be preprocessed in order to be useful for downstream NLP tasks. There already exist a number of libraries that can easily extract the plain text from the meta data of PDF's [1]. Though, for this project the goal was to create a parallel, multilingual corpus, which relies on more information than just the plain text.

	AEG	Braun	Delonghi	Jura	Krups	Melitta	Miele	Philipps	Russelhobbs	Sage
Header AP	68.7	64.3	65.1	72.3	70.4	54.2	65.1	64.8	70.4	78.0
Subheader AP	70.4	63.6	58.6	67.0	57.4	65.3	65.8	68.0	61.1	71.0
mAP	70.0	64.0	62.3	70.0	65.7	59.6	65.5	66.4	65.8	74.5

Table 1: Average precision (AP) and mean average precision (mAP) in % obtained by the individual models on the the two classes Header and Subheader.

Definitie 1 *A parallel, multilingual text corpus consists of text segments in various languages that are aligned at a specific level such that each segment has translations or equivalents in other languages.*

In our case, alignment within the resulting corpus means having the same structure of headings, subheadings and paragraphs for every language available in the manuals. In order to obtain such a structure, additional segmentation of the documents is required.

3.2.1 Segmentation

Segmenting documents is not a new problem and a lot of different solutions have already been proposed. These solutions range from strictly rule-based segmentation to solutions where machine learning is involved. After initial investigation of the different kind of document layouts we are dealing with, we decided on using object detection methods to detect the headlines and sub-headlines within the manuals. In particular, the LayoutParser toolkit [20] was used for this purpose. They are building on top of the Detectron2 library [23] and provide various tools and resources [19] that help to reduce the amount of effort that is needed to train, evaluate and use custom object detection models.

In order to train such a custom model, several preparation steps needed to be completed (Figure 6). The first step for training a custom model was to convert the individual pages of the manuals into images. For the reason that the different manufacturers sometimes had very different layouts, a separate training set and model is created for each manufacturer. In a second step, the headings and subheadings had to be manually annotated within the selected images. Because of the fact that the labeling process was very time consuming, only around 100 images per manufacturer were annotated and used for the training of the individual models. For the purpose of annotating we used the open source data labeling platform Label Studio [21], which in the end provided a file export for the labeled data in COCO format.

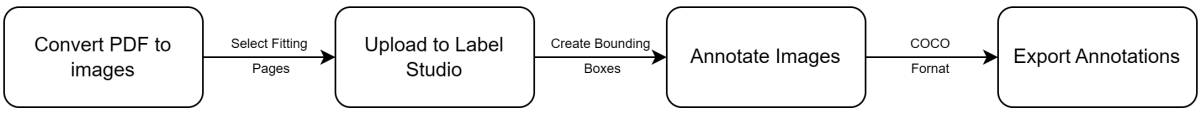
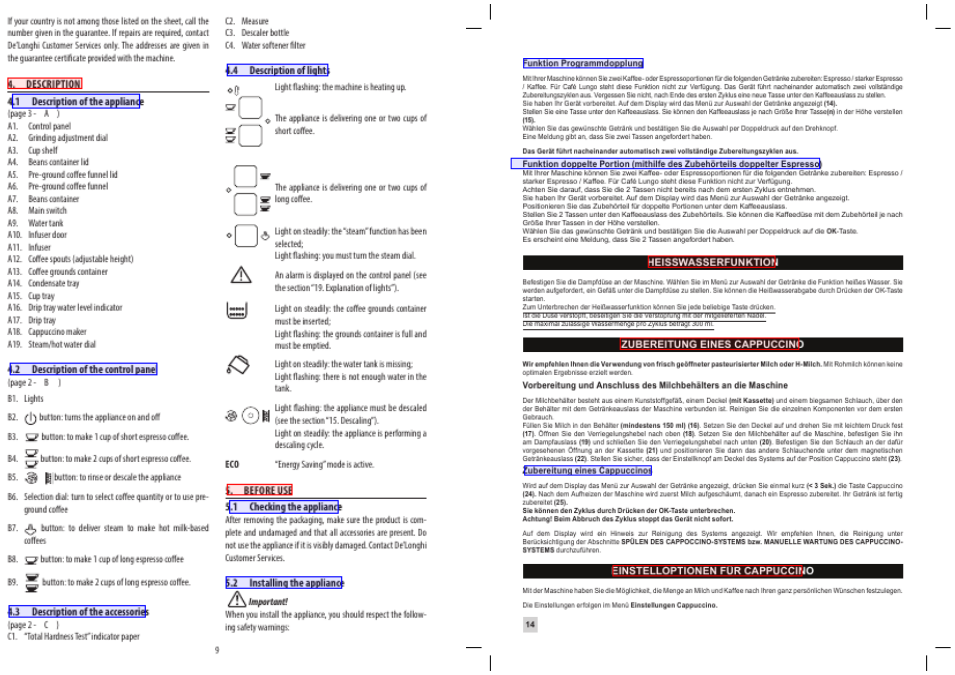


Figure 6: Pipeline for the preparation of training a custom object detection model.

The obtained COCO files for each manufacturer had to be split up into training and testing sets. These sets, together with an accompanying config file for the model, could then be fed into the training script that is provided in [19]. In the config file it is specified what model architecture should be used and how it should be trained. In our case, we used a Fast R-CNN with a ResNet-50 as the backbone [7], where large parts of the ResNet weights used were already pretrained on the ImageNet database [4] and only finetuned on our created training data.

For evaluating how good the individual models perform, we used the Average Precision (AP) and the mean Average Precision (mAP) of the predicted bounding boxes as described in [12]. Over all manufacturers combined we average a mAP of 66.4 (see Table 1). Considering that we only used around 100 images per manufacturer as training data, the results are pretty decent. As most deep learning architectures need a lot of data to really excel at their given task, having more training samples and more variations within our training data would probably increase the performance of the individual classifiers greatly.

Figure 7 showcases how the trained models predict the coordinates as well as the labels for the bounding-boxes for one page at a time. Comparing only the two manufacturers shown, it becomes



(a) Bboxes obtained for one of the Delonghi manuals.

(b) Bboxes obtained for one of the Krups manuals.

Figure 7: Obtained bounding boxes of the models on the example of two manufacturers. Red bounding boxes are the Header class. Blue bounding boxes are the Subheader class.

apparent how much the header and subheader representations can differ, which raised the need to train separate models. With the information about the location of the headers and subheaders on the pages, we can segment all the documents and store the information in a structured representation.

3.2.2 Corpus

Combining the plain text information obtained through the PyMuPDF library [22] as well as the bounding-box information obtained through the object detection, the parallel, multilingual corpus was built.

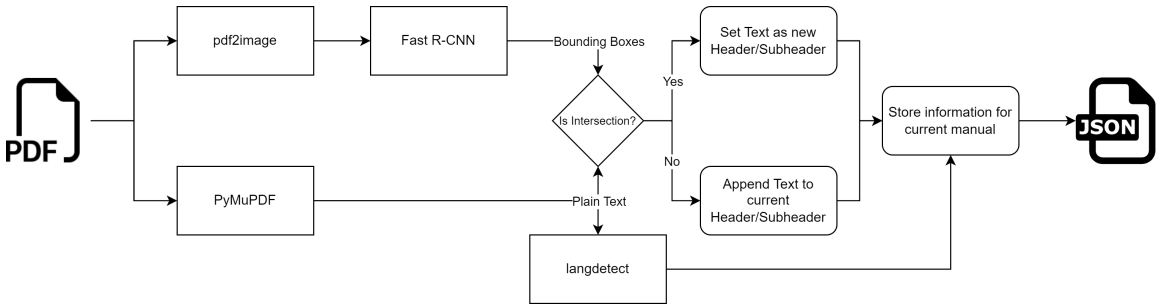


Figure 8: General preprocessing procedure for one manual.

The process, as seen in Figure 8, was performed for every manual that was scraped from the manufacturers website. Through PyMuPDF we read in every page iteratively and got the textual information as well as their respective coordinates on the page. Each text block was compared with the detected

bounding-boxes on the same page to check whether they intersected. If that was the case, a new current header/subheader was set. If they didn't intersect, the text block was appended to the currently set header/subheader. This way we could save the text blocks belonging to specific segments of the document correctly. Through another library called *langdetect* we also detected the language of the read-in manuals, which provided additional information for the structure of the resulting corpus.

3.2.3 Embedding

Additionally, to select only the most relevant paragraphs in the corpus for a given question, a Similarity Search was implemented. For that, all processed paragraphs within our corpus had to be embedded into some kind of vector representation, which is described in more detail in section 4.2.

4 Q&A Service

This section describes how the Q&A-Service utilizes the processed data from the previous section 3 to answer user questions.

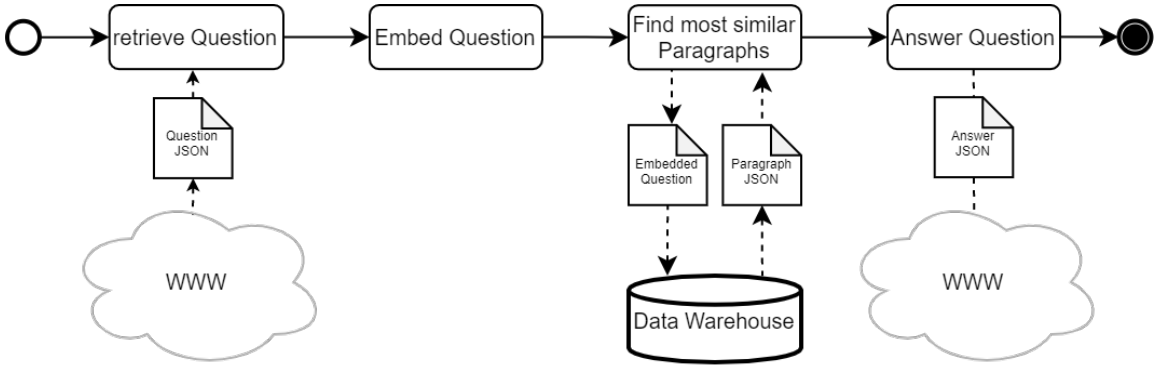


Figure 9: Question Answering Procedure

Figure 9 visualizes the general function of the Q&A-Service. First, the request containing the question as well as additional information is received by the API (see section 4.3. As described in section 4.2, the question is embedded in order to retrieve an appropriate context from the data warehouse. This context is then passed to the question-answer model (see next section 4.1) to extract an answer. The answer is then sent back as a response.

4.1 Model Training

In order to deliver precise and efficient answers to technical questions about coffee machines in our Question Answering System, we utilized Natural Language Processing (NLP). Within the NLP community, there are three distinct variants of Question Answering (QA) models:

- **Extractive QA**: The model extracts the answer from a provided context.
- **Open Generative QA**: The model generates free text based on the context.
- **Closed Generative QA**: In this scenario, no context is given, and the answer is generated entirely by the model.

Our approach adopts the Extractive Question Answering variant, where the context is delivered to the model via our data warehouse. Our data pipeline enables us to gather comprehensive and relevant

documents from various manufacturers and products. By providing the most appropriate context, we anticipate to achieve a high level of accuracy in our results.

An important framework in the field of NLP is *HuggingFace* which serves as a comprehensive resource, providing not only datasets and (pre-trained) models, but also a community and educational resources [9].

For our experiment, we chose to test BERT-based Transformers, specifically BERT, DistilBERT [18], and RoBERTa [11]. BERT, which stands for Bidirectional Encoder Representations, was first introduced in [5]. These Transformers can be fine-tuned to meet specific task requirements with the addition of a single output layer, which *HuggingFace* makes accessible and convenient. We also utilized models that were pre-trained on the SQuAD 2.0 dataset [15] and fine-tuned them for our task. SQuAD, or Stanford Question Answering Dataset, comprises over $100,000$ questions based on Wikipedia articles, with answers drawn from corresponding passages [16]. SQuAD 2.0 expanded the dataset by adding an additional $50,000$ unanswerable questions.

To perform the fine-tuning, we created a dataset of 653 question-answer pairs, which were manually constructed based on the information in our data warehouse. The dataset was divided into 80% for training and 20% for testing. We evaluated the various models without fine-tuning and with fine-tuning using the F1 and Exact-Match metrics, which are the most commonly used evaluation metrics in QA.

Exact-Match The Exact-Match metric is calculated by comparing the predicted answer and the actual answer character by character. If the predicted answer exactly matches the actual answer, the Exact-Match score is 1. If there is any difference, no matter how small, the Exact-Match score is 0. This metric is strict and considers an answer either correct or incorrect with no room for partial accuracy.

F1 F1 score is a common metric in classification problems and widely used in QA. It combines the precision and recall metrics to provide a measure of the overlap between the predicted answer and the true answer:

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2\frac{precision * recall}{precision + recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

Precision is the ratio of the number of correctly predicted words to the total number of words in the prediction, while *recall* is the ratio of the number of correctly predicted words to the total number of words in the true answer. The F1 score represents the percentage of shared words between the prediction and the true answer.

Model	with Fine-Tuning		without Fine-Tuning	
	F1	Exact-Match	F1	Exact-Match
bert-base-uncased	61.7	54.6	10.4	0.3
distilbert-base-uncased-squad	66.4	57.0	49.9	21.1
bert-base-cased-squad2	67.6	55.8	52.3	22.0
roberta-base	66.7	57.2	14.6	0.15
roberta-base-squad2	**68.3**	**59.8**	**58.3**	**27.5**

Table 2: Evaluation Results of different Models on our Dataset

Based on the results in Table 2 where we tested base and already fine-tuned variants without our fine-tuning and with fine-tuning the RoBERTa-base-SQuAD 2 performed the best and is our deployed model for later use.

4.2 Similarity Search

Similarity search is a technique used in a wide range of applications, such as information retrieval, recommendation systems, image and audio search, and natural language processing. In our project, we applied similarity search to enable efficient and accurate search through a large corpus of text paragraphs.

The first step in our approach was to generate vector representations of the paragraphs using an embedding technique. Embedding techniques are widely used in natural language processing to represent words or sentences as dense vectors in a high-dimensional space. These vector representations capture the semantic meaning of the text in a way that can be compared using mathematical operations. We used a sentence-transformer model to generate vector representations of our text paragraphs as an embedding. We then stored these embeddings in Elastic Search, a highly scalable search engine that allows for fast and efficient search operations.

When a user enters a query, we first encode it as a vector using the same embedding technique used for the paragraphs. We then search through all the stored paragraph vectors using the cosine similarity metric. Cosine similarity is a measure of the similarity between two vectors that ranges from -1 to 1. A score of 1 indicates that the vectors are identical, while a score of -1 indicates that they are completely dissimilar. A score of 0 indicates that the vectors are orthogonal, i.e., they have no relation to each other.

We retrieve the paragraphs with the highest similarity scores and use them as input for the question-answering model (see section 4.1). By using similarity search, we can efficiently search through a large corpus of paragraphs and retrieve the most relevant ones in a matter of seconds.

Overall, the use of similarity search allowed us to build a highly accurate and efficient search engine that can retrieve the most relevant paragraphs to a user's query in real-time. By using vector representations of the paragraphs and cosine similarity as the search metric, we were able to efficiently search through a large corpus of text and retrieve the most relevant paragraphs for the user's query.

4.3 User Interface and API

For the User Interface and the REST API we decided to use Django, a Python web framework for building web applications. Django alleviates the burden of writing repetitive and standardized code from developers by promptly generating a functional web application programming interface (API) that comprises popular features, such as user management, thereby streamlining the development process. Since we only use the API to receive and answer questions from the user, we did not include any functionality to manage and operate the app. This left us with the two REST functions GET and POST.

Method: GET Retrieves a list of all products, grouped by their respective manufacturers, using a pre-configured factory meta-client. Returns the data in JSON format.

Method: POST Accepts a request with a set of parameters—manufacturer, product, language, and question—to instantiate a `QuestionAnswerer` object. If valid, the object processes the question and returns the answers in JSON format.
The user interface has been kept minimal in order not to distract from the actual function and to make it self-explanatory. In addition to the input field for the question, the user can enter all the information he has (manufacturer, product name) via various dropdowns to further narrow down his results. The results are displayed after a short period of time below the input fields.

5 Conclusion

The aim of this project was to build a question-answering system for coffee machines. In a first step, the manuals were scraped from their respective manufacturers website. In a second step, the content of the manuals was preprocessed into structured data, which made it easier to work with. The structured data then provided the basis for the question-answering system. A user can ask specific questions regarding his coffee machine and our system will extract suitable passages from said data basis. To reduce computational time and accuracy of the question-answering system, we find only the most relevant passages through similarity search. The results of the system demonstrated its ability to provide accurate and relevant answers to user queries.

The system's success highlights the value of NLP in the development of knowledge-based systems, particularly in the field of machine maintenance. The development of such systems could greatly improve

the overall user experience, reducing the time and effort required to find information, and improving the efficiency of servicing and maintenance.

In future iterations, the system's capabilities could extend beyond merely extracting specific phrases from paragraphs to answer queries. Leveraging recent advancements in generative AI, we could facilitate the creation of comprehensive and grammatically accurate responses. Furthermore, to enhance efficiency and reduce processing time, we aim to implement a caching procedure that will store pairs of user questions and the model's responses. It's also worth noting that, due to time limitations, the custom built datasets we utilized were relatively small and leave room for improvement.

Overall, this project provides a proof-of-concept for the development of a practical and effective question-answering system for coffee machines, and demonstrates the potential for NLP-based solutions in the field of appliance maintenance.

Acknowledgement: Special thanks to our project supervisor Dr. *Sebastian Furth*.

References

[1] Hannah Bast and Claudius Korzen. "A Benchmark and Evaluation for Text Extraction from PDF". In: *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*. 2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL). Toronto, ON, Canada: IEEE, June 2017, pp. 1–10.

[2] Marius Benkert, Jochen Schmidt, and Lennard Rosse. *Question Answering System for Coffee Machines*. `https://github.com/LennardRose/Coffee_Corpus_QnA`. Accessed on: Sep. 26, 2023. 2022-2023.

[3] *DB-Engines ranking of search engines*. `https://db-engines.com/en/ranking/search+engine`. Accessed: 2023-02-08.

[4] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[5] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[6] *ElasticSearch documentation*. `https://www.elastic.co/guide/index.html`. Accessed: 2023-02-08.

[7] Ross Girshick. *Fast R-CNN*. 2015.

[8] *Hadoop documentation*. `https://cwiki.apache.org/confluence/display/HADOOP2`. Accessed: 2023-02-08.

[9] Shashank Mohan Jain. "Hugging Face". In: *Introduction to Transformers for NLP: With the Hugging Face Library and Models to Solve Problems*. Springer, 2022, pp. 51–67.

[10] Kenton Lee et al. "Learning recurrent span representations for extractive question answering". In: *arXiv preprint arXiv:1611.01436* (2016).

[11] Yinhan Liu et al. "Roberta: A robustly optimized bert pretraining approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[12] Rafael Padilla et al. "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit". In: *Electronics* 10.3 (3 Jan. 2021), p. 279.

[13] Kate Pearce et al. "A comparative study of transformer-based language models on extractive question answering". In: *arXiv preprint arXiv:2110.03142* (2021).

[14] Sujay Raghavendra and Sujay Raghavendra. "Introduction to selenium". In: *Python Testing with Selenium: Learn to Implement Different Testing Techniques Using the Selenium WebDriver* (2021), pp. 1–14.

[15] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know what you don't know: Unanswerable questions for SQuAD". In: *arXiv preprint arXiv:1806.03822* (2018).

[16] Pranav Rajpurkar et al. "Squad: 100,000+ questions for machine comprehension of text". In: *arXiv preprint arXiv:1606.05250* (2016).

[17] Leonard Richardson. *Beautiful soup documentation*. 2007.

[18] Victor Sanh et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter". In: *arXiv preprint arXiv:1910.01108* (2019).

[19] *Scripts for Training Layout Detection Models Using Detectron2*. `https://github.com/Layout-Parser/layout-model-training`. Accessed on: Feb. 7, 2023. layout-parser, Feb. 2, 2023.

[20] Zejiang Shen et al. "LayoutParser: A Unified Toolkit for Deep Learning Based Document Image Analysis". In: *arXiv preprint arXiv:2103.15348* (2021).

[21] Maxim Tkachenko et al. *Label Studio: Data labeling software*. `https://github.com/heartexlabs/label-studio`. Accessed on: Feb. 7, 2023. 2020-2022.

[22] *Tutorial — PyMuPDF 1.21.1 Documentation*. `https://pymupdf.readthedocs.io/en/latest/tutorial.html`. Accessed on: Feb. 7, 2023.

[23] Yuxin Wu et al. *Detectron2*. `https://github.com/facebookresearch/detectron2`. Accessed on: Feb. 7, 2023. 2019.

Marius Benkert
Technical University Würzburg-Schweinfurt
CAIRO
97082 Würzburg
Germany
E-mail:
marius.benkert@study.thws.de

Jochen Schmidt
Technical University Würzburg-Schweinfurt
CAIRO
97082 Würzburg
Germany
E-mail:
jochen.schmidt.1@study.thws.de

Lennard Rose
Technical University Würzburg-Schweinfurt
CAIRO
97082 Würzburg
Germany
E-mail:
lennard.rose@study.thws.de

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Navigating the Future: an approach of autonomous indoor vehicles

Chandu Bhairapu, Christopher Neeb, Fatima Mohamed, Julian Tilly, Indrasena Reddy Kachana, Mahesh Saravanan, Phuoc Nguyen Pham

Abstract

In this project, we explored the ability of Reinforcement learning (RL) in driving an indoor car autonomously. RL has proven its good performance in solving challenging decision-making problems. Therefore, RL can be a promising solution for autonomous car to deal with complex driving scenarios. As hardware a model car eqipped with sensors and powerful computational unit has been used. We also utilized SLAM for environment mapping and a combination of lidar data and Wi-Fi technology for localization. The experiment showed that the model can perform very well in simulation. Although the model lacks the ability to drive the car as smoothly along a route, the car is still able to avoid obstacles and walls in an unknown real-world environment.

Keywords— Autonomous vehicles, SLAM, indoor localization, reinforcement learning

1 Introduction

For several years, autonomous vehicles, also known as self-driving cars, have been an active area of scientific research. Every year, approximately 1.3 million people worldwide die as a consequence of traffic accidents [1]. The development of autonomous vehicles has the potential to revolutionize the entire transportation system and increase safety on the roads. The goal of this research sector is to develop vehicles that can drive autonomously from a starting point to a destination without human intervention. Numerous studies and experiments have already been carried out in this area, making an important contribution to the development of autonomous vehicles [2]. The Society of Automotive Engineers (SAE) has categorized autonomous driving into six levels. The lowest category, level 0, describes the state in which there is no automation. The top category, level 5, is reached when driving is fully automated. Many commercial cars nowadays reach level 2 in this classification [3].

Despite the progress made in the field of autonomous vehicles, the need for further research and development remains as high as it has always been. There are still multiple challenges to overcome, associated with autonomous driving. One of these challenges is developing vehicles, methods and techniques that can be used effectively in complex indoor environments, for example, a campus. Companies such as Waymo and Cruise have already driven several 100,000 km autonomously [4]. However, this experience only applies to outdoor environments. The majority of corporate research and development, has focused on the outdoor environments, because economically, the most benefit is available there.

The motivation for using automated driving in complex indoor environments is particularly high in manufacturing and distribution enterprises. Many use cases involve the transport of materials, not people. The purposeful transport of goods and commodities takes a lot of time and resources. Automation of logistics would increase productivity and reduce costs.

The development of autonomous vehicles for an indoor environment, such as a campus, is expected to contribute to filling the research gap. The system consists of a JetRacer Pro equipped with a propulsion control system and a Jetson computer for data processing. The vehicle was also enhanced with an RPLidar and an Intel

Real Sense camera. Our assembled car is shown in figure 1. These sensors allow the vehicle to collect data about its surroundings. This data is used to create a map of the environment and to locate the vehicle within the map. The vehicle uses this information to plan the best route to its destination using reinforcement learning. Safety is an extremely important aspect that must be taken into account. The vehicle should ensure the safety of people or objects in its vicinity. Therefore, the autonomous vehicle is equipped with an object detection and avoidance system that allows it to detect and avoid any objects that are in its immediate surroundings. This work fills the niche by offering a unique prototype for the indoor autonomous driving challenge and providing new insights into the field of autonomous vehicles. The prototype is adapted to the SHL campus and reaches level 3, of the SAE classification.

Figure 1: Autonomous car with the JetRacer Pro platform, advanced sensors RPLidar and Intel Real Sense camera

2 The car's architecture

Before the work on algorithms can begin, the topic of architecture have to be discussed first. Obviously, a carrier unit is needed, which can drive indoors, carry all sensors needed onboard and which is capable of computing and/or communicating with algorithms to make it drive the way it is supposed to. Beside of that, we need to have a look at the question, what sensors will be needed to fulfill the task efficiently and what software architecture can be used to combine all this. Since factors as time, money and know-how are limited in our project, we chose to use the robot operating software (ROS, 2.4) to handle our codes as well as the internal hardware communication. With this decision in place, the following hardware choices have been made.

2.1 Nvidia Jetson Nano

First of all we chose to use a mobile computing unit, which can do both, efficiently compute algorithms on the car itself as well as granting remote access and data transfer, so that the computation of more complex algorithms can be outsourced if needed. The Jetson Nano by Nvidia is a highly compact platform with a lot of computational resources, which makes it a great tool for autonomous driving. It combines CPU and GPU processors and supports a wide range of sensors. In addition, a custom Linux Ubuntu system is provided as software, so that the usage of ROS (see 2.4) is covered as well.

In detail, the Jetson Nano Dev Kit B01 is based on the same architecture as more powerful Jetson platforms like the Jetson Xavier AGX, which are widely used in robotics and in the autonomous driving vehicle industry. A quad-core ARM A57 CPU with 1.43 GHz and 4 gigabytes of DDR4 RAM are combined with 128-core Maxwell GPU [5]. Especially the GPU lets the Jetson Nano be more useful than a Rasberry Pi and it provides the parallel processing power needed to handle large amounts of data created by sensors such as lidars or cameras. In addition, the Jetson nano provides multiple interfaces for connecting a wide range of sensors and peripherals, for example two CSI slots for cameras and digital pins for optional sensors like super sonic [5].

2.2 Waveshare JetRacer Pro

The JetRacer Pro by Waveshare is an autonomous racing car kit that is designed to be used with the Nvidia Jetson Nano (see 2.1) and it includes a battery pack, a power management system, a motor controller for steering and throttle, four wheels and tires embedded in an all-wheel undercarriage, a simple camera as well as a small display. The special Ubuntu version for this car includes all important software packages and example code, with which the car can be trained via supervised learning to follow a line autonomously. Some adjustments had to be made from our side, since we do not simply follow some tutorial. To fit in a lidar and a depth camera, we removed the included camera and its holder. Instead a custom 3D-printed rig has been installed, which offers mounting points for a camera and a lidar while offering storage for cables and keeping the Jetson Nano´s cooling intact.

2.3 Sensors

Two main sensors have been used in this project. An Intel RealSense camera as close to mid range depth sensor and a lidar for more exact long range measurements.

The Intel RealSense D455 is a depth camera that uses stereo vision to capture a three-dimensional image of the world. By combining two infrared cameras with a colour camera, not only pure depth perception is possible but the computation of coloured point clouds as well. The depth perception can be used for close range collision avoidance up to a distance of 6 meters while the coloured point clouds open up the possibility of generating coloured 3D-maps. The infrared cameras have a global shutter, which is special in comparison to other systems which usually have rolling shutters and therefore tend to miscalculate some distances while in motion. Also build in is an inertial measurement unit (IMU), a combination of a three-axis gyroscope and accelerometer which detects changes in position and rotation of the camera in dependence of time. The calculation of the depth image stream with up to 90 frames per second and a resolution of 1280 to 720 pixel takes place onboard [6]. This way some computational resources of the Jetson Nano (see 2.1) can be saved other uses. Additionally, the manufacturers' software offers full compatibility with Python, our mainly used programming language, as well as an full inclusion into ROS.

The RPLidar A3M1 is a low-cost laser rangefinder device used in robotics and autonomous driving. Commonly used in devices like indoor vacuum cleaner robots, lidar sensors stand out do to there high accuracy and 360 degree measurement angle. The RPLidar used in this project is lightweight and offers a maximum range of 25 meters with 16000 samples per second [7]. It comes with a software embedding for ROS.

2.4 ROS

ROS (Robot Operating System) is an open-source software framework for building and controlling robots. It offers a collection of libraries and tools for creating robot software, including communication, simulation, and visualization tools. The fundamental concept of ROS is the communication of nodes. A ROS node can be any algorithm which sends or receives data. Sensors and their software drivers for example can be included into a node, which will publish their measurements in a predefined syntax, the messages. To make sure, that the published data finds its ways to the right receiver, the so called ROS master or ROS core is used to distribute that data. A node does not contact another node directly. Instead they inform the ROS core about their capability of sending or receiving data of a certain kind, the so called topics. Only if there is at least one sender (a ROS publisher) and one receiver (a ROS subscriber) on the same topic, the core connects those directly and data can be send. Alternatively, service nodes can be used, when specific calculations are needed only momentarily and not during the whole run. Instead of connecting two nodes permanently, a service will be called once by a special message and it will return its results only when called [8].

In addition to those communicational core capabilities of ROS there are includable packages, which expand the functional scope of actions. All kinds of software packages are available for free, for example sensor drivers, example code for hardware or software tools such as path planing nodes for mobile robots. Beside of installable packages, ROS already included some of the most useful packages into their own framework. ROS provides a transformation tool, which allow the user to easily construct robot models with many movable parts, which do not act individually but depend on each other. The visualisation tool RViz allows to not only visualize such a model itself but their sensor data and environment as well and with Gazebo ROS even included their own simulation software. These software packages cover a lot of problems which commonly occur in the field of robotics, but their use is optional and can be replaced by own solutions. In this project, ROS enabled our teams to work independently on our tasks and merge it effortlessly afterwards.

3 SLAM

SLAM is short for simultaneous localization and mapping and describes a well known problem in robotics. For localization on its own a map is needed to point out where in it something (f.ex. a robot) can be found. In contrast, during the whole mapping process one needs to know the exact position and orientation said thing. The problem arises especially in robotics, where both techniques are needed at the same time, although each of these processes relies on data generated from the other one. There are many approaches of different kinds to solve the problem of SLAM with extended kalman filtering, particle filtering and graph based approaches being the most popular ones [9].

In this paper the open source tool Hector SLAM has been used which follows the extended kalman filter approach. The choice of Hector SLAM among the many available solutions results from two reasons. First of all, the vehicle described can only move in a plane environment, so only a 2D map is needed to compute its path. Although having sensors on board capable of creating a 3D map as well, 2D mapping is sufficient and easier to compute, so we chose to save computational capacities by using Hector SLAM. Secondly, most of the SLAM approaches use odometry data as mandatory input. Due to software limitations this was no option for our case. Hector SLAM is not only very efficient regarding computational resources. It publishes a 2D map and is one of the rare open source options which only needs lidar data without odometry [10].

3.1 Mapping

In this section, mapping of an unknown indoor environment is explained in more detail. The legitimate question arises, why do we need a map? An autonomous vehicle will be unable to plan its route safely and efficiently, resulting in delays as well as increased energy consumption. Therefore, the availability of an accurate and detailed interior map is essential for the successful operation of an autonomous vehicle. The map serves as a reference for the localization and navigation algorithms and enables the autonomous vehicle to understand its position and plan its movements within the environment to efficiently perform the tasks at hand [11]. To capture the environment, we use a lidar sensor mentioned earlier. Only the sensor data of the lidar is used, no Inertial Measurement Unit (IMU) or other sensors. For our use case, the system is accurate enough and no loop closure procedures are needed [12]. Because the world is complex, we use approximations that require certain assumptions. One assumption we make, is a discretization of the environment into independent cells. In this way we obtain a cell structure of the environment. This has advantages for our further steps.

We estimate the state of each cell using a binary Bayes filter. Each cell is a binary random variable estimate whether the cell is occupied or free. The map m is created from the measured sensor data z and the positions of the car x. Calculations are performed in log odds notation to improve efficiency. The log odds notation calculates the logarithm of the ratio of the probabilities. This ratio can also be expressed as a sum. Formula 1 shows the occupancy mapping in log odds notation. The complete derivation and further details can be found in [13].

$$l\left(m_i \mid z_{1:t}, x_{1:t}\right) = \underbrace{l\left(m_i \mid z_t, x_t\right)}_{\text{inverse sensor model}} + \underbrace{l\left(m_i \mid z_{1:t-1}, x_{1:t-1}\right)}_{\text{recursive term}} - \underbrace{l\left(m_i\right)}_{\text{prior}} \tag{1}$$

For the creation of the map, we needed several attempts to find the right settings in Hector map that we could achieve the best quality. For this we took the car in our hands and walked from the starting point (Fig. 2, bottom right) to the end point (Fig. 2, top left). In our experience, we walked with the car because it is a smoother movement than driving the car remotely. Due to the steady movement, the quality of the map was better and contained fewer errors. However, we made the assumption that the map would look the same at 0.2 m height when the car is moving as it does at 1.2 m height when we are carrying it. This assumption is acceptable for our map. The map was created in real time while the car was moving. The final result is shown in figure 2.

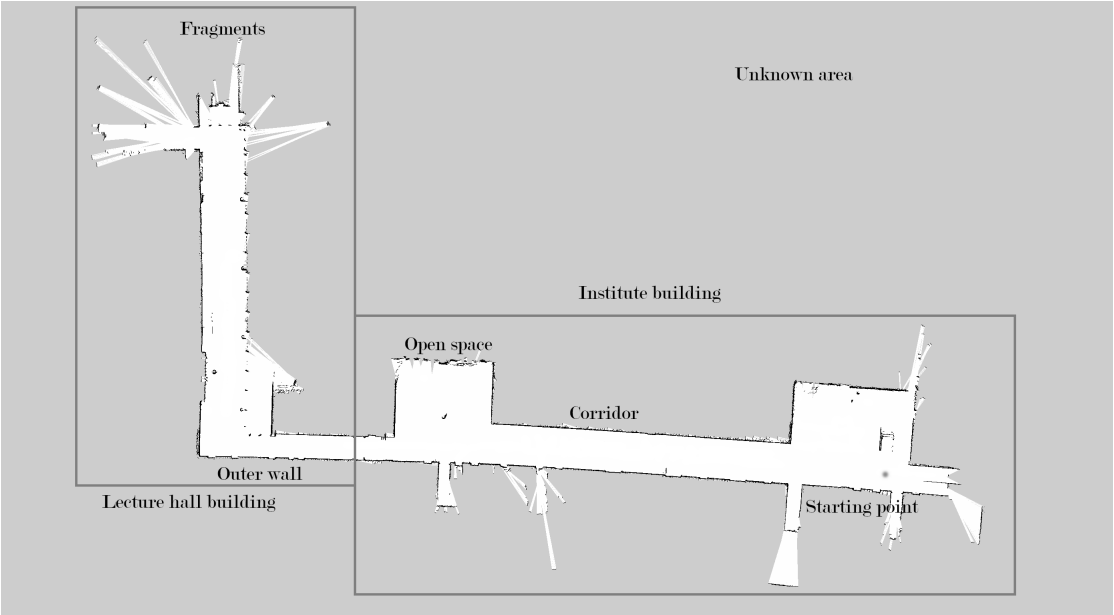


Figure 2: The map shows the first basement of the SHL campus. The horizontal corridor is part of the Institute Building and the vertical corridor belongs to the Lecture Hall Building. The black dots or lines represent obstacles and walls. The outer walls of the campus are clearly visible. The free space in between, shown in white, are the corridors and rooms inside the building. The unknown area is shown in gray, which is the outside of the basement in our case. There are fragments where the map is not completely accurate. This is the case where, for example, doors of rooms were open or corridors were not fully explored.

Once the map is created, it can also be used for training in a simulation. In this way, possible problems can be identified and corrected before the vehicle is used in a real environment. In the simulation, the map was revised and the fragments were removed to use an ideal map. In addition, the map can also be applied to validate the performance of various algorithms used by the autonomous vehicle and test diverse capability in a simulation. Thus, various scenarios such as unexpected obstacles or changes in the environment can be specifically tested, evaluated, and improved without causing a hazard in the real environment.

3.2 Localization

Accurate navigation of autonomous vehicles in indoor environments is a crucial requirement for their effective functioning. Therefore, the implementation of a reliable localization system is essential. This system determines the position and orientation of the vehicle, enabling it to navigate in a building or indoor environment effectively.

Various techniques have been developed to achieve accurate localization in different environments. There are five common techniques in total, namely Global Positioning System (GPS), visual-based systems, lidar-based systems, inertial measurement units (IMUs), and Wi-Fi-based positioning. The GPS is a commonly used localization method, but its application in indoor environments is limited due to several reasons. First, GPS signals in indoor environments are weakened by the presence of walls, roofs and objects, which interfere with the signal [14]. Second, the use of GPS in indoor environments often results in low-accuracy position information. Visual-based systems employ computer vision algorithms to determine the precise location of a mobile robot based on visual information in an indoor environment or building [15]. Lidar-based systems, on the other hand, use lasers to measure distances and provide position information for the mobile robot. Inertial measurement units (IMUs), which typically include accelerometers and gyroscopes, are also used for indoor localization and orientation of mobile robots [16]. Finally, Wi-Fi-based positioning utilizes the strength of Wi-Fi signals to determine the location of a mobile robot in an indoor environment.

In this project, we relay on Wi-Fi based localization to provide reliable real-time location information for indoor autonomous vehicles. We proposes a Wi-Fi positioning system for localizing autonomous vehicles to navigate in

indoor environments. The system relies on the signal strength of pre-installed Wi-Fi access points to determine the vehicle's location. A real-time Wi-Fi scanner was implemented in Python to detect Wi-Fi access points and their relative signal strength. The scanner compares the signal strength of the Wi-Fi access points to determine the car's position and direction. The proposed algorithm provides medium accuracy, but uncertainties can arise due to environmental factors or objects. These uncertainties should be considered when using the algorithm for real-time positioning of the autonomous vehicle. Specifically, Wifi localization is used as a coarse localization method, which provides an accuracy of approximately 9 meters. To achieve fine localization, the existing lidar system is utilized. This enables precise measurement of the distance between the vehicle and surrounding objects, facilitating accurate determination of the vehicle's position.

4 Object detection

Object detection is a widely studied and rapidly evolving computer vision technique that allows the detection of meaningful objects in digital images or videos. It has numerous real-world applications, such as self-driving cars, security and surveillance systems, and medical imaging. With the recent advancement of deep learning techniques, object detection has become an essential task in the field of computer vision [17].

Object detection algorithms typically rely on machine learning or deep learning approaches to identify relevant objects in an image. These algorithms learn to recognize objects by analyzing large data sets of training images and identifying unique patterns or features that are characteristic of each object. They then use this learned information to detect instances of these objects in new images or videos.

There are two main categories of object detection methods: Image-level detection and Instance-level detection. Image-level detection methods classify an entire image into a fixed set of predefined classes without identifying the location of the objects in the image. In contrast, instance-level detection techniques discover and classify numerous instances of objects in an image and produce a bounding box around each object.

Several state-of-the-art object detection methods have been proposed in recent years, including Faster R-CNN, You Only Look Once (YOLO), Single Shot MultiBox Detector (SSD), and RetinaNet. These methods utilize deep learning networks to detect objects accurately and efficiently [18]. For example, YOLO has achieved impressive performance on real-time object detection tasks [19], while RetinaNet has been shown to produce more accurate detections of small objects than previous methods [20].

Overall, object detection has proven to be a crucial tool in computer vision with many practical applications . Ongoing research efforts aim to improve the accuracy and efficiency of object detection algorithms, further expanding the range of applications for this important technology [21].

The object detection approach used in this paper is the mask R-CNN. Mask R-CNN is a computer vision architecture based on deep learning that is commonly used for segmentation. It is an extension of the popular Faster R-CNN network for object detection.

Mask R-CNN is built to handle two tasks at once: object detection and semantic segmentation. The task of recognizing and localizing things of interest in an image is known as object detection. The task of classifying every pixel in a picture into a preset set of categories is known as semantic segmentation. Mask R-CNN expands Faster R-CNN by including a network branch that generates binary masks for each instance of an item in an image.

Mask R-architecture CNN's is made up of three major components: a backbone network, a Region Proposal Network (RPN), and a mask branch. The backbone network captures information from the input image and sends them to the RPN, which provides a set of region suggestions. The RPN-identified regions of interest are then sent to the mask branch, which builds binary masks for each instance of an object in the image.

5 Reinforcement learning

Reinforcement Learning (RL) is a significant machine learning algorithm that differs from other learning algorithms in that it does not require prior training data. RL is utilized for sequential decision-making tasks and is formulated as a Markov Decision Process (MDP). In MDP, an agent interacts with an environment by following a policy,

receiving numerical rewards, training the policy and repeating this process until the policy converges to the optimal policy, which maximizes the cumulative reward. The fundamental components of an MDP include the environment, agent, policy, state, action space, and reward function, as illustrated in Figure 3 [22].

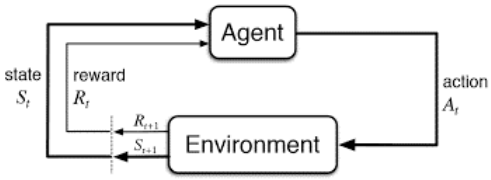


Figure 3: Markov decision process of Reinforcement Learning

5.1 Training

The policy must be constantly optimized using the knowledge acquired from experiences to maximize cumulative rewards and minimize penalties during simulations. Over time, the policy will improve and converge towards an optimal policy that guides the agent towards maximizing its rewards. The policy is updated after every action taken by the agent. Since the number of possible state spaces is vast and uncountable, the policy must approximate and adapt to similar state spaces.

5.1.1 Policy

The policy for the agent is defined by a simple neural network model consisting of three hidden dense layers. The input layer has 361 nodes, where the first 360 nodes receive the distance measurement at each angle around the car and the last node receives the car's speed (state space). The output layer has five nodes, each mapping to a different action in the action space. TensorFlow library is used to build, train and test the model.The model predicts the logits (Q-values) for each possible action. The car then executes the action with the highest predicted Q-value. In this way, the network guides the car's behavior by determining which action will result in the greatest reward based on the current state of the environment after training.

The epsilon-greedy approach is used to balance exploration and exploitation in the decision-making process of the agent. Exploration refers to the process of randomly selecting an action from the action space in order to gain knowledge about different states and actions taken from that state. On the other hand, exploitation refers to the process of following the policy and maximizing the reward. The epsilon-greedy policy manages this trade-off by allowing the agent to initially explore more, and then gradually exploiting more as the agent's knowledge of the environment increases.

The degree of exploration versus exploitation is controlled by the exploration-exploitation ratio ϵ. When ϵ is higher, the agent explores more and when ϵ is lower, the agent exploits more. The value of ϵ decreases over time at a rate determined by the epsilon decay rate, ultimately approaching close to zero as the final episode is reached. This approach ensures that the agent gains a thorough understanding of the environment while maximizing its rewards.

Experience memory is a buffer which stores tuples containing the history of each time step. The tuple B_t representing the time step t contains state of the car at time t (S_t), action taken by the car from S_t at time t (A_t), Reward received by the car for the action A_t at t (R_t) and next state of the car after executing A_t from state S_t at time t (S_{t+1}) [23].

$$B_t = (S_t, A_t, R_t, S_{t+1}) \tag{2}$$

The buffer appends every entry at the end and removes the first entry once it reaches the storage threshold value of 1024 entries.

5.1.2 Bellman Optimality Equation

The training process in reinforcement learning is to approximate the policy to an optimal policy which yields maximum cumulative reward. An Optimal model q_* should follow Bellman's Optimality equation

$$q_*(s,a) = E\left[R_{t+1} + \gamma \cdot \max_{a'} q_*(s',a')\right] \tag{3}$$

Expected cumulative reward for an action 'a' taken from state 's' and following the optimal policy q_* thereafter should be sum of immediate reward R_{t+1} and maximum possible future discounted reward ($max\ q_*(s\prime, a\prime)$) when following policy q_*. γ is the discount factor where each consecutive expected reward is discounted by γ^t.

5.1.3 Policy and target network

The batch of data is sampled from the replay memory buffer for training. The data are shuffled in order to break the corelation between the data. The current state (S_t) is fore-propagated into the policy network and it predicts the q values (logits) for each action. The next state (S_{t+1}) is passed into the new model called "Target network". The target network is the clone of policy network and their weights are locked. The second pass of (S_{t+1}) to the new network is to calculate the target q-value from the bellman's optimality equation which require optimal q value ($max\ q_*(s\prime, a\prime)$). The target q value is calculated using the same equation [24].

$$q_*(s,a) = E\left[R_{t+1} + \gamma \cdot \max_{s',a'} q_*(s',a')\right] \tag{4}$$

The loss between the target q value and the predicted q value from the policy network is calculated using the mean square error (MSE) loss function. The weights of the policy network are optimized using the Adam optimizer, an gradient-based optimization algorithm. To further stabilize the training process, the weights of the target network are updated with the weights of the policy network every 25 episodes. This separation of the target and policy networks is necessary to prevent any instability that could occur if both the target and predicted values were calculated within the same network using the same weights.

5.2 Reinforcement learning in simulation

Using simulation for training reinforcement learning model is essentially efficient in this project. Simulation enables the car to avoid collisions and reduces training time. Additionally, the car can explore more scenarios in simulation than in hand-designed situations during training. Therefore, our driving model is trained only in simulation with virtual lidar data.

Our simulator is developed by using a python library "Pygame", which is an open-source game environment. Pygame is extremely useful for training and testing reinforcement learning models with simple visualization (Fig. 4).

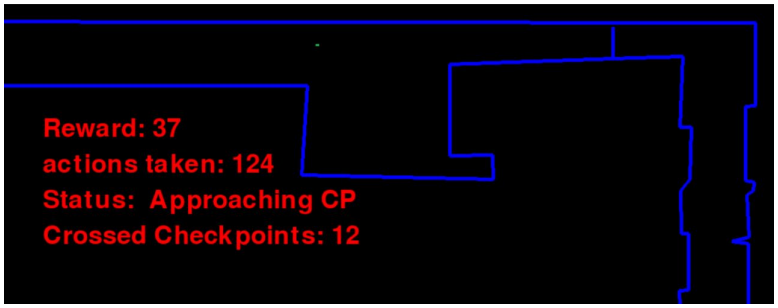


Figure 4: Pygame visualization of the buildings floor plan used for simulation

5.2.1 Environment

The environment in the simulator is the virtual space in which a agent (car) operates, which includes walls, obstacles, and other features. This representation is created by combining the floor plan of the building and a 2D map generated from lidar data using simultaneous localization and mapping (SLAM). The floor plan provides the correct geometric proportions, while the lidar map ensures a realistic representation of the real-world scenario. The resulting image was scaled to a resolution of 5 cm per pixel using Adobe Photoshop. Figure 5 illustrates a part of merged map of the environment. The starting and ending coordinates of each line in the merged picture are extracted using OpenCV Hough line transform function.

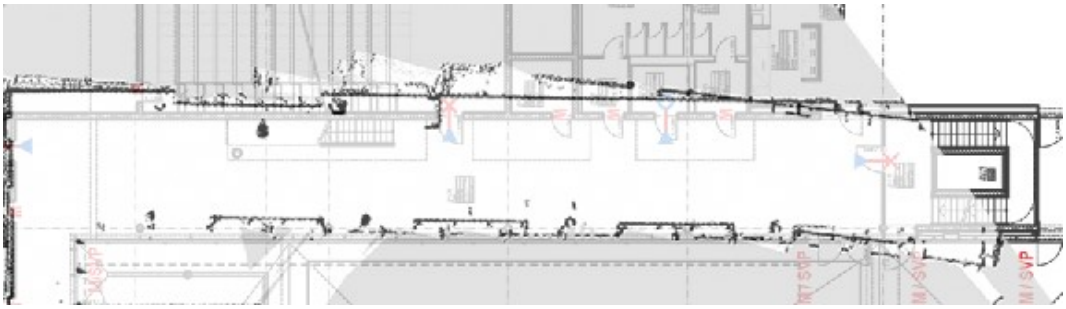


Figure 5: This figure shows the combined map created by combining lidar indoor mapping and the blueprint. This map enables them to provide better navigation and path planning for autonomous vehicles operating in this environment.

The agent in our simulation is a car that moves within the environment, which is defined by lines marking the boundaries of the car. The car is 30 cm by 15 cm in dimension and is scaled to match the resolution of the environment map. The car's state changes in response to action commands, which are realized by updating the pixel values of its corners. The car can move a maximum of 10 cm (2 pixels) in a single time step and can turn up to 10° in either direction. To simulate the effects of real-world uncertainty, Gaussian noise with a standard deviation of 2 pixels (10 cm) is added to the car's position and a noise of 5° is added to its orientation.

The state space represents the car's position and orientation within the environment. This is calculated by projecting 360 rays at different angles around the car and calculating the distances between the car and the points of intersection of the rays and walls or obstacles. This calculation was performed using Cramer's rule. The resulting distance measurements form a geometrical representation of the car's state within the environment. Let R_i and W_i be the equation of line denoting i^{th} Ray and i^{th} wall respectively. The parameters a, b, c, d, e, f have been determined by many experiments.

$$R_i = ax + by + c \tag{5}$$

$$W_i = dx + ey + f \tag{6}$$

The point of intersection (x_i, y_i) can be calculated by equating both the equation.

$$(x_i, y_i) = \left(\frac{bf - ec}{ae - db}, \frac{cd - fa}{ae - db}\right) \tag{7}$$

The distance between closest intersection point and the car is passed as input to corresponding node. This process is repeated for all the rays and walls. Figure 6 represents the rays and the intersection points with walls.

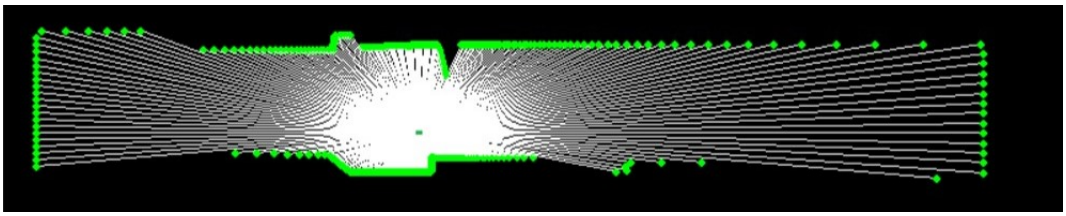


Figure 6: Visualization of the state representation during simulation. The lidar sensor emits laser beams in all directions around the car, which are represented in white. The green dots indicate the end of the beams where they collide with a wall or an obstacle.

The calculated distance data is subject to zero-centered Gaussian noise with a standard deviation of 10 cm (2 pixels) to simulate the real-world measurement uncertainty. Additionally, some random subsets of rays are omitted to further mimic the limitations of real-world sensors. These modifications add realism to the state space representation and help ensure the model's robustness to real-world conditions. The action space is the set of all

possible actions set the car can execute which is given by Move Forward, Move Backward, Turn left, Turn right, Stay

5.2.2 Reward function

The reward function evaluates the action taken by the car and returns numerical rewards and penalties. The route followed by the car is divided into a series of consecutive checkpoints. The car receives a reward of +1 for each time step in which it moves closer to the next checkpoint, and -1 for each time step in which it moves away from the checkpoint. When the car successfully crosses a checkpoint, it is rewarded with +10, and when it crosses a checkpoint that it has already passed, it receives a penalty of -10. This reward structure provides incentives for the car to move towards the next checkpoint and avoid backtracking.

A region of 30 cm (6 pixels) from any wall is defined as a danger zone. If the car enters this region, it receives a penalty of -1 for each time step spent in the danger zone. Additionally, if any part of the car intersects with a wall, the car is immediately penalized with a penalty of -10 and that marks the end of an episode. These collisions are calculated by checking for existence of any intersection points between the edges of the car and all walls in the environment. These penalties serve to discourage the car from colliding with walls and from lingering in dangerous regions near walls.

The car was not trained with additional obstacles, but learned to detect and avoid obstacles using lidar data. The vehicle was successfully tested in practice and avoided obstacles based on its skills acquired in the simulation environment. This was successful not only with obstacles encountered in training, such as walls, but also with unknown structures and people. The fact that both learned obstacles and new obstacles were successfully avoided illustrates the performance of the RL model applied.

6 Conclusion

In conclusion, the results of this study provide valuable insights into the feasibility and effectiveness of using RL in combination with SLAM and other techniques to solve the problem of autonomous driving in indoor scenarios. Special attention should be given to the following points.

The creation of the map works great after some tests. The result can be seen in figure 2. It shows that using lidar technology alone can provide highly accurate indoor maps with a resolution of up to 5 cm. The maps appearance is very similar to the original building plan. Compared to the ground-truth measurements, we achieve an average accuracy of 87%. The SLAM algorithm itself was able to handle the whole environment very well. In contrast, the RL algorithm had issues handling black surfaces and glass fronts, due to the sole usage of the live lidar data. There it is necessary to use additional sensors like the depth camera feed or ultrasonic sensors. Beside of such conceptual issues the hardware itself showed some minor difficulties as well. The wheels on the axis are not completely parallel. Due to the toe-in, the car does not drive straight even with 0° steering. This interference also affects the performance.

With regard to the capabilities of the RL algorithm, it can be shown that the trained model was able to successfully navigate through the environment in the simulator after 200 episodes, reaching the destination while avoiding walls and recovering from unfavorable situations. The model was tested in various conditions, including adding more noise to the input data, to validate its performance. In most of the tests, the car can navigates smoothly in the simulation. Positive tests in real-world scenarios were not as frequent and smooth. To increase the safety the simulation for the RL algorithm can be modelled closer to real life conditions, assisted by the addition of collision avoidance using object detection in the live run.

Finally it is quite surprising, how far a team of students can get when challenging the wide topic of autonomous driving. The combination of lots of open source code, the availability of efficient and affordable hardware and wealth of knowledge fostered by our study course in general have driven our approach of autonomous indoor vehicles to our personal success.

Acknowledgements

We would like to express our gratitude to Prof. Dr. Arndt Balzer and Prof. Dr. Magda Gregorová for their valuable contributions to this research. We also thank THWS for their financial support. Additionally, we extend our appreciation to our colleagues and collaborators for their helpful discussions and suggestions throughout the project. Finally, we are grateful to the reviewers for their constructive comments that have improved the quality of this paper.

References

[1] *Road traffic injuries*. 2022. URL: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (visited on 02/04/2023).

[2] Saeid Nahavandi et al. "A Comprehensive Review on Autonomous Navigation". In: *arXiv preprint arXiv:2212.12808* (2022).

[3] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. 2021. URL: https://www.sae.org/standards/content/j3016_202104/ (visited on 02/04/2023).

[4] David Fickling. *Self-Driving Cars Need to Slow Down After Uber Crash*. 2018. URL: https://www.bloomberg.com/opinion/articles/2018-03-20/uber-crash-shows-need-for-collaboration-in-self-driving-cars (visited on 02/04/2023).

[5] *Jetson Nano Developer Kit*. 2022. URL: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (visited on 02/04/2023).

[6] *Intel D455 Depth Camera*. 2022. URL: https://www.intelrealsense.com/depth-camera-d455/ (visited on 02/04/2023).

[7] *Slamtec RPLidar A3*. 2022. URL: https://www.slamtec.com/en/Lidar/A3 (visited on 02/04/2023).

[8] *ROS Concepts*. 2021. URL: http://wiki.ros.org/ROS/Concepts (visited on 03/04/2023).

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[10] S. Kohlbrecher et al. "A Flexible and Scalable SLAM System with Full 3D Motion Estimation". In: *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE. 2011.

[11] Syahrul Fajar Andriawan Eka Wijaya et al. "Research Study of Occupancy Grid map Mapping Method on Hector SLAM Technique". In: *2019 International Electronics Symposium (IES)*. 2019, pp. 238–241.

[12] Stefan Kohlbrecher et al. "A flexible and scalable SLAM system with full 3D motion estimation". In: *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. 2011, pp. 155–160.

[13] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. Cambridge, Mass.: MIT Press, 2005.

[14] Yuxiang Sun, Ming Liu, and Max Q.-H Meng. "WIFI signal strength-based robot indoor localization". In: *2014 IEEE International Conference on Information and Automation (ICIA)* (2014).

[15] Ming Liu et al. "The role of homing in visual topological navigation". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012).

[16] Giovanni Fusco and James M. Coughlan. "Indoor localization using computer vision and visual-inertial odometry". In: *Lecture Notes in Computer Science* (2018), pp. 86–93.

[17] David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. prentice hall professional technical reference, 2002.

[18] Median Hardiv Nugraha and Dina Chahyati. "Tourism object detection around monumen nasional (monas) using YOLO and retinanet". In: *2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*. IEEE. 2020, pp. 317–322.

[19] J Terven and D Cordova-Esparza. "A comprehensive review of YOLO: From YOLOv1 and beyond. arXiv 2023". In: *arXiv preprint arXiv:2304.00501* ().

[20] Zihao Zhou et al. "Image processing: Facilitating retinanet for detecting small objects". In: *Journal of Physics: Conference Series*. Vol. 1815. 1. IOP Publishing. 2021, p. 012016.

[21] Xiaoyue Jiang et al. "Deep learning in object detection and recognition". In: (2019).

[22] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Second. The MIT Press, 2018.

[23] Maxim Lapan. *Deep Reinforcement Learning Hands-On*. Packt Publishing, 2018.

[24] Richard S. Sutton et al. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. MIT Press, 1999.

Chandu Bhairapu
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
chandu.bhairapu@study.thws.de

Christopher Neeb
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
christopher.neeb@study.thws.de

Fatima Mohamed
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
fatima.mohamed@study.thws.de

Julian Tilly
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
julian.tilly@study.thws.de

Indrasena Reddy Kachana
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
sivasaikumarreddy.kachana@study.thws.de

Mahesh Saravanan
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
mahesh.saravanan@study.thws.de

Phuoc Nguyen Pham
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
phuocnguyen.pham@student.fhws.de

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Autonomous driving simulation utilizing DNNs in Unreal Engine 5

Cristian Nicolae Cîndea, Stephanie Fîșcă

Abstract

The rapid advancement of deep learning techniques and the growing importance of virtual environments in research and development have enquired the integration of deep neural networks (DNNs) into various platforms. This paper presents an approach for the integration, manipulation, and utilization of DNNs in Unreal Engine 5, focusing on the integration of autonomous lane-keeping control for vehicles. We create and train a DNN outside of the engine, employing a combination of convolutional and fully connected layers to process and interpret visual driving lane information. The proposed network is then integrated into Unreal Engine 5, utilizing a custom plugin with support from ONNX runtime and an OpenCV dedicated communication channel to bridge the gap between the two environments. The resulting system demonstrates the effectiveness of the DNN-based lane-keeping controller, showcasing robust performance. Our work contributes to the phenomenon of integrating deep learning techniques in game engines, offering valuable insights for future developments in the fields of autonomous vehicles, robotics, and virtual prototyping.

1 Introduction

This article contributes to the rapidly progressing field of autonomous vehicles, presenting a novel approach to the simulation of self-driving cars using Deep Neural Networks (DNNs) and the emerging capabilities of Unreal Engine 5. The fusion of advanced machine learning techniques and state-of-the-art game engine technologies offers a robust platform for generating, processing, and interpreting data, facilitating the development of an autonomous driving model.

The research of artificial intelligence (AI) has seen significant improvements in numerous sectors, one of which is the automobile industry. The concept of autonomous vehicles, once considered a fabrication of futuristic fantasy, is steadily transforming into reality. However, despite the considerable progress, there still exist numerous challenges and complexities to be overcome, primarily concerning safety, reliability, and efficiency.

Machine learning, particularly deep neuronal networks, have played a pivotal role in addressing these challenges by their superior capabilities in interpreting complex and high-dimensional data. Meanwhile, simulation environments have emerged as indispensable tools, offering controlled, diverse, and scalable platforms for generating, training, and testing AI systems. Specifically, they provide realistic scenarios for collecting data and modeling the real-world dynamics that a self-driving car might encounter.

1.1 State of the art

The article “Robust Reinforcement Learning-based Autonomous Driving Agent for Simulation and Real World”, published in 2020, describes a novel approach for training autonomous vehicles using deep reinforcement learning (DRL) [2]. The authors propose a method called "Robust Virtual Reality Training Platform" that combines DRL with a virtual reality (VR) environment to train autonomous vehicles in a safe, simulated setting. The DRL algorithm enables the vehicle to learn how to navigate through a variety of virtual scenarios, including different road conditions, traffic situations, and weather conditions. By training the vehicle in VR, the authors argue that the vehicle can learn to handle a wide range of scenarios, improving its robustness and safety.

The authors of the paper “Autonomous driving in simulated environments using reinforcement learning”, propose a novel approach for training autonomous vehicles in simulated environments using reinforcement learning [3]. They combine computer vision, sensor fusion, and a deep reinforcement learning agent to enable the vehicle to navigate in a range of driving scenarios, including urban streets and highways. The evaluation of the agent's performance shows that it is able to navigate safely and efficiently, making it a promising approach for improving the safety and performance of autonomous vehicles. The authors suggest that this method can be used to test and refine algorithms before they are deployed in real-world settings, potentially leading to safer and more efficient autonomous vehicles.

1.2 Our solution

The project detailed in this article brings these two powerful tools together, employing Unreal Engine 5 for simulation and data generation, and python with Keras, numpy and other libraries for data preprocessing, model training, and exporting. Unreal Engine 5, with its highly detailed and dynamic environments, presented in fig. 1, offers a platform that closely mimics real-world scenarios. At the same time, the use of Keras in python simplifies the complex task of training convolutional models.

Fig. 1: Screenshot from Unreal Engine showing a realistic looking environment.

The key components of the project include a simulation module for generating environment images and associated data (steering, throttle, brake, speed) and a python script for data

preprocessing and model construction and training. The trained model is then converted to ONNX format [1] and reintegrated into the Unreal Engine, completing the loop. A significant part of the process involves image preprocessing using OpenCV before it is fed into the DNN.

Titled "Autonomous driving simulation utilizing DNNs in Unreal Engine 5," this work underscores the idea that through a well-integrated and intelligent system, an autonomous car can be trained to navigate realistically within a virtual environment. It highlights two significant perspectives: first, the innovative use of DNNs for developing autonomous driving systems, and second, the potential of game engine technologies like Unreal Engine 5 in providing realistic, dynamic, and scalable platforms for training such models.

This article is structured as follows: Section 2 provides a detailed description of the simulation module and the python script, elaborating on the data generation, preprocessing, and model training processes. Section 3 offers insights into potential future developments and improvements to the system, and Section 4 summarizes the conclusions derived from this project.

2 Design and concepts

To bring the concept of autonomous driving simulation to fruition, this project is divided into two primary modules: an Unreal Engine 5 application for data generation and simulation, and a Python-based script for data preprocessing and artificial neural network model training.

2.1 Unreal Engine 5 Application

This segment of the project harnesses the power of Unreal Engine 5, an advanced game engine that is able to generate detailed and dynamic environments that closely mimic real-world scenarios. The application is instrumental in generating data - comprising images of the environment and associated parameters such as steering, throttle, brake, and speed values. This data serves as the primary learning material for the DNN.

2.1.1 Components

This module uses a few components such as:

- Virtual car with attachments:
  - User camera
  - Data processing camera
  - PID controller and Pure Pursuit Controller [4,5]
- Environment: [6]
  - Procedurally generating foliage
- Road

The virtual vehicle, as depicted in Fig. 2, is a high-definition model of a sports car, utilizing a skeletal mesh for animations. A skeletal mesh, in simple terms, is a structure used to create complex, articulated animations with multiple moving parts. The core logic of the car model is programmed in C++, supplemented by additions in Blueprints [7]. These are a visual scripting system that enables, as shown in Fig. 3, developers to build game logic without the need for extensive coding.

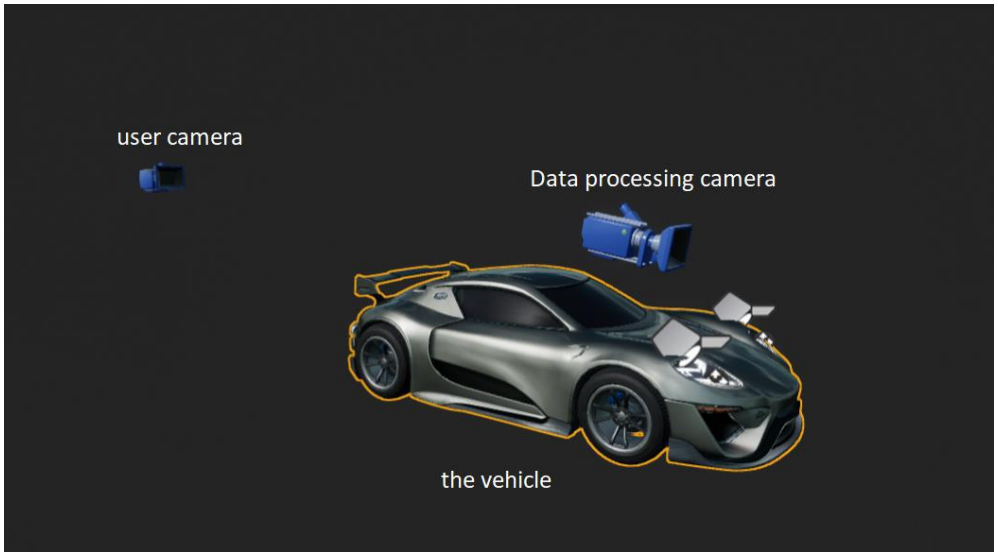


Fig.2: The vehicle with its components.

In C++, we construct and attach necessary components to the vehicle model, and manage the majority of the underlying logic for movement. The car model inherits from AWheeledVehiclePawn, a predefined class in Unreal Engine. Leveraging the UChaosWheeledVehicleMovementComponent, the virtual car simulates realistic physics behavior. Its front wheels control steering, while throttle and acceleration dynamics are simulated accurately. The brake system operates as expected in a real-world car.

Each frame, the car's behavior is dictated by either following the world's road blindly, responding to user input, or leveraging an ONNX model to navigate using AI. When operating in the first mode, the car asks the Road class for the nearest point ahead and uses a PID controller to steer towards it. If under user control, conventional gaming controls (i.e., W, A, S, D keys or a gamepad) are employed to maneuver the vehicle. The third driving mode, the AI-driven mode, will be elaborated on later in the paper.

The user camera, essentially a virtual viewpoint, defines what the user will see. It is tethered to the car via a string, thereby facilitating smooth transitions when the car is moving. Overlaid on this camera view, using widgets, are several vital parameters (Fig. 3): current speed, maximum permissible speed for the current section of road, and, when in AI mode, the network output and the frame being sent to it. In Unreal Engine, widgets are essentially user interface elements that enable the display of interactive content, including text, images, buttons, and more.

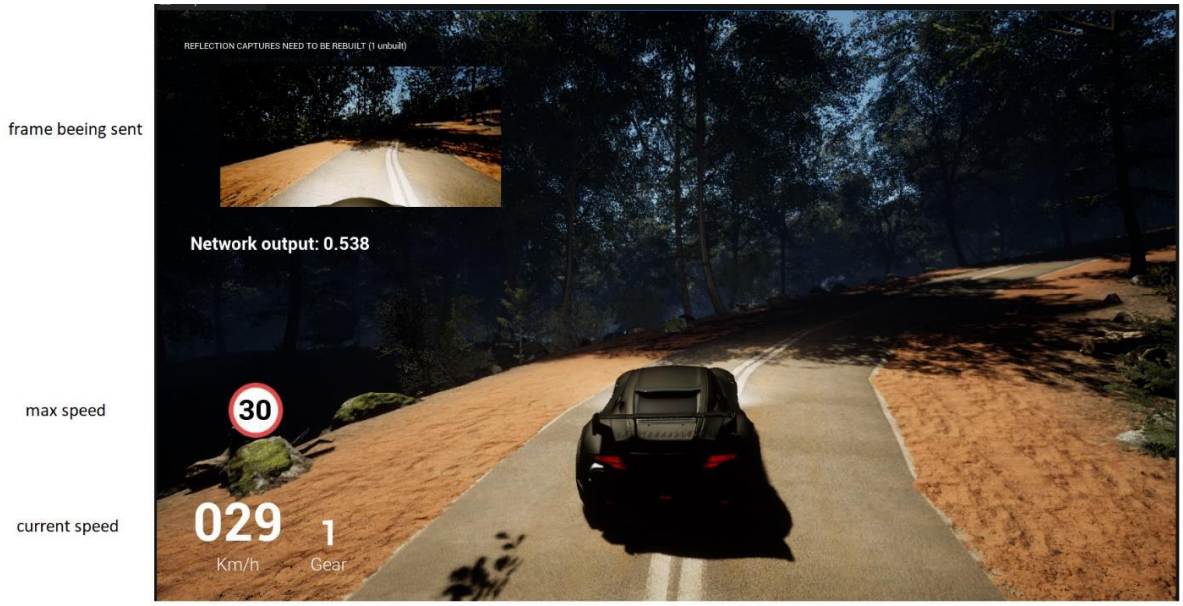


Fig. 3: Showing the user view experience

The Data Processing Camera is a critical component mounted on top the car, capturing the scene ahead. It has the capability to perform various tasks with the captured image. If required by the user, the captured image, along with corresponding data, can be saved to disk. If this is the case, the image is packaged and transferred to another class, named "ABrain," which automatically saves it to disk. This mechanism is designed to facilitate data transfer from multiple car instances to disk, a feature that, while not currently implemented, is planned for future developments. The Data Processing Camera also handles the selection of image resolution and format, default is 512x256 and RGBA8. If the user decides to run the AI model, the Data Processing Camera creates a specialized object for network interaction and forwards the frame for processing.

The class responsible for managing neural networks is named NNI_CNN. As a UObject, the base class of the engine, it employs a reference to UNeuralNetwork. This third-party experimental class enables the execution of ONNX models in C++. Upon receiving the image, this class utilizes the OpenCV library for preprocessing. The image, a TArray of FColors (a basic 2D array of 4-float vectors), is converted into an OpenCV mat. To accomplish this, we initialize an empty array with fixed length equivalent to the original array's length, specify the image format (RGBA8), and feed it the array pointer. After cropping, color space conversion to RGB, blurring, and resizing, the image data must be manually mapped to resemble a python numpy blob, a process demonstrated in Fig. 4. The image is then forwarded as a void pointer to the UNeuralNetwork reference for feedforward processing. Once a result is returned, it is relayed back to the Data Processing Camera, which in turn sends it back to the car.

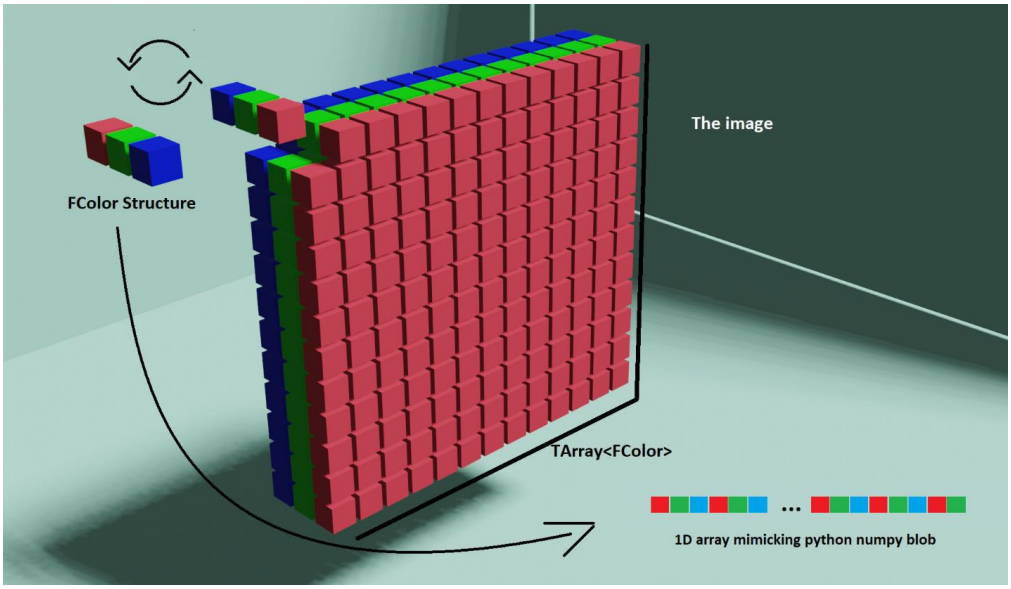


Fig. 4: The process of mapping the image as a numpy blob

The PID Controller, an acronym for Proportional, Integral, Derivative Controller, is a feedback mechanism utilized in control systems to adjust the output based on the difference between the desired setpoint and the measured process variable. In essence, it regulates a system's operation to eliminate errors over time. This internal component guarantees that the car maintains a constant speed, equal to the maximum speed allowed for that particular section of the road. If it detects a sharp turn, it adjusts the maximum speed to ensure the turn can be safely navigated.

The Pure Pursuit Controller is an algorithm designed to guide the vehicle along a predetermined path. This algorithm operates by continuously adjusting the vehicle's steering angle based on the current vehicle position and the target, or "pursuit", point on the path. This component is used when the vehicle is set to follow the road spline. The target point is dynamically selected based on a certain look-ahead distance along the spline. By consistently steering towards this target point, the vehicle is able to smoothly follow the designated path.

The environment of the project showcases a detailed landscape - a roadway through a forest set against a mountainous backdrop. The design aims to present a variety of driving challenges, including navigating left and right turns of varying degrees, passing through valleys and climbing uphill slopes, traversing through shadowed regions, and through sections where foliage reaches the roadway, exemplified in Fig. 5.

Fig. 5: Depicting challenging terrain

The landscape was generated using the default Unreal Engine map generator with a height map. Procedural generated foliage is used to ensure efficient resource management, enabling trees and other decorative elements to spawn within a certain radius around the player, demonstrated in Fig.6.

Fig. 6: Presenting the procedural generating foliage

Placed along the road are trigger boxes, as shown in Fig. 7, which are used to dynamically adjust the maximum speed for the vehicle. This adjustment is achieved by detecting an overlap in the vehicle's code, which then triggers a change to the new speed limit as determined by the trigger box.

Fig. 7: Speed changing trigger boxes

The road visible to the user is essentially an extension of the map. Given its integration with the map, direct queries from the car can be resource-intensive. To address this, if the car is set to follow the spline, a block of code in the ABrain class is executed. This code collects all the spline points from the map and crafts a new spline curve, which is then stored in an ARoad class object. This new spline is highly optimized and can effortlessly provide the nearest point to any given world location. Fig.8 illustrates how the vehicle's pure pursuit controller determines its target point on the spline. It requests the ARoad class to identify the nearest point on the spline to point A, and gets the point B in return.

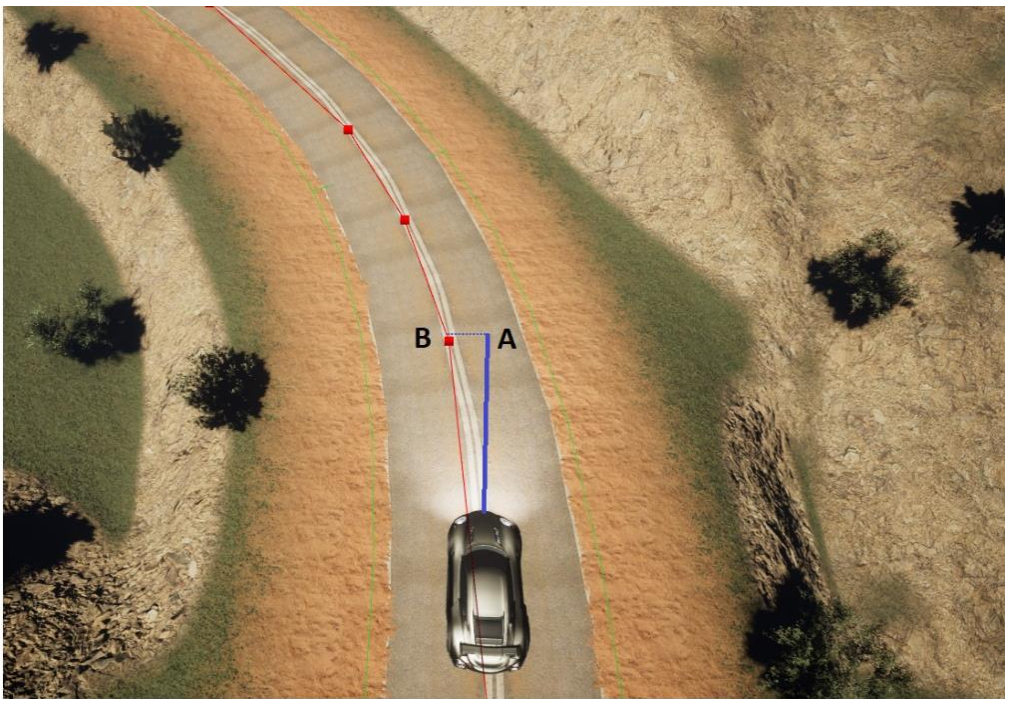


Fig. 8: depicting how the pure pursuit gets its target point

2. 2 Python Script

This part of the project leverages the Python programming language, which is known for its ability to quickly develop complex algorithms and its compatibility with powerful libraries such as Keras and OpenCV.

2.2.1 Data preparation

The data is archived on disk as individual JPEGs, along with a CSV file which includes the path to these images as well as information on steering, throttle, speed, brake, and reverse speed. However, for our purposes, we'll solely utilize the images and their associated steering data.

Step 1 – Data Loading: We import the CSV into a pandas dataframe. The steering values appear to be heavily concentrated into two specific ranges, thus we eliminate some data to achieve a more uniform distribution across all values. Fig. 9 highlights the contrast before and after this adjustment. Then the data is split into two sets - training and testing, using an 80/20 ratio. At the end of this step, we are left with two columns: one containing the image path and another with the corresponding steering value.

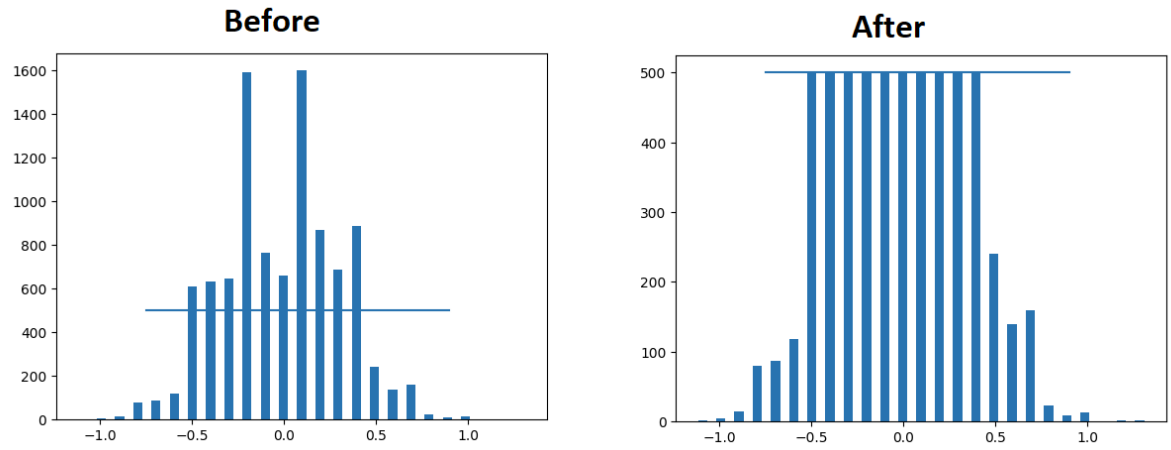


Fig. 9: Difference between before and after data drop

Step 2 – Image Preprocessing: Initially, we load the images from their respective paths into numpy arrays for each data set (training and testing), marking the beginning of the preprocessing phase. Each image undergoes a series of transformations: cropping to exclude the sky portion, blurring with a Gaussian filter of size 3x3, resizing to dimensions of 100 x 100 pxs, and finally normalization. Fig. 10 provides a comparison between an original and a preprocessed image.

Fig. 10: Image preprocessing

Step 3 – Model Construction and Training: The model is built using the Keras library with Tensorflow 2.10.0 as backend, starting from an empty sequential model. The architecture of the model, shown in Fig 11, is as follows:

- The input layer is a conv2D layer with 24 filters, each with a size of 5x5 and a stride of 2x2. The activation function used is ReLU (Rectified Linear Unit). The input shape is defined as 100x100x3, indicating that the model accepts images of size 100x100 pixels with 3 color channels (RGB).
- The subsequent 4 layers are of the same type, but have different numbers of filters, sizes, and stride values. The activation function remains ReLU.
- After these convolutional layers, there is a flatten layer whit purpose of flattening the output of the convolutional layers so that it can be fed into the dense layers.
- Following the flatten layer, there are 5 dense layers. The neurons in these layers utilize the ReLU activation function, except for the last dense layer, which uses the tanh activation function.

To compile the model, the Mean Absolute Error (MAE) loss function is used, with MAE also used as the evaluation metric. The model is optimized during training using the Adam optimizer, with a learning rate of 0.000001.

The model underwent training using 5900 images, employing a batch size of 128, for a total of 500 epochs. During this training process, it achieved convergence with a mean absolute error of 0.01 on the training data and 0.03 on the validation data. The dataset ranges from a minimum value of -1.5 to a maximum of 1.5, resulting in a span of 3. Given an error of 0.03 on this range, the precision can be approximated to 0.99 or 99%.

```
def construct_model():
    model = Sequential()
    model.add(Conv2D(24, (5, 5), strides=(2, 2), activation='relu', input_shape=(100, 100, 3)))
    model.add(Conv2D(36, (5, 5), strides=(2, 2), activation='relu'))
    model.add(Conv2D(48, (5, 5), strides=(2, 2), activation='relu'))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(Conv2D(64, (3, 3), activation='relu'))
    model.add(Flatten())
    model.add(Dense(1164, activation='relu'))
    model.add(Dense(100, activation='relu'))
    model.add(Dense(50, activation='relu'))
    model.add(Dense(10, activation='relu'))
    model.add(Dense(1, activation='tanh'))
    optimizer = Adam(lr=1e-5)
    model.compile(loss='mae', metrics=['mae'], optimizer = optimizer)
    return model
```

Fig. 11: The model architecture

Step 4 – Model Conversion to ONNX Format: The model obtained in the previous step is saved to disk as a ".h5" file. However, since the libraries utilized for converting the model to ONNX format require Tensorflow 2.12.0, which is incompatible with CUDA 11.2.2 (the latest version supporting the NVIDIA GeForce 1050Ti GPU), it was necessary to switch to a different Python kernel. The provided ".h5" model is then imported into this new environment. Using the tf2onnx library, the model is converted to ONNX format and subsequently saved to disk. At this point it can be used by Unreal Engine 5, as detailed in the previous chapter.

3 Future developments

Reflecting upon the future extensions of this research, several pivotal aspects emerge that can elevate the functionality and efficiency of our autonomous driving simulation.

Primarily, the implementation of traffic sign recognition capabilities forms an imminent objective. Creating this functionality would bring about a remarkable advancement in the environmental comprehension and decision-making faculties of the vehicle.

Subsequently, the creation of object detection and collision evasion systems form another key objective. While the existing design effectively adheres to lane demarcations, identifying and responding appropriately to dynamic entities in the environment - such as other vehicles, pedestrians, or unexpected obstructions - remains a critical component of an autonomous navigation system. The integration of a vision-based detection mechanism, in conjunction with radar and lidar systems, could provide a robust solution for this challenge.

Moreover, the incorporation of intersection management capabilities would introduce a significant degree of sophistication to our system. Navigating intersections is a complex endeavor as it demands the recognition and appropriate reaction to traffic signs, vehicular movement, and pedestrian activities. This will call for advancements not merely in sign recognition, but also in prediction and decision-making capacities of our artificial intelligence mechanism.

Furthermore, the exploration of unsupervised learning methodologies is an intriguing future trajectory. The present system predominantly relies on supervised learning, demanding extensive annotated data. Through the application of unsupervised learning techniques, we could utilize vast quantities of unlabeled data and potentially uncover hidden patterns and dependencies that remain elusive in the supervised context.

Finally, it is crucial to consider the development of an intuitive user interface for the system. This would offer users the facility to monitor and control certain aspects of the navigation

process, bridging the gap between intricate AI technologies and everyday users, and thereby enhancing the system's user-friendliness and accessibility.

4 Conclusion

We are firm in our belief that the autonomous navigation system outlined in this study has considerable potential and shines a beacon towards what the future holds in this domain. Presently, even though the system is in its prototype stage and the augmentation proposed in the future directions will undoubtedly enhance its capabilities, the current progress is significant.

Furthermore, we perceive this project as a stride forward towards inclusivity. It aims to address challenges that may seem trivial to in paving the way for technological advancement and accessibility. The system is a testament to the necessity of developing software that creates new experiences and simplifies tasks for everyone, irrespective of their technological acumen or physical abilities. This is more than just progress; it's a step towards a more inclusive and empathetic technological future.

In conclusion, while there is still a journey ahead in terms of perfecting and extending the system's capabilities, we are satisfied with the progress achieved so far. The foundation is sturdy, the direction is clear, and with continued effort and resourcefulness, we are confident that this system has the potential to make significant contributions to the field of autonomous navigation and AI-driven mobility. The insights gathered through this project will not only pave the way for more sophisticated autonomous navigation systems but will also serve to inspire innovation and inclusivity in AI technology and beyond.

Acknowledgement: This work was supervised by Lecturer PhD. Ralf Fabian Phd., from “Lucian Blaga” University of Sibiu.

References

[1] T. Ogata, Y. Yamamoto, Y. Ueno, and Y. Nakamura, *ESPnet-ONNX: Bridging a Gap Between Research and Production*, Proceedings of the 27th International Conference on Artificial Intelligence (IJCAI), Berlin Heildelberg New York, Springer Verlag, 1993.
[2] Almási, P., Moni, R., and Gyires-Tóth, B. *Robust Reinforcement Learning-based Autonomous Driving Agent for Simulation and Real World.* TSC, pp. 1063-1068., 2020
[3] F. Cai, T. Chen, A. Rus, and D. E. Koditschek. *Autonomous Driving in Simulated Environments Using Reinforcement Learning.* IEEE Robotics and Automation Letters, Vol. 2, No. 3, 2017.
[4] F. Li, Z. Wang, and L. Wang, *PID Controller Design: A Survey of Recent Advances*, IEEE Transactions on Industrial Electronics, vol. 61, no. 6, pp. 4396-4405, 2014.
[5] Yang Yu., *Implementation of the Pure Pursuit Path Tracking Algorithm*. Carnegie Mellon University, 2011.
[6] Youtube, *How to create your MAE Oak Forest under 15 minutes! | Unreal Engine 4*, [Online]. Available: https://www.youtube.com/watch?v=VgV2Dj6MRuw&feature=youtu.be
[7] Unreal documentation, *Using vehicles*, [Online]. Available: https://docs.unrealengine.com/5.0/en-US/how-to-set-up-vehicles-in-unreal-engine/

Cristian Nicolae CINDEA
Lucian Blaga University of Sibiu
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: cristi21122001@gmail.com

Stephanie FISCA
Lucian Blaga University of Sibiu
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: stephanie.fisca@ulbsibiu.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Ensemble Deep Learning for Tobacco Crop and Weed Detection

Zeynep Dilan Daşkın

Abstract

In recent years, agricultural farming has focused more on producing high-quality crops to reduce the waste of overall production. Scientists began to research and apply new techniques, such as artificial intelligence, to meet the massive demand for food as the population increased. In this study, the author has focused on improving the performance of an already-existed Convolutional Neural Network model by using Ensemble Technique. A model called AlexNet was first applied to a dataset to achieve this goal. Later, the same network was ensembled and tested with the same data. At the end of the study, the model evaluation has been discussed and compared according to the prediction accuracy.

1 Introduction

As the population grows, the need for high-quality plants and crops in agricultural organizations has gained more importance proportionately. Companies have started to invest more in finding new ways to improve the quality and safety of plant health to protect crops in order to meet this demand. The introduction of modern technology to the agriculture sector has shown some promising results regarding the yield and quality of crops, fruits, and vegetables (He et al. 2021; Fuglie et al. 2019; De Clercq, Vats, and Biel 2018).

One of the best ways to accurately increase crop production is to apply various machine learning and vision-based computer techniques to detect the crops in question. Currently, existing computer vision-based weed identification study methods are divided into two main parts: traditional and deep learning. Convolutional Neural Networks (CNNs) are one of the most used deep learning methods for vision-based analysis and classification. Although conventional methods have proven their effect on the field, more advanced methods such as CNNs have shown the most optimized ways of what can be done better. CNNs can be applied to several image-processing techniques, such as segmentation, detection, and classification of an object within an image or a video.

This study focuses on increasing an already existing CNN model by using the Ensemble method to distinguish tobacco crops from weeds. The dataset used in this work has been gathered manually in real-life conditions; therefore, another challenge was to minimize the effects of the environmental complexities and conditions.

This paper is organized as follows: Section 1 introduces the proposed study and the literature review. Section 2 presents the dataset and reviews the two frameworks and methods used. The benchmarks are discussed in Section 3. The results obtained after applying the proposed models are

analyzed and compared in Section 4. Section 5 describes each step followed throughout this research work, and finally, Section 6 concludes the proposed work and discusses possible future directions.

2 Methodology

2.1 Dataset Acquisition

To obtain more realistic results, an image dataset called TobSet has been used in this study which consists of 3000 tobacco and 1000 weed images taken within two months in different growth stages, varying lighting and weather conditions on the actual fields in Swabi, Khyber Pakhtunkhwa, Pakistan (34° 09′ 07.3″ N 72 21′ 36.2″ E). The images were captured with the resolution of 640 x 480 by using a 13-megapixel color digital camera possessing a CMOS-image sensor (IMX258 Exmor RS by Sony, Japan), 28 mm focal length, 65.4° horizontal FOV, and 51.4° vertical FOV (Alam, M. S. et al., 2022)

Data augmentation has been one of the crucial steps in image processing to increase the diversity of the training data. Augmented data can be obtained through various transformations (such as resizing, rotating and flipping, etc.) of the original dataset. It is mainly applied to prevent problems such as overfitting and helps to obtain more robust results. Different augmentation techniques have been applied to the weed dataset using Image Data Generator; the images were flipped, zoomed, skewed, sheered, and rotated. Finally, the 1000 original grass images were expanded to 3000 images so that the number of weed and tobacco images was equalized.

2.2 Convolutional Neural Network Based Classification

Different machine learning methods have revolutionized and dominated the image processing field since the development of artificial intelligence algorithms to solve various problems such as segmentation, detection, and classification. CNNs are one of the most effective ways of the recently developed machine learning techniques to capture, detect, and predict objects in images and videos.

This study uses an already existing CNN model, AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), to detect tobacco crops. AlexNet won the ImageNet Large-Scale Visual Recognition Challenge in September 2012, classifying 1.2 million high-resolution images. It consists of five convolutional layers, three fully connected layers, and one SoftMax layer, as shown in Figure 1.
For this research, the input image size has been defined as 256 x 256 x 3 pixels. ReLu has been used as an activation function for all convolutional layers, and the dropout value is applied as 0.4.

2.3 Ensemble Method

Ensemble method has been developed to solve the high variance of the CNNs, based on the fact that they are mainly nonlinear models. It does so by training and predicting multiple models instead of one and combining the outcomes to achieve the collective performance of the proposed models. By reducing the variance of the results, the generalization of the errors aimed to be reduced as well. Different Ensembling learning methods are used for CNN models, such as Stacking, Blending, Bagging, and Boosting. For this study, the Weighted Average Ensemble method has been used, which improves the accuracy of prediction by combining the contribution of each model by weighting proportionally according to its capability or skill. This work utilizes the Ensemble models using two AlexNet networks by uniformly averaging their weights. The Ensemble model architecture is presented in Figure 2.

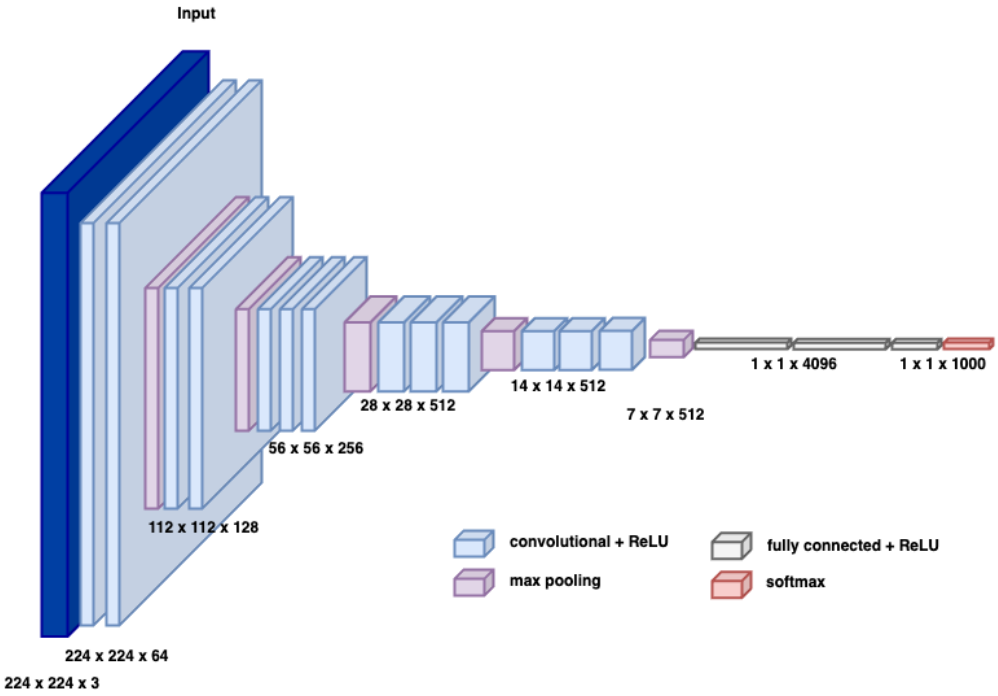


Figure 1. Architecture of AlexNet

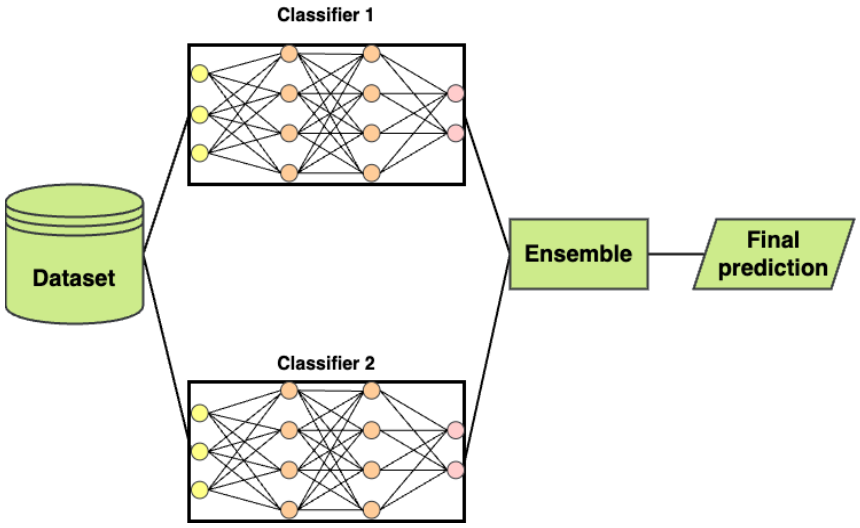


Figure 2. Ensemble Model Architecture

3 Comparison Criteria

The Ensemble model has been evaluated by considering the following performance matrices: accuracy, loss, specificity, recall, precision, and F1-score. For this, a Confusion Matrix has been utilized to analyze the classification algorithms further. This matrix measures how the performance of the model is successful at predicting data that belong to various classes. For this, four criteria are being used: True Negative (TN), True Positive (TP), False Negative (FN), and False Positive (FP). TN represents the number of negative examples classified negatively. TP stands for the number of

samples classified positively. Similarly, FN and FP represent the inaccurate classification of the given data.

3.1 Accuracy and Loss

Accuracy and loss values are the primary evaluation matrices of any CNN model. Accuracy represents the number of errors the model receives, and loss is the measure between the actual values and the predictions the model has made. For accuracy, the main goal is always trying to get close to 1, whereas, for loss, it is precisely the opposite.

3.2 Specificity, Recall, Precision, and F1-score

Although accuracy and loss are the two main bases for measuring the performance analysis of a CNN model, they are not enough to analyze alone. Four other measures are also being introduced.

Specificity is the ratio between true classified as negative to how our data was negative.

$$Specificity=TN/(TN+FP) \quad (1)$$

Recall helps us to understand how well the system labeled the true positives as actual positives.

$$Recall=TP/(TP+FN) \quad (2)$$

Precision is being used to determine if the predicted positives are labeled correctly.

$$Precision=TP/(TP+FP) \quad (3)$$

Finally, F1 Score combines two decision parameters, precision and recall scores, to correctly calculate how positive predictions are labeled.

$$F!\text{-}Score=2\ x\ Recall\ x\ Precision\ /\ (Recall+Precision) \quad (4)$$

4 Results & Discussion

To analyze the performance analysis of the models, the epoch numbers and batch sizes were kept constant at 20 and 25, respectively. The batches consisted of samples from each class. After generating and augmenting the dataset with weeds and tobacco crops, the training, testing, and validating sets are constructed with a ratio of around 0.15 (900 images), 0.2 (1200 images), and 0.65 (3900 images). A learning rate annealer has been used to boost the model's performance and reduce the computation time. This annealer helps to adjust the learning rate during the training process.

4.1 Comparison of AlexNet and Ensemble Model

The accuracy and loss values obtained for training and validation have been provided in Table 1 for AlexNet and Ensemble models. As can be deduced from the results, both models have predicted the data with high accuracy of 98% and 96% for training and 80% and 93% for validation, respectively. As for their loss values, it was expected that the Ensemble model would outperform AlexNet. It is apparent that training loss has resulted in as 0.03 for AlexNet and 0.01 for the Ensemble model. Their validation results are 0.1592 and 0.05, accordingly.

From the accuracy result of the AlexNet model, it can be observed that although the training accuracy is pretty high (96%), the validation accuracy is less than expected (80%). This means the model is overfitting, which means that the model does not perform well in the training data. By this, we can infer that the model cannot perform precisely while predicting and classifying the labels of the dataset. Applying the Ensemble method not only improved the performance of the single AlexNet model but also reduced the overfitting problem.

Table 1. Accuracy and Loss evaluated for 20 epochs

Performance Measures / Classifiers	Accuracy		Loss	
	Training	Validation	Training	Validation
AlexNet	0.9607	0.8081	0.0374	0.1592
Ensemble Model	0.9844	0. 9341	0. 0118	0.0540

After receiving the discussed results for accuracy and loss values, it was anticipated that the specificity, recall, precision, and F1-score values of the Ensemble model would be higher than the AlexNet model. As can be deduced by Table 2, Ensemble Model has overperformed AlexNet by 6% specificity, 5% recall, 11% precision, and 10% F1-score difference.

Table 2. Precision, Recall, F1-score, and Specificity for 20 epochs

Classifiers / Performance Measures	AlexNet	Ensemble Model
Specificity	0.87	0.9371
Recall	0.87	0.92
Precision	0.8201	0.9323
F1-Score	0.8156	0.9202

The results validate that the Ensemble method produces higher and more enhanced performance when compared with a single modality. Applying this network also reduced the overfitting and generated a smoother model.

Computation time is also another critical issue when it comes to CNNs. This paper calculates the computation time for training and testing images to validate the model performances. For a single AlexNet model, it took 3 minutes to complete the entire progress, while for the Ensemble model, it was almost twice of the first model with 6 minutes. This result was expected since our Ensemble model structure is more complicated than AlexNet; nevertheless, it can be said that the execution time for both models is optimal.

5 Program Code

The calculations were conducted on Jupyter Notebook and the Keras and TensorFlow deep learning development framework. During this research, the below steps were followed.

- Building a Dataset: Generating a database called TobSet from images of tobacco plants and weeds. In this stage, different growing stages of tobacco plants and weeds have been procured from various angles, under varying lighting and weather conditions.
- Data Pre-processing: Preparing the raw data to obtain reliable, accurate, and solid outcomes. This process enables the classifier to make more accurate predictions by allowing it to learn from a wider range of variations in the data. This step involves re-scaling the pixel values of the input images to a certain range (normalization), resizing, data splitting, and data augmentation.
- Image Segmentation: Breaking each image into its individual segments to decrease the complexity of the image. It involves several transformations, such as rotation, translation, flip, and changes in brightness or contrast.

- Feature Extraction: Extracting features of each images using convolutional layers to screen and capture key patterns and features from the raw input data.
- Classification: Classifying each pixel into its respective classes in order to assign and a label or category to an input image.
- Performance Analysis: Calculating the performance metrics, evaluating the models, and comparing the obtained results.

6 Conclusion

This paper aimed to improve the performance of an existing CNN model by using the Ensemble model for detecting and classifying tobacco plants from weeds. The proposed Ensemble model uses the Weighted Average Ensemble technique, which consists of two AlexNet models and incorporates the estimation of each model with equal weights. Loss, accuracy, precision, recall, F1-score, specificity, and computation time are used to extract and compare the performance of the results. It is concluded that the ensemble method outperforms the single AlexNet model in terms of all comparison metrics and reduces the problem of overfitting. In the future, we will apply different Ensemble techniques, such as stacking or bagging, to further investigate whether performance can be improved.

Acknowledgment: This work was supervised by Assoc. Prof. Dr. *Muhammad Umer KHAN* from Atılım University and Prof. Dr. *Alina Bărbulescu* from Transilvania University of Brasov, Romania. The TobSet dataset was curated by *Muhammad Shahab* and used in this study only for educational purposes.

References

[1] M.S. Alam et al. Tobset: A new tobacco crop and weeds image dataset and its utilization for vision-based spraying by agricultural robots. *Appl. Sci.*, 12(3): 1308, 2022.
[2] J. Baeten et al. *Autonomous fruit picking machine: A robotic apple harvester*. In: C. Laugier, R. Siegwart, R (eds). Springer Tracts in Advanced Robotics, vol 42. Springer, Berlin, Heidelberg, pp. 531–539, 2008.
[3] Krizhevsky, Alex, Sutskever, Ilya & Hinton, Geoffrey. *ImageNet Classification with Deep Convolutional Neural Networks.* Neural Information Processing Systems. 2012.
[4] Fuglie, K. et al. *Harvesting prosperity: Technology and productivity growth in agriculture*. World Bank Publications, 2019.
[5] T. van Klompenburg, K. Ayalew, C. Catal, Crop yield prediction using machine learning: A systematic literature review, *Comput. Elect. Agr.*, 177: 105709, 2020.
[6] S. Lawrence, C. L. Giles, *Overfitting and neural networks: conjugate gradient and backpropagation, Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks.* IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Como, Italy, 2000, vol.1, 2000, pp. 114-119.
[7] C. Szegedy et al. Going deeper with convolutions, 2015 IEEE Conference Proceedings on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1-9.
[8] J Iqbal, A. Rauf, Tobacco Revenue and Political Economy of Khyber Pakhtunkhwa, *FWU J. Soc. Sci.*, 15(1): 11-25, 2021.
[9] T. Talaviya et al. Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides, *Artif. Intell. Agr.*, 4: 58-73, 2020
[10] Z. Wang et al. Review of smart robots for fruit and vegetable picking in agriculture, *Int. J. Agr. Biol. Eng.,*15(1): 33-54, 2022.
[11] A. Bechar, C. Vigneault, Agricultural robots for field operations: Concepts and components, *Biosyst. Eng.*, 149; 94-111, 2016.
[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556, 2014.

[13] M. Shahhosseini et al. Corn yield prediction with ensemble CNN-DNN, *Front.Plant Sci.*, 12: 709008, 2021.

Zeynep Dilan DAŞKIN
Atılım University
Department of Mechatronics Engineering
Kizilcasar Mahallesi, 1184.
Cad No:13, 06830 Incek,
Golbasi/Ankara
TURKEY
E-mail: daskin.zeynepdilan@gmail.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Advanced Vision-Guided Autonomous Navigation Robot

Stephanie Fîșcă, Cristian-Nicolae Cîndea

Abstract

This article presents an advanced autonomous navigation robot designed as a four-wheeled vehicle, powered by a Raspberry Pi, primarily focused on enabling smooth and intelligent navigation through a test environment. The proposed cyber-physical system uses a camera for environmental perception and employs computer vision techniques for lane detection and traffic sign recognition. To generate a comprehensive understanding of the environment, the system utilizes computer-vision based techniques. Moreover, it incorporates a deep learning generated model trained on a diverse dataset of traffic signs, enabling the robot to accurately classify and interpret detected signs in real-time. By recognizing and reacting to these, the robot can effectively follow traffic rules, and adjust its behavior accordingly. This will ensure safe and reliable navigation in various situations. Our research contributes to the development of cost-effective, autonomous navigation and sign recognition capabilities, paving the way for potential applications in urban transportation, last-mile delivery, and intelligent mobility systems.

1 Introduction

In today's rapidly evolving world, autonomous navigation systems have gained significant interest due to their potential applications in diverse domains such as urban transportation, last-mile delivery services, and intelligent mobility systems. Several key works, including those by Thrun et al. (2006) on Simultaneous Localization and Mapping (SLAM) [1], Fox et al. (1998) [2] on the Monte Carlo Localization (MCL) algorithm, and the influential work of Karaman and Frazzoli (2011) on sampling-based motion planning [3], have shaped the trajectory of autonomous navigation research. Additionally, the advancement of deep learning techniques, particularly in the form of convolutional neural networks (CNNs) and recurrent neural networks (RNNs), as demonstrated by Bojarski et al. (2016) [4] in the development of self-driving cars, has opened up new avenues for autonomous navigation. To address the rising need for efficient and reliable autonomous navigation, we propose an advanced four-wheeled robot, powered by a Raspberry Pi, a small single-board computer (SBC) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom [5]. This robot leverages computer vision techniques and deep learning algorithms for accurate environment perception and intelligent navigation.

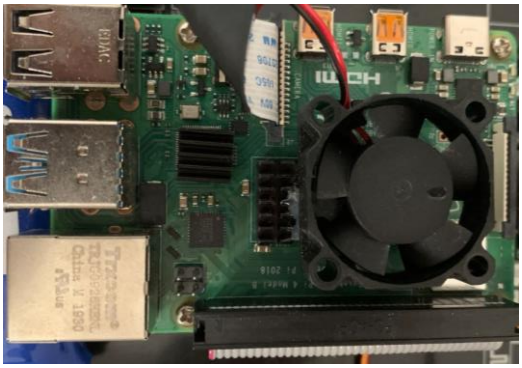


Fig. 1: Raspberry Pi 4B

Presently, various automotive enterprises such as Tesla, Inc. [6], General Motors (GM) [7] are fervently engrossed in the pursuit of manufacturing their own fleet of autonomous vehicles. In anticipation of a not-so-distant future where our urban landscapes are seamlessly traversed by self-driving cars, we have embarked upon an ambitious undertaking: the realization of our very own self-driving vehicle, integrating cutting-edge advancements in Artificial Intelligence and Computer Vision.

The project's title, "Advanced Vision-Guided Autonomous Navigation Robot," encapsulates the multifaceted technologies harnessed to bring this visionary concept to life. By utilizing intelligent algorithms and sophisticated visual perception, the goal is to create a marvel of engineering: a vehicle that operates autonomously, unencumbered by human intervention.

The two key technologies utilized in this project are focused on perception and navigation: computer vision and deep learning.

1.1 Computer Vision and Deep Learning

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information, whilst Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. [8]

These neural networks attempt to simulate the behaviour of the human brain—albeit far from matching its ability — allowing it to “learn” from large amounts of data [9]. By leveraging the Raspberry Pi's processing capabilities, the robot uses computer vision techniques to interpret the images captured by the camera, enabling it to understand and navigate its environment. Concurrently, deep learning algorithms are used to recognize and classify traffic signs, which allows the robot to adjust its behavior accordingly and follow traffic rules.

In summary, our work represents a significant advancement in the quest for autonomous navigation. It underscores the crucial role of artificial intelligence and robotics in shaping the future of transportation, delivery services, and mobility systems.

2 Robot Design and System Overview

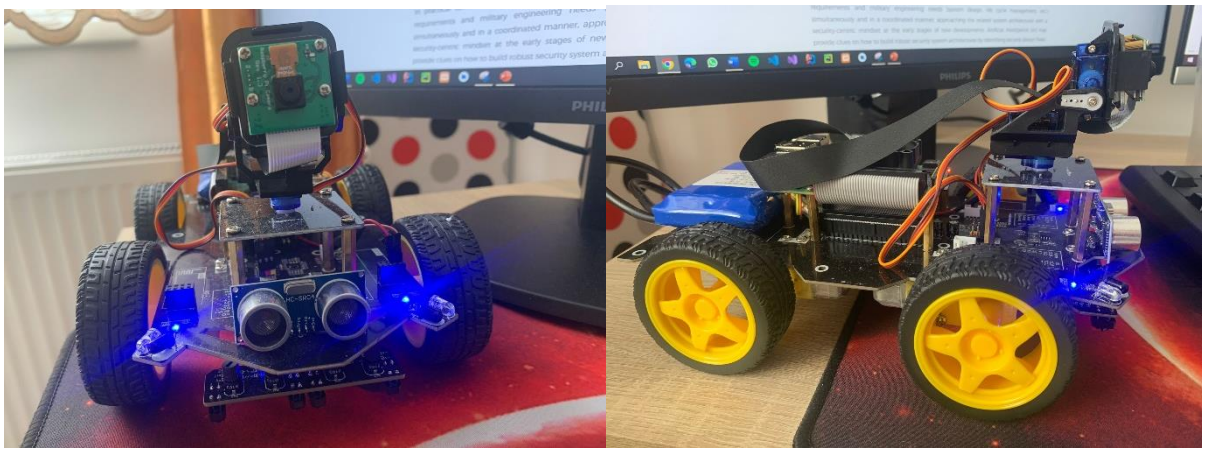


Fig. 2: Robot's Architectural Design

The architecture of an autonomous navigation robot is intricate and requires a variety of seamlessly functioning components. The Raspberry Pi 4B development board serves as a critical component in the architecture of such a UGV system. This powerful board provides the necessary processing power to run the robot's software and control its various components.

A crucial element of a self-driving robot is its chassis, which serves as the foundation for the entire robot. In this case, a multifunctional expansion board is employed as the chassis. This expansion board provides ample space for the robot's different components and enables easy expansion and customization.

Four TT DC motors are utilized to drive the robot's movement, while a high-definition camera ensures visual input. This camera is essential for the robot's ability to navigate its surroundings and detect traffic signs. Additionally, infrared and ultrasonic sensors are integrated into the robot, with future aspirations of aiding in navigation and collision avoidance.

The STM8 microcontroller plays a key role in the architecture of a self-driving robot. This microcontroller communicates with the Raspberry Pi through IIC and controls the TB6612 chip, which, in turn, actuates the motors. This communication enables precise control of the robot's movement and ensures that it can navigate its environment safely and efficiently.

The intricate design of an autonomous navigation robot necessitates a harmonious integration of its components. Anchored by the robust Raspberry Pi 4B development board, this intelligent system harnesses the boundless potential of advanced technology to pave the way for a future where autonomous robots traverse our world, harmoniously merging precision, perception, and safety.

2.1 PID Control System

In the realm of self-driving robots, where precision and accuracy are paramount, the implementation of sophisticated control systems is crucial. Among the many control techniques available, one stands out for its effectiveness and versatility: Proportional-Integral-Derivative (PID) control [10]. PID control is a feedback mechanism widely used in engineering and robotics to regulate a system's behavior based on feedback from sensors. It operates by continuously measuring the error between the desired setpoint and the actual output, and then adjusting the control input accordingly to minimize the error. PID control is particularly effective in scenarios where precise and stable control is required.

2.1.1 Proportional (P) Control

The proportional component generates a control response proportional to the error. It contributes to reducing the error by applying a corrective action based on the current discrepancy between the desired setpoint and the actual output. A higher proportional gain results in a more significant correction, albeit with the potential for overshoot or oscillation.

$$P = K_p * e$$

Where P represents the value of the proportional term, Kp is the proportional constant, and e is the control error (the difference between the desired reference, r, and the measured value, y).

2.1.2 Integral (I) Control

The integral component accumulates the error over time and integrates it to provide long-term correction. It is responsible for eliminating steady-state errors that may persist due to system biases or external disturbances. The integral gain amplifies the integral response, but excessive gain can lead to instability or oscillation.

$$I = Ti * \int e(t)dt$$

Where I represents the value of the integral term, Ti is the integral time constant, and e(t) is the control error over time (the difference between the desired reference, r, and the measured value, y).

2.1.3 Derivative (D) Control

The derivative component predicts the future trend of the error by calculating the rate of change. By considering the rate of error change, the derivative component provides a damping effect, reducing overshoot and oscillation. The derivative gain determines the strength of this damping action.

$$D = Td * \frac{de}{dt}$$

Where D represents the value of the derivative term, Td is the derivative time constant, and de/dt is the rate of change of the control error over time.

The output of the PID controller is the sum of the contributions from the three factors, each multiplied by their respective gain values:

Output = (P * Proportional Gain) + (I * Integral Gain) + (D * Derivative Gain)

3 Computer Vision for Lane Detection

Computer Vision, the first primary component of our system, plays a crucial role in enabling the robot to detect and follow lanes effectively [11]. As the robot navigates through its environment, the mounted camera operates ceaselessly, capturing real-time images of the path ahead. The camera focuses predominantly on the ground immediately in front of the robot, as it is this area that contains the lane markings and other information critical for navigation.

Fig. 3: Image Processing for Lane Detection

Once these images are captured, they are processed by a sophisticated computer vision algorithm. At the heart of the robot's perception lies the fusion of color and edge detection algorithms, working in tandem to decode the visual cues encoded within the captured images. This task can be accomplished through a variety of techniques. Thus, for this project we chose a common method called 'edge detection', where the algorithm identifies abrupt changes in pixel intensity that typically correspond to lane markings.

These algorithms serve as gatekeepers of knowledge, identifying the distinctive hues and gradients that delineate the road lines, while also determining the robot's spatial relationship to them.

Once the lane markings have been detected, the algorithm calculates the robot's position and orientation relative to the lanes. This involves determining the robot's lateral position within the lane (i.e., how far it is from the center of the lane) and its angular orientation (i.e., whether it is aligned with the lane or if it is veering to one side).

By leveraging computer vision in this way, the robot can accurately detect lanes in a variety of conditions and maintain an appropriate course. This enables the robot to navigate its environment safely and efficiently, closely mirroring the behaviours of a human driver.

The intricate interplay of image processing and cascade classifiers enables the robot to perceive and interpret its visual surroundings, mirroring the capabilities of a discerning human eye. Through the lens of its camera, the robot captures a continuous stream of images, which serve as the foundation for its navigation prowess.

By using sophisticated image processing algorithms, the robot systematically analyses each frame, extracting key features and discerning the presence of road lines amidst the visual clutter. This intricate dance between algorithms yields precise line detection and tracking, enabling the robot to traverse the simulated traffic lane with unwavering precision.

3.1 XML file generation for Traffic Sign Detection

The XML file utilized for the detection of stop signs implements a sophisticated image processing routine to identify and extract these signs from an input image. The algorithm employs various

techniques, including color normalization, histogram analysis, thresholding, morphological operations, contour detection, and extraction of regions of interest (ROI) [12].

- **Image Loading and Color Normalization:** The training begins by loading an image specified by its file path (Figure 4).

Fig. 4: Input image with a stop sign used for training

To enhance color information, we perform color normalization on the red channel of the image. The red channel value of each pixel is divided by the sum of the red, green, and blue channel values, and the result is stored in an array.

- **Histogram Analysis and Thresholding**: Next, the histogram is calculated of the normalized red channel. It represents the distribution of pixel intensities in the red channel. The histogram is then inverted to facilitate further analysis, and by summing the values of the inverse histogram, the code obtains the total number of pixels. Subsequently, an adjustable threshold is determined based on a percentage (0.2% in this case) of the total number of pixels, where pixels above this threshold are considered part of the traffic signs. The technique of thresholding the normalized red channel is applied, resulting in a binary image (thresholded) where the foreground (traffic signs) is white, and the background is black.

- **Morphological Operations and Contour Detection:** To refine the binary image and remove noise, morphological operations are applied such as erosion to erode the white regions and dilation to slightly expand these regions. Then, a median blur effect is applied to further smooth the image. After these operations, the code detects the contours in the binary image, which represent continuous curves along the boundaries of the white regions.

- **Bounding Rectangles and ROI Extraction:** The code traverses the detected contours and obtains the bounding rectangles. For each contour, a rectangle is drawn on the original image to visualize the region of interest (ROI) corresponding to the traffic sign. The coordinates of the ROI are used to extract the corresponding region from the original image, and the extracted ROIs are stored in the ROI list.

- **Visualization and Result:** To visualize the results, the code displays the histogram of the normalized red channel.

Fig. 5: Applying trained model on input image for stop sign detection

The final image is displayed as shown in *Figure 5*, where the highlighted region indicates the detected traffic sign, displayed in green.

4 Deep Learning for Traffic Sign Recognition

Fig. 6: Stop Sign Recognition

Deep learning [13], a subfield of artificial intelligence, has shown tremendous success in various domains, especially those requiring high levels of perceptual understanding, such as image and speech recognition. In our system, we utilize a deep learning model to further boost the robot's navigation capabilities, specifically in recognizing and interpreting traffic signs.

Our chosen model is a cascade-classifier, a type of deep learning model particularly effective at handling image data. Cascade-classifiers are adept at identifying hierarchical patterns in images, making them well-suited for tasks like traffic sign recognition.

The artistry unfolds as the robot applies cascaded classifiers to identify stop signs, strategically placed along the route. By training the classifiers on a wealth of stop sign examples, the robot develops an astute understanding of the shape, color, and textual elements unique to these traffic symbols. As the robot navigates, the camera captures images in real-time, feeding them through the classifiers, which promptly trigger alerts upon detecting a stop sign. This critical feature ensures the robot's compliance with traffic regulations and enhances its overall safety.

Thus, through the integration of cutting-edge image processing techniques and cascade classifiers, the robot gains an unparalleled visual acuity, navigating the simulated traffic lane with poise and dexterity. Armed with the ability to discern road lines and detect stop signs, the robot showcases its mastery of perception, contributing to the advancement of autonomous navigation in a world brimming with complex visual stimuli, being able to simulate a stop at the detection of stop signs for 5 seconds.

In essence, the integration of a deep learning model for traffic sign recognition provides the robot with a more nuanced understanding of its environment. This empowers it to interact with the environment in a more sophisticated and reliable manner, similar to a human driver.

4.1 Stop sign creation

The detection of the stop sign is performed on an unaugmented real-time image. The image in *figure 7* illustrates an actual case of stop sign detection, where the image itself is blurry and distorted due to the robot navigating on the track, resulting in camera movement. To enable the detection of a stop sign, it had to be constructed.

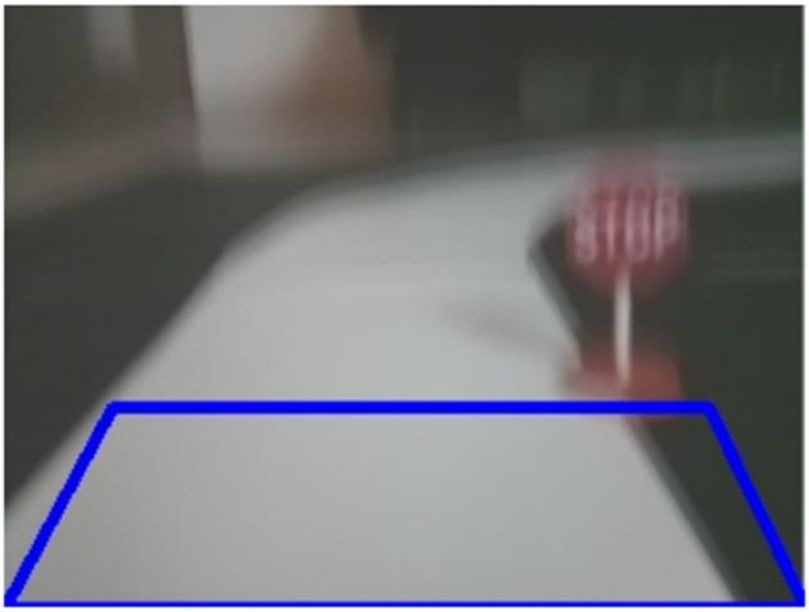


Fig. 7: Real time stop sign detection

To construct the stop sign, the SketchUp program in *Figure 8* was utilized for modelling an octagon, representing the desired shape of the stop sign, as well as for drawing the supporting base of the sign.

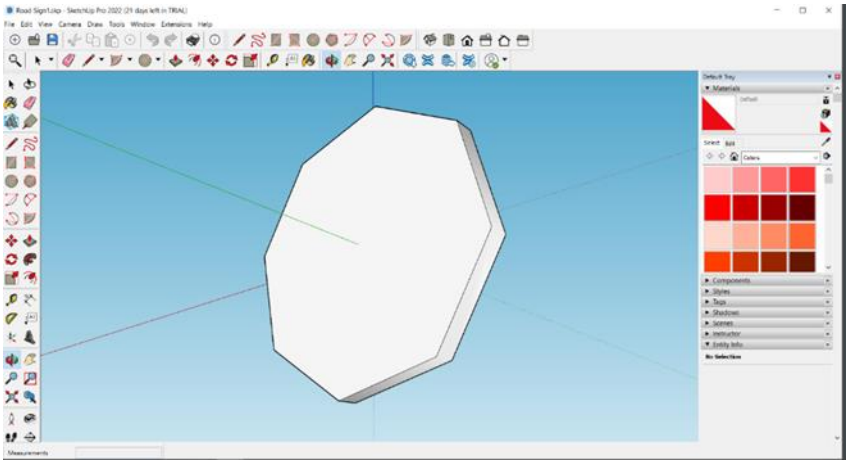


Fig. 8: Designing the stop sign in SketchUp

Once the components of the traffic sign were created, they were exported in STL format, which was then imported into the Simplify3D program. Simplify3D is a software tool used for 3D print preparation, allowing us to adjust printer settings and generate the necessary GCODE for printing the model. *Figure 9* displays the base of the stop sign in the Simplify3D program.

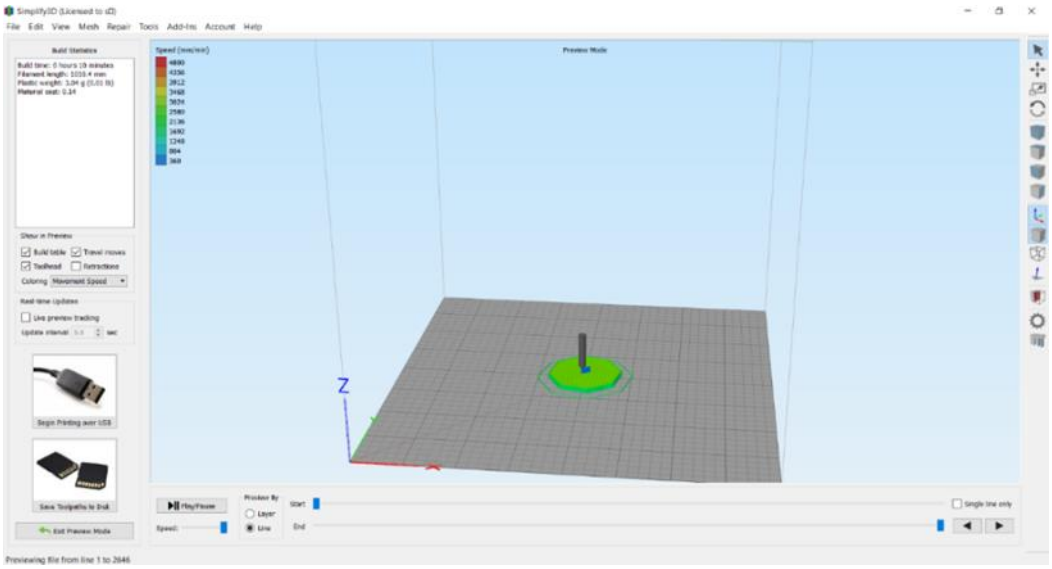


Fig. 9: Imported design in Simplify3D program

After verification and simulation, the necessary adjustments were made to generate the GCODE file, enabling the printing of the model created in the previous stages. This file contains instructions for the 3D printer, such as the printing head's movement speed and the amount of material to be extruded. Figure 10 demonstrates the printing of the components using an ANET A6 3D printer.

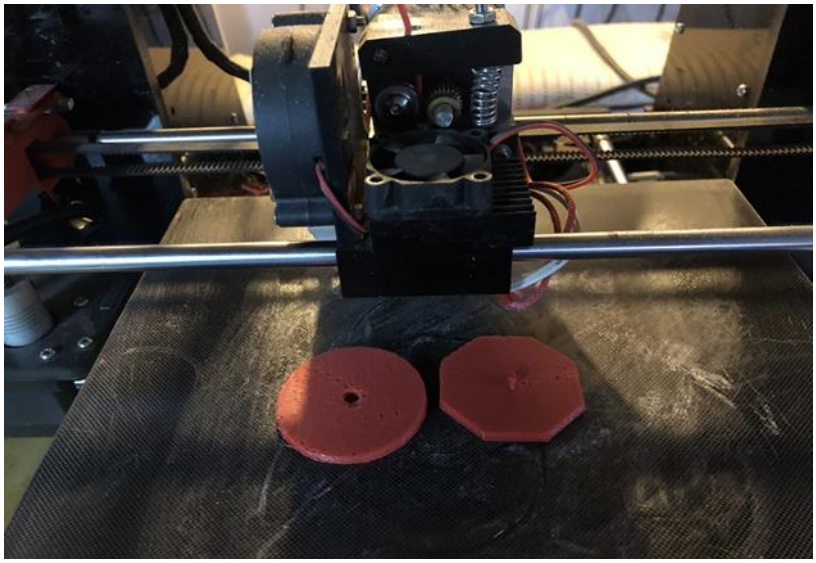


Fig. 10: Stop sign 3D printing process

5 Real-time navigation simulation

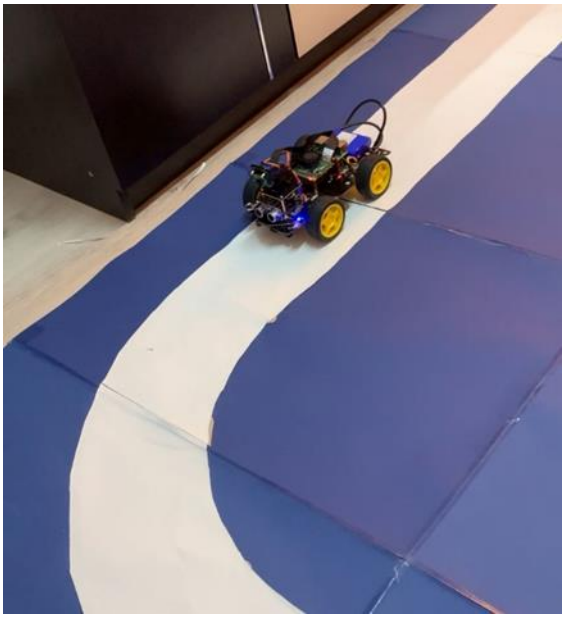


Fig. 11: Robot navigating in real-time

For the simulation, we constructed a platform using multiple dark-colored A2-sized papers as the base, which were taped together. The primary purpose of the platform was to ensure accurate lane detection and eliminate any issues that could arise from floor/parquet light reflection. On top of the base paper, we attached several A3-sized white sheets. The white strip serves as the path for the robot to follow.

One of the challenges encountered during the simulation of the robot's functionality was ensuring a stable internet connection for the Raspberry Pi. During the project presentation, we can utilize a separate router to ensure a reliable internet connection for the Raspberry Pi. This is

crucial as the robot requires a stable internet connection to communicate with other devices and services.

In addition to the separate router, I also ensured that the Raspberry Pi has the latest software updates and is correctly configured. This helped avoid any potential issues that could arise from outdated software or incorrect configurations. Furthermore, I carefully tested the robot to ensure its proper functioning and its ability to follow the white lane.

6 Results and Discussion

Upon reflecting on the development process of the autonomously navigating robot, I am proud to announce that many of the initially proposed objectives have been successfully achieved.

First and foremost, we successfully implemented an artificial intelligence-based lane guidance script, enabling the robot to stay within a predetermined lane. This script receives data from the camera and employs image processing techniques to identify lane markings and adjust the robot's direction accordingly. Thorough testing has demonstrated its capability to keep the robot on track while navigating both straight sections and curved portions of the road.

Secondly, we managed to develop and utilize a traffic sign detection module using a cascade classifier. This module utilizes computer vision techniques to identify traffic signs and their respective meanings. While our initial aim was to detect a broader range of traffic signs, the implementation of the cascade classifier allowed us to focus on detecting stop signs, a crucial traffic regulation. The module has been tested and has demonstrated high accuracy in stop sign detection.

Thirdly, we were able to create a robust platform on which the robot operates. This involved carefully selecting hardware components, utilizing sturdy materials, and fine-tuning software features to ensure optimal performance of the robot in the created environment. The platform has been thoroughly tested and proven its ability to support the autonomous navigation capabilities of the robot as well as the detection of traffic signs.

Lastly, we have successfully simulated a real-life driving scenario in which the vehicle must stay within their lane and adhere to traffic rules. This entailed integrating the lane guidance and traffic sign detection modules, as well as simulating a road environment. The simulation has been tested and demonstrated the robot's ability to navigate the surrounding environment while simultaneously adhering to traffic rules.

These positive outcomes underscore the significant potential of our autonomous navigation robot. However, it's important to acknowledge that the tests were conducted in a controlled environment. To further validate the robot's performance and reliability, future work will involve testing in more diverse and unpredictable environments. Nevertheless, our current findings demonstrate that a cost-effective, yet capable, autonomous navigation system is feasible. This opens numerous exciting opportunities for practical applications in urban transportation, last-mile delivery services, and intelligent mobility systems.

6 Conclusion and Future Work

For future work, we aim to enhance this smart system with autonomous navigation and traffic sign detection by utilizing ultrasonic and infrared sensors to simulate collision avoidance. This feature can be simulated by introducing an obstacle during the robot's movement, placed within the detection range of the sensors, triggering the command to stop the motors of the unmanned ground vehicle (UGV).

Overall, this feature will improve the simulation performance of a potential real-life collision, with the behaviour of the system integrated with the single-board computer resembling current technologies implemented in well-known vehicles.

Furthermore, to enhance traffic sign detection, I intend to experiment with the development of the cascade classifier to detect a wider range of traffic signs. For example, this includes detecting various speed limits, allowing for an increase or decrease in the robot's driving speed accordingly.

These features will be implemented and simulated for the dissertation project, our aim being to continue this project for the master's program within the Faculty of Science. The goal of these improvements is to create a more efficient and safer autonomous system capable of navigating traffic, avoiding collisions, and adhering to traffic laws. Ultimately, this system has the potential to revolutionize the field of autonomous technology and the simulation capabilities of real-life traffic scenarios, even within our own homes. Additionally, this work was supervised by Professor Fabian Ralf from the University of Sibiu, 'Lucian Blaga'.

In conclusion, our study has successfully demonstrated a cost-effective and intelligent solution for autonomous navigation, setting the stage for potential applications in various domains. This work contributes significantly to advancing technologies in autonomous vehicles and robotics. Future research will focus on enhancing the robot's capabilities, including obstacle detection and avoidance, and optimizing the system for real-world deployment. We envision our work to be a steppingstone for the broader adoption of autonomous navigation systems in urban transportation, last-mile delivery, and intelligent mobility systems.

Acknowledgement: This work was supervised by Lecturer PhD. Ralf Fabian Phd., from "Lucian Blaga" University of Sibiu.

References

[1] Wikipedia, *Raspberry Pi,* Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Raspberry_Pi.

[2] IBM, *What is Computer Vision?,* IBM, [Online]. Available: https://www.ibm.com/topics/computer-vision.

[3] IBM, *What is Deep Learning?,* IBM, [Online]. Available: https://www.ibm.com/topics/deep-learning.

[4] F. Dellaert, *Monte Carlo Localization for Mobile Robots,* [Online]. Available: https://www.ri.cmu.edu/pub_files/pub1/dellaert_frank_1999_2/dellaert_frank_1999_2.pdf.

[5] S. Thrun, *Simultaneous Localization and Mapping,* [Online]. Available: https://www.cs.columbia.edu/~allen/F19/NOTES/slam_paper.pdf.

[6] S. Karaman, *Sampling-based Algorithms for Optimal Motion Planning,* [Online]. Available: https://people.eecs.berkeley.edu/~pabbeel/cs287-fa19/optreadings/rrtstar.pdf.

[7] Tesla, *Tesla,* [Online]. Available: https://www.tesla.com/en_eu.

[8] G. Motors, *General Motors,* [Online]. Available: https://www.gm.com/.

[9] M. Bojarski. [Online]. Available: https://scholar.google.com/citations?user=5isUxCgAAAAJ&hl=en.

[10] Medium, *PID control,* [Online]. Available: https://medium.com/autonomous-robotics/pid-control-85596db59f35.

[11] IBM, *What is computer vision?,* [Online]. Available: https://www.ibm.com/topics/computer-vision.

[12] A. Vidhya, *Haar Cascade Algorithm with Object Detection,* [Online]. Available: https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/.

Stephanie FÎȘCĂ
Lucian Blaga University of Sibiu
Faculty of Sciences
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: stephanie.fisca@ulbsibiu.ro

Cristian-Nicolae CÎNDEA
Lucian Blaga University of Sibiu
Faculty of Sciences
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: Cristian-nicolae.cindea@ulbsibiu.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

The game theory behind a 2D action role-playing game

David-Stelian Hoka

Abstract

This article introduces a game-theoretic perspective to explore behavior in a 2D Action Role-Playing Game (ARPG). The primary objective is merging the timeless gameplay style of the Zelda series with contemporary gaming mechanics to create a richer, more challenging, and addictive gaming environment. The paper assists us to understand game theory by creating a 2D action role-playing game which demonstrates the successful application of game theory principles in the development of an ARPG game, providing an innovative and enjoyable gaming experience for players.

1 Introduction

The purpose of this project is to develop a 2D virtual environment encompassing a diverse range of subjects. These subjects include game theory, the properties and principles of object-oriented programming (OOP), design patterns/strategies, adherence to the SOLID principle [7], artificial intelligence [8], memory management within the constraints of the Java language [9], and the incorporation of features commonly found in classic 2D RPG action games [6].

The chosen theme of this article represents an idea inspired by the popular game "The Legend of Zelda" [5]. It is one of the most iconic and beloved game franchises of all time, having sold over 100 million copies worldwide.

The series is known for its captivating stories and immersive worlds that players can explore at their own pace.

The series is also known for its innovative gameplay mechanics, such as puzzle solving and dungeon crawling, as well as memorable boss fights.

This game had a significant influence on the video game industry and inspired many other games and game developers.

As a main objective is to merge the classic gameplay style of Zelda with modern game mechanics, providing players with a more intricate and immersive gaming environment. In order to perform this, I followed up to enhance the gameplay experience by introducing elements such as inventories, leveling systems, skill progression, and complex character interactions.

The author aimed to develop a robust inventory system that allows players to collect and manage various items, weapons, and tools throughout their journey. By implementing a leveling system, players will have the opportunity to develop their character's abilities and skills, offering a sense of progression and customization.

In addition, I strive to create a rich and dynamic world where players can interact with diverse non-player characters (NPCs). These NPCs will possess unique personalities, quests, and storylines, contributing to a more engaging and lifelike experience. By integrating advanced AI algorithms, I aim to make the behavior and responses of NPCs more realistic and adaptive, enhancing the overall immersion and storytelling aspect of the game.

Overall, my goal is to bridge the gap between the classic Zelda gameplay and the contemporary gaming landscape, providing you, as the player, with a captivating and multifaceted gaming experience that combines the beloved elements of the original franchise with the innovations of modern game design.

2 Theoretical perspectives

2.1 Application design

The application design architecture is designed to provide a robust and scalable platform to create a dynamic and engaging player experience. It uses a variety of techniques and technologies to achieve this goal, including a complex graphics system, sound rendering, and sophisticated AI systems. By combining these components in a cohesive and integrated way, the game offers players a rich and immersive gaming experience.

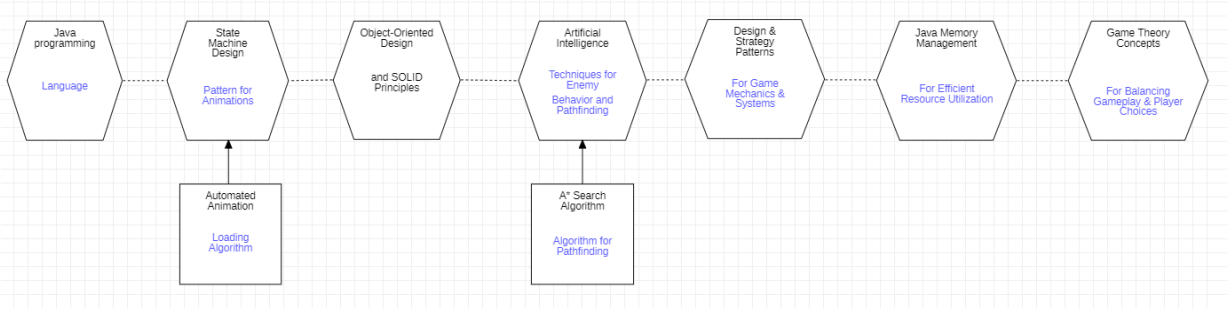


Fig. 1: Technologies & Techniques used

The game engine: is the core game software component that provides the functionality needed to create the game world, manage player inputs, and manage game mechanics. It also deals with graphics and sound rendering, physics simulation and other aspects. In order to perform a high quality game engine, The author studied and documented from the tutorial from [2]. Also, the 2D physics engine (the logic pattern) was inspired from the tutorial from [3], which presents 2D Physics concepts such as collisions between objects.

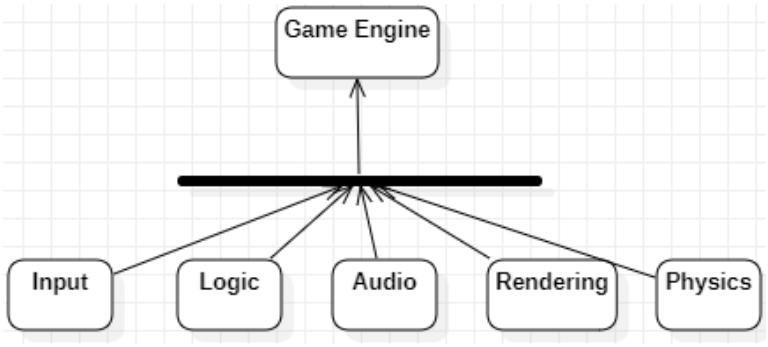


Fig. 2: The core game software component

The graphics and sound engine: is responsible for rendering the game world, characters and objects, as well as rendering sound effects and music. It uses various techniques such as sprite rendering, particle effects and 2D animation to create an attractive and immersive experience for the player [1].

AI system: controls the behavior of NPCs and monsters in the game, allowing them to move, attack, and interact with the player and the game world. It uses various techniques such as path-finding algorithms and behavior trees to create dynamic and immersive interactions between the player and the game world [10].

Game logic: the logic component of the game contains the rules and mechanisms that govern the behavior of the game world, including combat mechanics, inventory management, and mission systems. It also deals with player progression and character development, allowing the player to level up and acquire new skills and equipment as they progress through the game.

Data storage: the data storage component manages the persistent data used by the game, including the player profile, game saves, and configuration settings. It may use a database or file system to store and retrieve this data.

User interface: provides the player with an intuitive and friendly interface to interact with the game world. It includes elements such as menus, dialog boxes, and HUD displays.

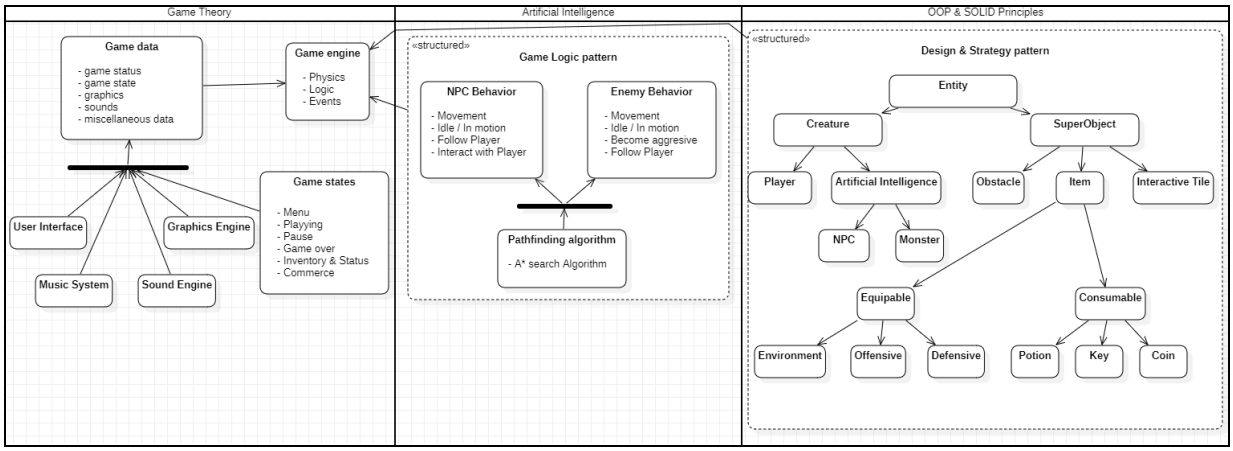


Fig. 3: Overview of Application Infrastructure

2.1.1 Technical Components of the Game

Java is a popular programming language for game development due to several key advantages it offers, which make it a suitable choice for implementing a game with the characteristics of a 2D action role-playing game:

- Compatibility across platforms.
- Memory management: Helps prevent memory leaks, which can cause crashes and performance issues.
- Object-oriented programming: Java's OOP programming features enable modular and scalable code, making it easy to add new features and modify existing ones. This can be particularly useful when implementing complex game systems such as path-finding algorithms or inventory management.

- Large developer community: There is a large community of developers with experience in the language. This can be a valuable resource for game developers who may need to troubleshoot issues or seek guidance on best practices.
- Third-Party Libraries and Frameworks: Java has a wide range of third-party libraries and frameworks available, many of which are designed specifically for game development.

The architecture of the project follows a well-structured object-oriented design, incorporating principles such as OOP (Object-Oriented Programming) and SOLID (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion). The hierarchy starts with the base class "Entity," from which two derived classes are formed: "Creature" and "SuperObject."

The "Creature" class represents entities capable of independent actions. It serves as the parent class for more specific entities like "Player," "Artificial Intelligence," "NPC," and "Monster." This hierarchical structure allows for the encapsulation of shared attributes and behaviors while enabling specialized functionality for different types of creatures.

On the other hand, the "SuperObject" class refers to objects that can be interacted with, provide utility, or present obstacles. It serves as the base class for three derived classes: "Obstacle," "Item," and "InteractiveTile."

The "Item" class represents objects intended for player use, which can be categorized into equipable items and consumables. Equipable items, such as offensive, defensive, or environmental items, provide the player with various abilities and advantages. Consumable items, including potions, keys, and coins, serve specific purposes and can be used or spent within the game world.

This architecture adheres to the principles of OOP and SOLID. By organizing entities into hierarchical relationships, it promotes code reusability and modularity. Each class has a well-defined responsibility, adhering to the Single Responsibility Principle. The open-closed principle is supported by the ability to extend the functionality of the classes through inheritance. The Liskov Substitution Principle ensures that derived classes can be used interchangeably with their base classes. Interface Segregation is maintained by defining specific interfaces for each class, enabling clients to interact with objects through appropriate interfaces. Finally, the Dependency Inversion Principle is upheld by ensuring that high-level modules depend on abstractions, allowing for flexibility and easier maintenance.

This architecture not only promotes a modular and scalable codebase but also facilitates the adherence to best practices in software development. It enables efficient collaboration among developers, enhances code maintainability, and supports future extensibility. By utilizing OOP and SOLID principles, the project aims to deliver a robust and well-structured implementation, fostering a successful and maintainable game development process.

In shaping the architecture of this project, inspiration was drawn from the [4]. This resource discusses the implementation of the Strategy pattern in Unity, which served as a valuable reference for designing the structure of the game. The Strategy pattern, known for its flexibility and extensibility, was leveraged to handle various aspects of the game mechanics and decision-making processes. By incorporating ideas and concepts from this resource, the project's architecture was enhanced to provide a solid foundation for the development of the game.

This section highlights the key technologies and design principles employed in the game's development, showcasing the importance of Java, object-oriented design, and the Strategy pattern. These choices have enabled the creation of a modular, maintainable, and extensible game architecture, laying the foundation for the game's innovative features and mechanics.

3 Scenes played

Scenes in the game serve as distinct environments where gameplay takes place and contribute to various aspects of the player's experience and game progression. They play a crucial role in storytelling, providing stages for narrative development, character interactions, and plot advancements. Additionally, scenes offer diverse gameplay experiences through challenges, puzzles, and obstacles. They promote exploration, discovery of hidden content, and reward-driven engagement. Furthermore, scenes often represent milestones, checkpoints, or hubs that mark player progression, unlock new abilities, and provide access to additional features. Overall, scenes play a vital role in shaping the player's journey, enhancing immersion, and facilitating game progression.

The inventory functions as the player's "backpack," containing useful items and equippable gear. Each item within the inventory is accompanied by a brief description and name. Currently, the inventory provides 30 available slots. Navigating through the inventory is accomplished using the W, A, S, and D keys. The left window, from Fig. 4, displays information about the player character, such as their current level, equipped gear, offensive/defensive values, and other pertinent details.

Fig. 4: Inventory window and character details

The trading system aims to provide players with a means to acquire new items, enhance their gameplay experience, and manage their inventory effectively. By utilizing coins as the trading currency and incorporating the Merchant NPC as a reliable source of transactions, players can engage in a dynamic and immersive trading experience within the game world.

The trading system within the game revolves around a Merchant NPC, serving as a central point for acquiring new items or selling objects from the player's inventory (Fig. 5). This system operates on a currency basis, utilizing coins as the primary medium of exchange. The Merchant NPC provides a variety of items for purchase, each accompanied by relevant descriptions and prices.

Fig. 5: Trading system, Merchant NPC

Players can interact with the Merchant NPC to initiate trading transactions. They can browse through the available inventory, select desired items for purchase, and complete the transaction by spending the appropriate amount of coins. Conversely, players have the option to sell unwanted items from their inventory to the Merchant NPC in exchange for coins.

Fig. 6: Trading system, The purchase scenario

Pathfinding in the game is implemented using the A* search algorithm, enabling monsters to dynamically become aggressive when the player approaches within a certain distance. This behavior triggers at each frame, continuously evaluating the player's proximity to nearby monsters. When the

player enters the aggressive range, the monsters enter an aggressive mode, actively pursuing the player (see in Fig. 7).

The A* search algorithm is employed to determine the most optimal path for the aggressive monsters to reach the player. It takes into account various factors, such as the terrain, obstacles, and the player's movements, to calculate the most efficient route. By utilizing heuristics and a cost evaluation function, the algorithm guides the monsters towards the player, considering both distance and potential obstacles.

As the player moves further away from the monsters, the aggressive mode has a chance to cease. The algorithm continuously recalculates the path and, if the player moves beyond a certain distance threshold, the monsters revert to their default behavior or cease their pursuit altogether.

This implementation of pathfinding and the A* search algorithm enhances the game's AI system, creating dynamic and engaging encounters between the player and monsters. It allows for intelligent decision-making by the monsters, enabling them to actively pursue the player while considering the environment and potential obstacles. The A* search algorithm proves to be a reliable and efficient method for generating optimal paths and ensuring smooth and responsive monster behavior in the game.

Fig. 7: Pathfinding, Monsters become aggressive

The Options Window serves as a central hub for players to customize various aspects of the game to suit their preferences. It empowers players to make personalized adjustments to display settings, audio levels, controls, and provides a convenient way to exit the game. This attention to customization and player comfort enhances the overall gaming experience and fosters player engagement and satisfaction.

Fig 8: Options window

Overall, scenes play a vital role in shaping the player's journey, enhancing immersion and facilitating game progression.

4 Conclusions and future developments

The originality of the application based on the implementation of a complex and seamless backend that supports all the features of a 2D RPG and action game. Some aspects of the back-end that contribute to the originality of the application include:

- NPC Behavior and Interaction: Developing a system that allows NPCs to interact with the player realistically and dynamically is a complex undertaking. This includes developing AI algorithms that allow NPCs to move, speak, and perform actions based on player input and environmental factors.
- Dynamic Monster Behavior: In addition to developing NPC behavior, the backend must also support dynamic monster behavior. This includes designing AI systems that allow monsters to move, attack, and react to player behavior in a realistic and challenging way.
- Seamless transition between maps: Developing a back-end that supports seamless transition between maps is essential to create a smooth and immersive gaming experience. This requires designing algorithms that allow the game to load and unload maps smoothly without interrupting gameplay.
- Sophisticated Battle System: A back-end that supports a sophisticated battle system is crucial to deliver engaging and challenging gameplay. This includes developing algorithms that support a variety of weapons, spells, and special abilities, as well as designing a combat AI system that allows monsters to react to player actions in a realistic and challenging way.
- Advanced Search: The backend must also support advanced search algorithms that allow NPCs and monsters to navigate complex maps and obstacles in a realistic and efficient way.

The presented ARPG demonstrates the application of the game theory concepts in game design and development. The innovative features and game mechanics provide a compelling and immersive gaming experience. The use of SOLID, OOP, AI, design patterns, and game theory principles improves code quality, maintainability and scalability. Further research and experimentation can enhance the game's complexity, replay ability and social aspects.

Future developments entail multiplayer functionality, new characters and quests, improved graphics and animations, an enhanced trading system, and advanced AI for non-player characters. These advancements will revolutionize the gaming landscape, providing an immersive and captivating experience.

Acknowledgement: This work was supervised by Assoc. Prof. PhD *Laura Stoica*, from *Lucian Blaga University of Sibiu*.

References

[1] OpenGameArt.org, https://opengameart.org (link accessed, July 2022)
[2] Intro | Coding a 2D Game Engine in Java, Author: "GamesWithGabe" https://youtu.be/VyKE7vz65rY (link accessed, February 2022)
[3] Intro and Roadmap | Coding a 2D Physics Engine in Java, Author: "GamesWithGabe" https://youtu.be/vcgtwY39FT0 (link accessed, May 2022)
[4] Change Behaviors with Strategy Pattern – Unity and C# https://youtu.be/yjZsAl13trk (link accesed, July 2022)
[5] The Legend of Zelda (NES) – 100% Full Game Walkthrough, https://youtu.be/6g2vk8Gudqs (link accesed, 20 January 2022)
[6] Strategy (game theory), https://en.wikipedia.org/wiki/Strategy_(game_theory) (link accesed, 17 November 2021)
[7] SOLID Principles for Game Development, https://game.courses/solid-principles-for-game-development/ (link accesed, 25 November 2021)
[8] AI Programming Overview, https://developer.valvesoftware.com/wiki/AI_Programming_Overview, (link accesed, 09 December 2021)
[9] Memory Management in Java, https://www.javatpoint.com/memory-management-in-java, (link accesed, 15 February 2022)
[10] A* Search Algorithm, https://www.geeksforgeeks.org/a-search-algorithm/, (link accesed, 12 November 2022)

David-Stelian HOKA
Lucian Blaga University of Sibiu
Faculty of Sciences
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: davidstelian.hoka@ulbsibiu.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

The Epidemics Toolset: An interactive web app for genome comparison and mathematical modelling in epidemiology

Augusta Ionaș

Abstract

The Covid-19 pandemic has shown us that access to quick, reliable results about viruses and possible outbreaks is essential. As such, my project "The Epidemics Toolset" aims to aid researchers and laypersons in discovering information about an unknown virus. The goal is to provide an interactive tool to both analyse an unknown virus genome and predict the evolution of an infectious disease. The user can upload a .fasta file with two virus genomes, a known one and an unknown one, for which the web app computes relevant elements such as the similarity percentage and the GC-content. Moreover, it calculates and graphs the possible spread of the infectious disease through mathematical models such as the SIR and SEIR. The web app was designed in Anvil using Python and integrating Google Colaboratory code.

1 Introduction

The last three years have been definingly shaped by the Covid-19 pandemic, certainly the most significant infectious disease of the 21st century. As such, the importance of finding fast, accurate results to understand both the construction of the virus and the evolution of the pandemic has been repeatedly proven. Although currently SARS-CoV-2 has attracted public attention, new types of viruses are constantly emerging in the world of epidemiology. The goal of my project is to provide an interactive way to analyse both an unknown genome and the evolution of an infectious disease using the Python programming language in Google Colaboratory and Anvil to ultimately provide an interactive web app. For this I used Covid-19 and RTG13 as virus examples. The theme of the analysis of infectious diseases was chosen due to the possibility of further developing the study in the field of epidemiology. Furthermore, the project is based on my knowledge in the field of bioinformatics formed within the Inspirit AI project, through which I studied the evolution of SARS-CoV-2 with the help of artificial intelligence.

Since the proposal of the term genome in 1920 by Hans Winkler, as a definition of the hereditary information contained in chromosomes [2] their study has been the basis of epidemiology. Currently, reliable, up-to-date information about them can be found in the database provided by the National Library of Medicine and the National Center for Biotechnology Information, which offers access to the genomic sequence and additional information [9]. In the field of bioinformatics specific for analysing such data is Biopython, which processes the information according to the specified database.

To predict the evolution of an epidemic, there are nowadays multiple mathematical models based on characteristics such as Lotka-Volterra equations [13], Markov chains [8] or SIR (Susceptible-Infectious-Recovered) ordinary differential equations [3]. The mathematical modelling part of my project is based on the SIR model and provides some improvements specific to the SEIR (Susceptible-Exposed-Infectious-Recovered) model.

From a structural point of view, the present work is divided into 7 sections. Section 2 introduces the design and implementation. The file format is described in Section 3. Section 4 details the comparison between two genomes, being divided into 3 subsections that describe the length of the genome of the unknown virus, the number of bases that differ and the similarity percentage. Next comes the 5th section which analyses the components adenine (A), cytosine (C), guanine (G) and thymine (T). Section 6 is divided into two subsections describing the mathematical models. Section 7 is the final section of the paper and presents both conclusions and future directions.

2 Design and implementation

Most of the code is written in Google Colaboratory as shown in Fig. 1. Google Colaboratory (or Google Colab) is an environment specific to data science that offers the possibility of developing projects in Python with information stored in cloud. The so-called "Colab notebooks" are a type of Juyter notebooks integrated with Google Drive on which collaboration through access to a shared link is possible. My project uses this element so that the user can upload information from their own computer.

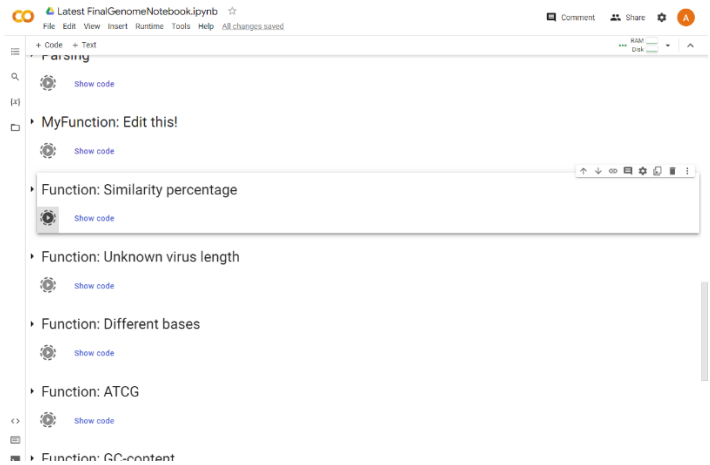


Fig. 1: The functions for genomic analysis in Google Colaboratory

The Google Colab code is then transformed into a web app called "Epidemics Toolset" through Anvil (Fig. 2). Moreover, the mathematical models are also coded using Anvil server modules.

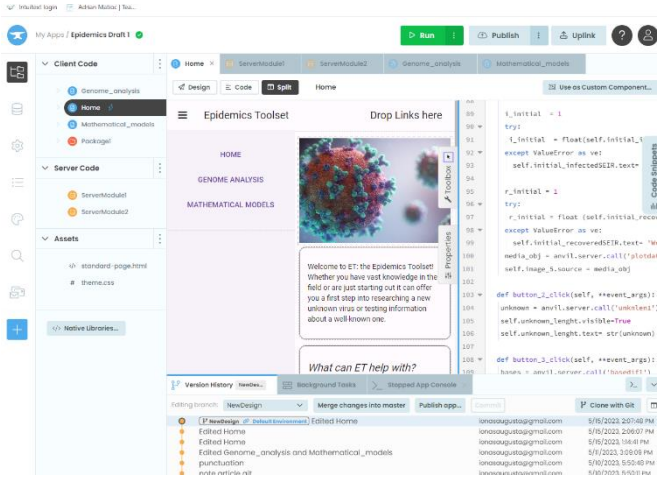


Fig. 2: Code and Design in Anvil

To connect Google Colab to Anvil the Uplink method, specific to the Anvil environment, was used [5]. In Colab this part of the code is:

```
!pip install anvil-uplink
import anvil.server
anvil.server.connect("uplink-link")
```

Code 1: Connecting Anvil to Google Colaboratory

In addition, before the functions the following call is used:

```
@anvil.server.callable
```

Code 2: How to call functions from Google Colaboratory in Anvil

so that they can be accessed by Anvil from Google Colab. As such, through Google Colaboratory, Anvil and Google Drive I have developed a web app with three page: Home, Genome Analysis and Mathematical Models which can help researchers and laypeople analyse a new virus. To sum up, the architecture of my web application can be understood as presented in the following figure (Fig. 3):

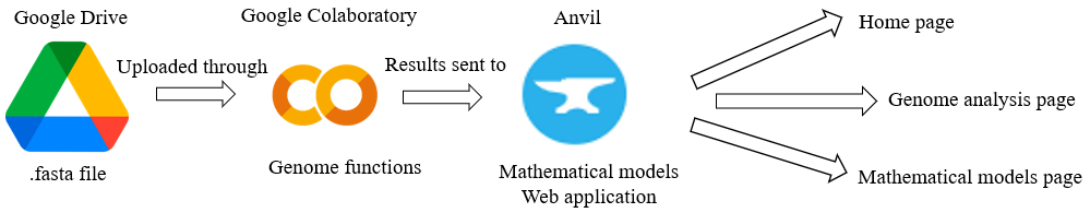


Fig. 3: The architecture of the web application

As such, the project demonstrates a unique integration of Google Drive, Google Colaboratory and Anvil to make a web application for genome analysis and mathematical modelling that allows collaboration between users.

3 File format

Initially, a file that contains information about Covid-19 and the RTG13 virus is already uploaded to Google Colab and connected to the web app. Those who want to experiment with other files can access the notebook by clicking the "Google Colab link" where they receive detailed explanations on how to upload their own file from Google Drive.

The model uses Covid-19 as an example of an unknown virus and RTG-13 as an instance of a known one. The file format is .fasta, a format specific to official databases such as the one provided by the National Center for Biotechnology Information [10]. A .fasta file can be created by editing a .txt and changing the ending to .fasta.

It is well known that for DNA the bases adenine (A), cytosine (C), guanine (G) and thymine (T) are used, and for RNA thymine (T) is replaced by uracil (U). In a .fasta file everything will be written according to A, T, C and G whether it is DNA or RNA. Furthermore, the file must have a specific format. As can be seen in the following image (Fig. 3), it starts with the symbol ">" followed by the name of the known virus, and on the next line the coded sequence in DNA. The information for the unknown virus is similarly inserted into the same file.

```
>NC_045512.2 Severe acute respiratory syndrome coronavirus 2 isolate
Wuhan-Hu-1, complete genome
ATTAAAGGTTTATACCTTCCCAGGTAACAAACCAACCAACTTTCGATCTCTTGTAGATCTGTTCTCTAAA
CGAACTTTAAAATCTGTGTGGCTGTCACTCGGCTGCATGCTTAGTGCACTCACGCAGTATAATTAATAAC
TAATTACTGTCGTTGACAGGACACGAGTAACTCGTCTATCTTCTGCAGGCTGCTTACGGTTTCGTCCGTG
TTGCAGCCGATCATCAGCACATCTAGGTTTCGTCCGGGTGTGACCGAAAGGTAAGATGGAGAGCCTTGTC
CCTGGTTTCAACGAGAAAACACACGTCCAACTCAGTTTGCCTGTTTTACAGGTTCGCGACGTGCTCGTAC
GTGGCTTTGGAGACTCCGTGGAGGAGGTCTTATCAGAGGCACGTCAACATCTTAAAGATGGCACTTGTGG
CTTAGTAGAAGTTGAAAAAGGCGTTTTGCCTCAACTTGAACAGCCCTATGTGTTCATCAAACGTTCGGAT
GCTCGAACTGCACCTCATGGTCATGTTATGGTTGAGCTGGTAGCAGAACTCGAAGGCATTCAGTACGGTC
GTAGTGGTGAGACACTTGGTGTCCTTGTCCCTCATGTGGGCGAAATACCAGTGGCTTACCGCAAGGTTCT
TCTTCGTAAGAACGGTAATAAAGGAGCTGGTGGCCATAGTTACGGCGCCGATCTAAAGTCATTTGACTTA
GGCGACGAGCTTGGCACTGATCCTTATGAAGATTTTCAAGAAAACTGGAACACTAAACATAGCAGTGGTG
TTACCCGTGAACTCATGCGTGAGCTTAACGGAGGGGCATACACTCGCTATGTCGATAACAACTTCTGTGG
```

Fig. 3: Example of a .fasta file for the unknown virus (Covid-19)

The .fasta file is uploaded to Google Drive, from where the information is accessed by the notebook. In Google Colab the connection is made via the following code sequence:

```
input_file = "File_name.fasta"
fasta_location = ("Google_Drive_link")
if not os.path.exists(input_file):
   gdown.download(fasta_location, input_file)
```

Code 3: Uploading the file in Google Colaboratory from Google Drive

The value for input_file should be the name of the file, and the value for fasta_location should be a link to the file on the personal drive. This connection is possible with the BioPython tool via the SeqIO library and via the gdown library. The link for the drive consists of the link obtained from using the "Get link" option and deleting the information after the last bar "/" that includes the text "usp=share_link". This connection represents a first novelty element of the project, being different from what I have identified in other articles and open-source projects on similar topics and the only one that makes it possible to compare two genomes using only Google Drive. More detailed information about this process is presented in the notebook.

4 Genome comparison

A first step in understanding a new virus is to analyse the RNA. Thus, the web app computes three relevant elements in epidemiology, namely the length of the genome of the unknown virus, the number of bases that differ and the similarity percentage (Fig. 4).

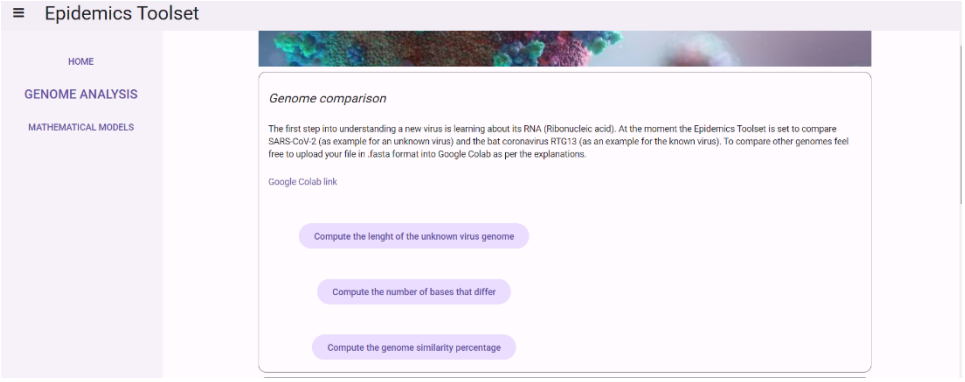


Fig. 4: Genome comparison page

For this purpose my project uses a parsing algorithm. The sequences are seen as strings and transformed into a numpy array consisting of the letters of the DNA by the code sequence:

```
sequences = [record for record in SeqIO.parse(input_file,'fasta')]
sequence_known = np.array(sequences[0])
sequence_unknown = np.array(sequences[1])
```

Code 4: The parsing algorithm

As the .fasta file consists of encoding the known genome and then the new one, we split these sequences in two to work with them later.

4.1 The length of the unknown virus

It is important to realize that in the genome description in the .fasta file "-" represents deletions. Thus, to calculate the length of the virus, the letters must be different from the symbol of the deletions. This can be easily achieved through the follwoing code-piece:

```
length_unknown = sum(sequence_unknown!='-')
```

Code 5: Computing the length of the unknown virus

4.2 The number of bases that differ

Similar to subsequence 4.1 most relevant here is the following code sequence:

```
bases_different = sum(sequence_known!=sequence_unknown)
```

Code 6: Calculating the bases that differ

Thus, the different bases between the known virus RNA (transformed into DNA) and the new virus RNA are summed.

4.3 The similarity percentage

Using the data obtained from subsequences 4.1 and 4.2, the percentage of similarity between two viruses can be calculated according to the formula:

$$P = 100 * \frac{bases_{same}}{lenghth_{unknown}} \quad (1)$$

The calculations performed in this section present an essential element in deducing the origin of a virus. A good example here is the comparison between Covid-19 and RTG13. The result for the similarity percentage is approximately 96%, which supports the conclusions of researchers in the field that the virus comes from a bat. This way, by comparing genomes, different hypotheses about the origin of a virus can be made.

5 An analysis of the A, T, C & G components of the unknown virus

The percentage in which the bases appear in the genome structure has several relevant consequences such as computing the GC-content or the number of start codons (Fig.5).

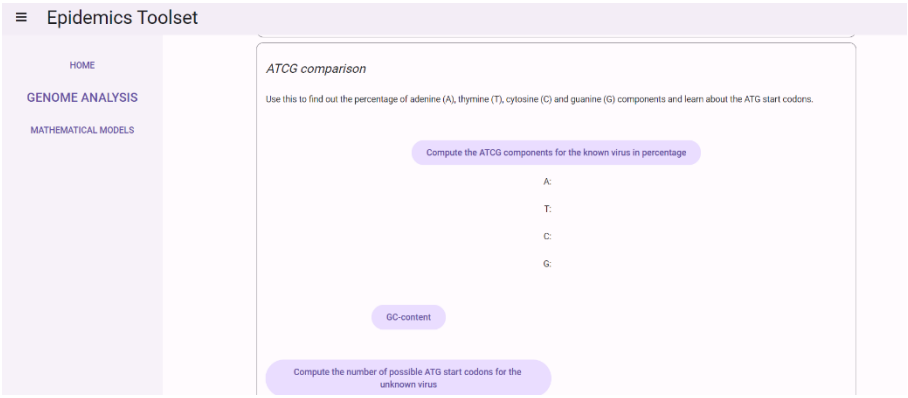


Fig. 5: Data about A, T, C &G

To determine the percentage of adenine (A), for example, we compare the genome sequence with the letter A and use the following formula that divides A, that is, the number of adenine components, by the length of the unknown virus in the third sequence and multiplies everything by 100:

$$\text{Percentage A} = 100 * \frac{A}{lenghth_{unknown}} \quad (2)$$

After calculating all the A,T,G and C percentages we can find out the GC-content. It represents the percentage of nitrogenous bases in a DNA or RNA molecule that are guanine (G) or cytosine (C) and is relevant in determining the thermal stability of the genome. As presented by Yoe Li & Co, it may also

be important in the adaptation of a virus at the lung level [7]. The equation used to calculate the GC-content element is:

$$GC - content = 100 * \frac{G+C}{A+T+G+C} \quad (3)$$

Next, the project determines the number of possible ATG start codons. For a better understanding of this process, a brief description of the codons follows. Through a reading frame we create a way to divide the nucleotide sequence in a nucleic acid molecule into a set of consecutive, non-overlapping triplets. These are called codons if the triplets are amino acids or stop signals. To calculate this number of start codons we use a counter start_codons initialized with 0 that increases every time it finds an "ATG" sequence.

Nonetheless, as a researcher in epidemiology, one needs to work with a variety of functions some of which may not be available in my project. As such, another item in the genomic analysis chapter is the MyFunction function. The user has access to the code from the Google Colaboratory, which is open-source, and can edit it to write their own function which is automatically linked to the web application.

6 Mathematical models for the transmission of infectious diseases

There are a multitude of mathematical models suitable for predicting the evolution of an infectious disease. For this project I chose the classic SIR model [1] and a SEIR model. I used the server modules in Anvil to write the code that forms the mathematical models presented in this chapter, each model being detailed in a different server module. In the design part, I worked with two images connected to the specific server module through media objects [4]. Thus, the functions in the server modules are called through media objects, so that they receive as parameters the values entered from the keyboard. In the following example one can see the connection between the first model and the image for it, i.e., the SIR plot, by calling the plotdata function.

```
media_obj = anvil.server.call('plotdata',R0,t_infective, i_initial, r_initial)
self.image_4.source = media_obj
```

Code 7:Server modules and media objects in Anvil

6.1 The classic SIR model

To code the SIR (Susceptible-Infectious-Recovered) model [6], a population is divided into three groups corresponding to the stages of the disease:

- S (Susceptible) - the subpopulation susceptible to the disease

For SARS-CoV-2, it is assumed that all those who have not previously had the disease are susceptible to infection.

- I (Infectious) - the subpopulation that is infectious
- R (Recovered) - the subpopulation that recovered from infection

It is assumed that they are no longer susceptible to the disease.

The model can be described by the equations:

$$\frac{dS}{dt} = -\frac{\beta SI}{N} \quad (4)$$

$$\frac{dI}{dt} = \frac{\beta SI}{N} - \gamma I \quad (5)$$

$$\frac{dR}{dt} = \gamma I \quad (6)$$

Here:
• S, I, R are the sizes of the susceptible, infectious, and recovered populations respectively
• β is the rate constant associated with transmission of infection
• γ is the rate of recovery from infections
• N is the total population
We use the usual notation for working with population fractions:

$$s = \frac{S}{N} \quad i = \frac{I}{N} \quad r = \frac{R}{N} \tag{7}$$

The system becomes:

$$\frac{ds}{dt} = -\beta si \tag{8}$$
$$\frac{di}{dt} = \beta si - \gamma i \tag{9}$$
$$\frac{dr}{dt} = \gamma i \tag{10}$$

To create the graph of this system it is clear that approximate parameters are needed to predict the course of the disease.

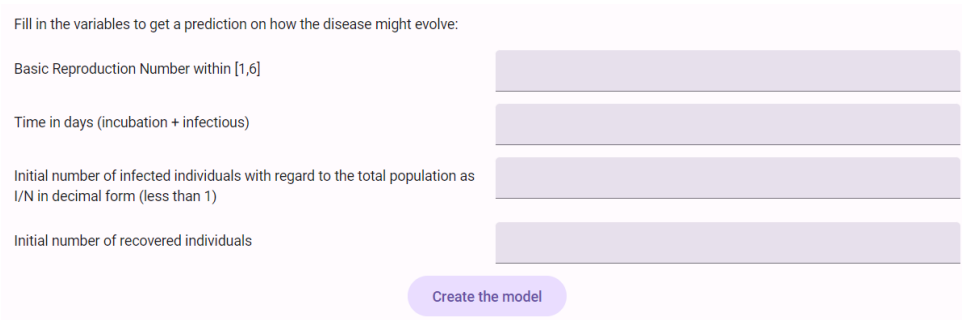


Fig. 6: SIR variables

As can be seen in Fig. 6, the chosen parameters are:

- The basic reproduction number
- The total time in days
- i: infectious subpopulation size with respect to N
- r: recovered subpopulation size with respect to N

Thus, we calculate:

```
s_init = 1 - i_initi - r_init
gamma = 1/t
beta = R*gamma
```

Code 8: The relevant parameters

The code that describes the system of equations is:

```
def sderivate(x, t, beta, gamma):
    s, i, r = x
    dsdt = -beta * s * i
    didt = beta * s * i - gamma * i
    drdt =  gamma * i
```

```
return [dsdt, didt, drdt]
```

Code 9: The system of equations

A possible disease evolution can be predicted, for instance, with the values 3.2, 7, 0.00001 and 0 to get the following plot (Fig.7).

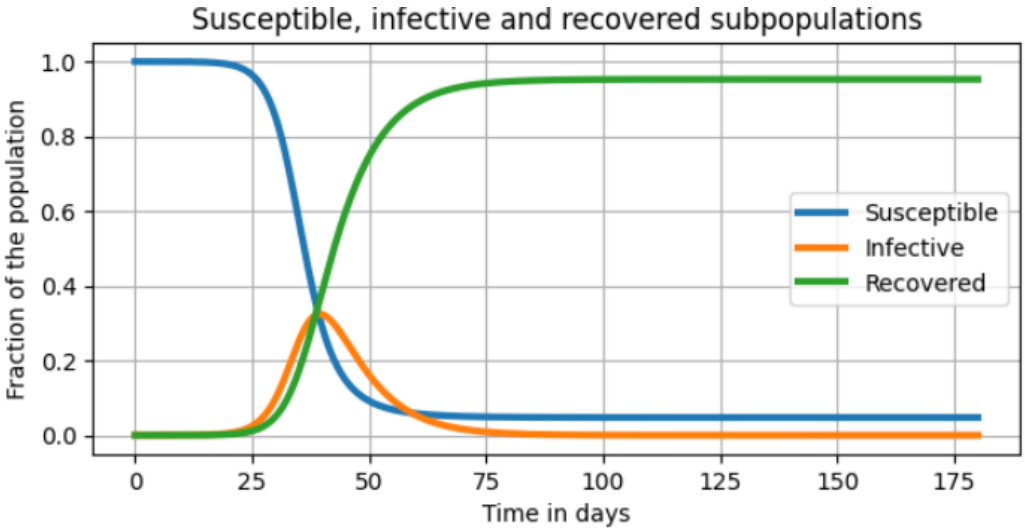


Fig. 7: SIR plot

These values can be changed by the user to get a different possible spread of the disease according to the new values. It is important to specify that the SIR model, altough widly used, does not always offer precise results, since it does not take into account data such as measures taken to prevent the spread of the disease. To do so, we must introduce another, more precise model, namely the SEIR. It is as such relevant to note important values, such as the peak of the infective category (0.3), which we can then compare to the results of the following model.

6.2 The improved model: SEIR

For the SEIR model both the variable E, of the exposed subpopulation that is not yet infectious, and u, the control parameter of the effectiveness of public health interventions to control disease transmission are added [11]. u=0 means no effective public health interventions and u=1 means total elimination of disease transmission [12].

The equations in this case become:

$$\frac{ds}{dt} = -(1-u)\beta si \tag{11}$$

$$\frac{de}{dt} = (1-u)\beta si - \alpha e \tag{12}$$

$$\frac{di}{dt} = \alpha e - \gamma i \tag{13}$$

$$\frac{dr}{dt} = \gamma i \tag{14}$$

Where:

$$\alpha = \frac{1}{time_{incubation}} \tag{15}$$

$$\gamma = \frac{1}{time_{infectious}} \tag{16}$$

Similar to the SIR model, the required information is (Fig.8):

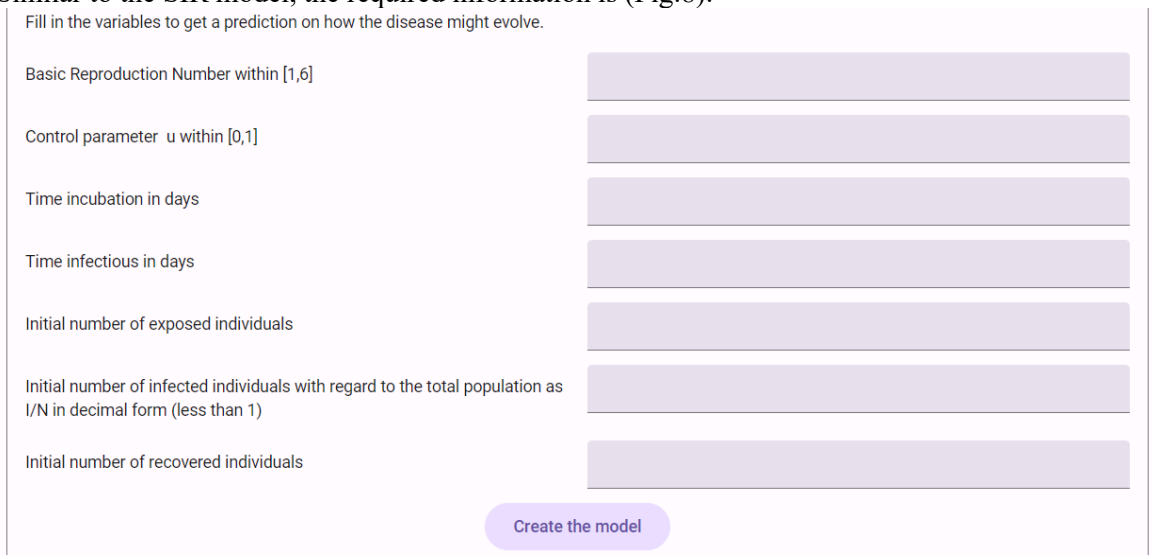


Fig. 8: SEIR Parameters

We can use the same values as before and add some new ones, namely 3.2, 0.2, 5, 2, 0.00001, 0.00001 and 0 to get the following plot (Fig. 9):

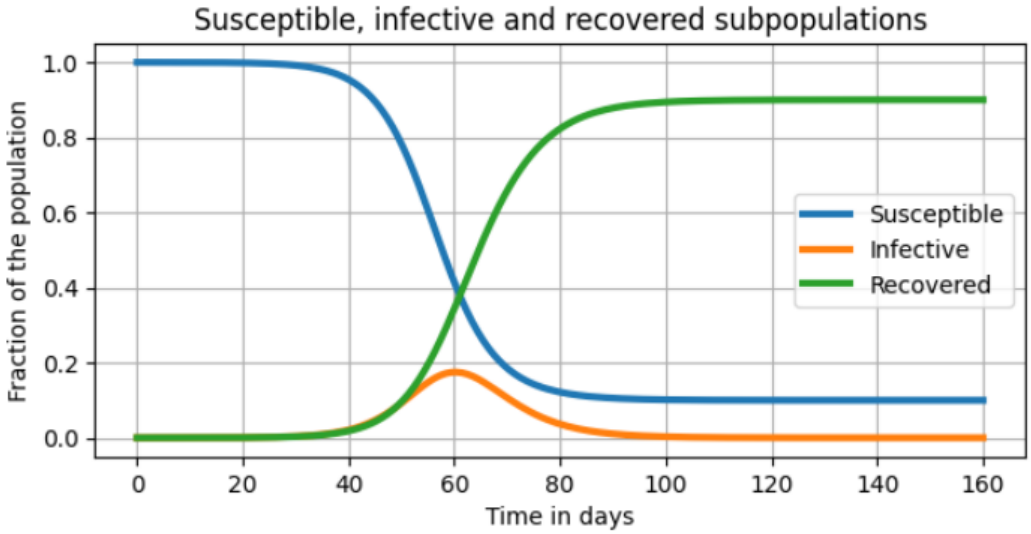


Fig. 9: SEIR plot

As expected, the results of the SEIR model differ slightly from those of the SIR model, mainly due to the introduction of the control parameter. While for SIR the y-axis showed around 0.3 as the peak for "infective", in this case the result is clearly lower, namely around 0.19, which means fewer people would get infected. As such, the difference between these two values goes to show the crucial importance of public health interventions when dealing with an epidemic.

7 Conclusions

The Epidemics Toolset is a novel web app that can be used to figure out information about the genome of a new virus and the possible spread of a disease. It offers users the possibility to compute relevant data about a genome such as the similarity to another virus, model the evolution of an epidemic, or even create user-defined functions through an interactive platform. As such, the app provides the necessary computation and coding, so that those with limited programming experience can easily analyse viruses, and offers users knowledgeable in programming the possibility to focus on specific functions instead of writing the complete code from scratch. Moreover, through the user-friendly interface and connections between different programming mediums, it builds a complex architecture that allows collaboration between users. As such, the project furthermore demonstrates the possibility of combining Google Colaboratory, Drive and Anvil to create an interactive web application written exclusively in Python.

The purpose of my study is to build a project with wide applicability in areas such as education or research, that can be improved with ease by adding new features or providing more complex and precise calculations. For instance, an improvement in the design of the app would be adding a direct upload element from the page to give users the possibility to upload the .fasta file directly from the site, instead of through Google Colaboratory. Some other relevant developments would be analysing possible mutations or introducing more advanced mathematical models.

Aknowledgement: This work was supervised by Professor Delilah Florea, from Samuel von Brukenthal National College of Sibiu.

References

[1] Abou-Ismail A. Compartmental Models of the COVID-19 Pandemic for Physicians and Physician-Scientists. *SN Compr Clin Med.* 2020;2(7):852-858, 2020 Jun 4. PMID: 32838137
[2] Cristescu Melania E. 2019. *The concept of genome after one century of usage*
[3] Cooper Ian., MONDAL Argha, & ANTONOPOULOS, Chris G. A SIR model assumption for the spread of COVID-19 in different communities. *Chaos, Solitons, and Fractals*, 2020, 139, 110057.
[4] Anvil: Server Modules, https://anvil.works/docs/server
[5] Anvil: Putting a web front-end on a Google Colab notebook, https://anvil.works/learn/tutorials/google-colab-to-web-app
[6] The SIR epidemic model, https://scipython.com/book/chapter-8-scipy/additional-examples/the-sir-epidemic-model/
[7] Li, Yue., Yang, Xinai., WANG, N. et al. GC usage of SARS-CoV-2 genes might adapt to the environment of human lung expressed genes, *Mol Genet Genomics* 295, 1537–1546, 2020
[8] Marfak, Abdelghafour, ACHAK, Doha, AZIZI, Asmaa & co The hidden Markov chain modelling of the COVID-19 spreading using Moroccan dataset, *Data in Brief*, 32.
[9] National Center for Biotechnology Information: https://www.ncbi.nlm.nih.gov/nuccore/NC_045512
[10] National Center for Biotechnology Information on the FASTA format: https://www.ncbi.nlm.nih.gov/genbank/fastaformat/
[11] Poonia , Ramesh Chandra et al. An Enhanced SEIR Model for Prediction of COVID-19 with Vaccination Effect, *Life (Basel)*. 2022 May; 12(5): 647
[12] University of Notre Dame, CBE 30338 Chemical Process Control (Modeling and Control of a Campus COVID-19 Outbreak)
[13] Younes, Ahmad Bani, and HASAN, Zeaid, COVID-19: Modeling, Prediction, and Control. Applied sciences, 2020, 10, 3666

Augusta IONAȘ
„Samuel von Brukenthal" National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: ionasaugusta@gmail.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

A User-Friendly Solution for streamlining Hotel and Restaurant Management with QR Code Integration

Corina Minodora Iordache, Felix Husac

Abstract

The management of hotels and restaurants can be a complex and challenging task, especially when it comes to keeping track of guest bookings, room availability, and food service. In this paper, we present a hotel and restaurant management application that utilizes Svelte.js and MongoDB to streamline these processes. The application allows hotel staff to easily manage bookings, track room availability, and monitor guest activities in the hotel and restaurant. The use of QR codes provides a seamless experience for the guests, allowing them to easily access areas of the hotel and check relevant information. In addition, the application can log the use of the canteen and the items served, allowing for accurate billing at the end of the stay. This approach eliminates the need for traditional canteen ordering and billing procedures, making the process faster, more efficient, and more secure. Guests are billed for their restaurant or canteen expenses when they check out. Our application provides a robust and scalable platform for hotel management, with potential for expansion into inventory management and other areas. Future improvements to the application could include additional features such as analytics and data visualization, further enhancing the value of the application for hotel owners and managers. It also has the potential for expansion into inventory management and other areas. Overall, this application offers a user-friendly and comprehensive solution for hotel and restaurant management, and can greatly improve the guest experience while streamlining operations for hotel staff.

1 Introduction

The management of small hospitality businesses, like family-owned and family-run hotels and restaurants poses various challenges, particularly in the areas of guest bookings, room availability, and food service. Effectively keeping track of these aspects can be complex and time-consuming. Some business owners even resort to inefficient manual methods like email, social media, and handwritten notes, leading to errors and delays. Traditional approaches like pen and paper book keeping and phone calls make it difficult to maintain accurate records of the hotel's activity. This highlights the need for a user-friendly and centralized solution that automates hotel and restaurant management processes.

Our proposed application, powered by Svelte and MongoDB, centralizes guest bookings, room availability, and food service management. The application provides real-time updates via the data fetching features provided by SvelteKit.js, a full stack web development framework discussed later in the paper. Accurate reports allow the hotel staff to ensure optimal occupancy management, preventing overbookings or room conflicts. The proposed application provides staff with insights into guest activities, and prevents the use of the hotel's facilities without the proper QR code ID, via the implemented duplicate scanning and invalid QR code detection.

The core functionality of the application revolves around the seamless integration of QR codes, which enhances the guest experience and simplifies access to various areas of the hotel. Upon check-in, guests are provided with unique QR codes that grant them access to their rooms and hotel facilities. In addition to guest management, the application aims to improve on the traditional canteen ordering and billing procedures. Guests can conveniently log their food orders using their issued QR code IDs

by scanning them at the canteen, and pay at the end of their stay. The application logs all canteen activities, ensuring accurate billing. The QR code IDs are printed on thermal paper, avoiding the inconvenience of yet another smartphone app for the hotel guests to download.

Deployed as a Progressive Web App, our application provides a clean and friendly user interface that features large fonts, pleasant colors, quick interactivity and well positioned controls for efficient guest management within hotel or restaurant premises. The application could be expanded to host analytics and payment processing facilities, or adjusted to suit the needs of inventory management.

The following section will present several solutions already in use in the realm of hospitality management. The rest of this paper will detail the system design, architecture, and user interface. The fourth chapter will describe the software stack used to build this application and why each technology was chosen. The conclusions chapter will highlight the application's strengths and suggest future development directions.

2 State of the art

Online Travel Agencies (OTAs) are integral to the hospitality industry, providing room booking services. However, they charge a commission, typically ranging from 15 to 30 percent of the total room price. OTAs enforce price parity, which mandates that hotels maintain identical pricing on their websites as on OTA platforms, prohibiting discounts. Furthermore, OTAs often restrict hotels from collecting valuable guest data, such as email addresses and phone numbers, during the booking process. This limitation hinders the creation of guest loyalty programs and personalized guest relationships. These OTA practices have forced hospitality businesses to use their own solutions or to accept more in-person reservations, at the expense of online bookings. [1] Within the domain of hospitality management software tailored for small and medium businesses, there exists a diverse array of options, each characterized by its unique merits and limitations. These software solutions significantly improve operational efficiency and comprehensive record keeping in a wide spectrum of hospitality establishments. We assessed various hospitality management tools, including conventional industry-standard choices and newer alternatives. Regrettably, none of these options aligned with our requirements. What follows is a concise description of some of the solutions we explored.

“OPERA Oracle Hospitality PMS” is a leading management service developed by the Oracle Corporation. It is a comprehensive solution famous for its many features, that include property management, point of sale, and customer relationship management. Its scalability, customization, and integration capabilities are noteworthy. However, it's essential to note that Oracle OPERA may be too expensive for smaller businesses. [2,3]

“Guestpoint” positions itself as a management service designed for small to medium sized hospitality businesses. It boasts an all-in-one solution, covering everything from initial booking, payment processing and guest checkout. Unfortunately, despite its extensive feature set, businesses may find themselves unable to customize the service to fit their particular needs. [4]

“HOTELTIME” is a cloud-based service focused on restaurant management, offering table management, inventory control, and reporting features. Its ease of use earned it the “Best choice” award from hoteltechreport [5]. Unfortunately, some clients have experienced technical difficulties, bugs and overall instability.

“LS Retail” is an all-in-one software solution that caters to restaurants, food service establishments, and hotels. It boasts features like point of sale, inventory management, and customer relationship management. The software's scalability, customization options, and integration capabilities are commendable, but its complexity may be off putting for smaller sized businesses. [6]

“Innkeeper's Advantage” is a software product designed specifically for small hotels, inns, and bed and breakfasts, emphasizes ease of use, affordability, and customization. However, it may not be the ideal choice for larger or more complex properties. [7]

The dynamic landscape of hospitality management software offers various options, each catering to distinct business needs. The management software chosen by hospitality establishments is beginning to become an important part of their business as the industry evolves, being a factor that can influence competitiveness and overall service quality.

3 Underlying Architecture of the Application

3.1 General Layout and User Interface Design

The hotel and restaurant management application was created with ease of use and responsive design principles in mind, ensuring adaptability across a wide range of devices and screen sizes. The interface incorporates user-friendly elements such as large fonts, a sticky navigation bar, an appealing color scheme, and clearly distinguishable controls. Information is presented in tiles, while data editing is conveniently performed within modal windows that dynamically adjust to the current page. These design choices, coupled with smooth animations and transitions for every component, contribute to an intuitive user experience. They allow the users to navigate the application with a small number of interactions.

The application is structured into five distinct pages, each serving a specific purpose. Out of the box, they cover the main aspects of hotel and restaurant management and can be easily expanded to fit the client's needs.

- The Home tab contains useful information and a brief description of the way the application works. It serves as an introductory welcome page and outlines the application's overall functionality.
- The Room management tab allows the manager to add or delete rooms, edit occupancy and check availability at a glance. Each room is represented by a tile, displaying useful information such as the room number, maximum occupancy, the current occupancy level and the names of the guests currently assigned to each room. Editing the guests hosted in each room can be done by clicking on the desired room tile. Selecting a tile or clicking the Add Room button opens a modal window, prompting the user to fill in or change the desired fields. For added convenience, each tile features a dedicated Delete button. (Fig. 1)
-

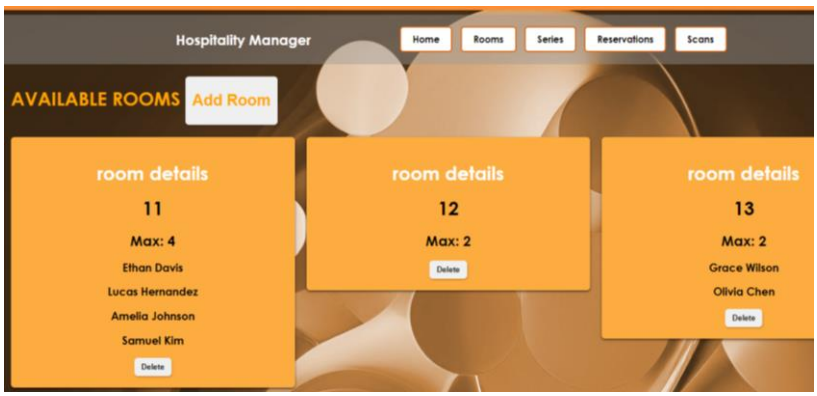


Fig. 1: Snippet of the Rooms tab, displaying tiles with information corresponding to each room

- The Series management tab offers the manager the option to group arriving guests into series of reservations, for better organization. It follows a similar layout to the previous page, using tiles to display the series name and the reservation starting date and ending date, which can be edited or deleted. The modal window adapts to the required fields for scheduling the guest series, featuring two date pickers and checking that the dates entered are in the correct order.
- The Reservation tab offers CRUD operations for each individual guest, also allowing for room and guest series reassignment. Guests are individually displayed on large tiles, presenting their name and the room number they are currently assigned to in a large white font for better readability. Additionally, important guest information, such as phone numbers, age, gender, and home address, is subsequently displayed (Fig. 2). Each tile also displays the series to which guests have been assigned by the manager. Here, right next to the Delete button, a Print button allows the manager to easily generate and print unique QR code IDs for each of the guests by adapting the modal window to generate the QR code and show the printing controls instead (Fig. 3). The printing modal window was customized to create 4 copies of the QR code.

-

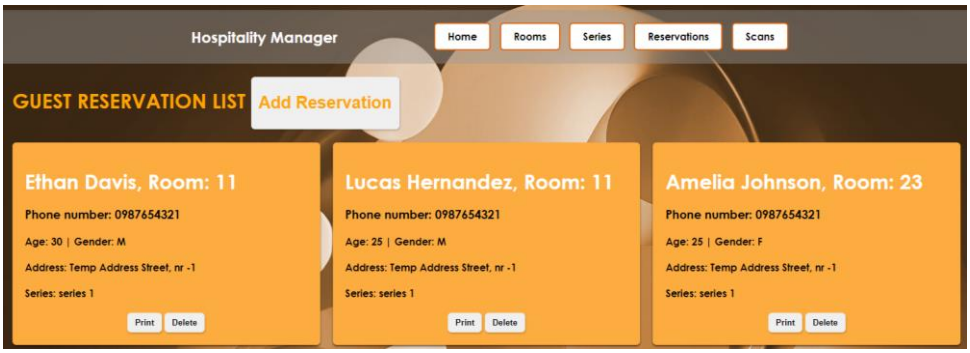


Fig. 2 - Snippet of the Reservation tab - allows basic CRUD operations for guest management

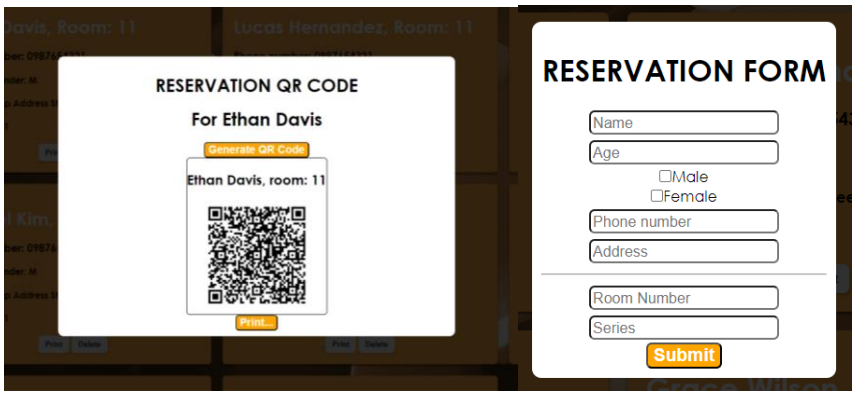


Fig. 3 - On the left - the QR code generation modal window; on the right - the Add Guest / Edit Guest modal window

- Lastly, the Meal Scans tab provides a real time report as the guests scan their QR code IDs when they serve their meal. Each scan is represented by a dedicated tile, displaying only the guest's name. This tab can easily be duplicated and adapted to suit the other amenities that need user logging that a hotel can offer.

3.2 Business Concept

Efforts have been directed towards simplifying the application's user experience and enhancing its user-friendliness. The primary goal of this application is to empower small hospitality business owners by streamlining their management processes, allowing them to focus more on improving the guest experience. This is achieved by offering an easy to use system for organizing rooms and guest information, making it easy for owners to quickly check room availability, guest details, and occupancy levels. Additionally, the application simplifies the management of larger groups of guests by grouping them together into series, simplifying the overall process of booking and room assignment.

Regarding guest reservations, the application requires only the most relevant client data, such as name, address, phone number, age and gender for a secure booking process. Managers can efficiently assign guests to rooms and groups using a user-friendly interface, and can generate unique QR code ticket IDs for each client. Additionally, the application tracks guest meal consumption for accurate billing by creating a report from QR code scans during their stay.

To make the most of the application, hotel managers should follow a straightforward set of steps. Firstly, staff should begin by adding or editing hotel room details, like room name and maximum occupancy, via the provided modal window that is displayed after clicking the Add Room button.. If applicable, they can also create or modify guest series with start and end dates in the second tab of the application. After successfully adding and configuring the rooms, managers can then proceed to the reservations page, where they can easily input guest information via the adapted modal window. They

can assign rooms and series during reservation creation and make any necessary changes. Room assignments are instantly updated in the Rooms tab, displaying the guest's name prominently on the corresponding room tile.

The Reservations tab offers convenient QR code generation and printing features. To ensure data privacy and confidentiality, the unique QR code is generated as a hashed value of the guest's data. To help prevent fraud, QR code IDs are then printed as tickets on thermal receipt paper, similar to supermarket receipts. These tickets include the guest's name, room number, and their unique QR code ID. The printing modal window offers four copies of the QR code IDs as default. Using thermal paper is both cost-effective and less wasteful for small tickets, while also offering better security against forgery, making it harder for imposters to impersonate genuine guests.

The application features a Meal Scans page that provides a live report of scanned QR codes at the hotel restaurant. This simplifies billing by allowing managers to monitor guests' meal consumption, ensuring accurate billing upon checkout. The scan page can be duplicated or adapted for other amenities, such as sports facility usage. The scan log incorporates a duplicate scan detection feature, which alerts the manager in case a guest checks in twice or attempts to forge their QR code ID. This safeguard prevents unexpected charges for guests and misuse of credentials for the benefit of another guest.

4 Technologies and Implementation

The development and implementation of this hotel and restaurant management application involved several steps for integrating the needed functionalities. The following subsection outlines the methodology used: It briefly describes the frontend and backend libraries, the database and how it is used in conjunction with SvelteKit's own data persistence mechanism, and the necessary packages for QR code generation and cryptography. The application is deployed locally as a Progressive Web App (PWA) for an enhanced user experience and offline access. PWAs are known to provide faster, near-native like performance on any device, be it a PC, smartphone or a tablet. The application uses the same runtime environment as the development version of the application, NodeJS. PWA deployment also enables installation and launch from a home screen icon. To achieve this, the application utilizes service workers, which are background scripts that cache the necessary assets and data when the user visits the app for the first time. This allows users to continue using the application even when they are offline or have a poor internet connection, enhancing accessibility and allowing the manager to use the application in all circumstances.

4.1 Brief Overview of the Technology Stack

The proposed hotel and restaurant management application makes use of a robust technology stack composed of Svelte.js, SvelteKit.js, and MongoDB, adapted to create an efficient solution.

Svelte.js, a powerful and modern JavaScript framework, serves as the foundation for developing the user interface of the application. With its reactive and component-based approach, Svelte.js enables the creation of dynamic and interactive user interfaces. The lightweight nature of Svelte.js ensures fast loading times and optimal performance, enhancing the overall user experience. Its declarative syntax and efficient reactivity system enable developers to write clean and concise code, resulting in a more maintainable and scalable application. Svelte.js is also one of the easiest to use JavaScript frontend frameworks.[1]

SvelteKit.js, a meta-framework built on top of Svelte.js, extends its features to encompass full stack web applications. It adds server-side rendering, routing, and code-splitting, enabling efficient and performant client-server interactions. SvelteKit.js simplifies the development process by eliminating complex configuration steps and boilerplate code, allowing developers to focus on building the core functionality of the application. SvelteKit.js has a modular architecture and easy to use APIs, which helps it integrate easily with other technologies, making it an great choice for building fast and interactive web applications.[2]

Svelte and SvelteKit provide a unique compilation process that sets them apart from other JavaScript frameworks. When deploying an application, the Svelte code is compiled into efficient and optimized vanilla JavaScript. This compilation step eliminates the need for a runtime framework, resulting in smaller bundle sizes and faster loading times for end-users. By generating highly optimized

JavaScript code during the build process, Svelte and SvelteKit ensure that the application performs optimally in the browser, delivering a smooth and responsive user experience. This compilation process not only enhances the performance of the application but also simplifies the deployment process by generating static files that can be easily hosted on any web server or CDN. SvelteKit can also generate static pages, providing significant advantages for a website's SEO (Search Engine Optimization) performance. The web pages are pre-rendered as static HTML during the build process, allowing search engines to easily crawl and index the content, thus leading to improved discoverability and higher rankings in search engine results.[3]

MongoDB, a popular NoSQL database, serves as the data persistence layer for the application discussed in this paper. MongoDB's flexible document model allows for the storage and retrieval of complex data structures, making it well-suited for handling the diverse needs of a hospitality management system. MongoDB organizes its data using a flexible and scalable document-oriented model, meaning that instead of using traditional tables with rows and columns, common to SQL databases, MongoDB stores data in collections, which are containers for documents. Each document is a self-contained unit that represents a record, stored as a JavaScript Object (JSON) and is analogous to a row in a relational database. Documents are also allowed to have different structures and fields. This enables the data to flexibly adapt as the application grows or is expanded upon. Its scalability and high availability features ensure that the application can handle a growing number of guests, bookings, and dining transactions. The seamless integration between Svelte.js, SvelteKit.js, and MongoDB enables real-time data synchronization and efficient data management, ensuring reliable record keeping for the hotel and restaurant staff. By making use of SvelteKit's prefetching feature, content can be loaded in advance and synchronized for all components of the application by using files called Stores. Stores are javascript files used for passing data between multiple components inside a SvelteKit application, without having to cascade the information through the component hierarchy as props.[4]

4.2 UI Components and Component Hierarchy

The application architecture consists of a collection of static routes, which are managed by SvelteKit and organized as separate directories within the "src/routes" directory. Each route directory corresponds to a specific page in the application, allowing for a clear and structured representation. Within each route directory, the central component defining the page is the "page.svelte" file. This file encapsulates the functionality, HTML structure, and styling of the respective page.[5]

To promote reusability and modularity, each page is composed out of components that are individual ".svelte" files containing JavaScript code, HTML markup, and associated styling. These components can be nested within each other or inherit data and functionalities from one another, establishing a flexible and adaptable component hierarchy. The global styling of the application is managed through the "global.css" file, located in the source directory, ensuring consistent visual presentation throughout the application.

Communication between components is facilitated through data binding and observables provided by Svelte, enabling bidirectional data flow and event handling.[6]

The pages of the proposed hotel management application are constructed using a variety of custom components, specifically designed and styled from scratch to enhance flexibility, adaptability, and ease of use.

These modifications are synchronized across all components that rely on the same Store, ensuring data consistency and facilitating future synchronization with the database (DB). On initial request, the pages are server-side rendered to optimize load times, while subsequent navigation triggers client-side rendering for seamless and dynamic user experiences. Every component retrieves data from a relevant Store file associated with its specific business logic, ensuring access to the required information for populating component fields and enabling real-time changes.

In Figure 4, the hierarchical arrangement of the custom components used in the application is illustrated, highlighting the interdependence among the various elements. This component-based approach enhances maintainability, extensibility, and scalability while catering to the unique requirements of hotel management operations.

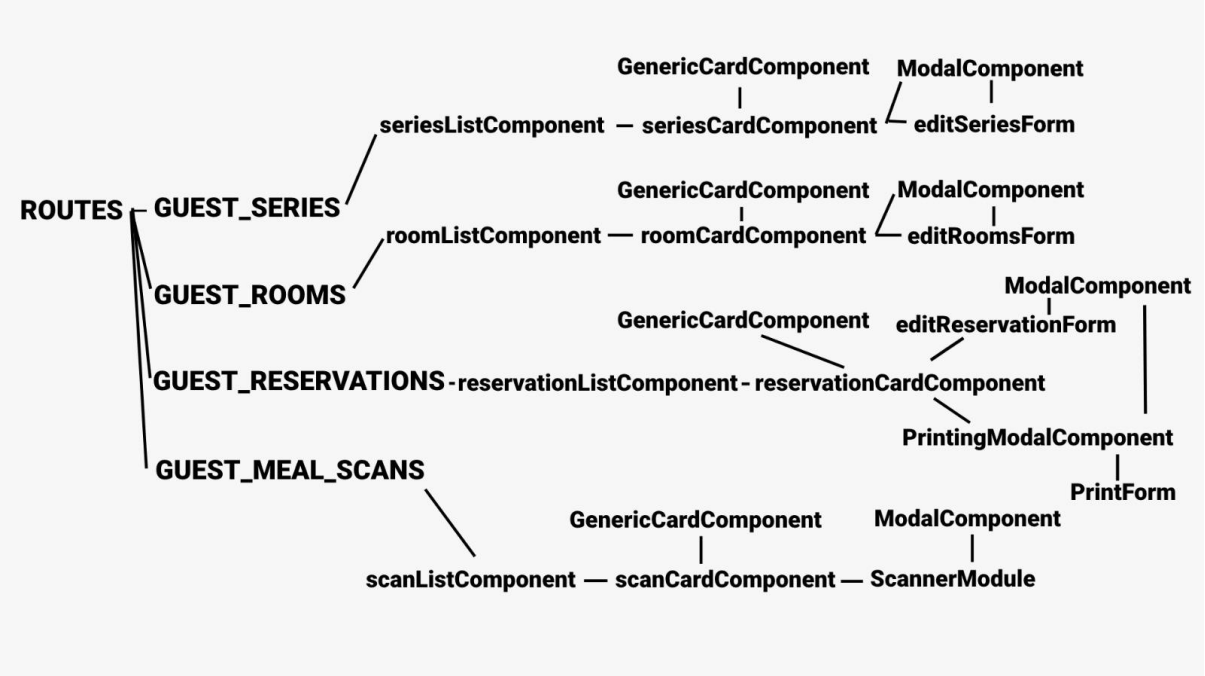


Fig. 4 - Hierarchical view of the components used inside the application - top-level components are made by compositing lower-level components or by inheriting features from other components

4.3 Plugin Packages

The application uses the "crypto.js" package to generate an MD5 hash of the serialized guest object. The JSON object corresponding to the desired guest is retrieved from the Store, transformed into a string format, and passed into the cryptographic hash function provided by "crypto.js". The resulting hashed value is then sent to the QR code generation module, which integrates the "qrcode.js" package. This module allows the application to not only generate QR codes but also customize them, enabling the programmer to change features such as color scheme, size, and data redundancy. Particularly noteworthy is the possibility of selecting the level of data redundancy, as it ensures the QR codes remain intact and legible despite potential damage that may occur during printing and use. By choosing the “High” setting, “qrcode.js” ensures that up to 30% of the surface of the QR code can be damaged without affecting its contents, guaranteeing readability even under less-than-ideal circumstances.[7,8]

A notable use case of external JavaScript modules is within the modal window equipped with access to the device's onboard camera. For integrating with the system's camera and making the scanning of QR codes possible, the application relies on the "ZXing.js" library. This library not only facilitates camera access but also allows the application to easily scan and extract data from QR codes. Prior to initiating camera access, appropriate permissions are needed from the user. Upon successful scanning, the application compares the scanned data against the stored hashes associated with each guest of the hotel. In the event of a match, the Scans page is updated, rendering a new component that displays the name of the user who has just scanned their QR code. To maintain data integrity and mitigate any potential duplications, the application's Scans page swiftly alerts the manager when duplicate scanning occurs within a short timeframe. If the QR code fails to resemble any information within the stored database records, the application labels it as invalid, initiating no further actions.[9]

The aforementioned modules play an important role within the modal windows dedicated to the creation and printing of QR codes. These modules allow the application to generate and display QR codes, enabling effortless scanning and extraction of information.

5 Conclusion and Future Developments

This paper presents a simple approach for creating a user-friendly, simple and efficient solution for small hotel and restaurant management, focused on streamlining staff operations. The application described in this paper seamlessly integrates Svelte.js and SvelteKit.js for smoother and more responsive navigation experience and employs MongoDB for scalable data management. Additionally, it generates unique QR code ticket IDs on thermal paper for enhanced data privacy and fraud prevention. The application helps automate various hotel and restaurant management processes, such as guest bookings, room availability tracking, and canteen operations. This will have a direct impact on efficiency, meaning a reduction in manual workload, time saving and minimized errors. However, the application relies on internet connectivity, making it vulnerable to disruptions during network issues, and there may also be a potential for temporary disruptions due to software bugs or compatibility issues.

The modular component design and flexible page architecture of the application allow for scalability and potential expansion into additional areas such as inventory management. The integration of analytics, reports and data visualization can provide valuable insights to hotel owners and managers, enabling data-driven decision-making and further optimization of operations. Adding a payment system and a revenue estimation component will also prove useful for making informed decisions as a manager. Data entry operations inside the application can be adapted to accept .csv files for simplifying the repetitive task of entering client data, rooms and reservations one by one. Reports can also be offered in printable formats.

The application incorporates simple yet useful protective measures to help guard guest information and follow basic data protection guidelines by publicly presenting only hashed data in the form of QR codes printed on thermal receipt paper.

Overall, this management application excels in its user-centric UI design, technological integration, and data security, providing a free alternative solution for small hospitality businesses.

Acknowledgement: This work was supervised by Professor *Dana Simian*, from *"Lucian Blaga" University of Sibiu*.

References

[1] *How to Get Your Hotel Off of OATs,* Stephen Alemar, accessed on 10.05.2023 https://www.canarytechnologies.com/post/how-to-get-your-hotel-off-of-otas
[2] *Best Hotel Management Software in 2023,* G2, accessed on 10.05.2023 https://www.g2.com/categories/hotel-management
[3] *OPERA Hotel Property Management SOlution (PMS),* Official Website, accessed on 10.05.2023 https://www.oracle.com/ro/hospitality/products/opera-property-services/
[4] *GuestPoint,* Official Website, accessed on 10.05.2023 https://www.guestpoint.com/
[5] *HotelTime solutions,* Hotel Tech Report reviews page, accessed on 10.05.2023 https://hoteltechreport.com/food-and-beverage/restaurant-management/vento-epos
[6] *All-in-one software solutions for restaurants, food service and hotels,* Official Website, accessed on 10.05.2023 https://www.lsretail.com/industries/hospitality-software
[7] *Innkeeper's Advantage,* Official Website, accessed on 10.05.2023 https://www.innkeepersadvantage.com/
[8] *Svelte - Cybernetically Enhanced Web Apps,* Svelte Official Web Page, accessed on 10.05.2023 https://svelte.dev/
[9] *SvelteKit - Web Development Streamlined,* SvelteKit Official Web Page, accessed on 10.05.2023 https://kit.svelte.dev/
[10] *SvelteKit Docs,* Official SvelteKit Documentation, accessed on 10.05.2023 https://kit.svelte.dev/docs/
[11] *MongoDB Docs,* Official MongoDB Documentation, accessed on 10.05.2023 https://www.mongodb.com/docs/manual/
[12] *SvelteKit Routing,* SvelteKit Official Documentation, accessed on 10.05.2023 https://kit.svelte.dev/docs/routing

[13] *Svelte Docs*, Official Svelte Documentation, accessed on 10.05.2023 https://svelte.dev/docs
[14] *CryptoJS*, Official Documentation Page, accessed on 10.05.2023 https://cryptojs.gitbook.io/docs/
[15] *QRCode.js*, Official qrcode.js Repository, accessed on 10.05.2023 https://davidshimjs.github.io/qrcodejs/
[16] *ZXing Library*, Official GitHub Repository, accessed on 10.05.2023 https://github.com/zxing/zxing

Corina Minodora IORDACHE
Lucian Blaga University of Sibu
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: corinaminodora.iordache@ulbsibiu.ro

Felix HUSAC

ROMANIA
E-mail: husacfelix@gmail.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Control and Simulation for an Electrically Controlled Pneumatic Suspension

Maria-Denisa Kiss

Abstract

This article presents the process of electronically controlling an air suspension system. The developed system enables the user to choose between different operating modes, such as Sport, Off-Road, or Normal, through buttons. Additionally, it includes a simulation function of a road with bumps, allowing the reaction of the pneumatic suspension to be visualized in such driving conditions. All of these actions can be viewed on an LCD (Liquid Crystal Display) where the graphical part of the suspension is implemented.

1 Introduction

Pneumatic suspension is a type of suspension system that has revolutionized the automotive industry, providing a smoother, more comfortable ride for passengers and improving the overall performance and safety of vehicles [1]. This innovative technology has been used in various industries, especially in the automotive sector, and has become an important feature in modern vehicles, particularly luxury cars and SUVs, as well as commercial vehicles such as buses and trucks.

The history of suspension systems dates back to the 19th century when the first rudimentary versions were introduced [2]. Since then, suspension technology has undergone significant advancements, with the development of pneumatic suspensions in the 1950s being a major breakthrough [3]. Pneumatic suspension uses compressed air to support the weight of the vehicle, providing a more stable and comfortable ride, regardless of the terrain or driving conditions.

Over the years, the pneumatic suspension system has become more advanced, and today's systems offer a range of features that improve ride quality, safety, and performance. For example, the adjustability of the system allows for changes in ride height, which is useful when driving on different types of terrain. Additionally, modern pneumatic suspension systems incorporate sophisticated computer control systems, which optimize the suspension for different driving conditions.

2 Description of the project

This article presents a demonstration stand featuring an electrically controlled air suspension system that provides users with the ability to select between different operating modes, such as

Sport, Off-Road, or Normal mode, using buttons or a user interface. The concept for this demonstration stand originated from the hardware team, who developed the physical components. However, in order to make the stand fully functional, the addition of a software program is required, so I required to work on the software program. The software program serves as the bridge between the user inputs and the air suspension system, allowing for control of the suspension's behavior. By integrating the software, the demonstration stand can showcase the capabilities of the pneumatic suspension concept to potential clients, offering a practical understanding of its benefits.

2.1 Used technologies

This project involves the integration of hardware and software technologies. The air suspension stand cannot function without both its hardware components, which ensure proper distance, motion, and control, and a software program that operates to achieve good functionality. The interdependence between the hardware and software highlights the importance of seamless integration between technology domains in modern projects.

2.1.1 Software technologies

The suspension control system has been developed in MATLAB Simulink, whereas the graphical interface displayed on the LCD screen has been created using the Arduino IDE. MATLAB Simulink is a software platform used for graphical simulation of dynamic systems. It enables the creation of system models and analysis of their behavior over time. In Simulink, models are represented graphically by interconnecting functional blocks that represent system elements. The main advantage of using model-based development for creating the functionality is the ease of manipulating functions [4]. The high-level language of MATLAB makes it easier to handle blocks and interconnect them to create desired functionality compared to writing code in C/C++. Additionally, MATLAB provides a wide range of built-in math, engineering, and plotting functions that can be used to quickly analyze and visualize data collected from the Arduino [5].

The second platform used for this project is the Arduino IDE, which is primarily written in C++. It provides a graphical text and drawing capability for the LCD screen, which is also connected to an Arduino board. This platform includes functions specifically designed for drawing on the LCD.

2.1.2 Hardware technologies

The hardware technologies employed in this project include: two Arduino Mega 2560 boards (Fig. 2), two ultrasonic sensors (Fig. 3), buttons (Fig. 6), two DC (direct current) motors (Fig. 11) and an LCD (Liquid Crystal Display) screen (Fig. 12).

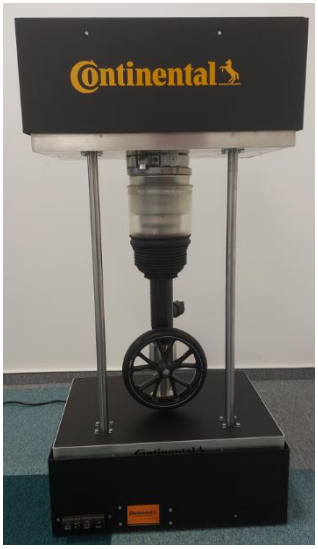


Fig. 1: The Air Suspension Demonstration-Stand

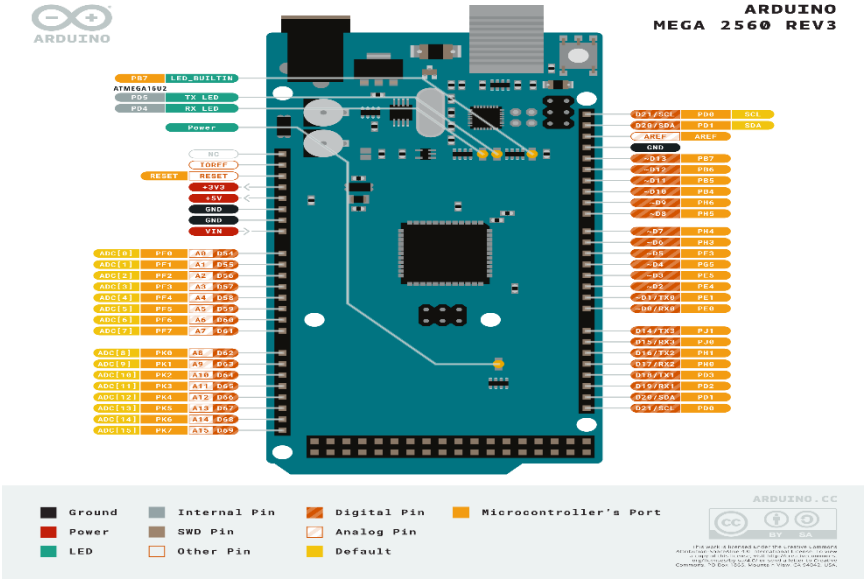


Fig. 2: Arduino Mega 256 [8]

The software program for the demonstration stand is deployed in the memory of an Arduino Mega 2560 board. In addition, a second Arduino Mega board is utilized specifically for the LCD since the LCD screen is directly connected to the secondary board. Embedded developers greatly appreciate the Arduino Mega 2560 due to its extensive range of ports and impressive processing power. This hardware part is quite helpful for carrying out complicated projects because it has adaptable I/O pins and strong performance characteristics [6]. Developers may easily incorporate various sensors and auxiliary devices with the Arduino Mega 2560, enabling the development of interactive systems.

The specifications of the Arduino Mega 2560 board are: Microcontroller: ATmega2560; Operating Voltage: 5V; Input Voltage (recommended): 7-12V; Input Voltage (limits): 6-20V; Digital I/O Pins: 54 (of which 14 provide PWM output); Analog Input Pins: 16; Hardware serial ports (UARTs): 4; DC Current per I/O: Pin 40 mA; DC Current for 3.3V: Pin 50 mA; Flash Memory: 256 KB of which 8 KB used by bootloader; SRAM: 8 KB; EEPROM: 4 KB; Clock Speed: 16 MHz [12].

The **ultrasonic sensors** are one of the important hardware components, which measures the current distance from the mobile plate to the fixed plate.

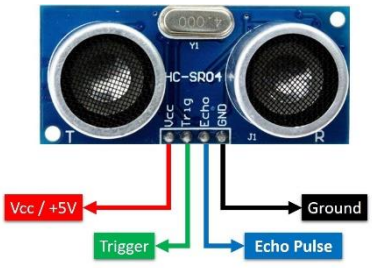


Fig. 3: HC-SR04 ultrasonic sensor [9]

The **signal processing** of the ultrasonic sensor occurs as can be seen in Figure 5: Upon the reception of a 10μs high-level signal at the I/O trigger (topmost signal line), the module commences the emission of eight 40 kHz pulses (middle signal line). Subsequently, it monitors for the presence of a returning pulse signal. Should it detect such a pulse signal, as indicated by a high-level signal, the duration of the high output IO (bottommost signal line) signifies the round-trip time for the ultrasonic waves traveling to and from the sensor and back [7].

Fig. 4: Fixed and mobile plate

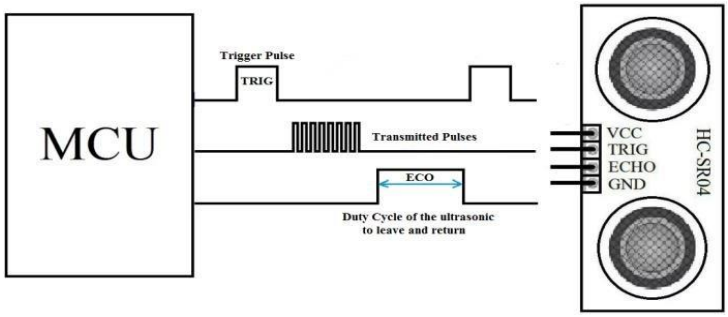


Fig. 5: Timing Diagram for trigger and echo signals [10]

The **specifications of the HC-SR04 ultrasonic sensors** are**:** Working Voltage: 5V; Working Frequency: 40 Hz; Operating distance: 2 cm – 4 m; Measuring Angle: 30 degrees.

The HC-SR04 ultrasonic module offers several advantages, including ease of use, precision of 3mm, low energy consumption, and the use of only four pins (5V and ground pins, and two digital pins for trigger and echo signals) [12].

As it is mentioned at the beginning of the article, **buttons** are used as a means of communication between the user and the demonstration stand. They make it possible to move the suspension to the desired Target level, and also the LEDs on it can indicate the target level we are in, as well as when the simulation of the bumpy road is active.

Fig. 6: Buttons (Down, Up, ESP)

As previously mentioned, three **target levels** have been established: *Normal*, *Off-Road*, and *Sport*. These levels simulate the function of adjusting the ground clearance of a car with air

suspension, aiming to enhance passenger safety and comfort. The *Normal Level* serves as the baseline, while *Off-Road* is used for navigating rugged terrain, and *Sport* is engaged to improve traction during high-speed driving.

In the following figures you can see the 3 ways of rides:

Fig 7: Normal Ride

Fig. 8: Sport Ride

Fig. 9: Off-Road Ride

The **ultrasonic sensor** measures a distance of 200 mm for *Normal Level*, which represents the height from the fixed plate to the mobile plate. Each level has a tolerance band of +/- 10 mm. If the measured height falls within this tolerance band, specifically for the *Normal Ride* where it ranges between 190 mm and 210 mm, it is considered to have reached the desired normal target level.

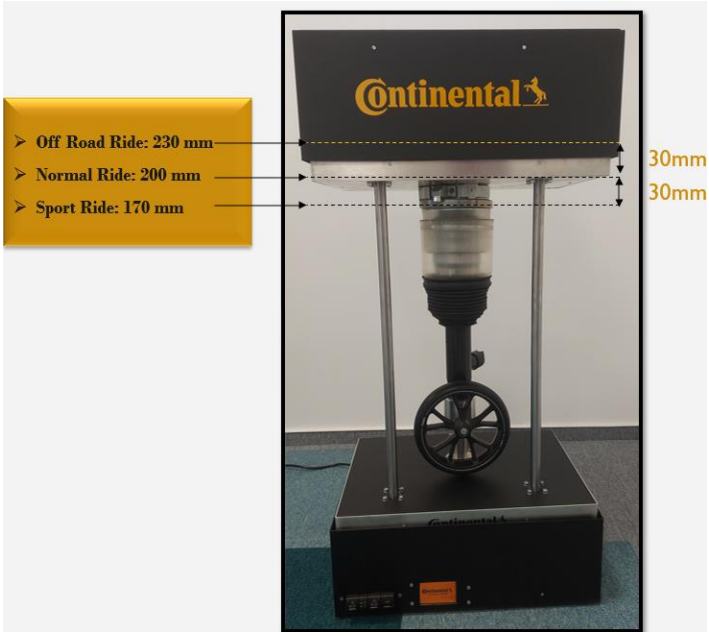


Fig. 10: The air suspension stand and the Target-Levels

DC (direct current) motors are used to facilitate the movement of the suspension up and down, allowing it to reach the desired target levels. Additionally, DC motors help simulate a road with uneven surfaces, providing characteristic motion for this type of road.

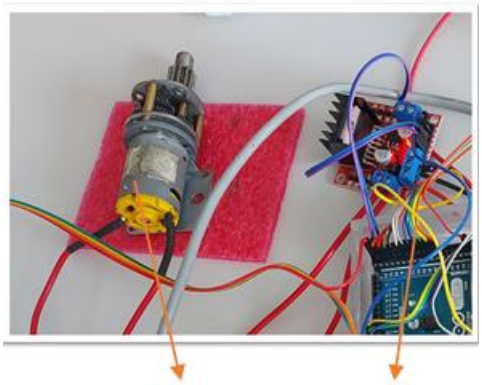


Fig. 11: DC Motor and H-bridge

By controlling the motors, the demonstration can realistically replicate the behavior of the suspension, enabling users to observe and understand how the pneumatic suspension system operates in various driving situations.

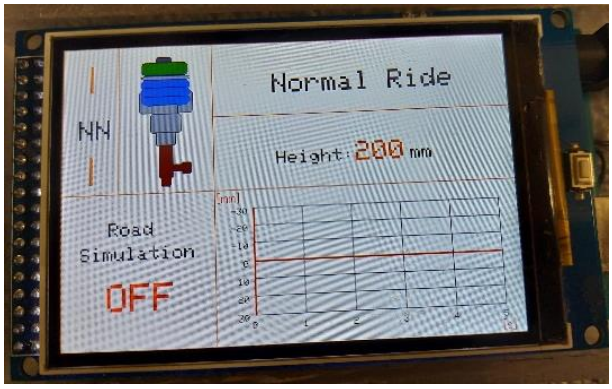
The H-bridge is responsible for controlling the motors and determining the direction in which the suspension moves. It plays an important role in managing the current flow to the motors, allowing them to rotate forward or reverse. By controlling the H-bridge, the demonstration stand can adjust the movement of the suspension system, enabling it to move up or down and providing flexibility in the direction of motion.

The implemented topics for DC motors:

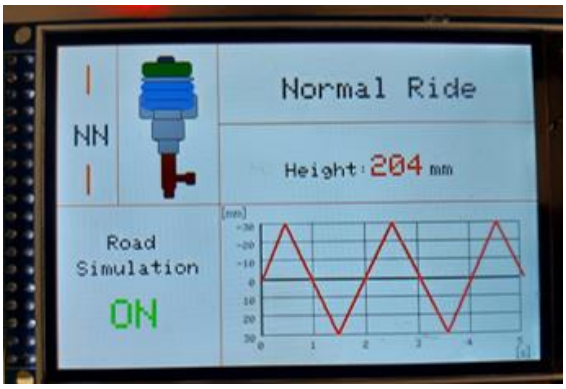
1. Ramp Up: This feature facilitates a gradual increase in the motor speed by 25%, ranging from 0% to 100%. It ensures a smooth and controlled acceleration, allowing the motors to reach higher speeds gradually.
2. Direction Change: The motors are designed to change their rotational direction to navigate from the Current Level to the desired Target Level. This functionality enables flexibility in motor movement and provides the ability to adjust the motor's position or trajectory as needed.
3. Software Protection: To safeguard the motors, a protective mechanism has been implemented. If the motors receive a continuous 100% duty cycle without any movement for an extended period, the software automatically reduces the duty cycle to 0%. This action stops the motors and prevents potential damage or overheating that could occur due to prolonged stalling, ensuring their safe operation.

The specifications of the DC motors are: Supply voltage: 12V; Dimensions: 12x36mm; Shaft diameter: 3x10mm; RPM (Rotations Per Minute): 300.

The **LCD (Liquid Crystal Display)** screen on the demonstration stand serves to provide real-time information about its current state. It displays the Target Level, indicating the desired suspension level, and a graphical representation of the suspension that visually shows its position. Furthermore, it indicates whether the road simulation is currently running or stopped.



a) Normal Ride without road simulation



b) Normal Ride with road simulation

c) Sport Ride without road simulation

d) Off-Road Ride without road simulation

Fig. 12**:** Color TFT LCD Display Module

The specifications of the LCD are: Size: 3.5"; Resolution: 320x240 dots (320x240 TFT); View Direction: 12H; Interface: SPI / MCU / RGB; Built-in Controller: SSD2119; Control-Board: Not included; Brightness: 400 cd/m²; Frame Through Hole: Optional; Touch Screen: No touch screen functionality.

2.2 Description of the Control System.

The project is divided into three parts: the functionality of the upper part of the suspension, the functionality of the lower part, which were implemented in Simulink, and the functionality of the LCD, which was implemented in the Arduino IDE.

2.2.1 Functionality of the upper suspension part

After the suspension is powered on from the source, regardless of the Target Level it is in, it will adjust to the Normal Target Level. The Normal Level means that when the ultrasonic sensor measures a height between 190 mm and 210 mm. To control the suspension in the upper part, the user will be able to use the Level Up and Level Down buttons. The Level Down button has been programmed to lower the ground clearance, while the Level Up button has been programmed to raise the ground clearance.

For these buttons, we have also implemented a function that will light up the LEDs on the buttons according to the Target Level that the suspension will reach (as shown in the previous section through photos).

In the upper part of the suspension, we have also defined three target levels, as mentioned earlier in this project. These are Normal, Sport, and Off-Road. Depending on the user's preference, they can reach the desired target level by interacting with these two buttons.

2.2.2 Functionality in the lower part of the suspension

This is achieved through the ESP button. When this button is pressed, the suspension will adjust to the Normal target level in the upper part, and in the lower part, a simulation of a bumpy road will begin. This will be done as follows:

In the upper part, the suspension will lower or raise the ground clearance, depending on which target level the road simulation is activated for, in order to reach the Normal target level. This will stop when the ultrasonic sensor measures a height between 190 mm and 210 mm.

In the lower part, the suspension will rise until it reaches a height between 200 mm and 220 mm, and it will lower until it reaches a height between 150 mm and 170 mm. This height is measured by the ultrasonic sensor fixed in the lower part. This motion will repeat until the ESP button is pressed again. When the ESP button is pressed for the second time, the suspension will rise to a height between 175 mm and 195 mm, which represents the Normal target level in the lower part of the suspension and in the upper part, the suspension will adjust from the Normal level to the target level it was in before the ESP button was pressed.

2.2.3 Functionality of the LCD

The functionality of the LCD was implemented in the Arduino IDE using the TFT library [11]. This library helped me construct the graphical image of the suspension, as well as the graphic indicating whether the road simulation is on or off. We drew the graphical part of the suspension using geometric shapes and calculated the pixels to ensure correct positioning.

Two dynamic arrows were drawn on the LCD to convey motion. One arrow, pointing downward, indicates the suspension moving down, while the other arrow, pointing upward, signifies the suspension moving up. To achieve the dynamic arrow effect, a black rectangular shape was drawn line by line, followed by a triangle drawn in the same manner. The process was repeated for a second arrow using an orange color. The arrows were drawn sequentially until a specific target level was reached, at which point the display showed the corresponding target level instead of the arrows.

The LCD also shows the height value obtained from the ultrasonic sensor located in the upper part of the suspension. Additionally, a graphical representation indicating the activation or deactivation of the bumpy road simulation was created using the same library used to draw the arrows from the graphic.

2.3 Software implementation

2.3.1 Software implementation of the upper suspension part

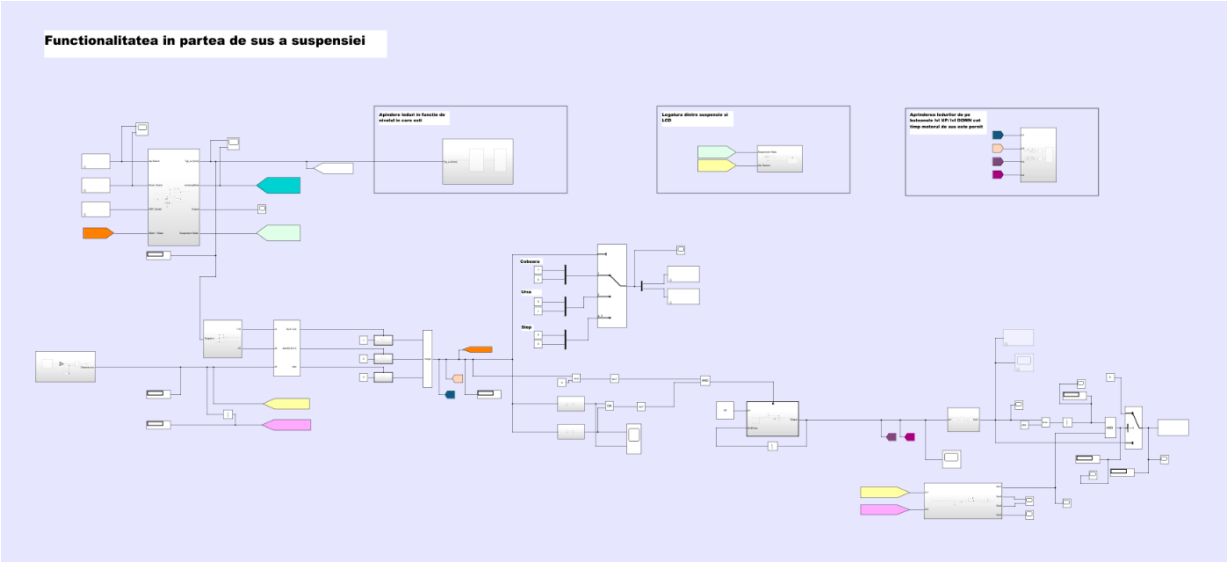


Fig. 13: Upper Part Functionality

To implement the upper part functionality, the process began with the programming of the buttons. Specifically, PIN 2 was assigned to the "Level Up" button, PIN 3 to the "Level Down" button, and PIN 4 to the "ESP" button, responsible for managing the lower suspension functionality.

For the "Level Up" and "Level Down" buttons, a counter was created to adjust the target level value by increments of 30. The initial program value was set at 200, corresponding to the "Normal" target level. When the "Level Up" button was pressed in the "Normal" state, the counter increased by 30, resulting in a value of 230, indicating the "Off-Road" target level. The counter was designed to remain locked at this level since it represents the uppermost level. Conversely, pressing the "Level Down" button in the "Normal" state decreased the counter to 170, which aligned with the "Sport" target level, and it remained locked at this value as it represents the lowest level. Users had the flexibility to reach any desired target level by utilizing these buttons.

Following this, values were transmitted to activate the LEDs based on the target level, with an additional tolerance margin of +/-10 target levels applied. Consequently, two values were generated. For instance, for the "Normal" target level with a value of 200, the values 190 and 210 were derived. These values were then compared with the data received from the sensor, which had been converted from meters to millimeters. If the sensor value exceeded the specified range, such as for the "Normal" target level, where the range was [190, 210], a signal of 1 was dispatched to lower the ground clearance. This action was achieved by setting PIN 23 to 1 and PIN 25 to 0, controlling the motor in the upper section to change its direction. Conversely, if the sensor value fell below the range, a signal of 2 was transmitted. This signaled the motor to adjust PIN 23 to 0 and PIN 25 to 1, subsequently raising the ground clearance. If the sensor value fell within the specified range, a signal of 3 was sent, denoting 0 for both pins and indicating that the motor had reached the desired target level. This same logic was applied to the "Off-Road" and "Sport" target levels.

To prevent motor overload, a counter was established, incrementing the duty cycle by 25% to gradually provide power ranging from 0% to 100%. These percentages were transformed into a PWM signal using the formula:

$$counter = \frac{(duty\ cycle * 255)}{100} \quad (1)$$

This formula was incorporated into a function to deliver the PWM signal to the motor situated in the upper section via PIN 8. Additionally, protective measures were implemented for both motors in the upper and lower sections. In the event that the motors encountered an issue and failed to respond for a duration of 1 second, an automatic shutdown mechanism was activated to prevent potential damage. This safeguard was achieved by monitoring the current and previous sensor values and calculating the derivative "df" using the formula:

$$df = \left|\frac{(current\ sensor\ value\ -\ previous\ sensor\ value)}{0.20}\right| \quad (2)$$

The value 0.20 represented the time interval between successive sensor readings for derivative calculation. This derivative value was subsequently processed and compared to a threshold of 5. If, for instance, the current sensor value equaled the previous sensor value, the derivative value would fall below 5, prompting the motor to cease operation. Conversely, if the derivative value exceeded 5, it did not impact the motor's behavior.

2.3.2 Software implementation of the lower suspension part

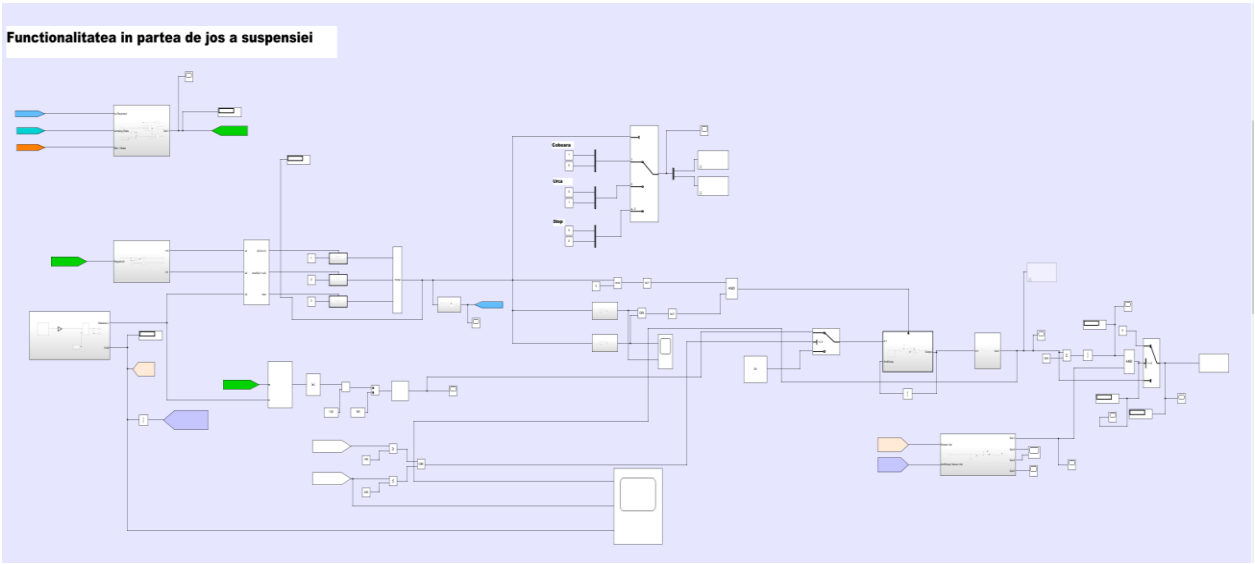


Fig. 14: Lower Part Functionality

To implement the lower suspension functionality, the system relies on the "ESP" button value. A counter was created for the "ESP" button, which toggles between 0 and 1. When the "ESP" button is pressed, the counter increments by 1, deactivating the "Level Up" and "Level Down" buttons. Pressing the "ESP" button again resets the counter to 0, reactivating the "Level Up" and "Level Down" buttons. Values are received from the "ESP" button counter and the state of the upper motor.

If a value of 3 is received from the upper motor, indicating that it has reached the "Normal" target level, adjustments are made in the lower part to simulate driving on uneven terrain. The

simulation begins by lowering the ground clearance to a level of 160 mm with a +/- 10 mm tolerance margin and then raising it to 210 mm with a similar tolerance margin. The same principles as the upper motor are applied in the lower part. If the sensor value is higher than the specified interval, the motor receives a signal of 1 to lower the ground clearance (PIN 27 set to 1 and PIN 29 set to 0). If the value is lower, it receives a signal of 2 (PIN 27 set to 0 and PIN 29 set to 1) to raise the ground clearance. If the value falls within the interval, the motor receives a signal of 3, indicating the target level has been reached, and it receives a signal of 210 to raise from the ground.

Similar to the 160 mm level, the 210 mm target level has a tolerance margin of +/- 10 mm, resulting in an interval of 200 to 220 mm. If the sensor value exceeds this interval, the ground clearance is lowered. If not, it is raised, and if the desired level is achieved, the value is reset to 160 mm to lower the ground clearance again. This process repeats until the "ESP" button is pressed again.

Upon the second press of the "ESP" button, the suspension adjusts to the "Normal" level in the lower part, defined by a value of 185 mm, also with a +/- 10 mm tolerance margin. If the sensor measures a value within this interval, the lower motor stops, and in the upper part, the system reverts to the target level it was in before the "ESP" button was pressed.

A protective mechanism for the lower motor was created, similar to the one in the upper part. It increases the duty cycle by 25% from 0% to 100% to avoid overloading the motor. This value is converted into PWM and transmitted on PIN 5 (ec.1), corresponding to the lower motor's PWM input.

In the lower part, a motor protection mechanism was also implemented to prevent overheating. It triggers a motor stop if no response is received for 2 seconds, considering the heavier load of the lower suspension. This protection is based on the derivative calculation mentioned earlier, and if the derivative value and the "Moving Average" value are both less than 10, the motor stops.

2.3.3 Software implementation of the LCD

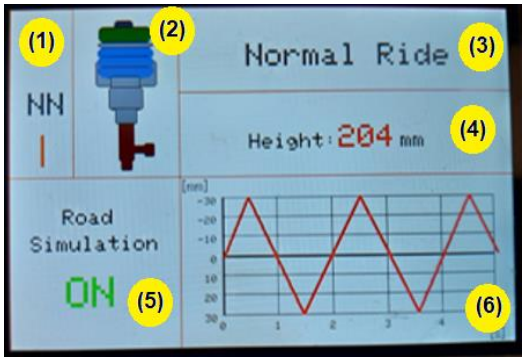


Fig. 15: The LCD with the 6 distinct frames.

The first frame, labeled as (1), corresponds to dynamic arrows and abbreviations indicating the current level reached. The second frame corresponds to various graphical representations of the pneumatic suspension based on the target level. The third frame displays the name of the target level at which the suspension is currently set. The fourth frame shows the value measured by the ultrasonic sensor. The two frames in the lower part correspond to frame (6), which represents whether the road simulation is on or off, and frame (5), which displays text indicating the status of the uneven road simulation. Additionally, when the suspension is activated, an image of the company logo where I work and my name will be displayed.

The application is structured into functions that create the components in the six distinct frames. The void setup(), is executed only once, while the void loop() function repeats as long as the Arduino board is powered.

Void setup() function

The void setup() function is a predefined function in Arduino, called only once when the program starts. In this function, the serial ports (Serial3 and Serial) are initialized with a baud rate of 38400 bits per second. Then, the TFT screen is initialized using the tft.init() function, and the SD card is checked and initialized with the SD.begin(SDC_CS) function. A command line is read from the Serial3 port, converted into an integer value, and stored in the variable command1. The orientation of the TFT screen is set using the tft.setRotation(1) function.

Next, a BMP (Bitmap) image is drawn on the TFT screen using the drawBMP("Conti.bmp", 0, 0, BU_BMP) function, where the image is placed in the upper-left corner of the screen. Finally, a 10-second delay is added using the delay(10000) function. All these actions prepare the necessary devices and settings before entering the loop() function, which runs in a continuous loop to execute the Arduino program.

Void loop() function

The void loop() function is a predefined function in the Arduino IDE that runs in a continuous loop until the Arduino board's power is turned off. The purpose of this function is to display components based on the values received from the application created in Simulink.

Firstly, with the help of the Serial3.readStringUntil('\n') function, I managed to read the values transmitted from the Simulink-created application. These values include data from the ultrasonic sensor, the state of the motor, and information about the activation of the uneven road simulation.

Using these values, I was able to call the necessary functions for each component and create appropriate logic. For example, when the suspension is activated, the transmitted value is 200, corresponding to the "Normal" target level. In this case, if the value measured by the ultrasonic sensor falls within the interval [190, 210], the motor will transmit the value 3, indicating the attainment of the target level. In this situation, the LCD screen displays the abbreviated name of the target level, the full name of the "Normal Ride" target level, the value measured by the ultrasonic sensor, the graphical representation of the suspension for "Normal Ride," and in the lower part, it indicates the inactivity of the uneven road simulation.

The components in the upper part, such as the abbreviated name of the target level, the graphical representation of the suspension, the full name of the target level, and the height, are updated based on the target level measured by the ultrasonic sensor. For example, if the motor value is 3, and the measured value falls within the interval [160, 180], the "Off-Road" target level is displayed (Fig. 12(c)). If the measured value is within the interval [220, 240], the "Sport" target level is displayed (Fig. 12(d)).

It's worth noting that I swapped the values of the "Off-Road" target level with that of the "Sport" target level to achieve a more realistic display. I did this to better reflect how a car's suspension height typically changes based on the selected mode. In the case of the "Off-Road" mode, the measured height is higher because the suspension raises the ground clearance, whereas in the "Sport" mode, the measured height is lower because the suspension lowers the ground clearance. In the Simulink program, it wasn't possible to implement this aspect realistically because the ultrasonic sensor couldn't be placed on the mobile part of the suspension to accurately measure its movement, as there was a risk of not measuring the height correctly.

The components in the lower part that indicate the inactivity of the uneven road simulation remain the same until the ESP button is activated.

In the case of pressing the Level Down and Level Up buttons, the motor will receive values 1 and 2, respectively. When the Level Down button is pressed, the program will receive value 1 and display a downward-pointing dynamic arrow instead of the abbreviated name of the target level. When the Level Up button is pressed and value 2 is received, an upward-pointing dynamic arrow will be displayed. These arrows will remain displayed until the program receives value 3, indicating the attainment of the target level. The graphical representation of the suspension and the name of the target level will remain unchanged until value 3 is received. Afterward, they will be modified based on the corresponding range of the value measured by the ultrasonic sensor.

Another scenario is when the ESP button is activated. In this case, the program has been implemented to transmit value 4. When the Arduino IDE application receives value 4, the following changes will occur: in the upper part of the screen, the suspension graphic will change to match the "Normal" target level, and the name of the target level will change to "Normal Ride." In the lower part of the screen, the text "Road Simulation ON" will be displayed (Fig. 12(b)), indicating that the road simulation is active, and the graph will display a linear function drawn using the graph() function. Upon pressing the button for the second time, the lower part of the screen will display the text "Road Simulation OFF," and the graph will no longer display the linear function (Fig. 12(a)).

3 Conclusion and future developments

In conclusion, the implemented system simulates the movement of an air suspension and enables control over its behavior using the software program. The demonstration stand is now utilized as a tool to showcase and to explain the functionality of an air suspension system.

As part of future hardware improvements, there are plans to enhance the components of the system. This includes upgrading the DC motor at the bottom part of the suspension to handle the weight more effectively and improve the accuracy of the road simulation. Additionally, integrating Bluetooth connectivity is being considered to enable wireless control and operation of the system. These developments aim to enhance the overall performance and functionality of the system, providing a more seamless and convenient user experience.

Acknowledgement: This work was supervised by Assoc. Prof. PhD *Laura Stoica*, from *Lucian Blaga University of Sibiu*.

References

[1] Prashant Lunawat, *How Does A Suspension System Work And What Are Its Various Types?,* https://www.scienceabc.com/innovation/how-does-a-suspension-system-work-and-what-are-its-various-types.html, (last update 22 Jan 2022)
[2] *Car suspension*, https://en.wikipedia.org/wiki/Car_suspension, (last edit, 17 May 2023)
[3] *Air Suspension: How Has it Developed over the Past 100 Years?*, https://blog.maxpeedingrods.com/air-suspension-how-has-it-developed-over-the-past-100-years/ (last edit, January 29, 2022)
[4] *Simulink, Simulation and Model-Based Design*, https://www.mathworks.com/help/simulink/, (last access, January 2023)
[5] *Arduino Programming with MATLAB and Simulink*, https://www.mathworks.com/discovery/arduino-programming-matlab-simulink.html, (last access, January 2023)
[6] *Arduino Mega 2560 Rev3*, https://docs.arduino.cc/hardware/mega-2560, (last access, January 2023)
[7] *Ultrasonic Ranging Module HC - SR04*, https://www.electroschematics.com/wp-content/uploads/2013/07/HCSR04-datasheet-version-1.pdf, (last access, January 2023)
[8] *Arduino Mega 2560 Pinout, Projects & Spec*, https://devboards.info/boards/arduino-mega2560-rev3, (last access, April 2023)

[9] *HC-SR04 Ultrasonic Sensor with ESP8266 NodeMCU – Measure Distance*, https://microcontrollerslab.com/hc-sr04-ultrasonic-esp8266-nodemcu-tutorial/, (last access, April 2023)
[10] Dlnya Abdulahad Aziz, *The Integration of Smart Anti-Collision Robotic System*, International Journal of Scientific and Engineering Research 9(7):335, 2018, https://www.researchgate.net/figure/H-C-SR04-Ultrasonic-Sensor-Mod-u-le-The-trigger-p-u-lse-of-d-u-ration-of-10-C-is-sent_fig3_326673128,
[11] *Arduino TFT graphical display*, https://www.arduino.cc/reference/en/libraries/tft/, (last access, January 2023)
[12] Constantin-Robert Ciolompu, *Autonomous navigation system based on data acquisition*, International Conference on Applied Informatics Imagination, Creativity, Design, Development, ICDD 2018, Sibiu, Romania.

Maria-Denisa KISS
Lucian Blaga University of Sibiu
Faculty of Sciences
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: mariadenisa.kiss@ulbsibiu.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

On Enforcing Database Constraints Using MS VBA Event-Driven Procedures and SQL Server

Diana Christina Mancas

Abstract

The goal of this paper is to provide a rigorous methodology for enforcing database constraints using MS VBA event-driven procedures for software applications built on top of MS SQL Server databases, in the framework of the software engineering Database Constraint-Driven Design and Development. The original contribution is a pseudo-code algorithm for assisting developers in this process. We exemplify the results of using it with some VBA code examples taken from a genealogy database software application designed and developed using this methodology. Consequently, it enforces all the database constraints -be they relational or not- which govern this sub-universe of discourse, thus guaranteeing the highest possible quality of its managed data. We also show that the proposed algorithm works fine not only for other relational database management systems, but even with NoSQL platform backends, and that the modifications needed to adapt it to other similar SQL embedding frontend platforms are minimal.

1 Introduction

In Mancas et al. [1] a Database Constraint-Driven Design and Development (DCDDD) methodology was introduced as a software engineering paradigm for guaranteeing the highest possible quality of the data managed by software applications. This methodology was very successfully used for the design and development of a software database (db) application managing genograms [2], with which the DCDDD approach was exemplified in [1].

DCDDD has the following 6 steps, in this order: sub-universe analysis; translation of the resulting Entity-Relationship data model [3] into a (Elementary) Mathematical Data Model ((E)MDM) [4] scheme; (E)MDM scheme validation and enhancement; corresponding Relational Data Model (RDM) [3] db generation; corresponding db software application generation driven by the non-relational db constraints; ergonomic polishing of the generated application Graphic User Interface (GUI).

The core last but one step has the following 3 sub-steps, in this order: for all constraints, detect all use cases in which they might be violated; establish the corresponding event-driven procedures needed to be coded; generate needed code for enforcing all db constraints that cannot be enforced by the host Database Management System (DBMS).

The first four steps, as well as the first sub-step of the fifth one are extensively discussed and illustrated in [1] with examples from [2]. The latter two sub-steps of the core step are only generally approached in [1], as they are heavily dependent on the technological platforms used.

This paper presents them in detail for the MS VBA and SQL Server platforms, with examples from [2], as well.

The *GENEALOGIES* [2] db software application manages genograms, i.e., genealogical trees extended with geographical, architectural, and touristic data. In the first DCDDD three steps, we identified 210 business rules that are governing this sub-universe [1, 2] and formalized them as db constraints. 172 of them are of the six RDM types: domain (range); not null; uniqueness (key); referential integrity (foreign key); tuple (check); and default values. The remaining 38 constraints are non-relational, but of the four (E)MDM categories: set; dyadic relationship; function; object.

Unfortunately, 19 relational constraints cannot be enforced by the MS SQL Server: 15 of them are keys that contain at least one column that is not null, and which might have more than one null value, and 4 are on VARBINARY objects used to store, via OLEDB embedding, high-resolution pictures of cities and monuments. Out of the latter 4, only 2, those of the not-null type, were enforced through VBA: those of the uniqueness type (i.e., no city or monument picture should be stored more than once) cannot be enforced through VBA either, as not even its Variant data type is capable of handling such huge memory objects.

Consequently, the VBA frontend *GENEALOGIES* application enforces 55 constraints, out of which 17 are relational and 38 are non-relational. VBA, the MS *Office Suite* programming language, was chosen especially for its rich panoply of db related event-driven procedure types, because it embeds SQL, but also for its simplicity and robustness.

1.1 Related work

The VBA programming platform is fully documented in [5]. There are thousands of papers on event-driven programming (e.g., [6 – 10]), a paradigm that arose from the finite state machines field.

Unfortunately, there are much fewer papers and books on non-relational database constraint enforcement.

Non-relational database constraints may be enforced in two ways: either using triggers of the DBMS' extended SQL (e.g., IBM's SQL PL, Oracle's PL/SQL, MS' T–SQL, etc.) or using event-driven procedures of a high-level programming language embedding SQL (e.g., MS VBA, C#, ASP, etc.).

For example, [11, 12] advocate the first one of these approaches, while [13, 14] advocate the second one. We prefer the second approach, for several reasons, namely:

1. Why passing to the backend DBMS unplausible values, when the frontend can much faster detect and reject them?
2. Generally, triggers are much slower than event-driven procedures.
3. Extended SQLs differ significantly between them (as there is no agreed extension standard), which makes triggers very hard to port from one DBMS to another. On the contrary, thanks to the *de facto* standards of ADOX and ODBC, any db software application needs only an almost insignificant programming effort to adjust its frontend to virtually any other relational DBMS.

1.2 Paper outline

The next section is the core of our original contribution, as it presents our proposed pseudo-code algorithm aimed to assist the design and development in VBA of the event-driven procedures needed to enforce both non-relational and relational constraints that cannot be enforced by the SQL Server (where ' followed by a space starts comments).

Section 3 provides corresponding code examples from [2], within the limits of the maximum allowed paper length: two apparently simple constraints were chosen for their coding complexity. The paper ends with a concluding and further work section, as well as references.

2 The algorithm for db constraint enforcement in VBA

Algorithm VBAConstraintEnforcement
Input: a constraint *c* and the set of forms $\{F_1, \ldots, F_n\}$, $n > 0$, containing at least one control whose data source is a db table column involved in *c*
Output: the set of forms $\{F_1, \ldots, F_n\}$ having their corresponding classes augmented with the VBA code needed for enforcing *c* as well
Strategy:
If c might be violated by time passing *Then FL*(*c*)
For i = 1 *To n*
  If c involves only column *f Then*
    If f is not a foreign key *Then ctrlBU*(*c*, *f*, F_i) *Else FC*(*c*, *f*, F_i) *End If*
  ElseIf c involves only columns *f* and *g* and *f* is the link column of sub-form F_i *Then*
    ctrlBU(*c*, *g*, F_i) *Else FBU*(*c*, F_i)
  End If
  FAU(*c*, F_i)
  FADC(*c*, F_i)
Next i
End Algorithm VBAConstraintEnforcement

Fig. 1. The proposed pseudo-code algorithm for assisting db constraint enforcement in VBA

FL(*c* AS Constraint)
‘ Use the *Form_Load* event procedure of the application’s *Menu* form to check and correct the db
‘ instance accordingly, if needed.
If the application’s *Menu* form does not have an associated *Form_Load* event procedure
Then add it to its class;
add in *Form_Load*() procedure of class *Menu* the code to check whether *c* is violated;
If this is the case *Then*
  adjust corresponding application parameters so that *c* is satisfied;
  display corresponding warning message;
End If
End Sub FL

Fig. 2. The method *FL* called by Algorithm *VBAConstraintEnforcement* from Fig. 1

ctrlBU(*c* AS Constraint, *f* As Control, F_i As Form)
‘ Use the *f_BeforeUpdate* event procedure to reject unplausible values.
If the application’s F_i form does not have an associated *f_BeforeUpdate* event procedure
Then add it to its class;
add in *f_BeforeUpdate*(*Cancel* As Integer) procedure of class F_i the following code:

```
If Not Cancel And f <> f.OldValue Then
  If f violates c Then
    Cancel = True
    Beep
    MsgBox “Corresponding error message”, vbCritical, “Request rejected…”
    f.Undo
  End If
End If
End Sub ctrlBU
```

Fig. 3. The method *ctrlBU* called by Algorithm *VBAConstraintEnforcement* from Fig. 1

FC(*c* AS Constraint, *f* As Control, F_i As Form)
‘ Use the *Form_Current* event procedure to eliminate unplausible values from combo-box *f*.
If the application’s F_i form does not have an associated *Form_Current* event procedure
Then add it to its class;
inject in the WHERE clause of the SQL SELECT statement that is the row source of *f* the code needed to eliminate from the *f*’s combo-box instance, for the current data row, all values that would violate *c* and save the resulting SELECT statement in a local string variable *s*;
add in *Form_Current*() procedure of class F_i the following code:

```
f.RowSource = s
f.Requery
```

End Sub FC

Fig. 4. The method *FC* called by Algorithm *VBAConstraintEnforcement* from Fig. 1

FBU(*c* AS Constraint, F_i As Form)
‘ Use the *Form_BeforeUpdate* event procedure to reject unplausible values.
If the application’s F_i form does not have an associated *Form_BeforeUpdate* event procedure
Then add it to its class;
add in *Form_BeforeUpdate*(*Cancel* As Integer) procedure of class F_i the following code (where f_1, …, f_m are the *m* columns involved in *c*, $m > 0$):

```
If Not Cancel And (f1 <> f1.OldValue Or … Or fm <> fm.OldValue) Then
   If f1, …, fm violate c Then
      Cancel = True
      Beep
      MsgBox “Corresponding error message”, vbCritical, “Request rejected…”
      f1.SetFocus
   End If
End If
```

End Sub FBU

Fig. 5. The method *FBU* called by Algorithm *VBAConstraintEnforcement* from Fig. 1

FAU(*c* AS Constraint, F_i As Form)
‘ Use the *Form_AfterUpdate* event procedure to synchronize data.
If the application’s F_i form does not have an associated *Form_AfterUpdate* event procedure
Then add it to its class;
add in *Form_AfterUpdate*() procedure of class F_i the code needed to enforce *c*;
End Sub FAU

Fig. 6. The method *FAU* called by Algorithm *VBAConstraintEnforcement* from Fig. 1

FADC(*c* AS Constraint, F_i As Form)
‘ Use the *Form_AfterDelConfirm* event procedure to synchronize data.
If the application’s F_i form does not have an associated *Form_AfterDelConfirm* event procedure
Then add it to its class;
add in *Form_AfterDelConfirm*(*Status* As Integer) procedure of class F_i the following code:

```
If Status = acDeleteOK Then
   ‘ add here code needed to enforce c
End If
```

End Sub FADC

Fig. 7. The method *FADC* called by Algorithm *VBAConstraintEnforcement* from Fig. 1

3 VBA code examples of enforcing non-relational constraints

The structural E-R diagram of the *GENEALOGIES* db [2] is the following:

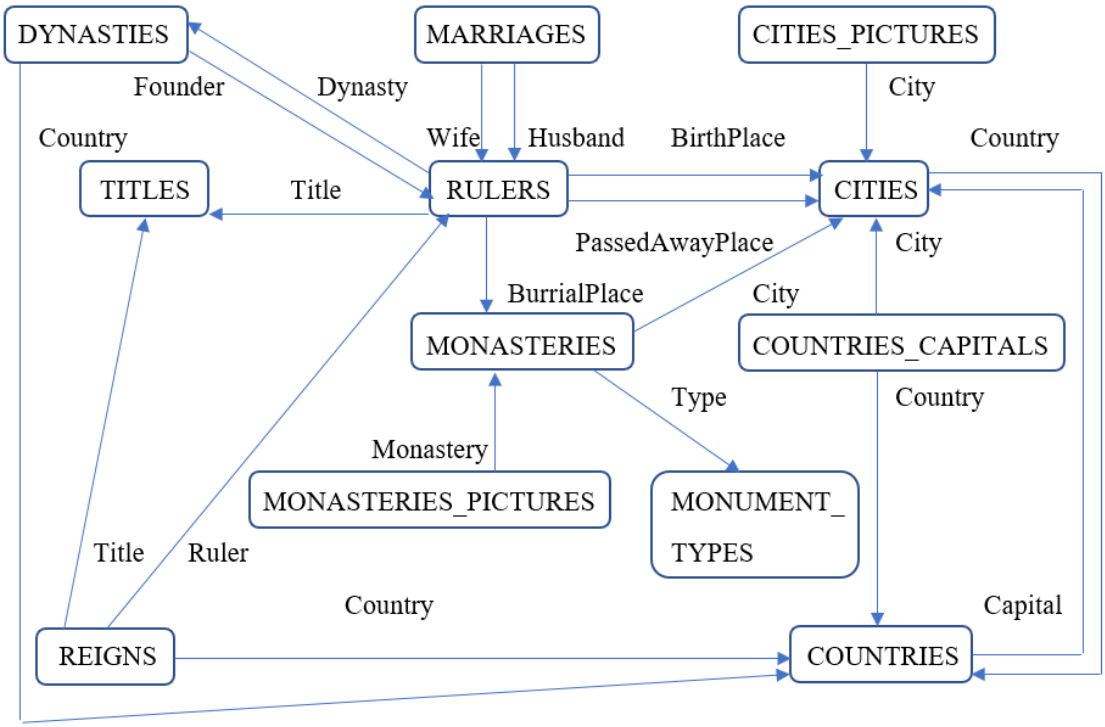


Fig. 8. The structural E-R diagram of the *GENEALOGIES* db

3.1 Constraint C_4 (nobody may live less than 0 or more than *maxLifeYears*)

This constraint might be violated either by the time that passes (as for living persons their pass away date is null) or by changing persons' birth or/and passed away dates [1, 2]. In theory, it is relational, but SQL Server may not enforce it as all 6 involved columns may store any number of null values. The corresponding VBA code for enforcing it is presented in Fig. 9, 10 and 11.

3.1.1 Code added to the *Form_Load*() method of class *Menu*

```
' enforce constraint C4
Dim maxAge As Variant
Dim Name As String
maxAge = DMax("Age", "Ages")        ' uses view Ages - see Fig. 9
If Not IsNull(maxAge) Then
  If maxAge > maxLifeYears Then     ' maxLifeYears is one of the app.'s parameters, stored in db table PARAMS
    Beep
    Name = DLookup("Name", "Ages", "Age =" & maxAge)
    MsgBox "At least " & Name & "'s age is " & maxAge & " years old, which is greater than " & maxLifeYears & _
        ", the current value of the parameter maxLifeYears: consequently, maxLifeYears has been automatically" & _
        " increased to & CInt(maxAge) + 1 & ", such as to satisfy constraint C4!", vbInformation, "Constraint C4 " & _
        "was violated by the passing time!"
```

Fig. 9. The VBA code added to the *Form_Load* procedure of class *Menu* to enforce constraint C_4

```
   maxLifeYears = CInt(maxAge) + 1
   DoCmd.RunSQL "UPDATE PARAMS SET maxLifeYears =" & maxLifeYears
  End If
End If
```

Fig. 9. (Continued)

This piece of VBA code calls the SQL view *Ages* (see Fig. 10), which computes the ages of all persons stored in the db (assigning -1 for those whose birth date is unknown).

This code is so complex because the *GENEALOGIES* application allows users to store data on persons born since the year 6,000 BC, with separate columns for their years, months, and days of both birth and death, as some of them might be only partially known (cases in which the application replaces months and days with guesses), as well as because VBA considers the minimum calendar date value as being January 1st 100 (also see Fig. 11).

3.1.2 Code added to the *Form_BeforeUpdate*(*Cancel* As Integer) method of class *PERSONS*

The *Form_BeforeUpdate* procedure from Fig. 10 was chosen because constraint C_4 concerns 6 columns of this table, namely *BirthYear*, *BirthMonth*, *BirthDay*, *PAYear*, *PAMonth*, and *PADay*. In fact, this code is also added to the *Form_BeforeUpdate* procedures of the classes of the *PERSONS* sub-forms *SIBLINGS*, *SIBLINGSofMOTHER*, *CHILDREN*, and *CHILDRENofMOTHER*, which are also based on the db table *RULERS* storing persons data.

```
SELECT Name, IIf([PassedYear]<0,Cdbl(Round((CDate(IIf(IsNull([BirthMonth]),"7",CStr([BirthMonth])) & "-" &
      IIf(IsNull([BirthDay]),"1",CStr([BirthDay])) & "-" & CStr(IIf([PassedYear]<=-100,-[BirthYear],100-[BirthYear])))-
      CDate(IIf(IsNull([PassedMonth]),"6",CStr([PassedMonth])) & "-" & IIf(IsNull([PassedDay]),"28",CStr([PassedDay])) &
      "-" & CStr(IIf(PassedYear<=-100,-PassedYear,100-PassedYear))))/365,2)),IIf(BirthYear<0,Cdbl(Round(((CDate(IIf(
      IsNull(BirthMonth), "7",CStr(BirthMonth)) & "-" & IIf(IsNull(BirthDay),"1",CStr(BirthDay)) & "-" & CStr(IIf(BirthYear<=-
      100,-[BirthYear],100-[BirthYear])))-#1/1/100#)+(CDate(IIf(IsNull([PassedMonth]),"6",CStr([PassedMonth])) & "-" &
      IIf(IsNull(PassedDay),"28",CStr(PassedDay)) & "-" & CStr(IIf([BirthYear]<=-100,[PassedYear],100+[PassedYear])))-
      #1/1/100#))/365,2)),Cdbl(Round((CDate(IIf(IsNull([PassedMonth]),"6",CStr(PassedMonth)) & "-" & IIf(IsNull(
      PassedDay),"28",CStr(PassedDay)) & "-" & CStr(IIf[BirthYear<100,PassedYear+100,PassedYear)))-CDate(IIf(
      IsNull(BirthMonth),"7",CStr(BirthMonth)) & "-" & IIf(IsNull(BirthDay),"1",CStr(BirthDay)) & "-" & CStr(IIf(BirthYear<
      100,BirthYear+100,BirthYear))))/365,2)))) AS Age, BirthYear, BirthMonth, BirthDay, PassedYear, PassedMonth,
      PassedDay, [#R]
From RULERS WHERE NOT BirthYear Is Null AND NOT PassedYear Is Null
UNION
SELECT Name, -1, BirthYear, BirthMonth, BirthDay, PassedYear, PassedMonth, PassedDay, [#R] FROM RULERS
WHERE BirthYear Is Null
UNION
SELECT Name, Cdbl(Year(Date()) - BirthYear), BirthYear, BirthMonth, BirthDay, PassedYear, PassedMonth, PassedDay,
      [#R]
FROM RULERS WHERE NOT BirthYear Is Null and PassedYear Is Null ORDER BY Age DESC , Name, [#R];
```

Fig. 10. The SQL code of view *Ages* invoked by the VBA code from Fig. 9

```
‘ enforce constraint C4
Dim by, bm, bd, pay, pam, pad As Integer          ' birth and passed away years, months, and days
Dim d1, d2 As Date                                ' adjusted min and max life dates
Dim d As Long                                     ' number of lived days for the current person
If Not Cancel And Not IsNull(BirthYear) And (BirthMonth <> BirthMonth.OldValue Or BirthDay <>
      BirthDay.OldValue Or PAYear <> PAYear.OldValue Or PAMonth <> PAMonth.OldValue Or PADay <>
      PADay.OldValue) Then
  by = BirthYear
  bm = IIf(IsNull(BirthMonth), 6, BirthMonth)          ‘ when not known birth month, assume it was June
  bd = IIf(IsNull(BirthDay), 1, BirthDay)              ‘ when not known birth day, assume it was 1st of bm
  pay = IIf(IsNull(PassedYear), Year(Date), PassedYear) ‘ current system year for persons alive
  pam = IIf(IsNull(PassedMonth), 7, PassedMonth)  ‘ when not known death month, assume it was July
  pad = IIf(IsNull(PassedDay), 28, PassedDay)      ‘ when not known death day, assume it was 28 of pam
```

Fig. 11. Code added to the *Form_BeforeUpdate* procedure of class *PERSONS* to enforce C_4

```
  If pay < 0 Then
    If pay < -100 Then
      by = -by
      pay = -pay
    Else
      by = 100 - by
      pay = 100 - pay
    End If
    d1 = CDate(pam & "-" & pad & "-" & pay)
    d2 = CDate(bm & "-" & bd & "-" & by)
  Else
    If by < 100 Then                                    ‘ translate birth and death dates with 300 years ahead
       by = by + 300
       pay = pay + 300
    End If
    d1 = CDate(bm & "-" & bd & "-" & by)
    d2 = CDate(pam & "-" & pad & "-" & pay)
  End If
 d = DateDiff("d", d1, d2)
  If d < 0 Then
    Cancel = True
    Beep
    MsgBox "Nobody may live less than 0 days!",vbCritical, "Please correct birth or/and passed away dates..."
  ElseIf d / 365.2425 > maxLifeYears Then
    Cancel = True
    Beep
    MsgBox "Nobody may live more than " & maxLifeYears & " years!", vbCritical, _
              "Please correct birth or/and passed away dates..."
  End If
  If Cancel Then BirthYear.SetFocus
End If
```

Fig. 11 (Continued)

3.2 Constraint C_{20} (For any marriage, both spouses must be simultaneously alive for at least one day.)

This constraint might be violated either by selecting as spouse of a marriage somebody who does not satisfy this condition or by modifying marriage and/or divorce dates or by modifying birth and/or death dates of a spouse [1, 2]. The corresponding VBA code is presented in Fig. 12, 13, 14, and 15.

3.2.1 Code added to the *Form_BeforeUpdate*(*Cancel* As Integer) method of class *PERSONS*

```
Dim q As String                          ' enforce constraint C20
Dim rs As ADODB.Recordset
' birth date > spouse passed away date?
If Not Cancel And Not NewRecord And Sex <> "N" And Not IsNull(BirthYear) Then
  If Not IsNull(DLookup("[#M]", "MARRIAGES", "Husband =" & [#R] & " Or Wife =" & [#R])) Then
    Set rs = New ADODB.Recordset         ' current person had at least one spouse
    If Sex = "M" Then                    ‘ compute the set of wives for the current male person
      q = "SELECT Name, PassedYear, PassedMonth, PassedDay FROM MARRIAGES INNER JOIN RULERS " _
                & "ON MARRIAGES.Wife = RULERS.[#R] WHERE Husband =" & [#R]
    Else                                 ‘ compute the set of husbands for the current female person
      q = "SELECT Name, PassedYear, PassedMonth, PassedDay FROM MARRIAGES INNER JOIN RULERS " _
                & "ON MARRIAGES.Husband = RULERS.[#R] WHERE Wife =" & [#R]
    End If
    rs.Open q, Application.CurrentProject.Connection, adOpenForwardOnly, adLockReadOnly
```

Fig. 12. The code added to the *Form_BeforeUpdate* procedure of class *PERSONS* to enforce C_{20}

```
    While Not rs.EOF And Not Cancel          ‘ check passed away dates spouse by spouse
     If Not IsNull(rs!PassedYear) Then       ‘ null values may not violate this constraint
       If rs!PassedYear < BirthYear Then     ‘ spouse dead before current person’s birth?
         Cancel = True
       ElseIf rs!PassedYear = BirthYear Then
         If Not IsNull(rs!PassedMonth) And Not IsNull(BirthMonth) Then
           If rs!PassedMonth < BirthMonth Then
             Cancel = True
           ElseIf rs!PassedMonth = BirthMonth Then
             If Not IsNull(rs!PassedDay) And Not IsNull(BirthDay) Then If rs!PassedDay < BirthDay Then _
                Cancel = True
           End If
         End If
       End If
    End If
     If Not Cancel Then
       rs.MoveNext
     Else
       Beep
       MsgBox Trim(Me!Name) & " was married to " & rs!Name & ", who died in " & rs!PassedYear & " before " & _
              IIf(Sex = "M", "him", "her") & " birth: please change birth date.", vbCritical, "In marriage, both " & _
              "spouses must be simultaneously alive for at least one day!"
     End If
    Wend
    rs.Close
    Set rs = Nothing
  End If
End If
' passed away date > spouse birth date?
If Not Cancel And Not NewRecord And Sex <> "N" And Not IsNull(PassedYear) Then
  If Not IsNull(DLookup("[#M]", "MARRIAGES", "Husband =" & [#R] & " Or Wife =" & [#R])) Then
    Set rs = New ADODB.Recordset          ' current person had at least one spouse
    If Sex = "M" Then                     ‘ compute the set of wives for the current male person
      q = "SELECT Name, BirthYear, BirthMonth, BirthDay FROM MARRIAGES INNER JOIN RULERS " & _
                                          "ON MARRIAGES.Wife = RULERS.[#R] WHERE Husband =" & [#R]
    Else                                  ‘ compute the set of husbands for the current female person
      q = "SELECT Name, BirthYear, BirthMonth, BirthDay FROM MARRIAGES INNER JOIN RULERS " & _
                                          "ON MARRIAGES.Husband = RULERS.[#R] WHERE Wife =" & [#R]
    End If
    rs.Open q, Application.CurrentProject.Connection, adOpenForwardOnly, adLockReadOnly
    While Not rs.EOF And Not Cancel       ‘ check birth dates spouse by spouse
     If Not IsNull(rs!BirthYear) Then     ‘ null values may not violate this constraint
       If rs!BirthYear > PassedYear Then  ‘ spouse born after current person’s death?
         Cancel = True
       ElseIf rs!BirthYear = PassedYear Then
         If Not IsNull(rs!BirthMonth) And Not IsNull(PassedMonth) Then
           If rs!BirthMonth > PassedMonth Then
             Cancel = True
           ElseIf rs!BirthMonth = PassedMonth Then
             If Not IsNull(rs!BirthDay) And Not IsNull(PassedDay) Then If rs!BirthDay > PassedDay Then _
                Cancel = True
           End If
         End If
```

Fig. 12. (Continued)

```
      End If
    End If
    If Not Cancel Then
      rs.MoveNext
    Else
      Beep
      MsgBox Me!Name & " was married to " & rs!Name & ", who was born in " & rs!BirthYear & _
        " after " & IIf(Sex = "M", "his", "her") & " death: please change passed away date.", _
        vbCritical, "In marriage, both spouses must be simultaneously alive for at least one day!"
    End If
   Wend
   rs.Close
   Set rs = Nothing
  End If
End If
```

Fig. 12. (Continued)

ADODB recordsets were needed in Fig. 12 also because both birth and passed away years, months, and days of persons, as well as marriage and divorce years might be null.

3.2.2 Code added to the *Form_Current* methods of classes *MARRIAGES* and *WMARRIAGES*

From Fig. 12 it should be noted that this constraint may not be totally enforced preventively in class *MARRIAGES*, for the same reasons as above. However, it is worth eliminating from the *Wife*'s combo-box, to begin with, at least all persons that do not have sex "F" and who were born after the corresponding divorce or husband's passed away years or were dead before the corresponding marriage or husband's birth years. Dually, this constraint may not be totally enforced preventively in the dual class *W_MARRIAGES* either, for the same reasons as above. However, it is worth eliminating from the *Husband*'s combo-box, to begin with, at least all persons that do not have sex "M" and who were born after the corresponding divorce or wife's passed away years or were dead before the corresponding marriage or wife's birth years. The corresponding code is almost identical: you should only replace *Wife* by *Husband* and 'F' by 'M'.

```
' partially enforce C20 (only for years)
Dim s, w, o As String
s = "SELECT RULERS.[#R], RULERS.Name & IIf(Not IsNull(FATHERS.Name),' of ' &  FATHERS.Name" & _
      ", IIf(Not IsNull(MOTHERS.Name), ' of ' & MOTHERS.Name,)) AS Ruler " _
    & "FROM (RULERS LEFT JOIN RULERS AS FATHERS ON RULERS.Father = FATHERS.[#R]) LEFT " _
     & "JOIN RULERS AS MOTHERS ON RULERS.Mother = MOTHERS.[#R] WHERE RULERS.Sex= 'F' "
o = " ORDER BY RULERS.Name, FATHERS.Name, MOTHERS.Name"
If Not IsNull(MarriageYear) Then
  If Not IsNull(DivorceYear) Then
    w = "And (RULERS.PassedYear Is Null or RULERS.PassedYear >= " & DivorceYear & _
                        ") AND (RULERS.BirthYear Is Null or RULERS.BirthYear <= " & MarriageYear & ")"
  Else
    w = "And (RULERS.PassedYear Is Null or RULERS.PassedYear >= " & MarriageYear & _
                        ") AND (RULERS.BirthYear Is Null or RULERS.BirthYear <= " & MarriageYear & ")"
  End If
Else                    ' current marriage year not known
  If Not IsNull(DivorceYear) Then
    w = "And (RULERS.PassedYear Is Null or RULERS.PassedYear >= " & DivorceYear & _
                        ") AND (RULERS.BirthYear Is Null or RULERS.BirthYear <= " & DivorceYear & ")"
```

Fig. 13. The VBA code added to the *Form_Current* procedure of class *MARRIAGES* to partially enforce constraint C_{20}

```
  Else
    If Not IsNull(Parent!PassedYear) Then
      If Not IsNull(Parent!BirthYear) Then
        w = "And (RULERS.PassedYear Is Null or RULERS.PassedYear >= " & Parent!BirthYear & _
                  ") AND (RULERS.BirthYear Is Null or RULERS.BirthYear <= " & Parent!PassedYear & ")"
      Else                  ' neither current person's marriage, divorce, or birth years known
        w = "And RULERS.BirthYear Is Null or RULERS.BirthYear <= " & Parent!PassedYear
      End If
    Else                    ' neither current person's marriage, divorce or passed away years known
      If Not IsNull(Parent!BirthYear) Then
        w = "And (RULERS.PassedYear Is Null or RULERS.PassedYear >= " & Parent!BirthYear
      Else                  ' neither current person's marriage, divorce, birth, or passed away years known
        w = ""
      End If
    End If
  End If
End If
Wife.RowSource = s & w & o
Wife.Requery
```

Fig. 13. (Continued)

3.2.3 Code added to the *Form_BeforeUpdate* method of class *MARRIAGES*

To fully enforce C_{20}, the code from Fig. 14 and 15 is absolutely needed in classes *MARRIAGES* (managing the set of wives of the current male person from *PERSONS*) and *W_MARRIAGES* (managing the set of husbands of the current female person from *PERSONS*), respectively.

```
' enforce C20
Dim v As Variant
Dim wby, wbm, wbd As Integer        ' wife's birth year, month, and day
Dim wpy, wpm, wpd As Integer        ' wife's passed away year, month, and day
Dim wb, wp As Date                  ' wife's birth and passed away dates
Dim hbm, hbd As Integer             ' husband's birth month and day
Dim hpm, hpd As Integer             ' husband's passed away month and day
Dim hb, hp As Date                  ' husband's birth and passed away dates
 If Not Cancel And (NewRecord Or Wife <> Wife.OldValue Or DivorceYear <> DivorceYear.OldValue) Then
  ' C20.1 wife born after divorce or husband's death?
  v = DLookup("BirthYear", "RULERS", "[#R] =" & Wife)
  If Not IsNull(v) Then
    wby = CLng(v)
    If Not IsNull(DivorceYear) Then
      If wby > DivorceYear Then     ' wife born after divorce?
        Cancel = True
        Beep
        MsgBox "For any marriage both spouses must be simultaneously alive for at least one day." & Chr(13) _
             & Wife.Column(1) & " was born in " & wby & " after divorcing!" & Chr(13) & "Please either " & _
             "change divorce year or choose another wife or first change her birth year accordingly.", _
             vbCritical, "Request rejected..."
        DivorceYear.SetFocus
      End If
    Else                            ' divorce year null: wife born after husband's death?
      If Not IsNull(Parent!PassedYear) Then
        v = DLookup("BirthMonth", "RULERS", "[#R] =" & Wife)
        wbm = IIf(IsNull(v), 6, v)      ' when not known wife birth month, assume it was June
        v = DLookup("BirthDay", "RULERS", "[#R] =" & Wife)
        wbd = IIf(IsNull(v), 28, v)     ' when not known wife birth day, assume it was 28 of wbm
        wb = CDate(wby & "-" & wbm & "-" & wbd)
```

Fig. 14. The VBA code added to the *Form_BeforeUpdate* procedure of class *MARRIAGES* to fully enforce constraint C_{20}

```
      ' when not known husband death month, assume it was July
      hpm = IIf(IsNull(Parent!PassedMonth), 7, Parent!PassedMonth)
      ' when not known husband death day, assume it was 1st of hpm
      hpd = IIf(IsNull(Parent!PassedDay), 1, Parent!PassedDay)
      hp = CDate(Parent!PassedYear & "-" & hpm & "-" & hpd)
      If DateDiff("d", wb, hp) < 0 Then
        Cancel = True
        Beep
        MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." _
          & Chr(13) & Parent!Name & " passed away on " & hp & ", while " & Wife.Column(1) _
          & " was born after that, on " & wb & "!" & Chr(13) & "Please choose another wife or " _
          & "first change these spouses passed away or/and birth dates accordingly.", vbCritical, _
          "Request rejected..."
        Wife.SetFocus
      End If
    End If
  End If
End If
If Not Cancel And (NewRecord Or Wife <> Wife.OldValue Or MarriageYear <> MarriageYear.OldValue) Then
 ' C20.2 wife dead before marriage date or husband's birth?
  If Not Cancel Then
    v = DLookup("PassedYear", "RULERS", "[#R] =" & Wife)
    If Not IsNull(v) Then
      wpy = CLng(v)
      If Not IsNull(MarriageYear) Then
        If wpy < MarriageYear Then      ' wife passed away before marriage?
          Cancel = True
          Beep
          MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." _
              & Chr(13) & Wife.Column(1) & " passed away in " & wpy & " before marrying!" & Chr(13) & _
              "Please either change marriage year or choose another wife or first change her passed away " & _
              "year accordingly.", vbCritical, "Request rejected..."
          MarriageYear.SetFocus
        End If
      Else                              ' marriage year null: wife passed away before husband's birth?
        If Not IsNull(Parent!BirthYear) Then
          v = DLookup("PassedMonth", "RULERS", "[#R] =" & Wife)
          wpm = IIf(IsNull(v), 7, v)        ' when not known wife passed away month, assume it was July
          v = DLookup("PassedDay", "RULERS", "[#R] =" & Wife)
          wpd = IIf(IsNull(v), 1, v)        ' when not known wife passed away day, assume it was 1st of wpm
          wp = CDate(wpy & "-" & wpm & "-" & wpd)
          ' when not known husband birth month, assume it was June
          hbm = IIf(IsNull(Parent!BirthMonth), 6, Parent!BirthMonth)
          ' when not known husband birth day, assume it was 28 of hbm
          hbd = IIf(IsNull(Parent!BirthDay), 28, Parent!BirthDay)
          hb = CDate(Parent!BirthYear & "-" & hbm & "-" & hbd)
          If DateDiff("d", hb, wp) < 0 Then
            Cancel = True
            Beep
            MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." _
              & Chr(13) & Parent!Name & " was born on " & hb & ", while " & Wife.Column(1) _
              & " passed away before that, on " & wp & "!" & Chr(13) & "Please choose another wife or " _
              & "first change these spouses passed away or/and birth dates accordingly.", vbCritical, _
              "Request rejected..."
            Wife.SetFocus
          End If
        End If
      End If
    End If
  End If
End If
```

Fig. 14. (Continued)

3.2.4 Code added to the *Form_BeforeUpdate* method of class *W_MARRIAGES*

```
' enforce C20
Dim v As Variant
Dim wby, wbm, wbd As Integer  ' wife's birth year, month, and day
Dim wpy, wpm, wpd As Integer  ' wife's passed away year, month, and day
Dim wb, wp As Date            ' wife's birth and passed away dates
Dim hbm, hbd As Integer       ' husband's birth month and day
Dim hpm, hpd As Integer       ' husband's passed away month and day
Dim hb, hp As Date            ' husband's birth and passed away dates
If Not Cancel And (NewRecord Or IsNull(Husband.OldValue) Or Husband <> Husband.OldValue Or
      IsNull(DivorceYear.OldValue) Or DivorceYear <> DivorceYear.OldValue) Then
 ' C20.1 Husband born after divorce or wife's death?
 v = DLookup("BirthYear", "RULERS", "[#R] =" & Husband)
 If Not IsNull(v) Then
   hby = CLng(v)
   If Not IsNull(DivorceYear) Then
     If hby > DivorceYear Then        ' Husband born after divorce?
       Cancel = True
       Beep
       MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." & _
          Chr(13) & Husband.Column(1) & " was born in " & wby & " after divorcing!" & Chr(13) & _
          "Please either change divorce year or choose another husband or first change his birth year " & _
          "accordingly.", vbCritical, "Request rejected..."
         DivorceYear.SetFocus
     End If
   Else                               ' divorce year null: husband born after wife's death?
     If Not IsNull(Parent!PassedYear) Then
       v = DLookup("BirthMonth", "RULERS", "[#R] =" & Husband)
       hbm = IIf(IsNull(v), 6, v)          ' when not known husband birth month, assume it was June
       v = DLookup("BirthDay", "RULERS", "[#R] =" & Husband)
       hbd = IIf(IsNull(v), 28, v)         ' when not known husband birth day, assume it was 28 of hbm
       hb = CDate(hby & "-" & hbm & "-" & hbd)
       ' when not known wife death month, assume it was July
       wpm = IIf(IsNull(Parent!PassedMonth), 7, Parent!PassedMonth)
       ' when not known wife death day, assume it was 1st of wpm
       wpd = IIf(IsNull(Parent!PassedDay), 1, Parent!PassedDay)
       wp = CDate(Parent!PassedYear & "-" & hpm & "-" & hpd)
       If DateDiff("d", hb, wp) < 0 Then
         Cancel = True
         Beep
         MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." _
           & Chr(13) & Parent!Name & " passed away on " & wp & ", while " & Husband.Column(1) _
           & " was born after that, on " & hb & "!" & Chr(13) & "Please choose another husband or " _
           & "first change these spouses passed away or/and birth dates accordingly.", vbCritical, _
           "Request rejected..."
         Husband.SetFocus
       End If
     End If
  End If
 End If
End If
If Not Cancel And (NewRecord Or Husband <> Husband.OldValue Or IsNull(MarriageYear.OldValue) Or
      MarriageYear <> MarriageYear.OldValue) Then
  'C20.2 husband dead before marriage date or wife's birth?
 If Not Cancel Then
   v = DLookup("PassedYear", "RULERS", "[#R] =" & Husband)
   If Not IsNull(v) Then
     hpy = CLng(v)
     If Not IsNull(MarriageYear) Then
```

Fig. 15. The VBA code added to the *Form_BeforeUpdate* procedure of dual class *W_MARRIAGES* to fully enforce constraint C_{20}

```
      If hpy < MarriageYear Then      ' husband passed away before marriage?
        Cancel = True
        Beep
        MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." & _
          Chr(13) & Husband.Column(1) & " passed away in " & hpy & " before marrying!" & Chr(13) & _
          "Please either change marriage year or choose another husband or first change his passed " & _
          "away year accordingly.", vbCritical, "Request rejected..."
        MarriageYear.SetFocus
      End If
    Else                              ' marriage year null: husband passed away before wife's birth?
      If Not IsNull(Parent!BirthYear) Then
        v = DLookup("PassedMonth", "RULERS", "[#R] =" & Husband)
        hpm = IIf(IsNull(v), 7, v)          ' when not known husband passed away month, assume it was July
        v = DLookup("PassedDay", "RULERS", "[#R] =" & Husband)
        hpd = IIf(IsNull(v), 1, v)          ' when not known husband passed away day, assume it was 1st of hpm
        hp = CDate(hpy & "-" & hpm & "-" & hpd)
       ' when not known wife birth month, assume it was June
        wbm = IIf(IsNull(Parent!BirthMonth), 6, Parent!BirthMonth)
        ' when not known wife birth day, assume it was 28 of wbm
        wbd = IIf(IsNull(Parent!BirthDay), 28, Parent!BirthDay)
        wb = CDate(Parent!BirthYear & "-" & hbm & "-" & hbd)
        If DateDiff("d", wb, hp) < 0 Then
          Cancel = True
          Beep
          MsgBox "For any marriage, both spouses must be simultaneously alive for at least one day." _
            & Chr(13) & Parent!Name & " was born on " & wb & ", while " & Husband.Column(1) _
            & " passed away before that, on " & hp & "!" & Chr(13) & "Please choose another husband or " _
            & "first change these spouses passed away or/and birth dates accordingly.", vbCritical, _
            "Request rejected..."
          Husband.SetFocus
        End If
      End If
    End If
  End If
 End If
End If
```

Fig. 15. (Continued)

4 Conclusion and further work

In the framework of the Database Constraint-Driven Design and Development methodology [1], we designed, used, and proposed an algorithm for assisting the process of designing the frontend MS VBA classes of the Windows forms, aimed to enforce both non-relational and relational constraints that are not enforceable by the backend relational DBMSes. Where needed and possible, we also provided actual corresponding VBA code patterns. We exemplified the result of applying this algorithm with the VBA code enforcing two such constraints taken from [2].

The algorithm proposed in Section 2, which is the main contribution of this paper, can be easily adapted for similar programming platforms. For example, in MS .Net there exist similar to VBA event-driven procedures, only having different names (e.g., *BeforeUpdate* is called *Validating*, *AfterUpdate* is called *Validated*, etc.). Dually, thanks to the *de facto* standardization of both ADOX and ODBC, this algorithm may be used without modifications for the design of VBA software frontends based on other relational DBMSes (e.g., Oracle's DB and MySQL, IBM's DB2, Postgres DB, MS Access, etc.): only minor changes in VBA coding are needed (e.g., correspondingly replacing constants storing DBMS error codes, taking care of particular behaviors, like the fact that autonumbering values are generated by MS Access immediately after users type a character on a new data line, which makes them available as early as the

Form_AfterInsert event-driven procedure, while MS SQL Server does it only after saving the new data line, which makes them available only in the *Form_AfterUpdate* event procedure, etc.).

Moreover, our proposed algorithm could also be used for software applications built on top of NoSQL DBMSes: as they lack support for even almost all the relational constraint types, the only difference is that the set of the relational constraints to be enforced by the frontend is much larger, but the algorithm of enforcing them is the same as for the relational DBMS ones.

In [2] we also analyzed 8 of the top 10 genealogical software applications of 2023, as considered by [15] (the other 2 are unavailable to us: one is meant only for Apple platforms and the other has no free version). Unfortunately, only a few of them are at most issuing a couple of warnings when trying to save unplausible data, which you can dismiss and save it, however. All of them have glossy interfaces, but are ok with managing aberrant data like people being born after their christening, last marriage, having children, and even being dead, or people being buried before death, or living some negative years, or huge positive ones, etc.

In sharp contrast, the *GENEALOGIES* application proposed in [2], which uses the DBCDDD methodology, and the algorithm proposed in this paper, provides the highest possible quality of its managed data, which is satisfying all constraints governing this sub-universe.

Further work could be done: first, to exemplify the power of our proposed algorithm even in the frameworks using NoSQL DBMSes (e.g., MariaDB, MongoDB, Cassandra, Oracle NoSQL DB, etc.); secondly, devising similar algorithms for other SQL embedding platforms (e.g., C#, Java, Python, etc.).

Acknowledgement: This work was supervised by Associate Professor Cristina Șerban, from the Math. & Computer Science Dept. of the Ovidius University at Constanta, Romania.

References

[1] Christian Mancas, Cristina Serban and Diana Christina Mancas. On Software Application Database Constraint-Driven Design and Development. *Journal of Computer Science Research,* 5(1): 31–45, DOI: https://doi.org/10.30564/jcsr.v5i1.5476, 2023.

[2] Diana Christina Mancas. *Design and Development of a Database Software Application for Managing Genealogical Trees*. MSc. Dissertation Thesis submitted to the Math. & Comp. Sci. Dept. of the Ovidius University at Constanta, Romania (in press, in Romanian), 2023.

[3] Christian Mancas. *Conceptual Data Modeling and Database Design: A Completely Algorithmic Approach. Volume I: The Shortest Advisable Path*. Apple Academic Press / CRC Press (Taylor & Francis Group): FL, USA, ISBN-13: 978-1774635452, 2015.

[4] Christian Mancas. *MatBase* Constraint Sets Coherence and Minimality Enforcement Algorithms. In: Benczur, A., Thalheim, B., Horvath, T. (eds.), *Proc. 22nd ADBIS Conf. on Advances in DB and Inf. Syst., LNCS 11019*, pp. 263–277. Springer: Switzerland, DOI: https://doi.org/10.1007/978-3-319-98398-1_18, 2018.

[5] Microsoft Corp. *Office VBA Reference*. Office Visual Basic for Applications (VBA) reference | Microsoft Learn, 2023.

[6] Kim B. Bruce and Andrea Danyluk. Event-driven programming facilitates learning standard programming concepts. In: *Proc. Conf. on O-O Prog. Syst., Lang., and App. (OOPSLA)*, ACM, pp. 96–100, DOI: 10.1145/1028664.1028704, 2004.

[7] Stephen Ferg. Event-Driven Programming: Introduction, Tutorial, History. Microsoft Word - event_driven_programming.doc (cuny.edu), 2006.

[8] The Open University. *Event-Driven Programming*. Open University Course Team, USA, ISBN-13: 978-0749219925, 2009.

[9] Peter Mehlitz, Oksana Tkachuk, and Mateusz, Ujma. JPF-AWT: Model checking GUI applications. In: *Proc. IEEE/ACM Intl. Conf. on Automated Softw. Eng. (ASE'11)*, pp. 584–587. DOI: 10.1109/ASE.2011.6100131, 2011.

[10] Nrusingh Prasad Dash, Ranjan Dasgupta, Jayakar Chepada, and Arindam Halder. Event Driven Programming for Embedded Systems - A Finite State Machine Based Approach. In: *Proc. Intl. Conf. on Systems (ICONS 2011)*, IARIA, pp. 9–23, 2011.

[11] Geraldo Zimbrao, Rodrigo Miranda, Jano Moreira de Souza, Mario H. Estolano, Francisco P. Neto. Enforcement of Business Rules in Relational Databases Using Constraints. In: Alberto H. F. Laender (ed.), *Proc. XVIII Simpósio Brasileiro de Bancos de Dados (SBBD 2003)*, UFAM, pp. 129–141, 2003.

[12] Reza Sanati-Mehrizy and Afsaneh Minaie. Writing Triggers to Implement Business Rules in a Relational Database. ASEE Annual Conf., Paper Id #34908, American Society for Engineering Education, https://peer.asee.org/writing-triggers-to-implement-business-rules-in-a-relational-database.pdf, 2021.

[13] Christian Mancas. *Matbase* Autofunction Non-relational Constraints Enforcement Algorithms. *Intl. J. of Comp. Sci. and Inf. Techn.*, 11(5): 63–76, DOI: 10.5121/ijcsit.2019.11505, 2019.

[14] Christian Mancas. On Detecting and Enforcing the Non-Relational Constraints Associated to Dyadic Relations in *MatBase. J. of Electronic & Inf. Syst.*, 2(2): 1-8, DOI: https://doi.org/10.30564/jeisr.v2i2.2090, 2020.

[15] No1Reviews.com. Reviews of the Top 10 Genealogy Software of 2023. https://genealogy-software.no1reviews.com, 2023.

Diana Christina MANCAS
Ovidius University
Mathematics and Computer Science Department
Bd. Mamaia 123, Constanta
ROMANIA
E-mail: diana.mancas@365.univ-ovidius.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Prism: Democratizing Artificial Intelligence through a Low-code Approach

Matei-Ioan Marin

Abstract

Prism is an innovative low-code web tool that enables users to create and publish custom artificial intelligence web applications without requiring extensive technical expertise. Despite AI being a powerful technology, its day-to-day accessibility remains limited for many individuals. This tool seeks to address this issue by providing a dual-interface system, combining a user-friendly drag-and-drop design interface for constructing the visual part of the app and a modular graph visualization interface for defining the app's functionality. Users can integrate features like computer vision, language models, and their custom AI models into their web applications using pre-built modules. Thus, Prism democratizes access to AI technologies, empowering a broader audience to create and deploy AI-driven web applications. The paper will discuss the design principles, architecture, and implementation of the proposed tool, highlighting its potential impact on the AI development landscape.

1 Introduction

1.1 Motivation

Artificial Intelligence has witnessed an unprecedented rise in popularity over recent years, with its adoption more than doubling from 2017 to 2022 (*McKinsey*). It sparked interest even among non-technical communities, with terms like *AI*, *no-code AI* and *AI automation* reaching all-time highs in 2023 (*Google Trends*), mainly thanks to the rise of LLMs like chatGPT, which managed to reach 100 million users just two months after its launch (*The Guardian*). Alongside the growth and increased popularity of AI, low-code/no-code technologies are also gaining more attention, as they allow anyone to create applications without coding knowledge, and as they can save up to 90% of development time, as per a *McKinsey* report.

Despite the increasing popularity of Artificial Intelligence (AI) technologies, there remains a challenge in developing AI applications for individual needs and tasks. Though powerful general AI tools exist, many users lack the technical expertise required to customize these tools or integrate them for tailored tasks. Prism aims to address this issue by offering an intuitive platform that simplifies the process of creating personalized AI-driven web applications without requiring extensive technical skills. However, the democratization of Artificial Intelligence is a vast and complex concept due to the diversity of possible solutions, approaches for implementation, target-users and use-cases. Some of the available platforms trying to make AI more accessible have indeed managed to enable non-AI experts to take advantage of this technology, but this solutions

typically target regular programmers, thought for production use instead of personal use and are still inaccessible to the general population. Other no-code AI tools, even though intended for the general user have different approaches and tend to be niched. Prism envisages an ecosystem where AI tools, from Large Language Models to Computer Vision, can be harnessed by anyone regardless of their digital expertise.

1.2 Overview

Prism is a low-code web tool that empowers a diverse range of users to create custom web applications by combining AI modules and functionalities with a user-friendly drag-and-drop interface. It enables users to easily connect various modules, adjust the use of AI technologies, and develop tailor-made web applications to suit their specific needs.

To create this approachable platform, Prism employs modern technologies such as Nuxt 3 for the application framework, TailwindCSS for styling, and Firebase for authentication, database, and storage solutions. For user web application design, Prism integrates Bootstrap 5, which offers a set of pre-made components and a grid system for building responsive web applications. Additionally, Tensorflow.js and OpenAI are used for incorporating AI modules within the platform.

This paper discusses the design principles, concept, and implementation of Prism, highlighting its potential impact on the AI development landscape by enabling users to create personalized AI-driven web applications with ease.

2 Editor

The core component of the Prism application is its editor, which merges two separate interfaces: the Design Editor, which allows users to build and style their web app and the Functionality Editor, which enables them to add functionality to their platform.

2.1 Design Editor

2.1.1 Component Editor

The Component Editor (Fig.1), a crucial part of the Design Editor, enables users to create the content and layout of their interface. This drag-and-drop reflects real-time changes, ensuring a seamless design experience. The editor comprises a component library, which includes various HTML elements, pre-made components (e.g., webcam view that can be combined with an Object Detection Module to highlight detected objects), and a grid system based on Bootstrap, allowing for mobile-friendly development.

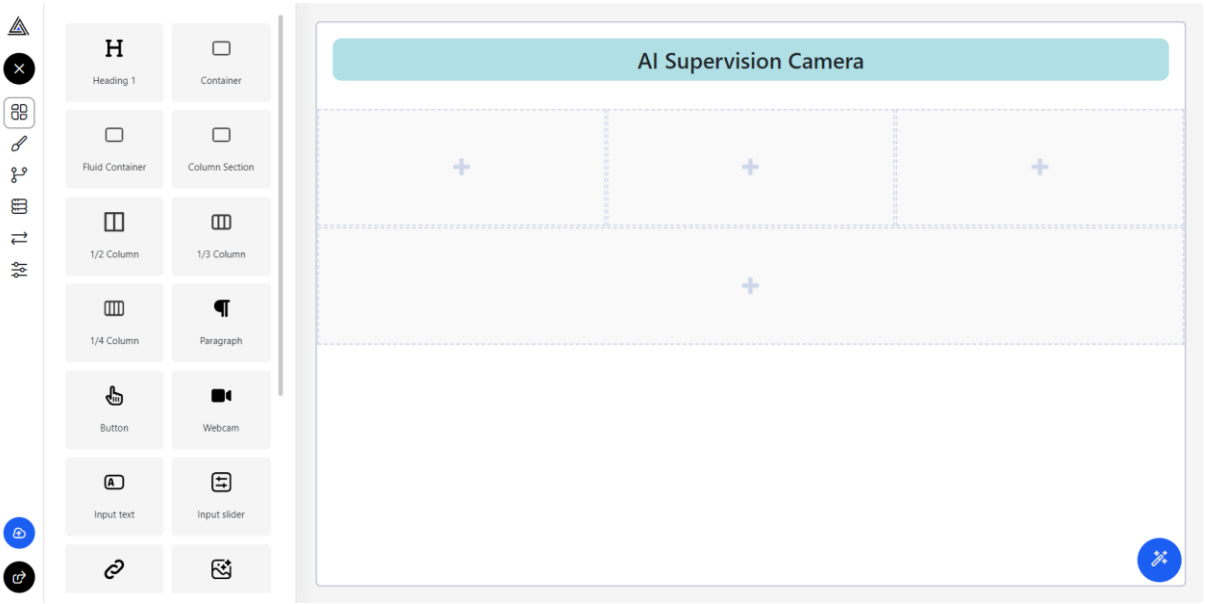


Fig.1 : Component Editor

2.1.3 Style Editor

The Style Editor (Fig.2) enables Prism users to modify the general styles of the selected component. This can be done by changing the value of the various input fields of the editor, each corresponding to a specific style attribute. When a change occurs, the appearance of the component is updated in real-time. Currently, the Style Editor supports several common style attributes, including margins, paddings, background color, text color, font size, font weight, text aligning, and border radius, with plans to expand its capabilities. This functionality is achieved by directly modifying the inline style of elements within the DOM.

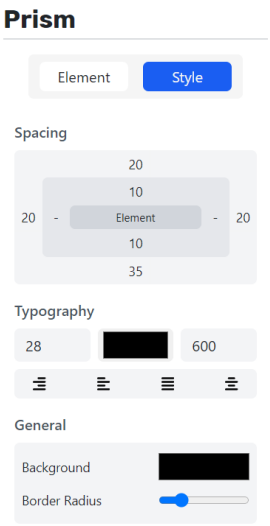


Fig.2 : Style Editor

2.2 Functionality Editor

2.2.1 Overview

Prism's functionality editor (Fig. 3) was designed to provide users with an intuitive and interactive platform for visually programming the logic and behavior of their AI-driven web application. It comprises a module library, which offers users a diverse array of pre-built modules. These are functions the user can easily integrate in his web app by means of the Graph Visualization Interface, which enables him to define the logic of his platform, just by visually connecting the modules of his choice.

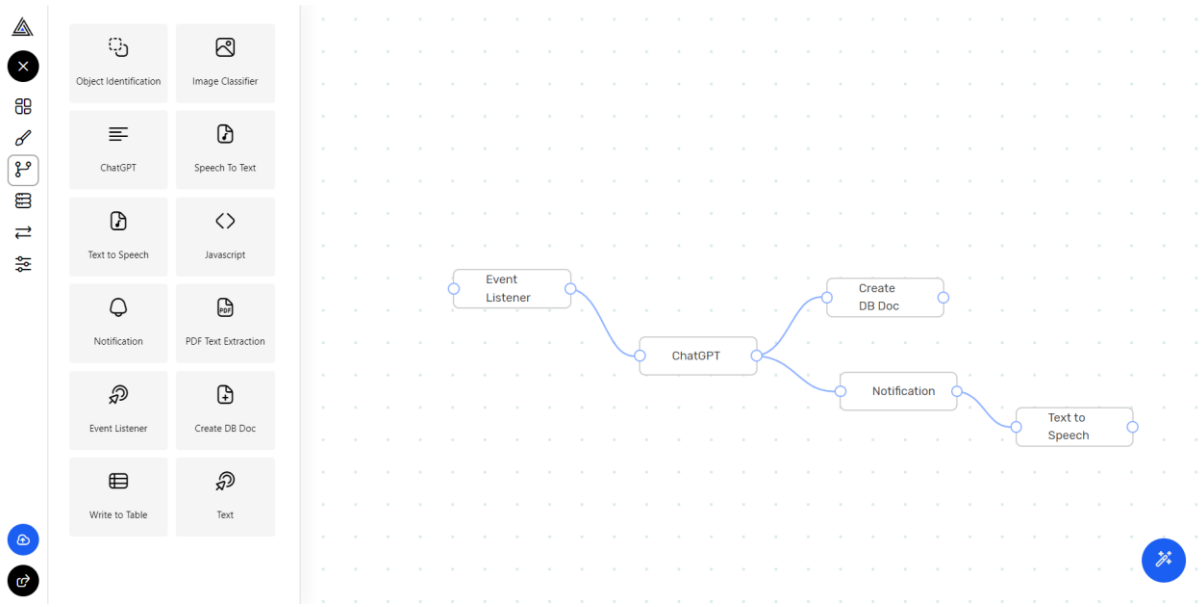


Fig.3 : Functionality Editor

2.2.2 Concept

The concept behind the Functionality Editor is its modular, visual approach to defining the logic of the user's application. It relies on three primary verticals: modules, connections and variables, which the user can visually customize.

The modules serve as building blocks of the applications' functionality, which the user can select from Prism's module library. Essentially, they are individual functions, each having its own purpose, which the users can combine to achieve the desired product. The modules reach a wide range of variety, from artificial intelligence functions (ex. Object Identification Module, chatGPT Module, Text-to-Speech Module) to simple conditional logic, thus allowing for great flexibility.

In order for the user to define in which order the modules are being executed he has to interconnect them. Connections depict the logic and flow of the user's application. Thus, by connecting two modules, users define how their program operates. Moreover, through connections, the variables available for the parent module and the ones returned by the parent module are passed down to the child module.

In the following example of a program built within Prism's Functionality Editor (Fig.4) there is a tree consisting of four nodes (modules). Thus, the flow of the program is defined: It starts with identifying the object from the camera, proceeds to execute some JavaScript code and ends by

converting some text to speech and sending a notification to the user. A connection doesn't also define which variables are used in a certain module, as this is up to the user. For example, the 'Text to Speech' won't automatically take the output of the 'Javascript' module as a parameter, but the user has to choose between the variable returned by the 'Object Identification' module and the one returned by the 'Javascript' module, as a module has access to all variables returned by the previous modules, which are on the same chain as it. The user can set which variables are used as parameters from the module's settings pop-up.

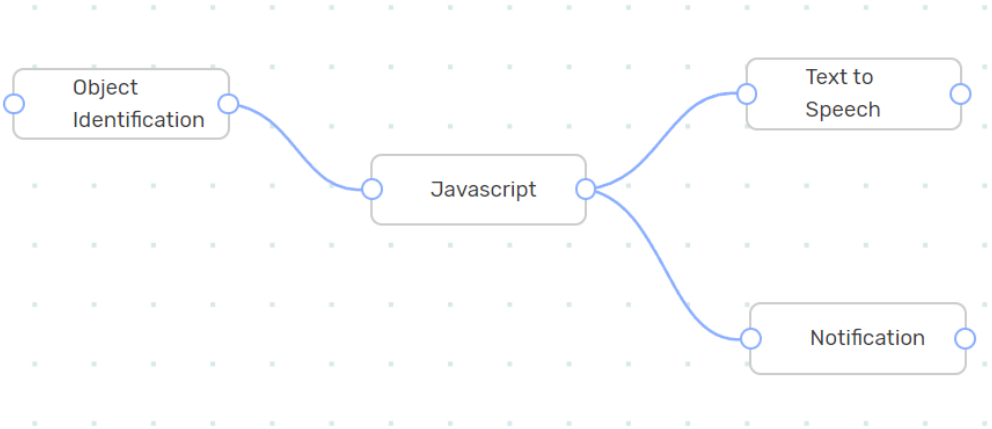


Fig. 4: Example of program built within Prism's Functionality Editor

2.2.3 Implementation of the Graph Visualization Interface

As mentioned before, the Functionality Editor relies on arrays of modules, connections and variables. The Graph Visualization Interface consists of modules and connections, both of which have to be drawn to the screen. This is done dynamically by drawing all modules and connections which are saved in their arrays. Modules are drawn as HTML elements and are positioned absolutely on the screen based on coordinates defined by the user's mouse position. Connections are depicted through a single SVG element containing multiple paths, with each path determined by a formula based on the coordinates of the start and end modules, which assures a smooth curve of the connection.

In order for the user to be able to interact with the interface the following primary functions were implemented:

- Creating a module: Upon dragging a module onto the canvas, a new object with specific data such as coordinates, module type, related code templates, inputs, and outputs is created. The object is then pushed to the module array and appears on the screen at the specified coordinates, which are related to the position of the mouse

- Adding a connection: Establishing a connection between two modules involves multiple steps and functions to manage the connection's temporary path, state, and completion. The primary function responsible for adding a connection takes the start module and the end module as parameters and calculates a custom path based on their coordinates. A new JavaScript object, containing the two modules and the calculated path, is created and subsequently rendered on the screen.

- Drawing the temporary path: As users initiate a connection, a temporary path is dynamically drawn on the screen to visually represent the ongoing connection. While the user moves the cursor, the path of the temporary connection is constantly updated on the screen. Upon releasing the mouse, the temporary path is replaced by a final connection, linking the modules.

- Updating the interface: When the user rearranges a module by dragging it, both the module's position and its connections have to be updated (for the connection the stick to the module). This function is triggered when a module is selected and the user moves his mouse. The module's coordinates are changed and the connection path recalculated accordingly.

- Deleting a module: The module, its connections and its variables are deleted from their arrays, effectively deleting them from the interface.

- Defining a module's behaviour: Some of the modules take in parameters or need to be provided some sort of data, to be customized, or to be linked to some element on the front-end. In the example below (Fig. 5), a chatGPT module is customized, by providing it with a dynamic prompt. The red chip is a variable which was passed to the chatGPT module by a parent module. These customizations can be done from a pop-up to be accessed by double-clicking a module.

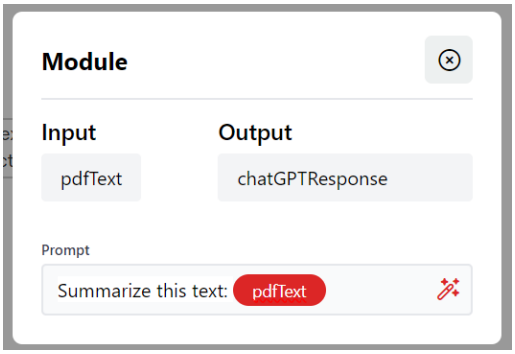


Fig. 5: Example of module configuration

3 User's Application Code Generation

3.1 Design Editor Code Generation

Initially, the approach for generating code in the Design Editor involved storing components as objects in an array and using an algorithm to construct the HTML code. However, a more efficient method was implemented: the editor captures user interactions and changes through JavaScript events, directly updating the Document Object Model (DOM). When the user publishes the application, Prism retrieves the HTML code from the editor preview window, ensuring all changes have been incorporated in the final code.

3.2 Functionality Editor Code Generation

The Functionality Editor generates JavaScript code using a recursive approach based on user-defined modules, connections, and variables. Modules and connections are stored as arrays of JavaScript objects, and variables are stored in a JavaScript Map. By storing this data in this way, Prism can efficiently generate the user's app code and restore the Graph Visualization Interface for the user to be able to go on editing from where he left in his previous session on the platform.

To convert this data into JavaScript code, Prism utilizes the Depth-First-Search (DFS) algorithm to iterate through all the modules. The Depth-First-Search algorithm is a graph traversal technique that explores as far as possible along a branch before backtracking (Fig.6). For optimized DFS traversal, an adjacency list representation of connections is preferred. However, the connections are initially stored as arrays of objects representing connections between Module A and Module B, and therefore must be first converted to an adjacency list format before generating the code. The primary advantage of using DFS for generating the code is that it follows the flow of the user-defined logic and allows for progressive generation of Javascript code during traversal. The traversal path can be observed in the following scheme: It starts off with the red path, exploring mod_1, mod_2 and mod_3, changes its path (green) until mod_4 and then backtracks once again and visits mod_5 on its last path(blue).

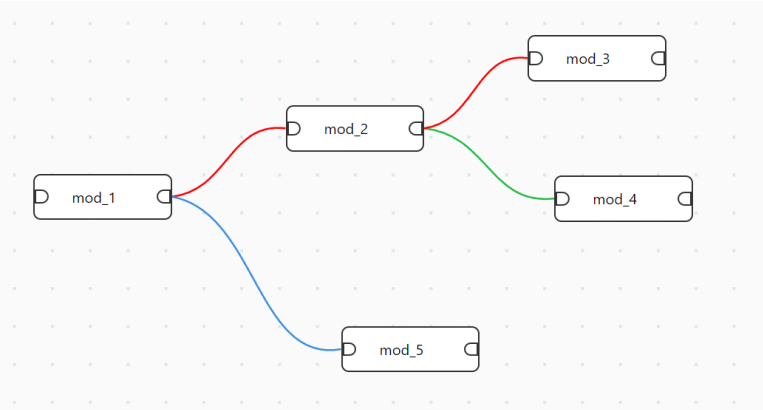


Fig 6: Highlight of the traversal path of the DFS algorithm

Thus, DFS follows each module's hierarchy, which enables Prism to naturally nest and put in order the functions and ensure proper variable scopes. In the following figure (Fig.7) a step-by-step example of how this process works is delineated.

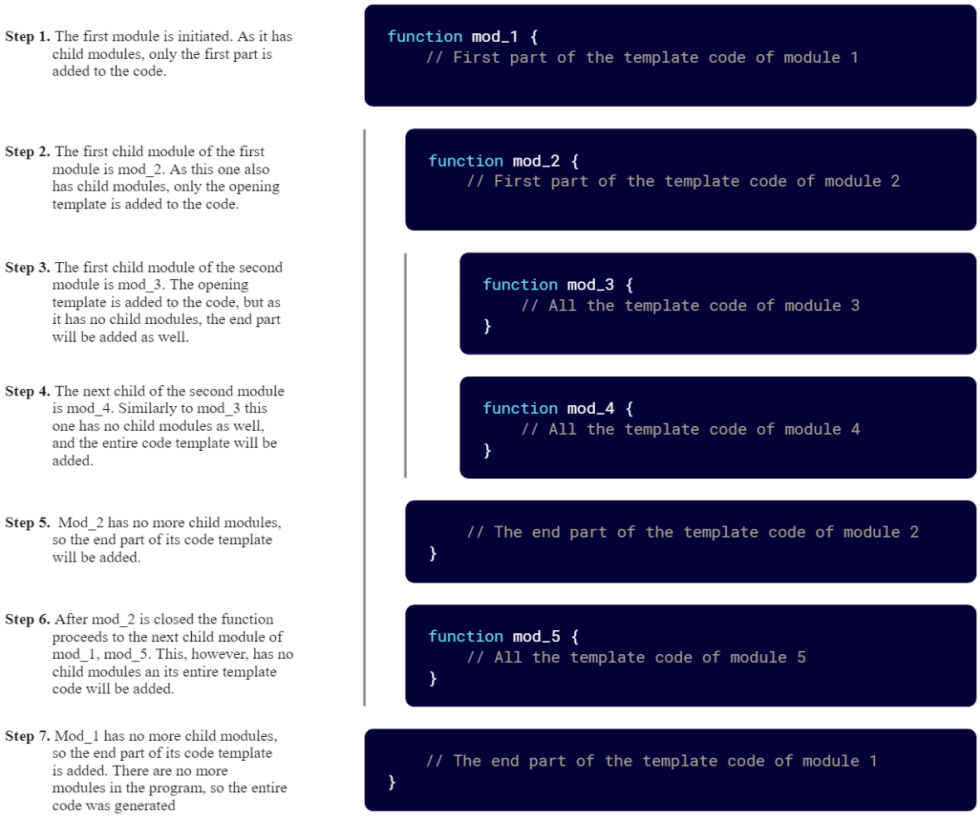


Fig 7: Example of Javascript code generation

Besides the ordering and nesting of the template codes using the DFS algorithm, the functionality code generation process in Prism also involves customizing the template code based on user-defined logic. In the template code of each module, there are designated keywords, to be modified by each user. Each module can be altered by certain parameters. Before generating the final code, Prism customizes each template, replacing the initial configuration with the user's, allowing for flexibility. Both the inputs and outputs of a module are stored in their respective objects, which are then used to dynamically modify the template code according to the user-defined variables.

Prism enables users to effortlessly publish their custom-built applications online, making them accessible to anyone with a unique access link. When the app is published, the final code of the user's app is generated by first retrieving the HTML code from the Design Editor and generating the JavaScript code for functionality. These two are then inserted in an HTML skeleton to obtain the complete code of the page.

This code, initially in the form of a string, is converted to a text/html blob, which is subsequently uploaded to Firebase Storage. When users access their app through the unique link, the stored code is retrieved, allowing them to interact with and utilize the app they created within the Prism editor.

4 Use Cases

Currently, Prism serves as a proof of concept in the realm of low-code AI application development. The complexity and diversity of applications that can be implemented by its users is

currently confined by the flexibility of the platform and the limited number of ready-to-use components and modules. Nevertheless, the potential to scale up Prism is enormous thanks to its modularity, which allows for the expansion of the components and modules library. The process can be sped up by implementing a marketplace, where a community of users with expertise in coding create and post their own components and modules to be used by everyone else, a widespread approach among other *low-code/no-code* platforms.

In the following, the process of building a practical application will be delineated to demonstrate Prism's current capabilities. The app that is created is designed to streamline the task of resume analysis and job applicant ranking. It utilizes Prism's pre-built modules to extract text from a multitude of PDF resumes, analyze the content with the help of a LLM and compare it against the requirements detailed in a job description. All this data is then displayed on the screen and simultaneously stored in the apps database for easy access and future reference.

The first step in building this app is designing its interface which will contain two main components (Fig. 8): a file upload, where the user of the app will be able to upload all the applicants' resumes, and a table where data about each applicant will be displayed like name, email, a relevant short description, and a score reflecting how suitable the applicant is for the job. The two components are placed in a container with two columns for them two be aligned and to ensure responsiveness.

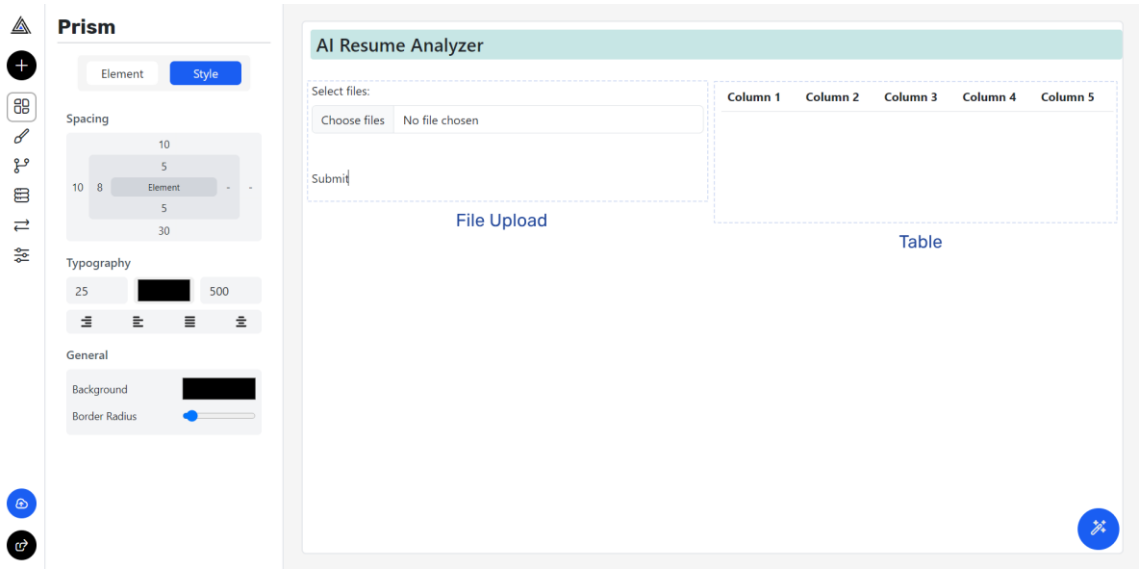


Fig 8: Design of the proposed app's interface

Following the creation of the app's interface, the functionality has to be defined, which is done as depicted in Fig. 9. The functionality of the application should start upon submission of the applicants' resumes through the file upload. Thus the first module is an Event Listener that will detect when a certain event occurs for a certain element from the interface. In this case, the application should detect when the form with the resumes is submitted. The connection between the design editor and the functionality editor is established by passing the unique ID of the element in the front-end into a designated field in the module customization pop-up (the ID of the element is displayed, among other attributes of the element, in the side panel of the design editor). Upon the submission of the resumes, the app should get the files uploaded by the user and then extract the text from them so that their content can be used by a LLM. This is done by using the PDF Text Extraction module which will separately return the texts of the PDFs. The texts will then be passed down to the chatGPT module

which takes in a custom prompt requesting the LLM to analyze the resume from the PDF Text Extraction Module by referring to a job description, extract data about the applicant, write a short relevant description about him and score how suitable he is for that specific job. The data the LLM returns should then be displayed in the table on the interface through the 'Write to Table' module and saved in the application's database for future reference.

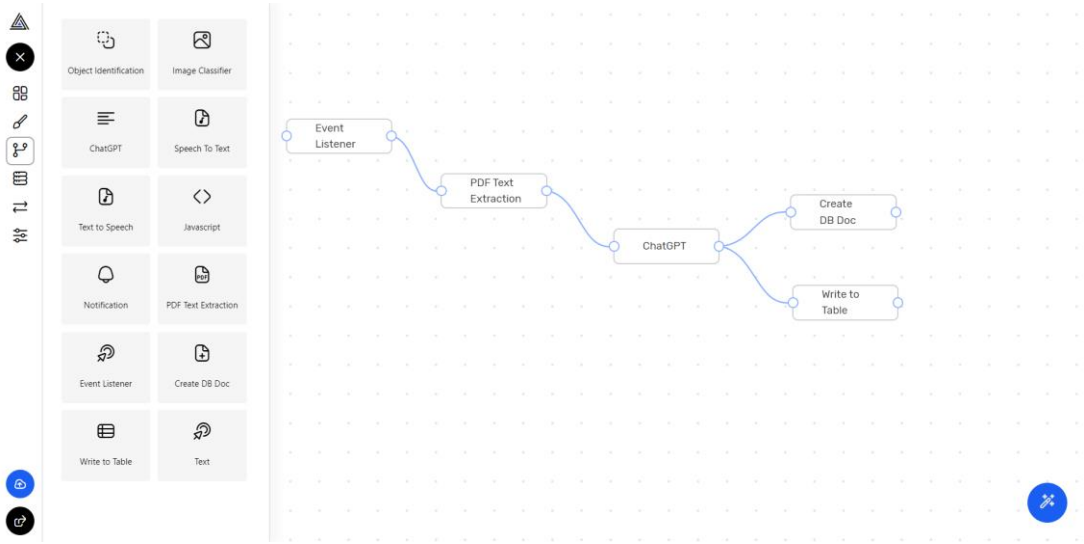
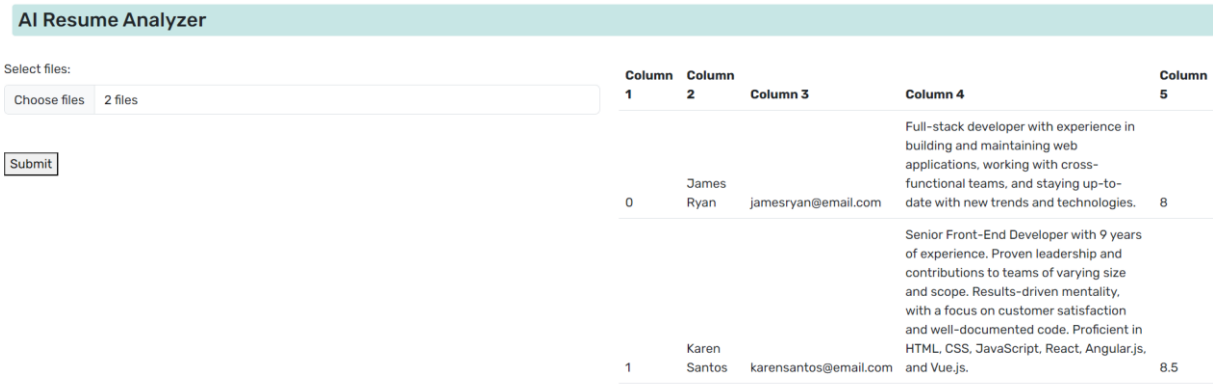


Fig 9: Functionality Graph of the proposed app

After building the desired app inside the Prism editors the user has to publish the app from the blue button at the bottom of the sidebar and can then access and use it anywhere, anytime from an unique URL. In Fig. 10 the application can be seen in action, with the interface created in the design editor. Two files have been submitted by the user and, on the right side of the screen, in the table, the data about the two applicants has been displayed in the table.



	Column 1	Column 2	Column 3	Column 4	Column 5
0		James Ryan	jamesryan@email.com	Full-stack developer with experience in building and maintaining web applications, working with cross-functional teams, and staying up-to-date with new trends and technologies.	8
1		Karen Santos	karensantos@email.com	Senior Front-End Developer with 9 years of experience. Proven leadership and contributions to teams of varying size and scope. Results-driven mentality, with a focus on customer satisfaction and well-documented code. Proficient in HTML, CSS, JavaScript, React, Angular.js, and Vue.js.	8.5

Fig 10: The final app developed by the user

This is just an example of what can be developed with Prism. The utility of Prism extends across a broad spectrum of fields, including parenting, education, agriculture, and work automation. Further enhancements in architecture and flexibility could enable the development of even more complex apps.

5 Conclusion and Further Developments

In conclusion, Prism represents a significant step forward in democratizing access to AI technologies and simplifying the development of AI-driven web applications. By offering a user-friendly, low-code platform, Prism empowers a diverse range of users to create personalized applications tailored to their specific needs.

Looking ahead, there are several opportunities for enhancing Prism's capabilities and versatility. Firstly, adding more modules and components to the platform will increase its scalability and address a wider variety of user requirements. Secondly, improving the app's complexity and flexibility will ensure that users can fully leverage AI technologies in their applications without facing any constraints. Lastly, integrating third-party services will greatly enrich the user experience by allowing seamless data import and export, as well as interaction with external services, enabling users to automate their daily work.

References

[1] *Depth-first search*. https://en.wikipedia.org/wiki/Depth-first_search , October 2023
[2] *Tech Trends Outlook 2022:* https://www.mckinsey.com/~/media/mckinsey/business%20functions/mckinsey%20digital/our%20insights/the%20top%20trends%20in%20tech%202022/mckinsey-tech-trends-outlook-2022-full-report.pdf , October 2023
[3] *Number of chatGPT users:* https://www.theguardian.com/technology/2023/feb/02/chatgpt-100-million-users-open-ai-fastest-growing-app , October 2023
[4] *McKinsey AI Survey 2022:* https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai-in-2022-and-a-half-decade-in-review, October 2023

Matei-Ioan MARIN
"Gheorghe Lazăr" National College
1-3 Gheorghe Lazăr str, Sibiu, 500035
ROMANIA
E-mail: matei.i.marin@gmail.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Domina - a puzzle game meant to improve mental health

Petar Mishev, Galina Atanasova

Abstract

This article presents the benefits of playing puzzle games and especially their function as a mental health improvement tool. Computer games are seen as tools for improving the brain's health, by challenging it with simple to learn, but hard to master challenges. The presented game by the author is an example of how such games work and help improve one's skills in special awareness tasks. Additionally, the author's game is also a way to test the capabilities of the Godot engine, a free and open-source game engine.

1 Introduction

Everyone wants to be happy, healthy, and smart. Many believe that a healthy body will be able to lead them to this state. Dr. Daniel G. Amen, a neuroscientist and prominent associate of the American Psychiatric Association, says that a healthy body is not enough [1]. He argues that a healthy mind is the most direct route to a productive and happy life. He adds that learning new things breaks the routine and keeps the brain alive.

The human brain is divided into areas responsible for sensation, movement, spatial perception, self-organization, attention control, and analytical skills. This division is loosely defined, however, the main point is still clear - if a zone isn't maintained, the neurons in it would undergo atrophy over time. This means that the person's ability to think critically, memory capacity, intelligence, and attention span are on the decline. Over the past 20 years, scientists have discovered that every time a person learns a new skill or takes on a new challenge, the brain's neuroscience builds new connections. The more often these connections are used, the stronger they become. This gives enough reason to assume that brain games are not only entertainment but also a means of improving the functions of the mind. The best thing is that, depending on their complexity, they are suitable for all ages.

Gaming has been a popular form of entertainment throughout the centuries of human history, and with the advent of new technologies, the world of gaming has expanded exponentially. From simple games with black backgrounds and white lines like Asteroids [2] to large worlds that fool even our senses for their existence like Hogwarts Legacy [3]. But the most interesting thing is that the most loved and iconic games are not among the most complex, on the contrary, they are straightforward tasks that excite people of different ages and cultures.

These are *puzzle* type games. Their idea is to present the player with a problem and let them solve it. The special part is that, although the presented problem is simplistic, and the tools provided have

a constant logical functionality, the initial start of every game is always different, and from this the player undertakes a unique sequence of actions, trying to find a solution to the problem. These types of games are versatile and can entertain players of all ages and skill levels. They are also available on a wide variety of platforms, including mobile phones, computers, and game consoles.
This paper outlines a student project that focuses on the design and development of a puzzle game with two main objectives - it aims to explore the practical uses of the Godot engine [4], an alternative to the more established engines such as Unity [5] and Unreal[6], which are commonly used by both independent creators and big companies in the video game industry, as well as gaining practical knowledge in the GDScript programming language. The introductory section presents the motivation for this work, including the origins, advantages, and beneficial effects of logic games for mental and long-term health maintenance. The puzzle game project Domina is then described in detail. The authors argue for their choice of toolkit for the project's development. The article concludes by summarizing the benefits of the current project and outlining future directions for work. Overall, this paper provides an insightful and comprehensive analysis of the development process for the puzzle game, Domina, and highlights the potential of the Godot engine and GDScript for independent creators and big companies in the video game industry.

2 Exposition

2.1 The origins of the puzzle

The game was first created in China and the oldest examples date back to the first half of the 19th century. One of the earliest published descriptions of a puzzle appeared in 1826 in the collection *Continuation to infinite amusement*, which contained 400 different experiments in various fields of science [7].
The puzzle became especially popular in the early 20th century when hundreds of different companies used it as a gift to promote their business or product. The pieces were made of paper or cardboard with an advertisement printed on them. They usually come in an envelope with instructions and an invitation to write or call the company or local dealer for its solution.
Featured in many printed publications - a monthly magazine for teachers *Primary Education* 1904, magazine *Our Young People* 1905.

2.2 Benefits of logic and puzzle games

Logical games provide an opportunity through entertainment to break away from reality and improve many of your abilities [8]. Here are some of them:
Improve concentration - Puzzle games require attention. Therefore, they help to increase the level of concentration in children and adults. In cases where the level becomes easy, you can always increase the difficulty of the game, which will require the brain to think more deeply.
Improve IQ - Each game contributes to the development of the speed of thinking, which is necessary in real life. Logical games successfully solve all problems related to erudition, reaction speed, and volume of information perception.
Improving memory - the most obvious, but no less important advantage of logic games. Nothing improves the ability to remember like systematically increasing one's efficiency in logic games. The positive effect can be seen already in the first month, as the benefit is especially clear in people who initially had a weak memory. Here everything is like in sports - progress comes with training!
Improve planning skills - In the process of playing it is necessary to count or think several moves ahead. This helps improve decisiveness and organizational skills, as well as how to break down a complex problem into smaller, solvable parts. The problem of procrastination, which is typical for quite a few people, will completely or partially disappear. Games also help build confidence to take on bigger challenges.

Improving social skills - The most important factor that can significantly change a person's life is the ability to communicate with people. Logical games are one of the main ways to overcome complexes and improve interpersonal relationships. Group games promote communication and teamwork.
Improves cognitive abilities - Logic tasks are reduced to finding a solution to a given problem. The use of critical thinking is required to find a solution, as the possibility of cheating is not available.
Improve motor skills - games, in addition to developing attention, also improve fine motor skills and hand-eye coordination. Mobility is especially important, both for children and for adults.

3 The Domina project

Inspired by the above facts, the authors created the Domina project. It is a type of logic game where the player has to fill a board with tiles. The boards can be any of the preset sizes 5x5, 7x7, 10x10, or any size the player wants. Tiles are fed as combinations in a 3x3 area (Fig. 1). In the center of these combinations there is always a blue tile that is surrounded by one or more red tiles. These combinations are 16 in total. Those combinations have been selected, because they would provide the most challenge and entertainment to the player. But the player gets a limited set of randomly selected combinations. If using the preset board sizes, the player gets several combinations corresponding to the size of the board (5 if 5x5, etc.), but if he wishes, he can choose his value from 1 to 16.

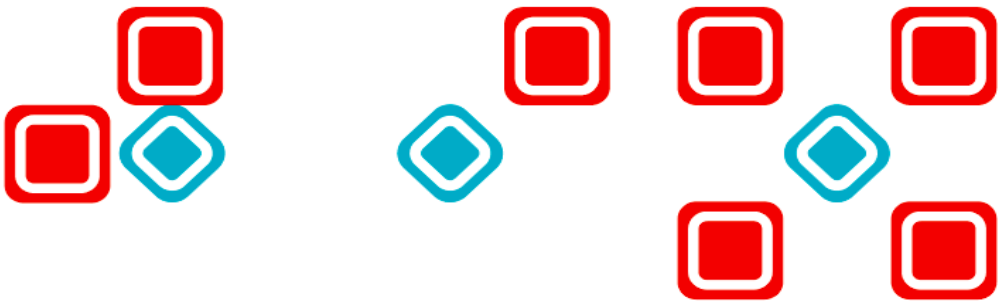


Fig. 1: Combinations of tiles

The challenge comes when placing the blue tiles, the incoming red tiles must not overlap the previously placed blue tiles. Of course, there are some exceptions to tiling. The first is that there can be a red tile on top of another red tile. The second exception is when placing tiles along the contours of the board. If any red tile from the placement combinations leaves the board, that tile will be removed without any penalties.
The idea of these two exceptions is to keep the game moving quickly, but also to help the player remove some tiles in case the game gives a sequence of combinations that will quickly fill the board.

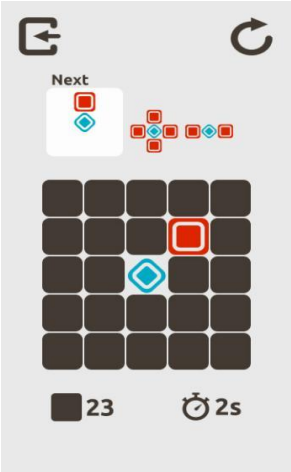


Fig. 2: New game with the first turn made

In Fig. 2 a new 5x5 board game is introduced. The player in this case wished to place the blue tile in the center, which is followed by a red tile in the upper right corner. The next blue tile will come with a red tile above it, and the next two will have more complex combinations

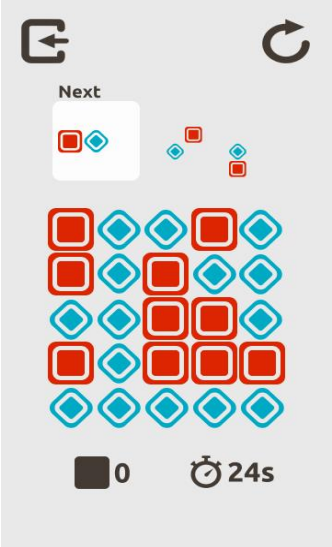


Fig. 3: A game won in Domina

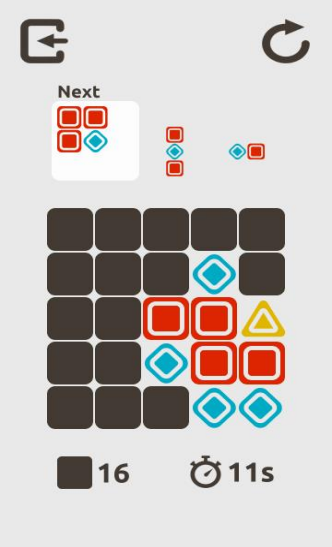


Fig. 4: A lost game in Domina

Fig. 3 presents a standard example of winning the game. The board is filled and no red tile overlaps with a blue tile. In Fig. 4. loss of the game or failure with the task is seen. The player placed a blue tile that had a red tile taped to it in the lower right corner. The red tile has replaced the previously placed blue tile.

It has to be mentioned that in Figures 2-4, on the bottom you can see two counters. On the left is a counter for all of the free tiles that are available and on the right, a measure for how long the game has been going. Both of these features are meant to give trivial information to the player and serve no greater purpose.

The project has been realized with the Godot engine. Godot Engine is a free, cross-platform game engine that allows users to create 2D and 3D games using a set of common tools provided by the engine. It offers a unified interface and can export games to various platforms, including desktop,

mobile, and web-based platforms. The engine is open-source under the MIT license, and users own their games completely. Godot's development is community-driven, allowing users to contribute to the engine's development.
Godot is easy to develop for small and medium projects. Everything in Godot is a *scene*, which can contain elements of any type (textures, models, logic, etc.) that can be inserted into larger scenes. In this way, the whole game is made up of parts that can be reused in a wide variety of situations. Another advantage of Godot is that it comes with it's programming language called GDScript. It's a scripting language that is much easier compared to C# for Unity and C++ for Unreal. GDScript is a high-level, object-oriented, imperative, and gradually typed programming language built for Godot. GDScript is a high-level, dynamically typed programming language used to create content. It uses an indentation-based syntax similar to languages like Python. Its goal is to be optimized for and tightly integrated with Godot Engine, allowing great flexibility for content creation and integration [9]. Finally, Godot is free with an open-source license. This allows the developer to customize the environment to their liking without the need to pay for licenses.

4 Conclusion

Puzzle games have great advantages that set them apart from others. They are preferred because of their interesting and complex levels, the need to create a strategy to win, and the fact that by relaxing one develops one's brain activity. They are the key to a successful life and make people smarter, more attentive, more focused. As a bonus, one retains mental health well into old age. Driven by the idea to enrich the world of entertainment, we started work on the Domina project. The implementation is based on new tools that have enriched our knowledge and capabilities. It is planned to develop some more functionalities, such as multi-user game mode, profile creation, progress tracking and ranking lists. There are also plans to create a Desktop version, and the final versions will be available on the Google Play Store and Steam.

References

[1] Amen, Daniel G., Personel Site, https://danielamenmd.com, 2013.
[2] Atari Inc, https://atari.com/products/asteroids-recharged, 2021
[3] Warner Bros. Games, https://www.hogwartslegacy.com/en-us, 2023
[4] Godot Foundation, https://godotengine.org/, 2014
[5] Unity Technologies, https://unity.com/, 2005
[6] Epic Game Inc, https://www.unrealengine.com/en-US, 2004
[7] Janakieva, V, *Puzzels' History and etymology*, https://nastola.bg/interesno/istoria-proizhod-na-puzelite/, 2022.
[8] Logicbg, A., *How do the logical games improve our brain?*, https://logicbg.com/2019/12/12/kak-logicheskite-igri-vliayat-na-mozaka-ni/, 2019.
[9] Godot Docs contributors, *GDScript Basics*, https://docs.godotengine.org/en/stable/tutorials/scripting/gdscript/gdscript_basics.html, 2023

Peter MISHEV
University of Ruse "Angel Kanchev"
Natural Sciences and Education
8 Studentska str., POB 7017, Ruse
BULGARIA
E-mail: s206252@stud.uni-ruse.bg

Galina ATANASOVA
University of Ruse "Angel Kanchev"
Natural Sciences and Education
8 Studentska str., POB 7017, Ruse
BULGARIA
E-mail: gatanasova@uni-ruse.bg

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Advanced Sentiment Analysis at Textual Conversation Level Using Modern NLP Techniques

Rareș-Gabriel Mușea

Abstract

In an era of digitalization, of perpetual technological advance, of remote communication achieved as efficiently as possible thanks to social media, Artificial Intelligence (AI) plays a crucial role in the development and implementation of systems capable of simulating intelligent human behaviour, so that users can have an immersive and close to reality experience as possible. Natural language processing (NLP) is one of the most notable attempts in the history of Artificial Intelligence to obtain smart human-like behaviours through computer system and specialized algorithms. The idea also led to the advent of sentiment analysis to investigate the frequency of good, negative, or neutral feelings within publicly expressed attitudes or even views and opinions. This paper aims to compare two modern sentiment analysis models, which are based on advanced NLP techniques. They can estimate, by means of a prediction, the polarity of a textual message within a conversation. For the experimental demonstration of these two methods of sentiment analysis, different datasets will be used, one from a social media application and the other from an existing training file. The prevalence of positive or negative texts, as well as their classification will prove to be beneficial, especially within social networks, where an intelligent system based on such conversation analysis could detect a malicious chat, or a possible fraud/harassment. In the past years, numerous users have been victims of such malicious activities, which could be prevented by warning users about the dangers to which they are exposed.

1. Introduction

The following article addresses the need for the development and implementation of an autonomous solution to prevent and combat the phenomenon of online violence, which has been gaining increasing prominence in recent times, primarily due to the emergence of virtual means of communication and information sharing, namely social media. Furthermore, the current solution, presented at an experimental level aims to provide users present on such platforms with an environment of utmost tranquillity and safety for interhuman interaction, one that is founded upon respect and civilized discourse.

Since ancient times and up to the present day, humans have strived to simplify their processes as much as possible to enhance their efficiency. This process of simplification has led to significant progress in numerous fields of activity, progress intended to facilitate and improve the quality of existing processes. This has led to the concept of emergent evolution. According to this notion, the emergence of new qualities is entirely spontaneous and unpredictable.

Communication, on the other hand is part of this idealistic theory of emergence. It represents the quintessence of human interaction, being wholly responsible for the way in which people socialize, relate to one another, and behave within society. As a complex and pervasive concept, the idea of communication lends itself to the use of multiple models from different theoretical perspectives. Psychologists may treat it as a distinct type of behaviour, sociologists view it as a crucial factor within the socialization process, anthropologists see it as an instrument for the formation and dissemination of culture, whereas semioticians perceive it as a process of shaping and reshaping the meanings of signs.[1]. The perpetual need for communication has been profoundly impacted by the challenge of distance, time and other decisive factors that often hinder the smooth exchange of information between several individuals. In response, humans have been motivated enough to develop ingenious solutions capable of facilitating communication, even when it seemed unattainable.

Postal services, telegrams and landline telephony are just a few of the key innovations that have drastically transformed the way society currently relates to the concept of nonverbal communication. However, by far the most ambitious and future-oriented solution in this regard is closely linked to the development and globalization of internet infrastructure, as well as the emergence of social-media in the digital realm. Indeed, in specialized literature, social media is referred to as the primary disruptor of global boundaries and cultural barriers. [2]

This idea of limitless communications and the expansion of existing knowledge through an easily accessible and intuitive virtual environment, alongside new methods of information dissemination have taken the industry by storm and captured the attention of the audience, who have long awaited a significant and beneficial change to their lifestyles. While it successfully complements analogue and traditional communication through a modern mode of expression based on the concept of social networks where content is shared through posts, messages, comments, and reactions, and undoubtedly facilitates access to information, it cannot be vehemently asserted that this element of social media does not have negative impacts or harmful effects both on society and individuals. During the 7th International Symposium on Frontiers in Ambient and Mobile Systems which took place in 2017, Martina Drahošová along with Peter Balco have demonstrated through detailed analysis that the major disadvantages of virtual information and communication environments arise from the lack of security, dependency on the online which they generate, the abundance of misinformation and unfiltered content, as well as the loss of genuine offline social connections. [3] In addition to this, there is also the concept of online violence/aggression, which is influenced by the increasing number of users and the emergence of new social networking platforms on the market, as well as the excessive permissiveness that these platforms provide to all registered individuals within their framework. Although at first glance it may seem that this consumption of social media would be relatively harmless, researchers indicate that the excessive use of these internet facilities can expose users even more to the concept of cyberbullying (also known as the most popular form of virtual harassment) which in recent years has been a major cause of suicide, especially among young people.[4]

Thus arises the need to prevent the damages caused by social media, which occur more frequently from year to year. While the decrease in the number of these unfortunate events primarily relies on the emotional maturity, ethics and morality of each individual within the online context, including social networking platforms, combating the malicious activities of some users does not imply that the platform itself, or the respective network cannot confront them, While the complete elimination of these networks would have no positive effect on society, the fight against these irregularities is highly likely to reduce the occurrence of such incidents, even if it may not completely eradicate them.

The solution is closely tied to the modelling and development process of such a social networking application, and it is strongly influenced by one of the computer science's domains that has gained

popularity due to the tremendous rise of this widely spread phenomenon of social media, namely *artificial intelligence.*

Sentiment analysis, as an applied form of natural language processing (NLP) and it uses machine learning/deep learning along with neural networks in order to perceive and classify the polarity of a given piece of text.[5] It is widely used amongst social media platforms, marketing solutions or even customer services applications in order to filter and predict the polarity of a given opinion or any other type of textual-based element.

As reflected in the title of the paper, the purpose of this study is to employ modern techniques of sentiment analysis and natural language processing to analyse a textual conversation between two users within a social platform, considering both behavioural and emotional aspects. Although sentiment analysis has traditionally been applied to individual elements one at a time, this article tackles the problem from a holistic perspective, specifically analysing an entire conversation consisting of multiple messages. By approximating the polarities obtained from analysing each message individually, an average global polarity of the conversation can be estimated, shedding light on the overall sentiments exposed. The obtained results can then be utilized to proactively detect potential virtual aggressions and prevent the spread of violence within the social environment. Given that the implemented model is purely experimental, the focus will be on a comprehensive comparison of two of the most popular libraries capable of performing such tasks.

The motivation behind working on this project stems for both the author's passion for current technology trends and a personal goal he set for himself from the very first line of code written: to implement an application that will prove beneficial to humanity in the not-so-distant future, assisting individuals in their daily challenges. Furthermore, the idea of developing this mechanism was also fuelled by a previously explored conceptualization of a potential social networking platform. Considering the current context where most newly emerging applications are often seen as mere clones of existing ones in the market, the purpose was to add a personal touch to this application by incorporating a functional element that would set it apart from all others.

2. Development

2.1. Concepts

At a conceptual level, the functionality of the solution is closely tied to the coexistence and synchronization of two separate workspaces that capture two distinct entities within the same standalone application. On one hand, the social networking application creates a virtual environment and provides features through which users can connect with each other and share content. It is also the main data provider for the AI models that I've been used. On the other hand, there is an AI model capable of intercepting the conversation between two users at a given moment and using sentiment analysis techniques to deduce the predominant sentiment conveyed by that conversation. It can also generate detailed reports on the trajectory of the conversation.

By integrating these two components together the solution aims to provide insights into the emotional dynamics of the conversation and other valuable information about the main feeling expressed throughout the interaction (positive/negative/neutral). The synchronized workspaces allow for a comprehensive analysis of the entire conversation, allowing users to gain a deeper understanding of the emotional context and detect any potential issues or instances of cyberbullying or aggression. The goal is to create a safer and more positive online social environment by leveraging advanced technologies and empowering users with valuable insights that can help them navigate conversations more effectively and foster healthy interactions.

2.2. The social media application

According to Gavin Wright's publication in TechTarget [6], social media applications were primarily built to support communication at a distance, as long as content sharing and social connections. Considering all those three fundamental aspects, the ReachMe application came into existence. It is a social networking, full stack, cloud based, single page web application designed for anyone, promoting civilized discourse, strong connections among users as well as emphasizing accessibility and intuitiveness. Its core premise revolves around creating an interaction environment that is both comprehensive and minimalist, ensuring that every user can effortlessly use it and familiarize themselves with it features in the shortest possible time.

Given the complexity and distributed nature of the application, each fundamental subcomponent of this application will be elaborated separately to succinctly describe its functionalities.

2.2.1. Frontend

When it comes to a web-based application, the frontend is undeniably one of the most crucial components as it defines how the application's UI will look and feel and how the basic interactions between the graphical elements and the user will take place. To achieve a simple and minimalist interface, the frontend part of this application has been developed by following a mobile-first approach, which entails first styling the application's appearance for the mobile version and then gradually building it in such a scalable manner that it accommodates different screen sizes and resolutions. This ensures universal compatibility between the application and various types of devices on which it can be accessed. By prioritizing the mobile experience, the application can provide a more responsive and user-friendly interface across different devices, including smartphones, tablets, notebooks, or even desktops.[7]

The ReachMe application has been developed in a rather unconventional manner compared to the most current web applications. Its frontend relies on a client-side rendering system, which means that the entire graphical context of the application is rendered while the page loads rather than being a response received from the server upon requesting a specific HTML resource.

Although the second approach mentioned above is more commonly used nowadays, as it provides superior SEO capabilities to the application, as search engines directly read the content of the server response in place of a page bundle, the overall performance of the application is greatly improved when using client-side rendering. [8] The initial page load time may be slightly longer compared to server-side rendering (SSR) applications, but subsequent changes between different contexts and sections are instant, eliminating the need to reload the page or request a new HTML document from the server to update the Document Object Model (DOM) and display the changes to the user. By using client-side rendering, the short waiting time caused by interacting with a specific feed section or component is replaced by smooth transitions and animations, interface changes are automatically managed and don't require refreshing the entire window and the user experience provided can be more closely compared to that of a mobile application rather than a traditional website.

This preference for the client-side rendering also led to the abandonment of the standard route-based navigation within the application. Although the route-based system was efficient and provided a well-structured access to different elements within the application, it wasn't feasible in direct relation to the development approach already adopted. Instead, a single page application (SPA) approach was chosen, where each transition to a new context is done through conditional rendering rather than routing. This decision sacrificed formal rigor in favour of performance and immersive user experience, providing the application and the entire project with incredible scalability. This approach also enables the application to be considered cross-platform and facilitates its migration to other platforms and ecosystems in the future. With its progressive web application capabilities (PWA), the application is designed to be highly

adaptable and compatible with various devices and operating systems. This scalability and cross-platform nature make ReachMe well-suited for future expansions and *reaching* a wider audience.

The development process of the web client involved the use of Reach.js along with modular JavaScript based on ECMAScript6 (ES6) modules. React was chosen as the main framework due to its numerous features for building progressive and scalable web apps based on reusable components. It provides architectural patterns, such as custom hooks, effects, contexts, queries and a simple yet efficient state management system. It also offers compatibility with other open-source component libraries and pre-defined graphical elements used in the page design, such as MaterialUI and HeadlessUI. These libraries provide a wide range of ready-to-use components and UI elements that enhance the development process and ensure a consistent and visually appealing user interface.

For styling and responsive design, SCSS (Sass) was used in combination with tailwindCSS.

The following screenshots capture some of the most important components within the application.

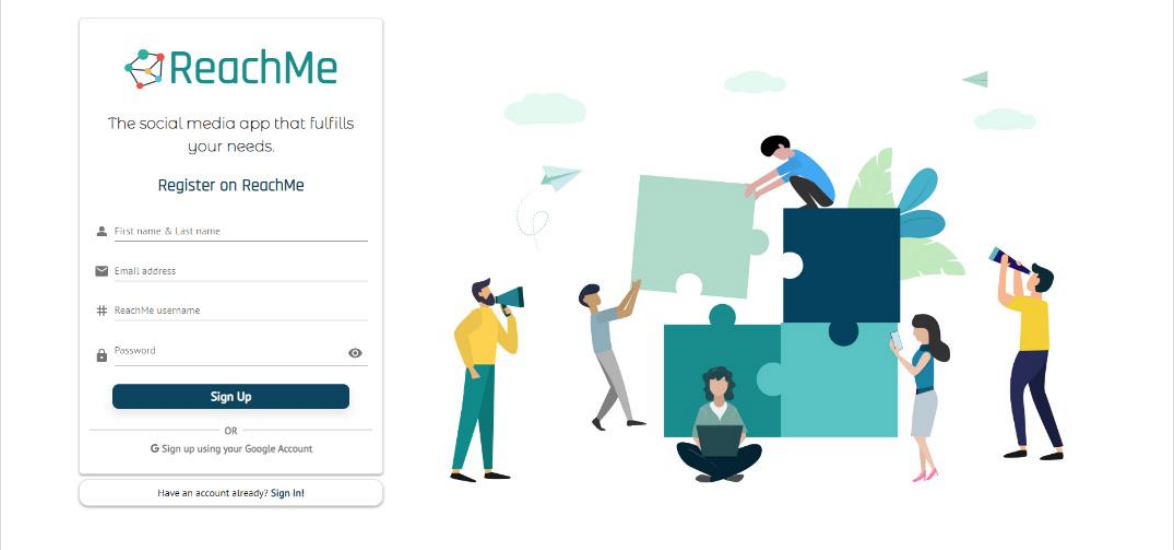


Fig. 1: ReachMe: Register component layout

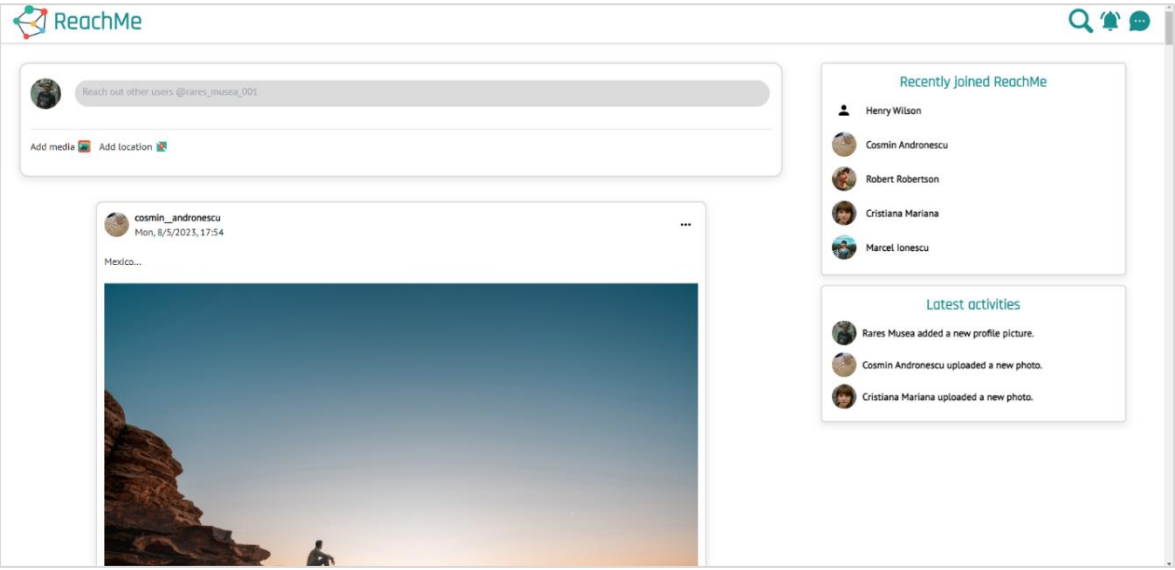


Fig. 2: ReachMe Feed component layout

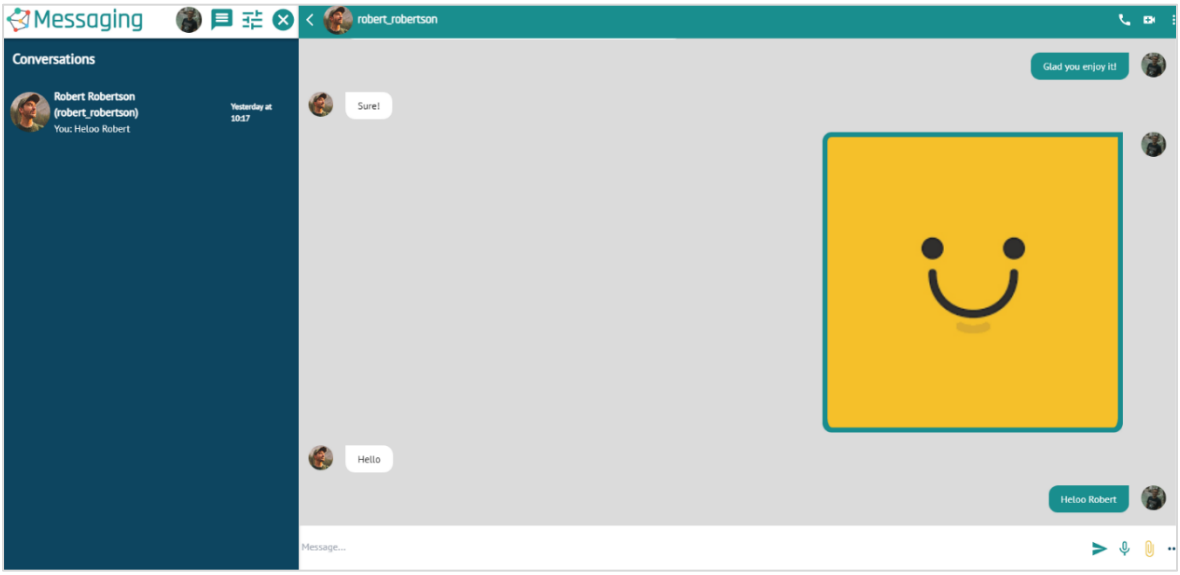


Fig. 3: ReachMe Messaging component layout

2.2.2. Backend

The backend of the ReachMe application consists of a Representational State Transfer (REST) API that maps all the requests sent from the client component of the application to server-side HTTP methods via endpoints, ensuring efficient bidirectional communication between the application's frontend and the storage layer, as well as data persistence.

For implementation, Java was used along with Spring Boot framework, which was configured to work alongside two others widely used established technologies within this tech stack. Specifically, the Hibernate ORM was chosen, which was declared the second most performant and widely used ORM in 2022, according to StackShare.[9] Additionally, the Java Persistence API (JPA) was utilized, which provides numerous capabilities for configuring and managing the logic behind the back-end's persistence layer, closest to the database.

Considering that the MVC (Model-View-Controller) model wouldn't be feasible due to the way the application framework was built, as the rendering is not done on the server side, a different architectural pattern, specifically DAO (Data Access Object Pattern) was chosen in designing the application's backend side. This pattern is based on segregating the business logic of the application from the persistence layer (which, in this case is the same as the storage layer), using an abstracted form of API that hides the complexity of performing CRUD operations within the storage mechanism. This way, both layers can be evaluated independently without being exposed to each other.

The Java language greatly facilitates the implementation of this pattern, particularly due to its object-oriented approach and dependency injection mechanism. The process starts with the implementation of a POJO (Plain Old Java Object) on which the necessary Spring annotations are applied to map it as a table in the persistence layer. This element can often be associated with the concept of an entity since it will later materialize as a table within the database schema.

Furthermore, a DAO interface, commonly known as a repository, is constructed, which contains all the method signatures responsible for performing CRUD operations. JPA already provides a base interface the new DAO interface can implement to inherit the standard CRUD methods, which can be used directly. Also, specialized methods can be defined based on the entity they are configured for. A service component uses the dependency injection mechanism to inject the DAO repository, which will be used as an instance to invoke the defined methods (functional interface). Finally, the highest-level

component in this hierarchy that describes the REST API is a controller which also exists in the MVC pattern. Its purpose is to provide an abstraction of the methods in the service class, which will be also introduced via dependency injection. At this level, it is necessary to declare the endpoints for each method. These endpoints are HTTP methods that are automatically called when a request is made to the specified endpoint from the client. Depending on the endpoint type and the HTTP method it describes (GET, POST, PATCH, PUT, DELETE), Hibernate will make changes to the database automatically. The results are persistent.

The following two figures (Figure 4, Figure 5) depict both the architecture of the Spring Boot library's flow and the UML class diagram associated with the application's backend.

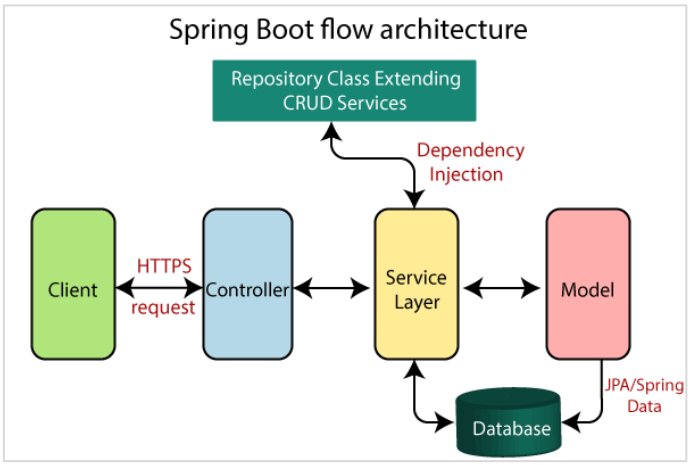


Fig. 4: ReachMe web service UML class diagram

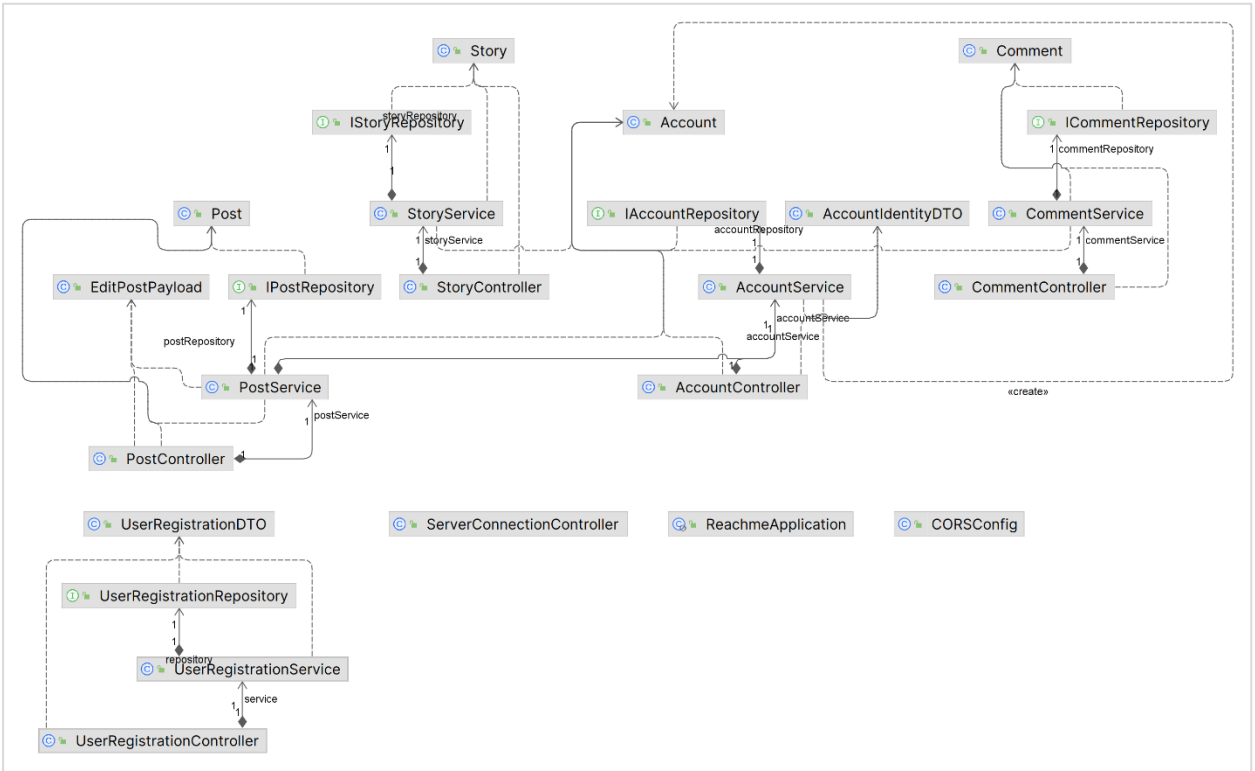


Fig. 5:- ReachMe Web services UML class diagram

2.2.3. Backend as a Service (BaaS)

Continuous technological development has not only allowed for the emergence of innovative concepts in every industry but has also facilitated access to the tools and instructions needed for anyone to come up with innovative ideas. The distributed application could not be conceived without client, server, and storage. This triad defines the three layers that are necessary for a networked application to be stable, facilitate user access through UI elements, and enable data persistence.
However, thigs were about to change with the emergence of Cloud Computing, in 2006. [10]

Cloud Computing is referred as the provision of various services over the Internet. These services encompass a range of resources such as data storage, servers, databases, networking, and even software. Instead of storing files on a local storage device or proprietary hard drive, cloud-based storage enables users to save their data to a remote storage. If an electronic device is connected to the Internet, it can access both the data and the software programs require to operate it. [11]

The introduction of these new remote resources has proven to be extremely beneficial, as the expansion of this phenomenon has sparked increased interest and a need for development. This development has led to the emergence of new concepts, specifically Platform as a Service (PaaS) and Software as a Service (SaaS). Both concepts refer to the ability to use platforms and applications through an internet connection, without the need to download a local copy.

This migration of many popular services towards the cloud has also brought about the need to patent platforms that ensure the functionality behind an application, which can be made available to users in the form of a private or public API key and further exposed through generic functions and methods that can be called directly. This has recently given rise to the concept of Backend as a Service (BaaS). It is an idea that allows anyone to build a full-stack distributed application without requiring knowledge of HTTP methods, network protocols or even architecture of infrastructure.

To design an application capable of sending notifications and updating certain parameters in no time when the database undergoes changes, the decision was to utilize the cloud services provided by Google for this social networking platform. Specifically, I utilized two platforms: Google Cloud Platform and Firebase.

Although the development of these functionalities could have been achieved on the existing backend, the implementation time would have significantly increased and the development itself would have much more cumbersome.

Firebase is a service offered by Google that can serve as a backend solution or web service for applications written in numerous languages and technologies. It provides users with numerous benefits, including cloud storage, real-time managed database (RTDB), a credential-based authentication system which supports registration with other compatible platforms and others.

While developing this application, several Firebase solutions were used. The following list describes both the services that were used, along with the reasons for choosing them:

- **Firebase Cloud Storage:** The decision was made to store all multimedia resources shared within the application in the cloud because managing such files in SQL-based management systems can be very challenging. Firebase, on the other hand offers the ability to upload one or multiple files to the cloud using an asynchronous function. The result of this operation is obtaining a URL for that resource which can also be used as a download link. To store the location of that resource, the resulting URL is inserted into the local database in a specific field for that resource.
- **Firebase Authentication:** The decision was made to prefer this authentication system over building one from scratch, primarily due to the enhanced security provided by this pre-implemented solution. Additionally, there was a desire to offer users more flexibility and extensibility by providing them with multiple registration options within the application.

- **Firebase Firestore Database:** This service was chosen due to its real-time updating capabilities and its use of a non-relational data organization. This solution was exclusively used for storing user messages and conversations, as well as volatile information that can be modified or deleted over time, such as notification lists, recent activities or even connected and active users lists. Although the real-time update feature could have been achieved through the application's main database, it would have required implementing web sockets at the server architecture level, significantly increasing the development time. By using Firebase each client instance obtains an observer after authentication within the application, which automatically updates the application's data when changes occur in the cloud database context, whether made by other users or by the currently connected user. As a result, the application constantly updates, allowing for the seamless reception of messages, notifications, or viewing of the latest uploaded posts without delays or the need to refresh the page.

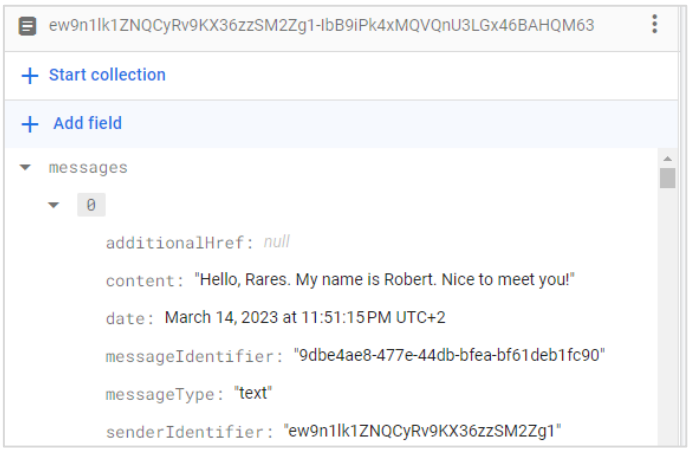


Fig. 6: Firebase Firestore message structure

2.2.4. Storage

Jerzy Letowski states in his article titled "Doing database design in MySQL" that database design is an integral part of the database development process, where the analysis of a given problem definition is conducted to obtain the necessary insights for constructing a logical data structure.[12]

In any modern application, the choice of technologies which will be used is a crucial decision that will impact the entire development process in terms of time, performance, or security. The persistence layer is no exception to this rule. Each database management system is different from another, and each has its own advantages and disadvantages.

For the storage part of this application, a relational MySQL storage was used. It gets automatically managed by the back end's ORM, Hibernate. The database schema consists of 15 tables, of which 6 trivial tables and 9 non-trivial. The relationships between these tables are simple, mainly consisting of one to one or one to many relationships. The referential integrity of the columns in the related tables is based on the concept of foreign keys. Triggers have been manually implemented on some tables, which are activated before or after executing operations such as INSERT, UPDATE, or DELETE.

For the information in this database, which is common to the cloud database, sequential updating is performed from the client-side functionality. An asynchronous function handles the simultaneous synchronization of the two storage spaces, either through the main local database using the REST API or by using a specific cloud function provided by the Firebase service for the NoSQL database.

The following picture (Figure 6) illustrates the database ERD diagram. The similarities between the fields of database tables and the entity class fields in the previously attached UML diagram highlight a close relationship between the web service and the persistence layer.

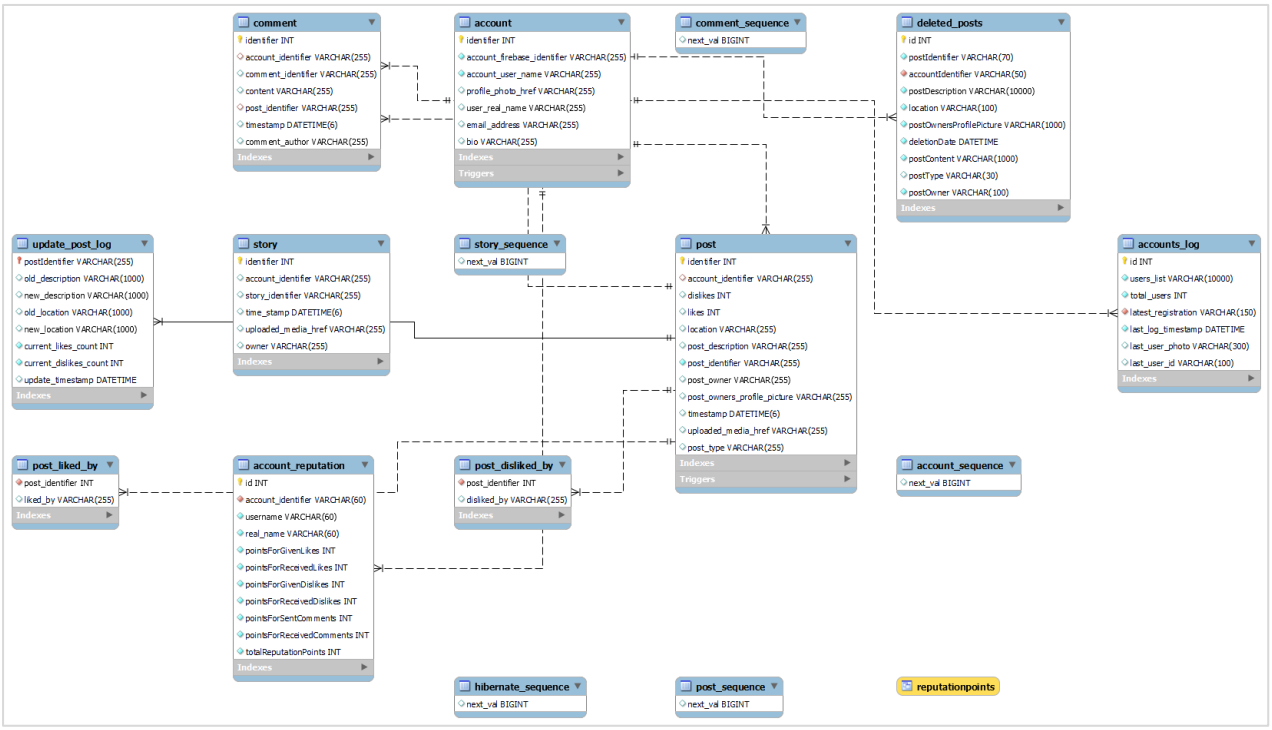


Fig. 7: Relational Database ERD Diagram

2.3. The AI Model

The second component presented in this paper is also the central topic and it is closely related to the utilization of algorithms capable of simulating intelligent human-like behaviour. Specifically, it focuses on natural language processing and its specific application, namely text mining, along with sentiment analysis.

The idea of utilizing this field of artificial intelligence to combat the phenomenon of virtual aggression, which has been increasingly prominent in recent years has been the subject of study in numerous scientific papers, presenting effective solutions for detecting these malicious activities in various virtual social contexts: on specific social networks, within a video game or in any other contexts.

Among the most significant works in this regard, the following titles are mentioned: “Sentiment Analysis for Effective Detection of Cyber Bullying” [13], “Detecting Online Harassment in Social Networks” [14], “Detection of harassment on web 2.0” [15].

The aim of the implemented model is to analyse a conversation between 2 individuals and to generate reports that illustrate the quality of that interaction in terms of emotional involvement and the predominant sentiments. The analysis categorizes the conversation into one of three possible general states: positive, negative, or neutral.

The artificial intelligence aspect relies on the use of two different unsupervised learning models that are currently highly utilized for sentiment analysis of text fragments. As a dataset, both models will utilize conversations among registered users within the ReachMe platform. Given the experimental nature of the study, a detailed comparison is desired in terms of accuracy of the models, to determine which one is more suitable for analysing an entire conversation consisting of multiple messages rather than just a single block of text. The emphasis will be on the precision of the predictions, directly comparing them to the true sentiment conveyed from the entire conversation as well as from a single

message, to identify which implementation has a higher success rate in detecting anomalies or unusual behaviours in online social interactions.

Finally, the obtained results will also be examined when using a very large dataset, designed to test the overall performance of both approaches in a scenario of stress and unfavourable execution conditions.

For the implementation of the two models, the Python programming language was used along with several libraries and external dependencies. These were utilized for tasks such as input serialization into data frames (pandas), complex matrix and vector computations (NumPy), input tokenization based on separators and normalization techniques (NLTK), tensor-based operations and processing (PyTorch, SciPy).

Furthermore, to access message-related information, a script was configured to intercept the requests defined at the backend level of the application. It accessed real-time data from cloud-based Firebase Firestore database, to locally fetch all messages within a specific conversation. For displaying the results, a geometric interpretation was chosen, aiming to graphically represent the polarity obtained from the processing of the entire conversation. This interpretation takes the form of a linear mathematical function, depicting the trajectory towards which the conversation is heading.

During the next section, the functioning of each of the two models will be detailed: Vader Sentiment Scoring and RoBERTa Pretrained Model.

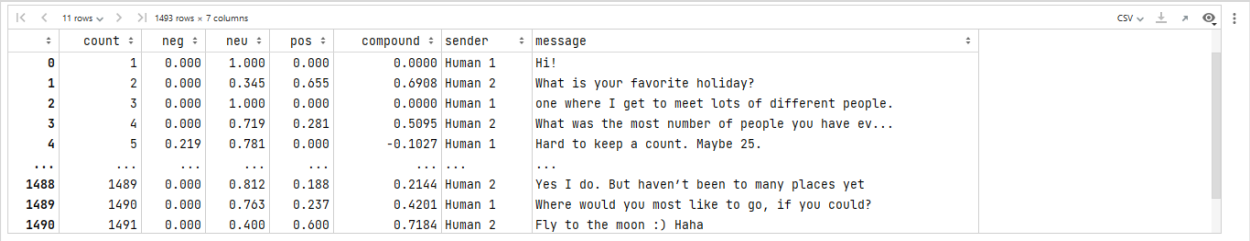
2.3.1. VADER Sentiment Scoring

Valence Aware Dictionary for Sentiment Reasoning, also known as VADER, is an NLP algorithm that combines a sentiment lexicon approach with grammatical rules and syntactical conventions to predict the polarity and intensity of a sentiment depicted from a piece of text. Included as a part of the NLTK (Natural Language Processing Toolkit) for Python, it is well-known for its high accuracy, especially for social-media posts. [16]

This lexicon-based approach utilized by VADER assumes the existence of a dictionary that maps primary sentiments (positivity, negativity, neutrality) to certain English words. The dictionary not only that includes individual words, but it may also map complex phrases, expressions, emoticons or even acronyms. Each component from the dictionary is associated with a polarity, which is marked on a scale ranging from -4 to +4, where the first one expresses extreme negativity, and the other refers to extreme positivity. Then, an average score is calculated as an indicator for each entry from that dictionary. As an example, VADER classifies the word “okay” with a positive score of 0.9, “good” receives a score of 1.9 and “great” has a positive score of “3.1”. On the other hand, the words related to negative emotions, such as horrible will be always negative, such as “horrible”, which receives a negative score of -2.5. [17] To establish the overall sentiment score of as given text, this algorithm analyses the text by identifying known emotional characteristics, adjusts the intensity and polarity of these features based on predefined rules, adds up the score of the identified features and then it normalizes the final score to a range of (-1, 1) using an approximation mathematical function.

The result of analysing a text block using the VADER Sentiment was, in this case a Python list of dictionaries, in which each entry incorporated a key-value quartet. The most conclusive result is by far the compound score, which also determines the nature the analysed input. However, the framework also returns the percentage of positivity, negativity, and neutrality that the text block highlights.

In this way, a result like the one presented in the image below (Figure 8) is obtained:



	count	neg	neu	pos	compound	sender	message
0	1	0.000	1.000	0.000	0.0000	Human 1	Hi!
1	2	0.000	0.345	0.655	0.6908	Human 2	What is your favorite holiday?
2	3	0.000	1.000	0.000	0.0000	Human 1	one where I get to meet lots of different people.
3	4	0.000	0.719	0.281	0.5095	Human 2	What was the most number of people you have ev...
4	5	0.219	0.781	0.000	-0.1027	Human 1	Hard to keep a count. Maybe 25.
...	...	...	...	...	...	...	...
1488	1489	0.000	0.812	0.188	0.2144	Human 2	Yes I do. But haven't been to many places yet
1489	1490	0.000	0.763	0.237	0.4201	Human 1	Where would you most like to go, if you could?
1490	1491	0.000	0.400	0.600	0.7184	Human 2	Fly to the moon :) Haha

Fig. 8: VADER Sentiment results on a textual conversation

However, the major disadvantage of this algorithm is that it was trained on relatively small text blocks with a maximum of 280 words. This makes it an ideal tool for analysing social media posts, where the description's length is usually limited. Despite this, it is not the most suitable candidate for analysing a social media conversation by individually scoring each message and then visualizing the oscillations of the compound values.

Furthermore, rule-based grammar analysis and word-based analysis that this model uses are not always the most accurate, as the relationship between the words in a sentence or phrase may convey a different message compared to how they are conveyed separately, as VADER evaluates them. This is why sarcastic replies or those with underlying meaning will never be adequately evaluated.

2.3.2. RoBERTa Pretrained Model

Robustly Optimized BERT Pre-training (RoBERTa) is a pretrained model developed in 2018 by HuggingFace. It is a transformers-based model, that uses the concept of MLM (Masked Language Modelling) to analyse blocks of text. It was intended to be an improved version of the Bidirectional Encoder Representations from Transformers model (BERT), which was initially introduced in the same year by the researchers from Google AI Language. [18]

RoBERTa is a transformers model that underwent pretraining on a large dataset of English text in a self-supervised manner. The pretraining process involved utilizing raw texts without human annotations, allowing the model to leverage publicly available data. The model employed an automated approach to generate inputs and labels from the texts. Specifically, RoBERTa was pretrained using the Masked Language Modelling Objective. During this process, the model randomly masks 15% of the words in a sentence or phrase from the input. In then processes the entire masked sentence through the model and predicts the masked words. This differs from traditional recurrent neural networks, such as GPT, which typically process words sequentially and autoregressive models that internally mask future tokens. This approach enables RoBERTa to learn a bidirectional representation of the sentence. [19]

By learning this internal representation, the model acquires knowledge of the English language which can be used to extract useful features for downstream tasks. Like any other Transformers-based model, RoBERTa employs tensors to analyse and produce results. These tensors can be converted into floating-point numerical values using the softmax function from the SciPy library. Subsequently, the results can be organized in a tabular format using a pandas data frame to facilitate an overall visual representation.

For the second AI model presented in this paper, the used model was the RoBERTa-base model which was trained on approximately 58 million tweets [20] and it's been optimized for sentiment analysis tasks based on the TweetEval benchmark. [21]

The result returned after analysing, following tensor normalization is a Python dictionary that captures three key-value keys (negative polarity, positive polarity, neutral polarity), as shown in the image below (Figure 9):

11 rows | 1493 rows × 6 columns | CSV

	count	r-neg	r-neu	r-pos	sender	message
0	1	0.052542	0.473377	0.474081	Human 1	Hi!
1	2	0.013626	0.660639	0.325734	Human 2	What is your favorite holiday?
2	3	0.009091	0.263943	0.726966	Human 1	one where I get to meet lots of different people.
3	4	0.032415	0.882687	0.084897	Human 2	What was the most number of people you have ev...
4	5	0.256742	0.695693	0.047565	Human 1	Hard to keep a count. Maybe 25.
...	...	...	...	...	...	...
1488	1489	0.267650	0.649337	0.083013	Human 2	Yes I do. But haven't been to many places yet
1489	1490	0.025601	0.886033	0.088366	Human 1	Where would you most like to go, if you could?
1490	1491	0.001485	0.067520	0.930994	Human 2	Fly to the moon :) Haha

Fig. 9: RoBERTa pretrained model results on a textual conversation

2.3.3. Results comparison

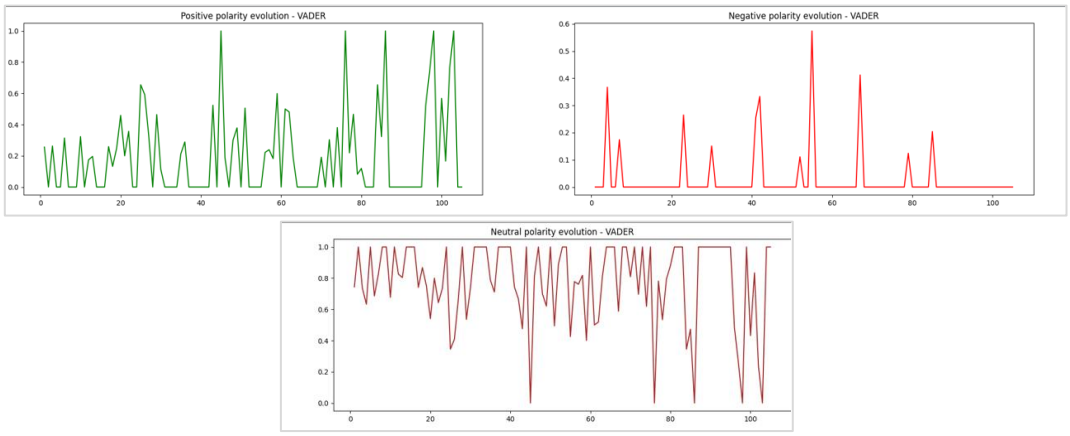


Fig. 10: Positive, negative, and neutral sentiment polarity evolution obtained using VADER.

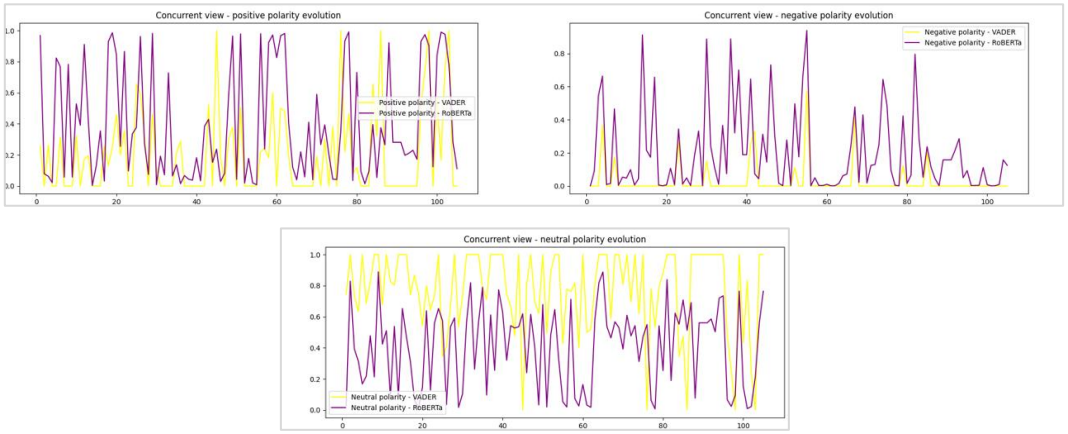


Fig. 11: Positive, negative, and neutral sentiment polarity evolution obtained using the RoBERTa pretrained model.

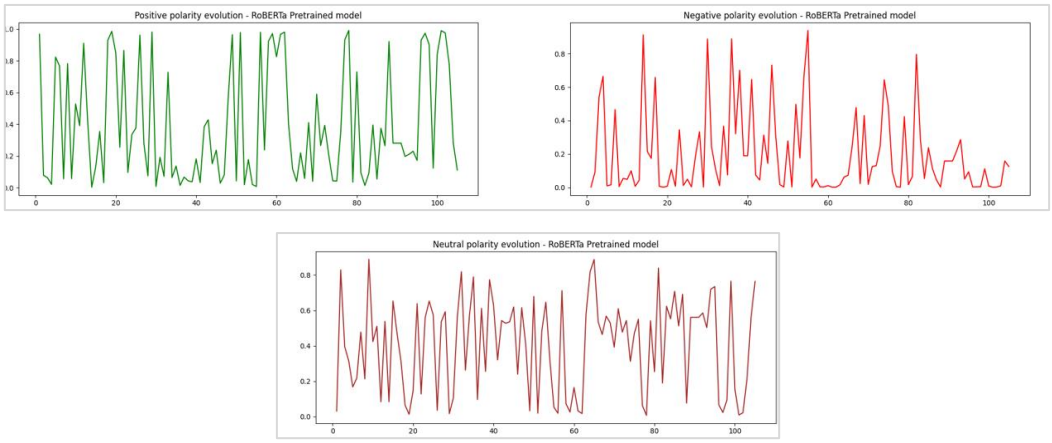


Fig. 12: Graphical comparison showing the results obtained using both AI models, in a concurrent manner.

The following graphs exposed in the above figures (Figures 10, 11 and 12) aim to compare the results obtained from using both previously presented models on 101-turn textual conversation between two users registered within the ReachMe application.

Considering both the format of the values represented in the tables of Figures 8 and 9 and the minimum and maximum points of the two graphs shown in each subplot of the third row in Figure 10 (the concurrent view), we can state that the RoBERTa model provides much higher accuracy in comparison to the VADER model. This is most likely a result of the significantly broader training that the RoBERTa model underwent. While the HuggingFace model was trained on around 58 million Twitter posts, forming a solid knowledge base, the VADER model's dictionary, constructed from the so-called "bag of words" consists of only 7500 entries responsible for extracting the results.

Despite this fact, it can be observed that, in the case of positive an neutral sentiments conveyed, both models have comparable results, although the intensity of positivity is much less evaluated by the VADER model. Regarding negativity, RoBERTa managed to identify certain areas in the conversation where the polarity of negativity was indeed more pronounced, making it a much better model for sentiment analysis of longer messages, not just descriptions of social media posts.

One last essential difference between the two can be attributed to the absence of a compound score in the results obtained by the RoBERTa model, unlike what the VADER algorithm offers. Even so, RoBERTa benefits from the advantage of being capable of providing a global score for the entire input received without the need to evaluate each input separately, as the first presented model does.

3. Future developments

The current project could be further developed by incorporating the pretrained RoBERTa model for sentiment analysis into the messaging component of the ReachMe social networking application. Through an efficient infrastructure and further developments in the full-stack application architecture, it could analyse the conversation in real-time with the purpose of alerting the user to the risk of aggression when a visible transition from predominantly positive to extremely negative sentiment occurs within a certain interval of messages.

The mechanism could prove highly beneficial in combating the phenomenon of virtual aggression and would represent an automated solution that could significantly reduce both the complexity of manual user reporting systems and the need for moderators to analyse these reports which are present in most social media applications.

4. Conclusion

Despite the purely experimental and comparative perspective it offers, this paper attests to the importance of artificial intelligence projects aimed at facilitating interpersonal interaction between individuals. It seeks to promote civilized discourse based on respect and ethics, as well as safety in these virtual environments.

Increasingly, numerous damages have been reported caused by certain ill-intention users who have abused the excess of freedom provided by these solutions to cause inconvenience to other users who simply wanted to connect with people they were separated from by distance or even time.

Although social media platforms currently have numerous disadvantages, we cannot overlook the fact that they have radically changed the way we, as human beings approach the communication process and the way we are constantly informed. Their elimination is not a viable solution, but constant improvement through exclusive attention to the issues that arise, and the needs of users could pave the way for the enhancement of these services in the long run.

Acknowledgement: This work was supervised by Lecturer PhD Ralf Fabian, from Lucian Blaga University of Sibiu.

References

[1] FÂRTE, Gheorghe-Ilie. *Comunicarea: o abordare praxiologică*. 2004.
[2] AMEDIE, Jacob. *The impact of social media on society*. 2015.
[3] Martina Drahošová, Peter Balco - *The analysis of advantages and disadvantages of use of social media in European Union* URL: https://www.sciencedirect.com/science/article/pii/S1877050917311286?ref=cra_js_challenge&fr=RR-1 (retrieved 4.12.2022)
[4] Attorney Matthew Bergman - *Social Media & Suicide*. URL: https://socialmediavictims.org/mental-health/suicide/ (retrieved 4.12.2022)
[5] PapersWithCode - *Sentiment Analysis*. URL: https://paperswithcode.com/task/sentiment-analysis (retrieved 6.01.2023)
[6] Gavin Wright - *What is social networking?* URL: https://www.techtarget.com/whatis/definition/social-networking (retrieved 25.01.2023)
[7] Bhanu Prakash - *Retrofitting Mobile First Design, Responsive Design: Driving Factors, Approach, Best Practices and Design Considerations.* URL: https://www.researchgate.net/publication/347601683_Retrofitting_Mobile_First_Design_Responsive_Design_Driving_Factors_Approach_Best_Practices_and_Design_Considerations (retrieved 1.02.2023)
[8] Taufan Fadhilah Iskandar *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* 801 012136. URL: https://iopscience.iop.org/article/10.1088/1757-899X/801/1/012136 (retrieved 1.02.2023)
[9] StackShare - *What are the best Object Relational Mapper (ORM) Tools?* URL: https://stackshare.io/orm (retrieved 1.02.2023)
[10] Research paper on Cloud Computing (https://www.researchgate.net/publication/352477780_Research_Paper_on_Cloud_Computing)
[11] Ashwini Sheth, Sachin Shankar Bhosale, Harshad Kadam - *What is Cloud Computing? Pros and Cons of Different Types of Services.* URL: https://www.investopedia.com/terms/c/cloud-computing.asp (retrieved 12.02.2023)
[12] Jerzy Letkowski - *Doing database design with MySQL*. URL: https://www.researchgate.net/publication/271910489_Doing_database_design_with_MySQL (retrieved 12.02.2023)

[13] Nahar, V., Unankard, S., Li, X., Pang, C. (2012). *Sentiment Analysis for Effective Detection of Cyber Bullying*. In: Sheng, Q.Z., Wang, G., Jensen, C.S., Xu, G. (eds) Web Technologies and Applications. APWeb 2012. Lecture Notes in Computer Science, vol 7235. Springer, Berlin, Heidelberg. URL : https://doi.org/10.1007/978-3-642-29253-8_75 (retrieved 12.02.2023)
[14] BRETSCHNEIDER, Uwe; WÖHNER, Thomas; PETERS, Ralf. *Detecting online harassment in social networks*. 2014.
[15] YIN, Dawei, et al. *Detection of harassment on web 2.0*. Proceedings of the Content Analysis in the WEB, 2009, 2.0: 1-7.
[16] Hutto, C.J. & Gilbert, Eric. (2015). *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.* Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.
[17] Ying Ma, NLP: *How does NLTK.Vader calculate Sentiment?* URL: https://medium.com/@mystery0116/nlp-how-does-nltk-vader-calculate-sentiment-6c32d0f5046b (retrieved 4.03.2023)
[18] Rani Horev, *BERT Explained: State of the art language model for NLP*. URL: https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270 (retrieved 4.03.2023)
[19] HuggingFace - RoBERTa-Base. URL: https://huggingface.co/roberta-base (retrieved 12.03.2023)
[20] HuggingFace - *Twitter RoBERTa Base Sentiment*. URL: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment (retrieved 12.03.2023)
[21] TweetEval (Findings of EMNLP 2020) – Reference paper. URL: https://arxiv.org/pdf/2010.12421.pdf (retrieved 13.03.2023)

Rareș-Gabriel MUȘEA
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: raresgabriel.musea@ulbsibiu.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

MemStego. Steganography and Memes combined.

Vladislav Nikolov, Galina Atanasova

Abstract

The paper examines the potential use of steganography in communication by a particular target group, as well as others. It proposes the use of funny images and memes that are frequently shared in daily communication. The paper introduces a new solution called MemStego, which is an amalgamation of steganography, cryptography, and memes. Unlike other software products that use images provided by the user, MemStego uses memes as well that are obtained from the Memegen API.

1 Introduction

Our online presence is an extension of our personal lives, as we share a vast amount of information, both explicit and implicit, that reflects who we are and what we do. While many people may feel that sharing some of their personal lives is harmless, the reality is that the information we share online can be used against us by others. This is especially true for people who express opinions that are not popular with certain groups. Journalists, researchers, and other truth-seekers are particularly vulnerable to monitoring and persecution.

The solution to this problem lies in using steganography, which is the art of hiding information within other information. By using steganography, individuals can communicate and share information without attracting unwanted attention. This technique has been used since ancient times, and it can be used effectively today to exchange messages without drawing attention to them.

One of the possible solutions to the problem we are going through in this article is the project “MemStego”. It is a Steganography tool that uses as base images, that could possibly be memes - those funny images shared throughout the Internet. Using them, with the combination of Cryptography and Steganography [1], could make a good, almost undetectable solution. The project itself solves not the root of the silencing problem, but protects the intermediate steps of every authority that could possibly be unlawfully punished.

The presentation is organised into several sections, beginning with an introduction, followed by a brief overview of shorthand and memes as their essence and benefits. The designed and implemented application for encrypting textual content in an arbitrary image is then described through a Use Case Diagram, with step-by-step instructions provided for both the encoding and decoding processes. The concluding section summarises the key points covered in the presentation.

2 Exhibition

2.1 Old Fashioned Steganography

In ancient times [2], when emperors and rulers held power across the world, messages were delivered from one location to another using human messengers. Unfortunately, these messengers were often intercepted, and their private messages read by unintended individuals. To address this issue, steganography was born. In this method, a secret message was tattooed onto the head of a person, and the person's hair was allowed to grow to cover the message. Once the message was hidden, the messenger was sent to deliver it, and their head was shaved at the endpoint to reveal the message.

Another example, that could be looked at as a modern one, could be the usage of an invisible ink. This represents among the earliest modern steganographic practices, where messages were crafted to be imperceptible to ordinary observation but could be unveiled through the application of a particular substance or method. Frequently employed substances for crafting invisible ink were from natural origin, like lemon juice, milk and others. Other times they were just artificial ones, usually consisting of a mixture of ferrous sulfate and water [3].

2.2 Modern Steganography

As mentioned earlier, steganography is a technique used to conceal one type of information within another type, such as images, videos, or sounds. [4] Typically, steganography involves hiding data within binary files, which are larger in size and contain encoded data that represents visual, auditory, or other sensory information. By embedding information within these files, steganography allows for covert communication. In the case of images, for example, the data is encoded in the pixels themselves. This means that the hidden information is embedded within the actual image data.

2.3 Memes

Nowadays, the term "meme" has already acquired an almost universal meaning for more and more people. For all of us, it is the funny collages and humorous pictures posted on the internet. In a more global sense, it is assumed that they can also be popular phrases or actions with an ironic purpose, which subsequently became a massive social trend.

The origin of memes dates back to 1976, when social networks were just a mirage. In his book "The Selfish gene", the biologist Richard Dawkins introduces the concept of "meme" as a unit of measure, accounting for the exchange of cultural information in a society. There, genetics is seen in a more modern guise and provides interesting explanations for how humans act (often consciously and self-centered) with the sole purpose of reproducing themselves, both biologically and ideologically, be it in an impure way. The ability of a society to transmit its traditions and cultural heritage between generations finds expression in memetics - a new, but no less important, scientific current. Memes are the transmitters and witness of all cultural and spiritual changes in a given group of people and are our processable visible genes [5].

Several straightforward facts about human nature can be noted in this theory. Embedded in the mind of a person is the desire that his existence should not be in vain, and hence the urge to create, explore and pursue his ideals. People's self-centred nature drives them to constantly seal every slice of their lives into time capsules and dream of a legacy that will make them immortal in someone's

anthology. Another DNA that expresses itself in long-standing traditions, customs and historical sources from forgotten periods. In this sense, memes are extremely valuable, because they are most often a version of already familiar images and can evolve over time, passing from one creative mind to a more creative mind, until they lose their current value, i.e. their qualities of sustainability. They can be used to track what is influential fashion and what is a fleeting trend and to measure trends around a given historical period.

2.4 MemStego

Memstego is a steganography tool that utilises memes and images, and is developed as a GUI application in Python. The application makes use of the Tkinter Framework for the graphical user interface, the PyCryptodome package for cryptography [6], and the Pillow package for image manipulation [7]. This software offers users three main types of operations.

The 'Crypt' operation/tab allows users to embed secret textual data into an image of their choice. The 'DeCrypt' operation/tab decrypts an image with embedded secret information, requiring a password for decryption. The 'MemeCrypt' tab is a variation of the 'Crypt' tab that uses a randomly generated meme from the Memegen API [8] for the next steps of the procedure. Below (Figure 2 is a Use Case diagram. The actor in this diagram could be anyone, either the sender or the receiver. The only requirement is that both parties exchange their used passwords through another channel of communication, such as face-to-face or another way of secure way of data transmission.

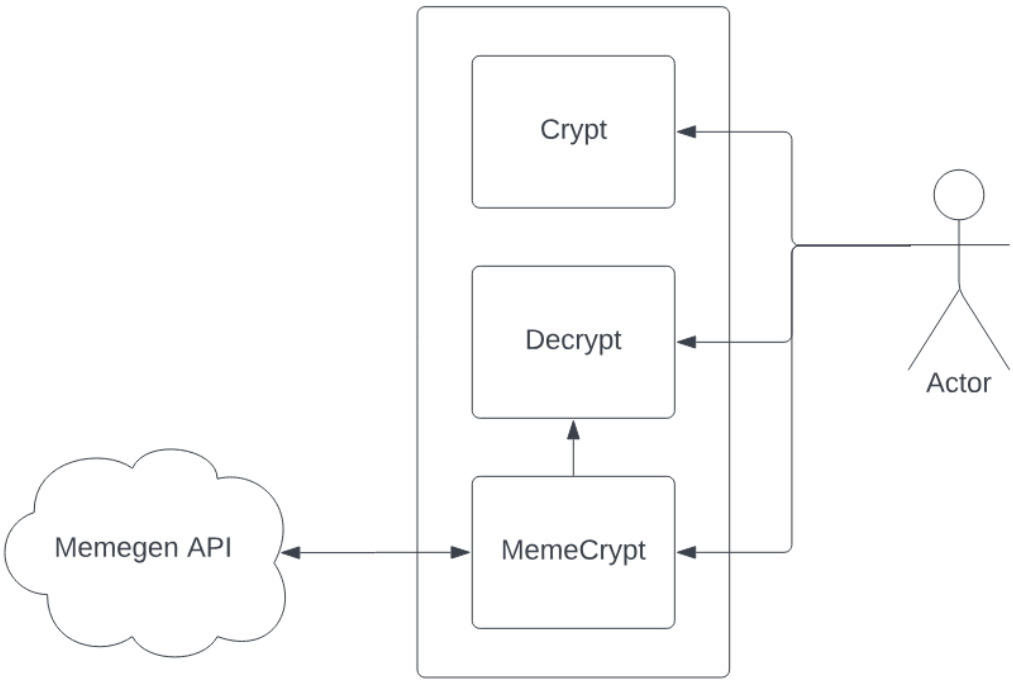


Fig. 1. Use Case diagram of the solution

To help visualise the embedding process, imagine a snake, where the head represents metadata about the embedded data and the tail represents the actual data. The metadata about the embedded data

is split into two parts: the "Tip" and the "Rest". The "Tip" section consists of 1 byte and is responsible for counting the number of bytes that contain the actual metadata. The "Rest" section is used to describe the number of characters embedded and takes up N bytes, where N is defined in the "Tip" section. It specifies the number of characters that have been embedded in the image. It's important to note that each character is embedded in exactly 2 pixels. Currently, Memstego only supports PNG images. This choice was made at the beginning of development because inserting 1 byte is accomplished using exactly two pixels, as each pixel has four color channels. However, a more sophisticated version that supports additional formats could be implemented as well. An image can contain a maximum number of message characters, represented as Base64 bytes, which includes emojis and text characters:

$$NumberOfBytes = Width\ of\ the\ image\ in\ pixels * Height\ of\ the\ image\ in\ pixels / 2 \quad (1)$$

2.5 MemStego Demo

Fig. 2. Before manipulation

Fig. 3. After manipulation

Fig. 2 and Fig. 3 show that an image before and after the manipulation is the same. After the manipulation, it's difficult to notice any difference between the original image and the modified one, as we have only altered the Least Significant Bit. This change is not easily perceptible to the naked eye, but it can be detected using specialised software designed to detect image manipulation.
For example, the above image was downloaded at a size of 735 by 500 pixels. Once the image is loaded into memory, the access time remains constant because each pixel has a known location index. Inserting text with a size of 3000 bytes took approximately 1500 milliseconds.

2.6 MemStego processes

2.6.1 Embedding data in image

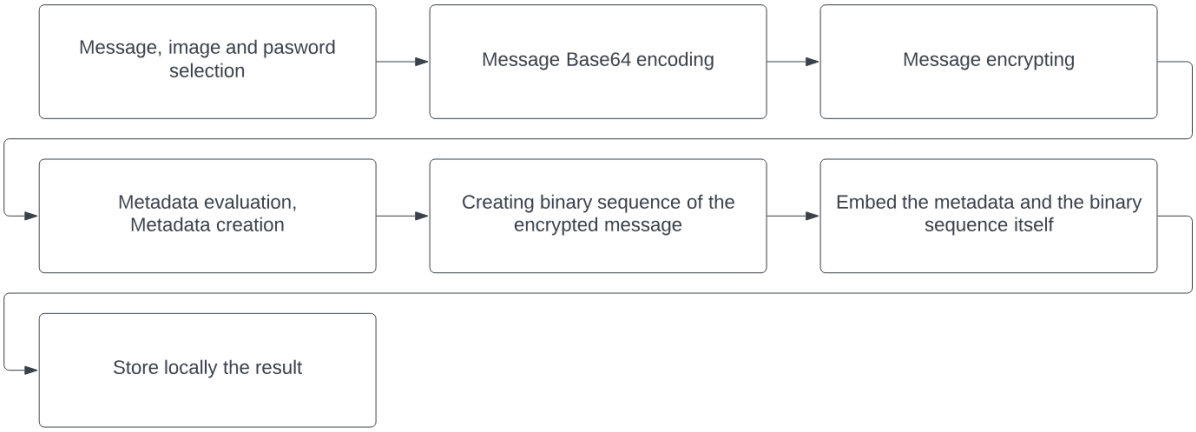


Fig. 4. Embedding data

Figure 4 depicts the following processes. The User begins by selecting a message file, an image, and a password. The message is then encoded using the proper encoding scheme to account for the possibility of any UTF-8 symbols being present. Once encoded, the resulting character sequence is encrypted. The length of the encrypted message is then evaluated, and a binary sequence representing it is generated.

Next, the first byte of the metadata header is written, followed by the corresponding number of bytes for the actual number of embedded characters. Then, the encrypted message is embedded into the image. Finally, the resulting image is exported and saved to local storage.

2.6.2 Extracting data from image

Extracting data, the process shown in Fig. 5, refers to the process of extracting embedded data from an image. To do this, we begin by selecting the image and inputting the password. Next, we retrieve the metadata from the image, followed by the encrypted embedded data. We then decode the binary sequence into characters and decrypt it before exporting the content to a text file.

Fig. 5. Extracting data

3 Conclusions

To align with the latest communication technology trends and address the need for secure text transmission over networks, we have developed and implemented an application that enables users to conceal any text within PNG images. The source code of the project is uploaded in GitHub https://github.com/cloudvlad/Memstego.

There are instructions available for installing and using the program, as well as a video demonstrating how to use the solution and the results obtained from its usage.
Planned for future development is adding more image formats, more media formats and GUI customizations.

Actual video demonstration could be found on - https://www.youtube.com/watch?v=O9hoZCktR8g.

The advantages of using this tool include its simple process and a straightforward codebase, which makes it easy to fork the project and start your own without building from scratch. Simplifying and clarifying the steps can assist individuals in understanding the underlying ideas of the concept, potentially aiding them in developing something on their own. Also the project, as noted before, is written in Python. Being one of the preferable languages among programmers, because it is easy to read and to maintain.

However, there are also several disadvantages. When one side creates steganographic messages, the other side must also use this software. Interpreted languages like Python tend to execute more slowly than compiled languages. Additionally, as with any encryption process, the password must be shared through another channel—either face to face or in another online environment. The second method makes the entire communication less secure, increasing the risk of a man-in-the-middle attack. Also, interpreted languages like Python tend to execute more slowly than compiled languages.

At the end of the day, not many people need to keep their thoughts completely private. However, delving deeper into how things work in the information era can be beneficial. While knowing exactly what to do is good, understanding it is even better. This way, you can become independent from external providers and keep your secrets truly yours.

4 Future development

The project serves as a good starting point for understanding the concepts. However, if the goal is to achieve better performance, it may not be the best trade-off. It is preferable to use tools that are better suited for this critical task, rather than ones as generic as Python and Tkinter. Utilizing Rust, one of the most performant languages available for writing safe code, in conjunction with Tauri as a Frontend Framework, could significantly enhance performance and security while still maintaining a relatively simple code base. This approach could reduce errors, improve efficiency, and enhance maintainability and customization possibilities, thanks to the utilization of popular web technologies.

References

[1] N. Subramanian, O. Elharrouss, S. Al-Maadeed and A. Bouridane, Image Steganography: A Review of the Recent Advances, *IEEE Access*, vol. 9, pp. 23409-23423, 2021, doi: 10.1109/ACCESS.2021.3053998.

[2] Guinness World Record, First use of steganography, https://www.guinnessworldrecords.com/world-records/first-use-of-steganography
[3] University of Michigan, William L. Clements Library Letters Exhibition, https://clements.umich.edu/exhibit/spy-letters-of-the-american-revolution/secret-methods/invisible-ink/
[4] Kaspersky Labs, What is steganography? Definition and explanation, https://www.kaspersky.com/resource-center/definitions/what-is-steganography
[5] Dawkins, R., *The Selfish Gene*, Oxford-Landmark-Science, 2016
[6] PyCryptodome's documentation, https://pycryptodome.readthedocs.io/en/latest/
[7] Pillow 9.4.0 documentation, https://pillow.readthedocs.io/en/stable/handbook/overview.html
[8] Meme generator, https://memegen.link/

Vladislav NIKOLOV
University of Ruse "Angel Kunchev"
Informatics and Information Technologies
Ruse, 17 Studentska street,
BULGARIA
E-mail: s206253@stud.uni-ruse.bg

Galina ATANASOVA
University of Ruse "Angel Kunchev"
Informatics and Information Technologies
Ruse, 17 Studentska street,
BULGARIA
E-mail: gatanasova@uni-ruse.bg

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Facial System Recognition in MATLAB

Victor Iulian Őtveş

Abstract

In this paper the author presents a safer and more efficient alternative that solves the problem of allowing a person's authorized access to a room, using artificial intelligence. The application is currently in the development phase. For this reason, this paper serves as documentation, providing an explanation of certain concepts related to the implementation of the application, the preparation of the data set, about how a model is trained and analyzes and observations based on the results obtained at this moment.

1 Introduction

Artificial intelligence and machine learning is a field that has seen great growth lately, being increasingly researched and published. These systems and applications are very useful in statistical analysis, event prediction, medical testing and object or person recognition.

The current work has just such a theme, more precisely the recognition of people in order to restrict the access in a room.

Normally, the classic verification involves showing an identity card or a badge. A big problem is that many of these documents do not have a picture of the owner on them and can be very easily transmitted. Thus, a person can lend his ID card to another person, and the latter can enter a room. If the ID does not have a photo, we assume that the person who has the borrowed ID is the one who has to enter the room and we give him access. This is how many fraud situations arise.

The application presented below proposes a safe and modern alternative, based on AI, created to solve this problem of classic (manual) identity verification. In this way, there is no need for additional identity documents (identity card or badge). The way the application works is simple: a list of people who should be present in the room is loaded into the application. Once the application is started, it receives images from the WEB camera. A person who wants to enter the room positions himself in front of the camera. Now the application will provide a PERMISSION message if the person has been identified and is on the list or REJECTED if the person could not be identified or is not on the list.

I chose this topic because I wanted to understand the AI field better: what are the stages of developing these applications, how to prepare and train the dataset and what results are obtained.

We know that artificial intelligence is a very controversial field, due to data protection. For this reason, the author would like to clarify a few aspects from the beginning. The application is

built to be used by private institutions (for example companies or education institutes). Since it is about personal data, it is important to state that there is a protection of user data. First of all, to start the application, an administrator or a person responsible for managing the application must first log in. Then, once the application is started, it does not store data (names or images), but basically just classifies the received information.

To train the model, the institution that uses the application needs a database with people. This database contains the name of the persons and a number of pictures with them. But this database is unique and private for each institution, and the information is used only by the respective institution.

The application is not intended to impose or oblige anyone, but rather to provide an optional tool for institution to use based on their own needs and preferences. Of course, if a person does not want to, then the verification will take place normally or alternative solutions will be sought.

2 Technologies used in application

As we have assumed that through this work we are trying to present in a way that is as easy to understand but at the same time as explicit as possible this complex field, we will start with some explanations for the concepts that define the title of this subsection. To better understand what artificial intelligence is, how it works, and what it deals with, it is important to see how it is presented in the specialized works of important authors in this field.

John McCarthy, one of the pioneers of the field of artificial intelligence, in his article *'What is Artificial Intelligence?*', published in 2007, defines artificial intelligence as "*the science and engineering of making intelligent machines, especially through programming.*" [1]. Authors Stuart Russell and Peter Norvig, in their work '*Artificial Intelligence: A Modern Approach*' define artificial intelligence (AI) as "*the study of rational agents and the development of systems that can perceive the environment, understand natural language, learn, and make autonomous decisions.*" [2]. On the other hand, in the work '*Pattern Recognition and Machine Learning*' author Christopher Bishop says that "*artificial intelligence refers to the creation of systems that can automatically learn to recognize and interpret patterns in data, and use this knowledge to solve problems and make decisions.*" [3]

We observe, therefore, that although we are talking about three different works, the approaches of the authors are roughly the same. They all talk about machines that, through programming, learn and are capable of making their own decisions. If we were to formulate our own theory based on the above, we can say that artificial intelligence is a field of science and technology that deals with the creation of systems and machines that can perform tasks and processes that were previously only solvable by humans. These systems are designed to learn from data, solve complex problems, make decisions, and interact with humans. They use algorithms and mathematical models to analyze information and understand and interpret the world around them. However, it is important to note that this field is very complex, but in general, it encompasses specific concepts and techniques depending on the type of application being developed. Thus, we talk about concepts such as machine learning, neural networks, natural language processing, computer vision, and others, some of these aspects being developed in the following pages of the work.

Machine learning is a subfield of artificial intelligence. Explaining this concept is easier and can be easily understood from specialized books in this field, due to the concrete way it operates. In the work '*Deep Learning*' authors Ian Goodfellow, Yoshua Bengio, and Aaron Courville define machine learning as "*a paradigm in which computers learn to solve problems through training on a dataset, rather than being manually programmed. This involves the development of deep neural network models that can learn and understand complex data representations.*" [4]. A similar definition can be found in the work '*Machine Learning: A Probabilistic Perspective*' by author

Kevin P. Murphy: "*Machine learning refers to the development of algorithms that allow computers to automatically learn from data and improve performance in solving specific tasks. This involves identifying statistical models and relationships in data to make predictions or decisions.*" [5]

In this application, the technologies we used are the following:

- as a development environment: MATLAB (abbreviation for "MATrix LABoratory") [8] software. The main reason why I developed the application in this software is the fact that I used this software before and I liked it. I think it is important to have diversity on the market and it is good to have applications in other programs than the established ones. Also, the installation of the toolboxes required for artificial intelligence is simple, and the program offers a very generous documentation that contains a lot of useful information and explanations.

- OpenCv library: which includes many functions and algorithms useful in various image manipulation operations (preprocessing, feature extraction).

- AlexNet Convolutional Neural Network. A convolutional neural network (CNN) "*is a type of deep neural network designed to process data that has a grid-like structure, such as an image. It uses multiple layers of small, trainable filters to hierarchically extract features from the input data. The filters are convolved with the input, computing dot products between the filter weights and local patches of the input. This operation allows the network to automatically learn and detect various features such as edges, textures, and shapes at different spatial locations in the image"* [6]. I chose AlexNet because of the large number of images it can handle and because of the very good results it offers - very low learning time and very high accuracy. AlexNet "*is a convolutional neural network that is 8 layers deep. You can load a pretrained version of the network trained on more than a million images. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals*" [7].

We created a diagram (Fig.1) in which the author exemplifies the technical side of the application and the logic behind it. The recognition process starts when a video frame containing a person is received. Then is the preprocessing phase in which the image is resized and the noise is removed with the help of some OpenCV functions. Also, with the help of OpenCV takes place the detection of the face in the image. Then follows one of the most important phases in the whole process - feature extraction, in which the information from the image is stored in a vector. In the 'Face Recognition' phase, algorithms are used to compare the input face representation with the stored representations in the database and determine recognition results. This result is actually the label with which the system assigns the respective image.

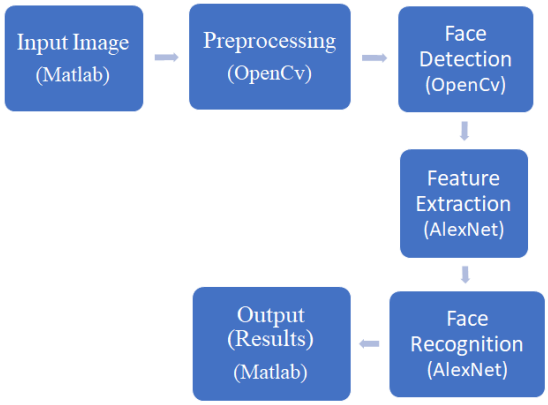


Fig. 1: The archietcture of the aplication

3 Model training

Training a facial recognition system involves feeding a convolutional neural network (in our case, an AlexNet) with a large set of labeled images. During training, the model iteratively adjusts its internal parameters based on the differences between predicted and actual labels.

To perform this training, it is necessary to prepare the training data. This involves collecting a dataset consisting of labeled images, where each image is associated with a specific class or category. Then, the dataset is split into training and validation sets. In our case, these are stored in a directory called "Database" (Fig. 2). Each person has a folder with their name and a certain number of pictures. Our dataset consists of 24 classes and includes images taken from the *Kaggle* website [9] featuring various actors, athletes, or politicians, as well as images captured by the author. The images are 227 x 227 x 3; this means that the input images are represented in a three-dimensional matrix with a height of 227 pixels, a width of 227 pixels, and 3 color channels (red, green, and blue). The images have been divided into training images (80%) and validation images (20%).

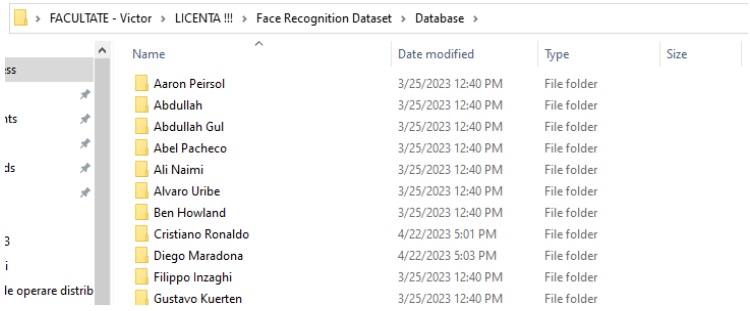


Fig. 2: The contents of the Database Folder

Network Architecture: AlexNet is a popular CNN architecture that consists of multiple layers (Fig. 3). The first few layers perform convolution and pooling operations to extract features from the input images. These convolutional layers are followed by fully connected layers that perform classification based on the extracted features.

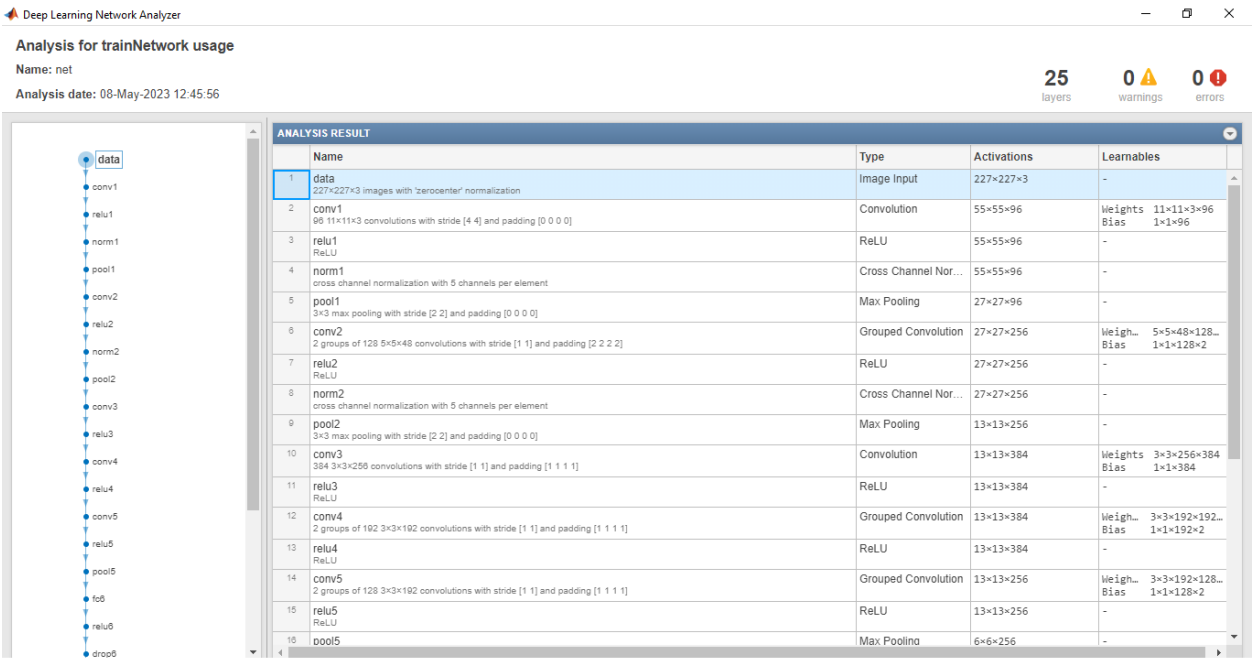


Fig. 3: AlexNet analysis

Initializing the Network: Before training, the network weights and biases are initialized randomly. This step ensures that the network starts with different parameters and can learn diverse features.

Forward Propagation: During training, forward propagation is performed. The input images are fed into the network, and the output is calculated layer by layer. The convolutional layers apply filters to the input, extract features, and produce feature maps. The fully connected layers combine the features and generate class probabilities.

Loss Calculation: The network's output is compared with the ground truth labels to calculate the loss. The loss function quantifies the discrepancy between the predicted and actual values. Common loss functions used in classification tasks include cross-entropy loss.

Backpropagation: Backpropagation is a crucial step for updating the network's weights and biases. It involves calculating the gradients of the loss with respect to the network parameters. The gradients are then used to update the weights and biases using an optimization algorithm, such as stochastic gradient descent (SGD).

Iterative Training (Epochs): Training a neural network involves multiple iterations, referred to as epochs. In each epoch, the entire training dataset is processed by the network, and the weights are updated based on the calculated gradients. The process of forward propagation, loss calculation, and backpropagation is repeated for each epoch.

Activation Functions - ReLU (Rectified Linear Unit): is an activation function commonly used in CNNs. It introduces non-linearity into the network, allowing it to learn complex patterns and relationships in the data. ReLU sets negative values to zero and keeps positive values unchanged, effectively introducing sparsity and improving the network's ability to learn.

Validation and Accuracy: During training, the validation set is used to evaluate the network's performance on unseen data. The accuracy metric is commonly used to measure the model's performance. It calculates the percentage of correctly predicted labels in the validation set.

Fig. 4 presents an image after the training was over. We can see that the model was trained for 10 epochs. Initially, the model had an accuracy of 60%, and during the iterations this increased and the losses decreased. Finally, the accuracy is almost 87%.

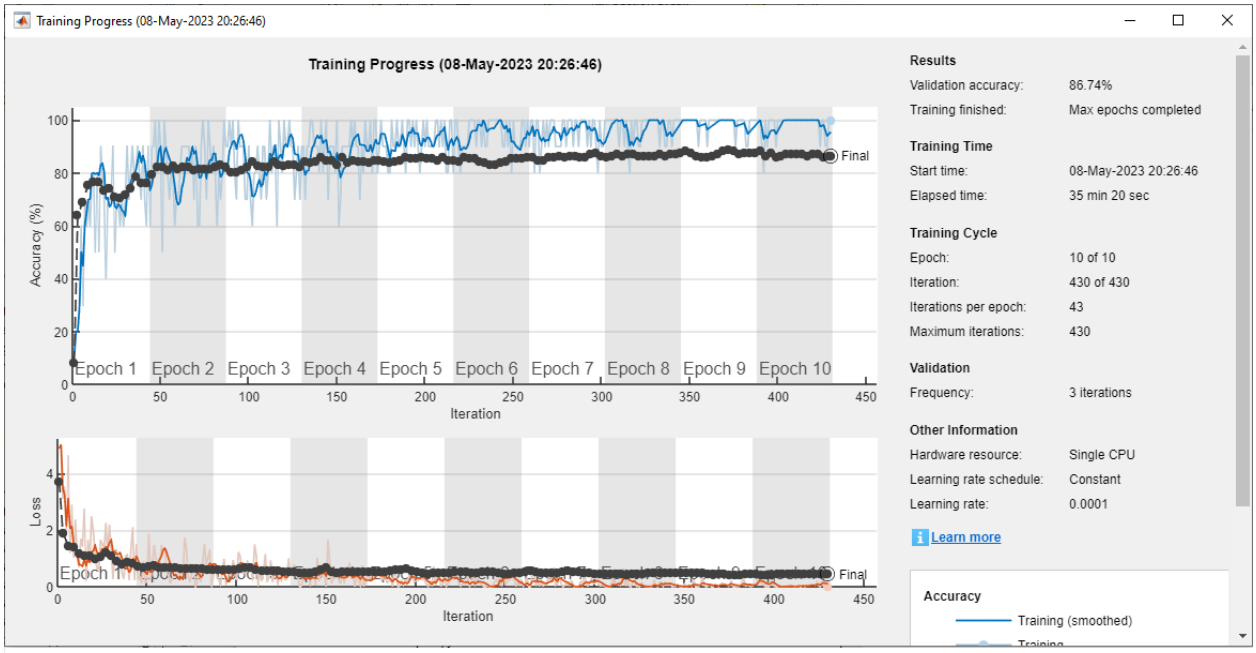


Fig. 4: The chart after finishing the training of the model

4 Presentation of the application

In this chapter the author will present the most important aspects of the application - the way it was built (which components I used), but also the way it works, exemplified by different execution scenarios.

When we open the application (Fig. 5), we have the following components:

- a panel-type object, in which the administrator/person authorized to manage the application will upload the list of persons who must be present in the room. This is done by pressing the "Read Data" button.
- another Panel object, called "Access" which includes two Edit Field (Numeric) type components. In these fields, the number of people who received access or who were denied access will be displayed.
- on the right side we have 2 buttons - "Start", which starts the camera and "Stop", which stops it. By default, these are blocked (cannot be pressed), being functional only after loading the list of people.
- below we have an Axes type object in which the frame captured from the WEB camera will be displayed by pressing the "Get" button on the right. And this (Get button) will be functional only after the list has been loaded and the camera has been turned on.
- the "Exit" button stops the program.

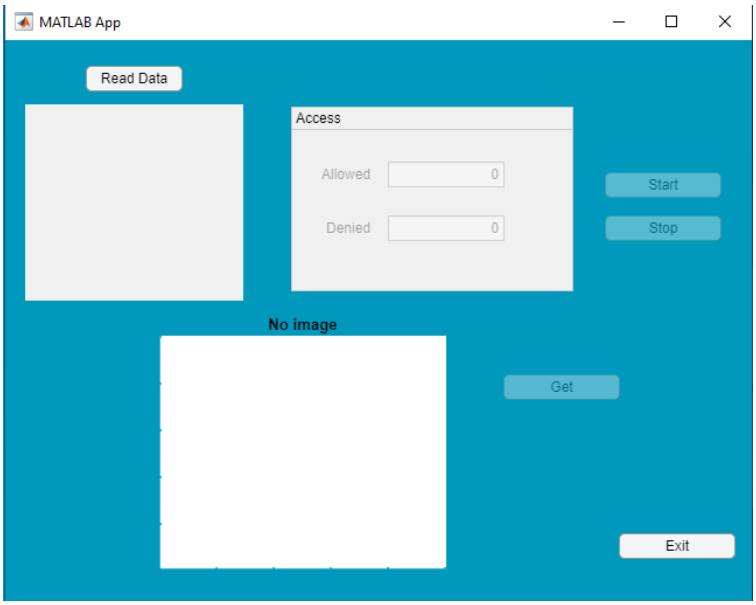


Fig. 5: The Desktop application upon opening

Next, the author will present step by step how the application works.

Step 1) pressing the "Read Data" button. At this moment, a dialog window opens in which the administrator/person responsible for the operation of the application is asked to log in to continue.

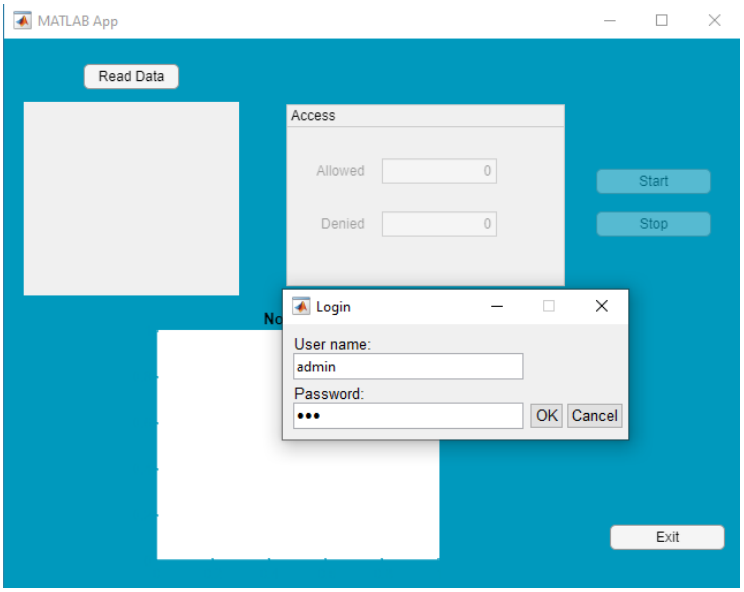


Fig. 6: Pressing the "Read data" button

After successfully logging into the application, the administrator will upload an .xlsx document containing the list of persons who must be present in the room. After the list with these persons has been loaded, the "Read Data" button is blocked (it can no longer be pressed). This was created in order not to be able to load another list once the program has started, so that fraud situations do not occur. Also, the list is not editable for the same security reasons - in order not to delete or add people so that they get access.

Step 2) Pressing the "Start" button (Fig. 7). At this moment, a Video Player object opens in which the images received from the WEB camera are displayed. In this video there is also a detection of people, so this is very useful for the administrator of the application, because when a person has been detected and it is framed in that yellow border, he can press the Get button.

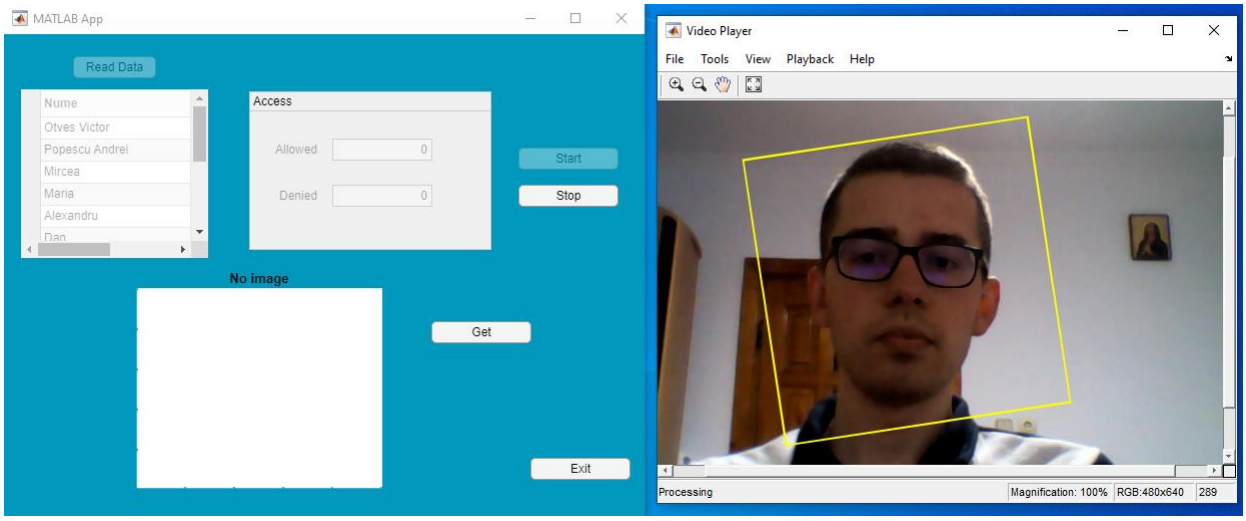


Fig. 7: Pressing the "Start" button

In this moment, after the camera has been turned on, press the "Get" button. Thus, the current frame from the camera will be obtained and it will recognize the person in the image.

Forwards, we will present three possible cases:

A. The situation in which the identified person is on the list (Fig. 8):

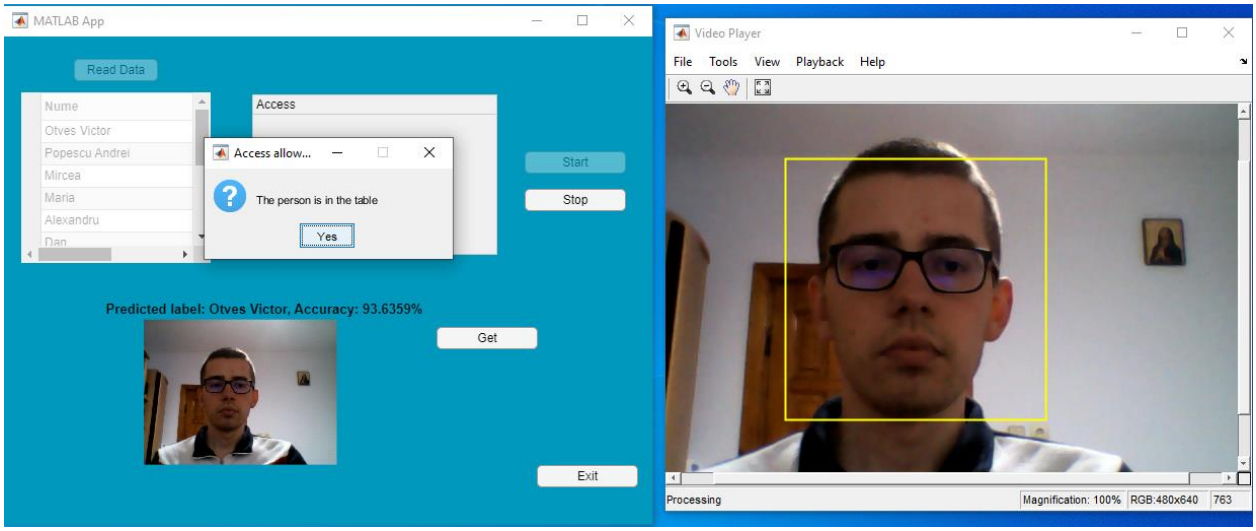


Fig. 8: Confirmation of the person's listing

We notice that the Axes component title has been changed - the application detected a person - Otves Victor (which is true), with an accuracy of 93.64%. A dialogue window also appeared in which it tells us that the person is in the table. Now, if we press "Yes" in the dialog window: The image in the Axes object has been deleted. The cell in the table corresponding to the identified person was colored with green, and the number in the box of people who received access was increased (Fig. 9).

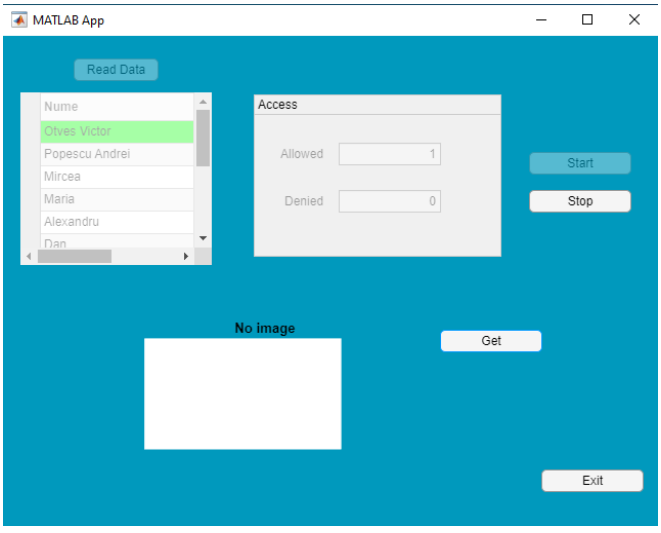


Fig. 9: Results after the person has been identified

B. The situation in which there is no person in the picture (Fig. 10):

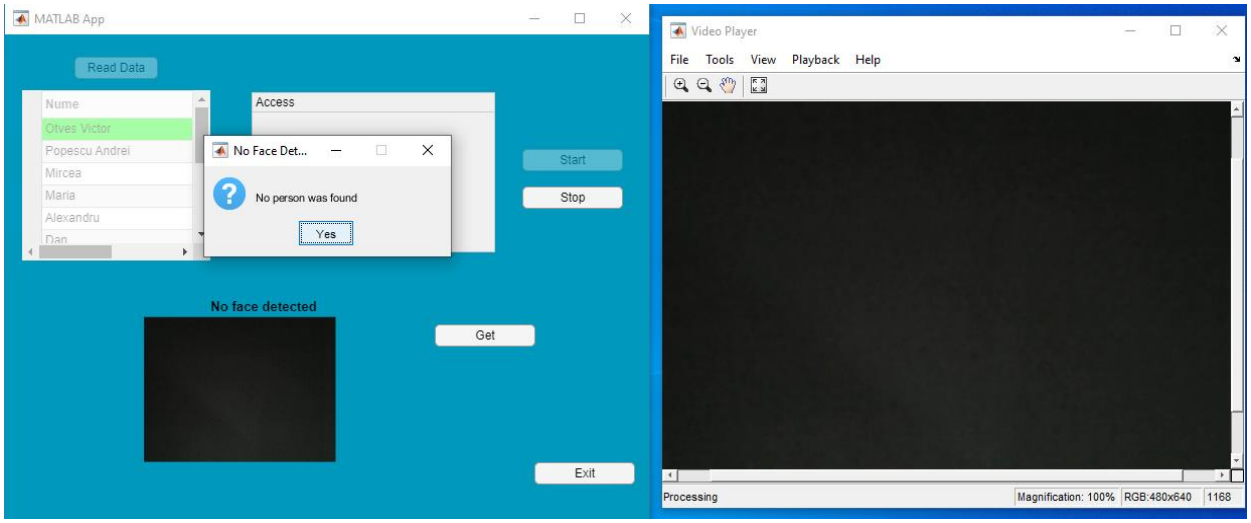


Fig. 10: No person was found in the image

In this situation, if no person was detected in the Video Player, but the “Get” button was still pressed, the title of the Axes object was changed - "No face detected", and the dialog box that appeared tells us that no person was identified. If we press "Yes", the image from Axes will be deleted, but the number of people who have or have received access will not be changed.

C. The situation in which a person is not in the database (Fig. 11):

In this situation, the person was identified in Video Player (detection was made). When the “Get” button was pressed, the image was displayed in the Axes object. The title is now "NOT in database", which means that the person in the image could not be recognized in the database. We also have the dialogue window that warns us about this aspect. If I press "Yes": the image was deleted from Axes, and the number of people with denied access was increased.

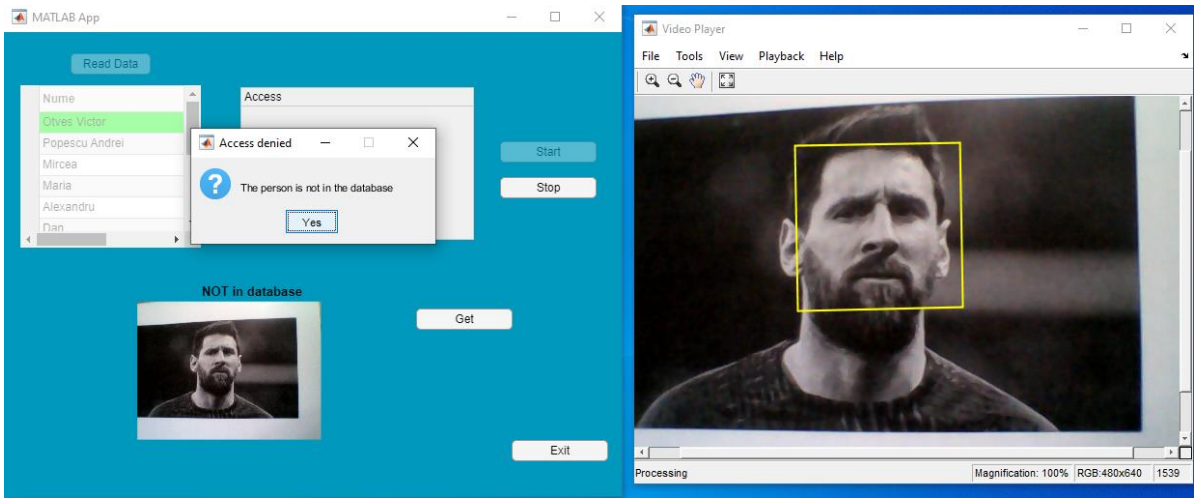


Fig. 11: The person is not listing. Access denied

5 Results

As the author said above, a very important thing when we talk about these intelligent systems is related to their accuracy and the results they can provide. I made a statistic of these results (Fig. 12) using an algorithm that takes 9 random images from the database and classifies them. I performed this operation on a sample of 99 images, and the results are as follows:

- 81 (almost 82%) images were correctly identified;
- 18 images were classified incorrectly.

These results are close to the accuracy obtained after training (87%).

Then, we tested the application on the concrete situation (in a live session) with 5 people (3 are in the database and 2 are not). The results obtained are the following:

- the 3 persons in the database were correctly identified;
- 1 person who is not in the database could not be identified (correct);
- the other person who is not in the database was identified as being in the database (wrong).

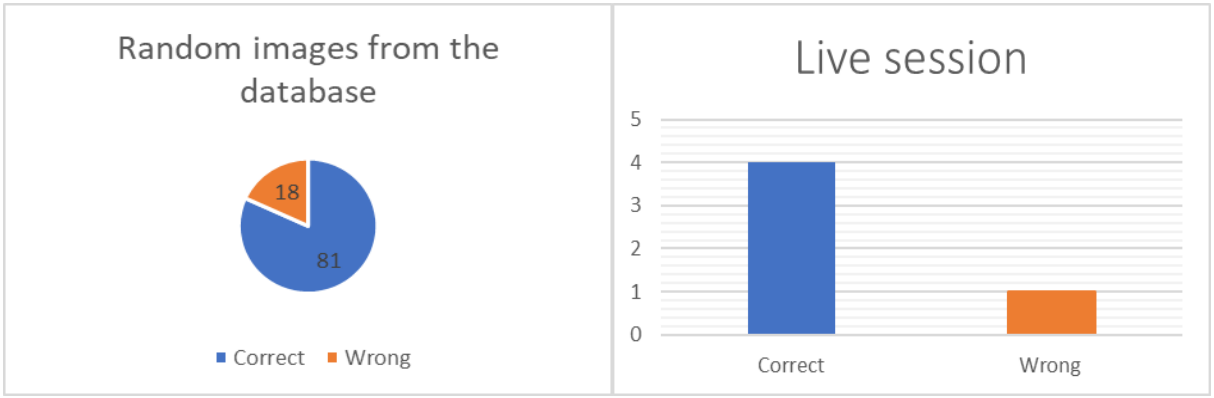


Fig. 12: Results of the person identification

One thing I noticed when I made this analysis of the results is that the application has problems recognizing that people have similar features: they have glasses, a beard, wrinkles or they are frowning.

It is also important to state that I tested the application with pictures from the phone or printed images, but the results were not very good. Images were not recognized, even if the person was in the database or not. This is related to the fact that the image must have some characteristics:

- image brightness can neither be very dark nor very bright.
- the person from image must sit straight and look at the camera. Profile pictures cannot be identified.

In order to improve the performance of facial recognition algorithms by obtaining clear images, in the future I want to use filtering methods to correct the brightness and sharpness of the captured images. These filtering methods, such as histogram filtering (to correct brightness) or average or median filters (to improve sharpness), will be applied and tested to improve the quality of saved images, thus reducing the negative effects of inadequate brightness, low contrast and clarity.

6 Conclusions and the future work

In this moment, it is important to remember that we cannot expect the results to be 100% correct. Just as the human mind does not work correctly all the time, these systems cannot be perfect either, being programmed by humans. Considering the calculation speed and the good results, I consider the application to be a valid alternative, which can treat the stated problem.

Of course, the application is still under development, for this reason there are still things that need to be added or improved: adding images to databases, to create a larger and more complex data set; the frontend part of the application; possibility to save the video stream (session) as proof or verification (feedback).

Acknowledgement: This study was supervised by Assoc. Prof. PhD Laura-Florentina Stoica, from Lucian Blaga University of Sibiu.

References

[1] John McCarthy, *What is Artificial Intelligence?*, Stanford University, SUA, 2007.
[2] Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson, UK, 2016.
[3] Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2009.
[4] Ian Goodfellow, Yoshua Bengio și Aaron Courville, *Deep Learning*, Cambridge, MIT Press, 2016.
[5] Kevin Murphy, *Machine Learning: A Probabilistic Perspective*, Cambridge, MIT Press, 2012.
[6] Ian Goodfellow, Yoshu Bengio, and Aaron Courville, *Deep Learning*, MIT Press, US, 2016
[7] *AlexNe*, https://www.mathworks.com/help/deeplearning/ref/alexnet.html (accessed on: November 2022)
[8] *Matlab*, https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab (accessed on: November 2022)
[9] *Kaggle*, *LFW - People (Face Recognition)*, https://www.kaggle.com/datasets/atulanandjha/lfwpeople (accessed on: December 2022)

Victor Iulian ŐTVEȘ
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: victorotves@yahoo.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Breezy-Air IoT app for monitoring air quality

Gabriel Claudiu Radu, George Bogdan Trîmbițaș

Abstract

Concerned about the environment and the impact of technology, we e came up with the idea of making an IoT device tailored for assessing air quality within our homes or workplace. "Breezy Air - Air Quality Monitor," is an Arduino-based project equipped with a range of sensors designed to monitor ambient air conditions. It not only provides users with essential information regarding the air they breathe, including temperature, humidity, pressure, and volatile organic compound (VOC) concentrations, but also incorporates alarm features to signal critical situations. In cases where measured values exceed safety limits, the device emits sound and light alerts, effectively warning users of potential danger.

1 Introduction

Today's technology allows users to build and program their own electronic devices. Like most of the inhabitants of planet Earth, we have often asked ourselves questions about the environment, about the quality of the air we breathe. We turned to the possibility of a scientific project aimed at measuring the quality of the air in our home or workplace. Breezy Air is an Arduino project that uses sensors, it's an ambient air monitor. In addition, it has alarm functions for critical situations, giving the user information about the atmosphere he is breathing (temperature, humidity, pressure, concentration of VOC, concentration of dust). It also provides sound and light warning signals when the values exceed the permitted maximums and constitute a potential danger of asphyxiation.

As objectives we set ourselves: the efficiency of the monitoring of the quality of the surrounding air; raising people's health awareness by monitoring the quality of the air they breathe; the development of theoretical, technical and computer skills through the use, assembly and programming of electronic components; extracting relevant scientific data from building a functional device applicable in everyday life.

2 Project description

We used an LCD screen to display data read by the sensors. We used a BME280 sensor that helps us simultaneously read temperature, humidity and atmospheric pressure. Most of the time, people are interested in knowing these values in their own homes or at the workplace. We have also made smoke and flammable gas detection functions based on the use of the MQ-2 sensor and detection functions of dust based on the use of the GP2Y1010AU0F optical sensor that detects light reflected from dust in the

air. In addition to the hardware components, the Arduino software-programming part plays an essential role in the operation of the project.

For portability we chose to monitor temperature, and humidity with ThingSpeak - an open data platform and API for IoT with which we collected, stored, analyzed and visualized data from sensors connected to Arduino [6] and created automatic reactions when the stored data meets certain conditions, specified by the user (for example the temperature measured by a sensor exceeds a threshold value).

Everything is customizable: elements on the screen, buttons, colors, menus, functions, etc. Because the hardware part is built from modules it can be improved to serve other purposes. The device used to measure atmospheric pressure is called a barometer. The sensor used in the project as a barometer works based on the piezoelectric effect and measures pressure in the range of 300 hPa-1100 hPa (hecto Pascals), with an accuracy of 0.03 hPa. The range in which the measured pressure can be found corresponds to an altitude between 9000 m and -300 m.

2.1 Hardware modules

a) WT32-SC01 V3.2 – Arduino kit (Fig. 1) equipped with:

- ESP32 - is a feature-rich dual-core microcontroller with built-in Wi-Fi and Bluetooth connectivity for a wide range of applications.
- Screen with a resolution of 480 by 320 pixels.

Fig. 1: WT32-SC-01

b) BME280 sensor (Fig. 2) - Used to measure:

- Temperatures between -40 °C and + 85 °C;
- Relative humidity in the range 0% - 100% (depends on temperature and pressure);
- Atmospheric pressure in the range of 300-1100 hPa;
  Communication with this sensor is done through the I2C protocol. We chose this sensor for its accuracy and low power consumption, providing the opportunity to make the project portable.

Fig. 2: BME280

The device that measures atmospheric pressure is called a barometer [3]. The sensor uses the piezo-electric effect to measure atmospheric pressure between 300 hPa – 1000hPa, with a precision of 0.03 hPa. The sensor reads the values in hPa and we transform it to mmHg using the following formula [4]:

$$p[mmHg] = p[Pa] \cdot \frac{0,760}{101,325} \quad (1)$$

The atmospheric pressure decreases with the increase of height.

$$p = p_0 \cdot e^{\frac{gM\,h}{RT0}} \quad (2)$$

c) MQ-2 sensor

Used to measure the concentration of volatile organic compounds (VOC). It can measure the concentration of LPG, Carbon Monoxide, Alcohol, Propane, Hydrogen, Methane and smoke, is suitable for air quality projects and can be used as a potential fire detector. The data reading is done from an analog pin that changes its voltage according to the concentration of these gasses in the air. It can be done from a digital pin with the on-off states according to the value set in the potentiometer. In our project we only use the analog pin but the digital pin is connected to an LED on the wiring that indicates if the digital output is enabled. We chose this sensor for the wide range of gasses to which it is sensitive, so we can monitor the atmosphere in great detail.

Fig. 3: MQ-2

The level of gasses in a room causes the MQ-2 to generate a voltage, low if low and high if high. The sensor's operating temperature is between -20°C and +50°C and the required current is 150 mA. In Pic.3 the sensor is mounted on a brick. Also, the sensor is accompanied by a potentiometer that allows adjusting the sensitivity to VOC. The sensor's supply voltage must be 5V.

d) GP2Y1010AU0F produced by Sharp

Used to detect dust particles. The sensor's operating temperature is between -10°C and +65°C, and the required current is 11 mA. The sensor supply voltage must be between 4.5V and 5.5V.

Fig. 4: Dust sensor

e) WeMos mini D1

Based on an ESP8266 microcontroller with WiFI connection. It is used to read the temperature, atmospheric pressure and humidity values using a proprietary BME280 sensor.

Fig. 5: WeMos mini D1

f) Buzzer

Used for sound alerts - it is an electronic device that acts as a translator. It produces a loud or humming sound while being supplied with energy. Integrated with Arduino, when an event is generated that we want to warn (e.g. temperature when it exceeds a certain value) we program the microcontroller to send a signal to the buzzer if that event happens and so it alerts us with a sound.

Fig. 6: Buzzer

g) Breadboard

Used for mounting components- a breadboard is a cheap, easy-to-use piece of hardware for wiring electrical circuits. I used it for quick assembly without the need to solder wires, for testing projects.

Fig. 7: Breadboard

h) WeMos mini D1 (Fig. 5), or like we call it Breezy mini

It is based on an ESP8266 microcontroller with Wi-Fi connection. It is used for reading temperature, atmospheric pressure, and humidity values through its own BME280 sensor.

3 IoT service

The development of IoT is based on two things: high-speed data transfer over the Internet and the development of chips with high processing power embedded in Internet-connected devices. One of the advantages of IoT is the possibility to have real-time access to data extracted from the surrounding

environment, provided by sensors (e.g. temperature, humidity, atmospheric pressure, concentration of volatile organic compounds (VOC) and concentration of dust (CPraf)).

ThingSpeak is an open data platform and API for IoT that allows the user to collect, store, analyze and visualize data from sensors connected to devices such as Arduino, Raspberry Pi, BeagleBone Black and others. In addition, automatic reactions can be created when the stored data meets certain conditions, specified by the user (e.g. the temperature measured by a sensor exceeds a threshold value). We chose to use IoT services that allow devices to be connected to the Internet, tracking for their carriers the temperature, humidity, pressure, VOC, CPraf, the differences between their minimum and maximum values , and the generation of useful statistics (Fig. 8).

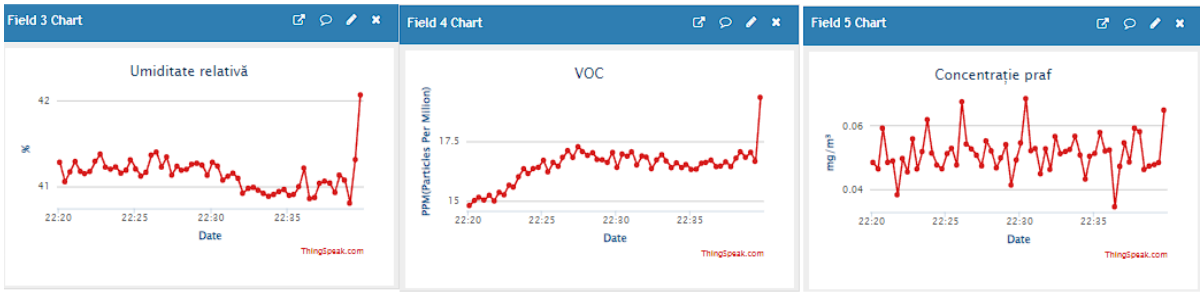


Fig. 8: Temperature, Humidity, Atmospheric Pressure graphs.

With the help of ThingSpeak, we were able to build an app (Fig. 9) that can show the real-time values from Breezy and Breezy Mini on mobile phones and send alerts if the value exceeds a level set by the user in the settings menu of the device.

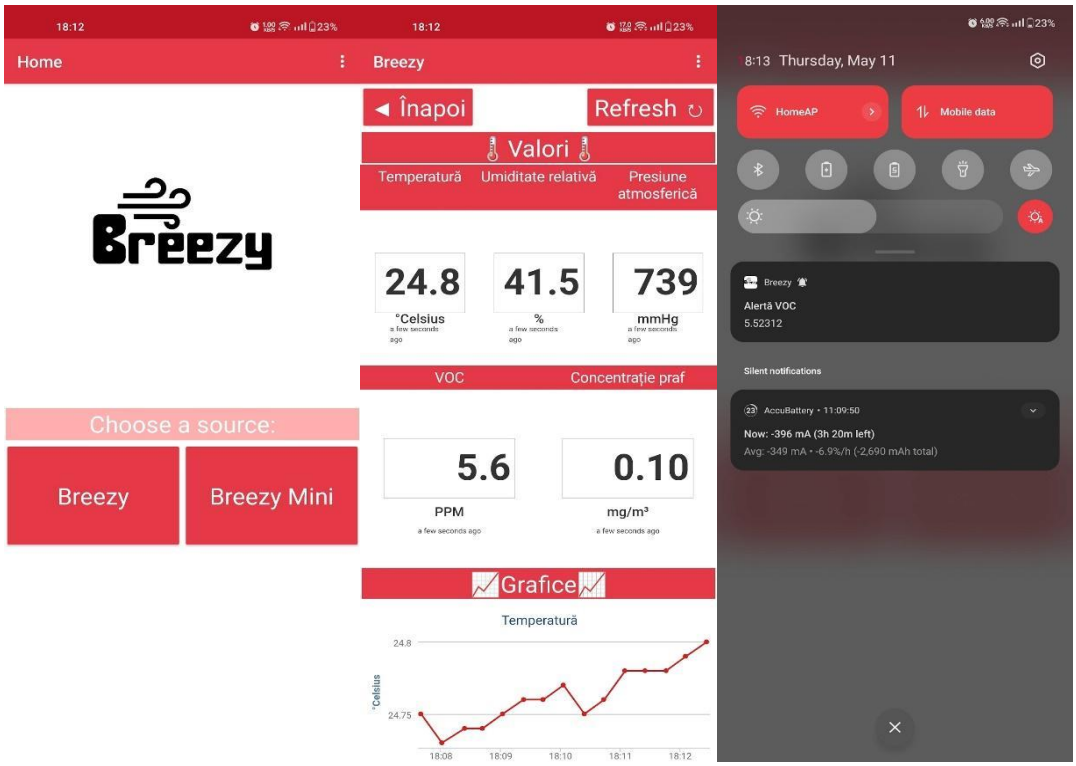


Fig. 9: The interface of the app and the notifications.

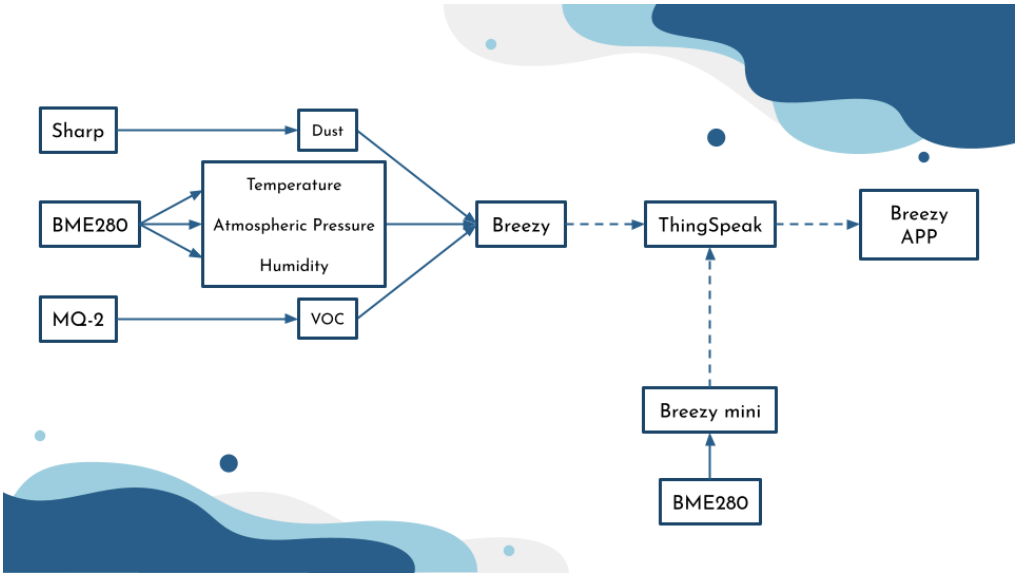


Fig. 10: Diagram showing how the data is transferred between devices

4 Code snippets

In addition to the hardware components, the software plays an important role in the project's operation. Below are key parts of the Arduino board programming [1] for the project. In Fig. 11, a code sequence is captured that reads the temperature, atmospheric pressure, and humidity values from the BME280 sensor.

```
void SenzorBME280() {

  Temperatura = bme.readTemperature();
  Presiune = bme.readPressure() / 100.0F;
  if (BmeUM == 2)Presiune = Presiune * ratioHpaMmHg;
  Umiditate = bme.readHumidity();
```

Fig. 11: Reading temperature, atmospheric pressure and humidity

The code sequence in Fig. 12 saves the minimum and maximum values for temperature, humidity, and atmospheric pressure into variables (tmin, tmax - for minimum and maximum temperature; umin, umax - for minimum and maximum humidity; pmin, pmax - for minimum and maximum atmospheric pressure).

```
if (Temperatura > tmax)tmax = Temperatura;
if (Temperatura < tmin)tmin = Temperatura;

if (Umiditate > umax)umax = Umiditate;
if (Umiditate < umin)umin = Umiditate;

if (Presiune > pmax)pmax = Presiune;
if (Presiune < pmin)pmin = Presiune;
```

Fig. 12: Determining the maximum and minimum values for temperature atmospheric pressure and humidity.

The code sequence in Fig. 13 reads the value of volatile organic compounds (VOC) concentration in the air using the MQ2 sensor.

```
void SenzorMQ2() {
  VOCvalue = 0;

  for (int i = 1; i <= 5; i++) {
    MQ2.update();
    VOCvalue += MQ2.readSensor();
  } VOCvalue /= 5;

  //MQ2.serialDebug();
  //Serial.println(VOCvalue);
```

Fig. 13: Reading the VOC concentration.

In the accompanying code sequence (Fig. 14), the value of the variable CPraf is set to 0, and then the program takes an arithmetic average of 5 values to obtain a stable value.

```
void Senzor_Praf() {
  CPraf = 0;
  for (int i = 1; i <= 5; i++) {
    digitalWrite(led_pin, LOW);
    delayMicroseconds(280);

    float mesured = analogRead(analog_pin);
    delayMicroseconds(40);

    digitalWrite(led_pin, HIGH);
    delayMicroseconds(9680);

    CPraf += (0.172 * (mesured * (3.3 / 4096)) );

  } CPraf /= 5;
```

Fig. 14: Calculating the dust concentration.

In the code sequence below, the program reads the maximum and minimum values for temperature, humidity, and atmospheric pressure from the external module connected to the ThingSpeak open data platform.

```
void ThingSpeakRead() {
  if (TSOn && (millis() - LastTSRead >= TSReadDelay)) {
    LastTSRead = millis();

    if (ReadIndex == 1) {
      ETemp = ThingSpeak.readFloatField(1748573, 1);
      if (ETemp > ETempMax)ETempMax = ETemp;
      if (ETemp < ETempMin)ETempMin = ETemp;
    }

    if (ReadIndex == 2) {
      EUmid = ThingSpeak.readFloatField(1748573, 2);
      if (EUmid > EUmidMax)EUmidMax = EUmid;
      if (EUmid < EUmidMin)EUmidMin = EUmid;
    }

    if (ReadIndex == 3) {
      EPres = ThingSpeak.readFloatField(1748573, 3);
      if (EPres > EPresMax)EPresMax = EPres;
      if (EPres < EPresMin)EPresMin = EPres;
    }
  }
}
```

```
void ThingSpeakWrite() {
  if (TSOn  && (millis() - LastTSWrite >= TSWriteDelay)) {
    LastTSWrite = millis();
    ThingSpeak.setField(1, Temperatura);
    ThingSpeak.setField(2, Presiune);
    ThingSpeak.setField(3, Umiditate);
    if (TimpIncalzireMQ2 < millis())ThingSpeak.setField(4, VOCvalue

    int x = ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);

    ThingSpeak.setField(5, CPraf);
  }
}
```

Fig. 15: Reading and Writing values on the ThingSpeak server.

In the "Settings" menu, we can modify various parameters using 4 buttons: +, -, <, >.

The < and > buttons are used to change the variable we want to modify. Some examples would be: time, values for triggering temperature, humidity, VOC alarms. The + and - buttons are used to increment or decrement the selected value.

The displayed text on the screen is divided into 2 categories:

1. Static - this text is written once on the screen and does not update.
2. Dynamic - this text updates (once per second) and is dedicated to values that change, such as temperature, and humidity. For these types of updates, we use a "One-Second Loop". This loop contains all the functions that read values from sensors and display dynamic information on the screen [2].

5. Conclusions and further development paths

The project is built from modules and parts that can be changed, together with the flexibility of the program generates a multitude of development directions. We plan to develop a portable version of the detector, another beneficial addition would be the use of a rain sensor useful in agriculture. On the IoT side, we would like to achieve remote monitoring of a large number of devices and the possibility to intervene/control as needed (smart home via the ESP-NOW communication protocol). We can make the project portable by tracking the altitude difference between successive positions and calculating altitude as a function of atmospheric pressure. The monitoring can be more extensive by displaying the grades based on the values read from the sensors (e.g. the value scale should be from 0 to 10: 0 for extremely contaminated air, respectively 10 for pure-ideal air).

The project allowed us to use our imagination, programming knowledge, to develop creativity, investigate scientifically, to enjoy the benefits of teamwork, carefully guided by the teacher and combining knowledge of electronics, physics, robotics, and programming we made a device that sensitizes us to breathe cleaner air. To maximize our performance, we worked in parallel with parts equivalent to those of the WT32-SC01 kit on a breadboard. The most significant difference was the screen (an ILI9488), having the same resolution, but with non-functional touch (we used physical buttons, for which functions can be found in the program).

The level of customization is a key feature of this project, because all the hardware is built from modules and can be easily modified and through the appropriate changes in the program many versions of the module can be created for different purposes. Everything is customizable: elements on the screen, buttons, colors, menus, functions, etc.

One of the advantages of our project is based on a development kit, and it is easy to put together, along with the program making it a very effective way to learn Arduino programming and electronics concepts.

Acknowledgement: This work was supervised by Professor *Ramona Humeniuc*, from *"Titu Maiorescu" National College Aiud, Romania*.

References

[1] Anghel,T., *Programarea plăcii Arduino*, Editura Paralela 45, Pitești, 2016
[2] Gregoire, M., *Professional C++*, JohnWiley & Sons, 2014
[3] SEARS, Francis, *Fizica*, Editura Didactică și Pedagogică București, 1983
[4] Stroustrup, Bjarne, *Programming: Principles and Practice Using C++(2nd Ed.)*, Addison Wesley, 2014
[5] Library for LCD display: https://github.com/Bodmer/TFT_eSPI (link accessed, January 2023)
[6] Arduino, https://www.arduino.cc (link accessed, January 2023)

Gabriel Claudiu RADU
"Titu Maiorescu" National College Aiud
Mathematics-informatics intensive English
ROMÂNIA
E-mail: radugabriel796@gmail.com

George Bogdan TRÎMBIȚAȘ
"Titu Maiorescu" National College Aiud
Mathematics-informatics intensive English
ROMÂNIA
E-mail: trimbitasb@gmail.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Learning to Unpermute

Muhammad Zakriya Shah Sarwar, Nehmiya Shikur

Abstract

This paper investigates the effectiveness of deep learning models in matching distributions, using the problem of unpermuting images as a case study. Unpermuting an image refers to the process of reversing the effects of image permutation, which involves randomly shuffling the pixels of an image. The research is divided into two phases. The first phase involves the development of a supervised learning model that trains a neural network on a paired data set of original and permuted images. The second phase focuses on the development of an unsupervised deep learning model that trains the network on unpaired images. The study demonstrates the ability of deep learning models to match distributions and address the problem of unpermuting images. The findings of this study can provide insights into the general problem of distribution matching in deep learning.

1 Introduction

Deep learning models have achieved remarkable success in learning to classify images and objects, enabling significant advancements in the field of computer vision. However, a less explored area is how deep learning models can learn to match the distributions of data sets. To investigate this topic, we chose the problem of unpermuting images as a toy example. Unpermuting images involves reversing the effects of image permutation, which is the process of shuffling the pixels of an image in a random order. In this paper, we demonstrate how deep learning techniques can be used to train neural networks in supervised and unsupervised settings to match the distributions of original and permuted images, thus addressing the problem of unpermuting images.

In the initial stage of our study, we concentrated on building supervised learning models. This involved training a neural network using a paired data set that consists of original and permuted images in a matched manner. The aim of training these models was to enable them to accurately generate the original image from their permuted counterpart by minimizing the difference between the unpermuted and original images. Through this approach, we sought to demonstrate the ability of deep learning models to learn the distributions in the images and to successfully unpermute them. The development of these supervised learning models laid the groundwork for our subsequent investigation into unsupervised deep learning models.

In the second phase of our study, we aimed to develop unsupervised deep learning models with an ambitious goal of training the network using an unpaired image data sets. The data sets consists of original and permuted images, where they are not matched to each other. This approach is challenging as it requires the model to learn to match the distributions of two sets of images without direct supervision.

In summary, section 2 of our paper provides a detailed overview of our image permutation techniques and the formation of our paired and unpaired image data sets. In sections 3 and 4, we discuss our supervised and unsupervised learning models, consecutively. Finally, section 5 presents the findings and interesting aspects of our study.

2 Data

For our study, we utilized both gray scale and color images, all of which underwent the same permutation process. Subsequently, we organized these permuted images into paired and unpaired formats for supervised and unsupervised models.

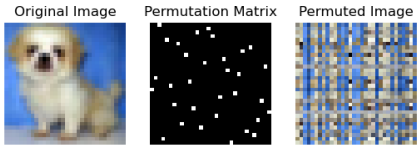
2.1 Permutation

Permuting an image involves rearranging the pixels of an image to create a new image that looks different from the original image. There are several methods to permute an image. For example, we could divide an image into smaller grid and then shuffle those grids. The other option is to scramble all pixels of the image, so that it is completely unrecognizable for human's eye. We took the later approach.

To permute the images on pixel levels we multiply the image with a permutation matrix from left and right. A permutation matrix is a square matrix that has exactly one entry of 1 in each row and each column and 0s elsewhere.

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(a) An example of 5×5 permutation matrix



(b) The permutation matrix in the middle shown as a binary image. Permuted Image is generated by multiplying original image with permutation matrix from left and right.

Figure 1

Let $\mathbf{P}$ be the permutation matrix and $\mathbf{X}$ an image. Then our permuted image $\mathbf{Y}$ can be described by equation (1).

$$\mathbf{Y} = \mathbf{PXP} \tag{1}$$

Where $\mathbf{X}, \mathbf{P}, \mathbf{Y} \in \mathbb{R}^{m \times m}$. Multiplying the image only from right side will permute the columns of the image and multiplying from left side will permute only rows. So we first multiply image from one side and then from other side with the same permutation matrix. Order of multiplication does not matter.

If the permutation matrix is known, then reverting to the original image is very simple.

$$\mathbf{X} = \mathbf{P^{-1}YP^{-1}} \tag{2}$$

where

$$\mathbf{P^{-1}} = \mathbf{P^{T}} \tag{3}$$

if the permutation matrix is not known, then to the best of our knowledge, it is not possible to solve equation (1) for $\mathbf{P}$. One way to figure out $\mathbf{P^{-1}}$ in equation (2), to get the original image back, can be trying all possible permutation matrices. But for a $m \times m$ permutation matrix, there are $m!$ possible different configurations (of which half matrices are transpose of the other half). For matrices of relatively large size trying to find the P^{-1} by going through all possible configurations will take huge time.

2.2 Paired and unpaired data sets

Paired data sets refer to a type of data sets in which permuted images are paired with their original images. In all training iterations, loss is calculated between the output image of the unpermuting model and the original image which was used to permute the input image of the model.

On the other hand, unpaired data sets do not have paired permuted and original images. Thus during training we can not match the output of the unpermuting model directly to it's original counterpart. Instead, a more sophisticated approach had to be developed which is presented in section 4.

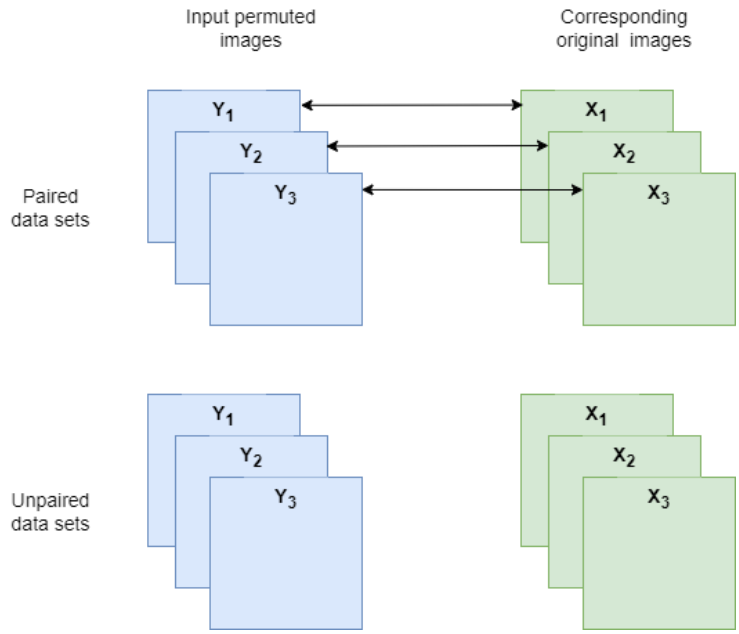


Figure 2: Illustration of paired and unpaired data sets.

Given that the images in the paired data sets are accurately matched, it enable us to use a supervised learning approach to train our models. In other words, we are able to provide the model with both the input (permuted image) and the expected output (original image). By doing so, we can optimize the model's parameters to minimize the error between the output and the expected output. On the other hand, for the unpaired data sets, we have to use an unsupervised learning approach since we don't have the matching expected output.

3 Supervised approach for paired data sets

In this stage of our research we utilized a paired data set to map permuted images to their corresponding original versions. This approach laid the foundation for our subsequent work in the unsupervised setting, and ensured the effectiveness of our models.

We have used supervised deep learning models, such as Single Layer Perceptron (SLP) [6] and Auto-encoders [3, p. 499], to address this problem by learning a mapping from permuted images to the original images. We trained the model on a large data set of permuted images, along with their corresponding original images. These models have been successful in recognizing patterns in the permuted images and were able to map them back to the original ones.

3.1 Classification

To understand and demonstrate the complexity of different type of Image data sets, we initially sought to develop several basic supervised learning models capable of classifying permuted images. In this pursuit, a cross-entropy loss function [2], as in equation (4), was utilized to evaluate the ability of a multi layer perceptron (MLP) [3, p. 164] model to correctly label permuted images, which correspond to their semantic content, for example 0-0-9 digits in MNIST data set [1]. This function employs the softmax [3, p. 180] function to predict the probability of various target categories and compares the distribution of the true labels c to that of the predictions $\hat{c}$.

$$L = \sum_{i=1}^{N} \sum_{j=1}^{M} c_{ij} log(\hat{c}_{ij}), \tag{4}$$

where N is the total number of examples and M is the total number of classes.

In this stage we trained four different models, namely Multi-Layer Perceptron (MLP), convolutional Neural Network (CNN) [3, p. 326], MLP with dropouts [7], and CNN with batch normalization[4]. These models were trained on three separate image data sets, namely MNIST [1], FashionMNIST [8] and CIFAR-10 [5], with the objective of classifying permuted images. MNIST and FashionMNIST has 70000 images each of size 28×28, of which 60000 images are used to train the models and 10000 images are used for testing. CIFAR-10 consists of 60,000 color images, each sized at $32 \times 32 \times 3$, distributed across 10 classes, with 6,000 images per class. To evaluate their performance, the training loss of each model on each data set was measured and plotted in a graph, which is shown below. By examining this graph, we can compare the training performance of the four models and determine which one performs the best for the task of classifying permuted images.

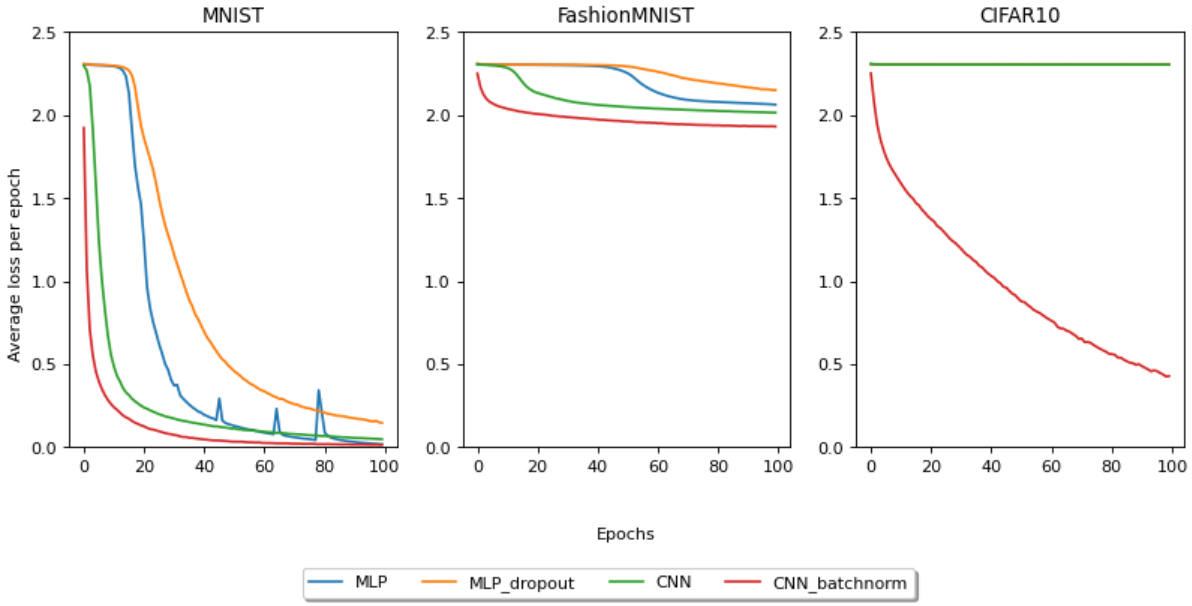


Figure 3: Classification Losses

We also sought to evaluate the accuracy of our classification models on the MNIST [1], FashionMNIST [8], and CIFAR-10 [5] data sets, in addition to analyzing their training performance. The table presented below indicates that all four models performed well in terms of accuracy, but the accuracy of the models on CIFAR-10 images was lower than that of MNIST and FashionMNIST. From these results, we can conclude that CIFAR-10 is a rather complex data set, as it has three color channels, while MNIST and FashionMNIST only have one. And also because CIFAR10 consists of realistic images and MNIST and FashionMNIST are synthetic sketches.

Model's Accuracy			
Model	**MNIST**	**FashionMNIST**	**CIFAR10**
MLP	0.9656	0.6441	0.0999
MLP + Dropout	0.9461	0.3668	0.1013
CNN	0.9728	0.7735	0.0999
CNN + Batch Norm	0.9581	0.7748	0.4885

Table 1: Our models accuracy on test data of different data sets. Best results are highlighted with boxes.

3.2 Unpermuting

Given that we have developed models capable of classifying various permuted images, the next step is to construct models that can reconstruct these permuted images to their originals. Our initial attempt at this involved using a Single-Layer Perceptron (SLP) model with the same dimensions for its input and output. We trained the SLP model using stochastic gradient descent [3, p. 149] and mean squared error [3, p. 131] as the loss metric to reconstruct the original image from its permuted form.

$$L = \sum_{i=1}^{N} \|\hat{\mathbf{X}}_\mathbf{i} - \mathbf{X}_\mathbf{i}\|_2^2, \tag{5}$$

where $\hat{\mathbf{X}}$ is model's output.

The SLP models underwent a training to reconstruct both gray-scale and RGB images, starting with MNIST and eventually progressing to CIFAR-10. Impressively, it was able to accurately reconstruct the permuted images that were given to it, as evidenced by Figure 4 and Figure 5.

Figure 4: Unpermuting single channel MNIST images with an SLP model. The images in the first column represent the original images while the images in the second and third column represent their corresponding permuted and unpermuted versions consecutively.

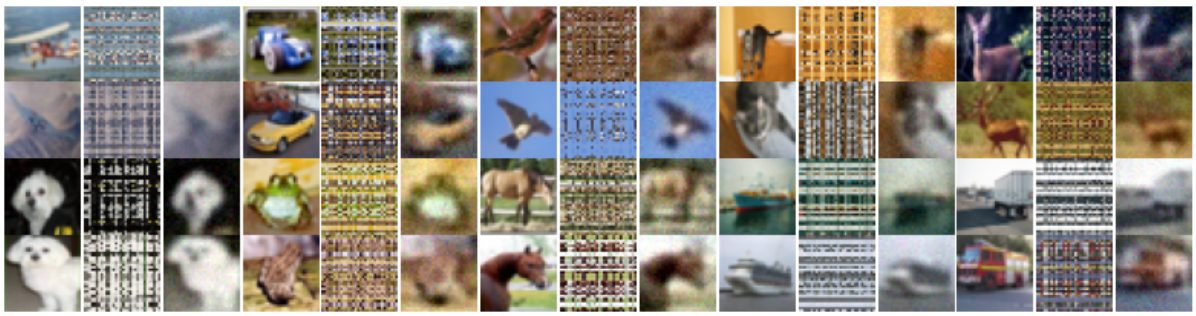


Figure 5: unpermuting multiple channel CIFAR10 images with an SLP model

In addition to the SLP model, an auto-encoder model that encodes the pixel values of the image by reducing it into a smaller latent representation and decodes this representation into a dimension that is equal to the input, as shown in Figure 8, were trained and tested on MNIST, FashionMNIST and CIFAR-10 data-sets. The model had the same input and output dimensions with a parameterized latent representation. Additionally, the mean squared error loss function metric as seen in equation (5) was used for the training of the model.

The auto-encoder model we built showed promising results in unpermuting images across different data sets, as depicted in Figure 5 and 6 for MNIST and FashionMNIST data sets, respectively. These results

demonstrate the model's ability to reconstruct permuted images, which further validates the effectiveness of our approach in handling permuted image classification and reconstruction tasks. Remarkably, the model achieved this despite being trained for only a few epochs. This is a significant finding that highlights the efficiency of our model and its potential for further development and improvement. Overall, our auto-encoder model has demonstrated its ability to perform well on permuted image reconstruction tasks.

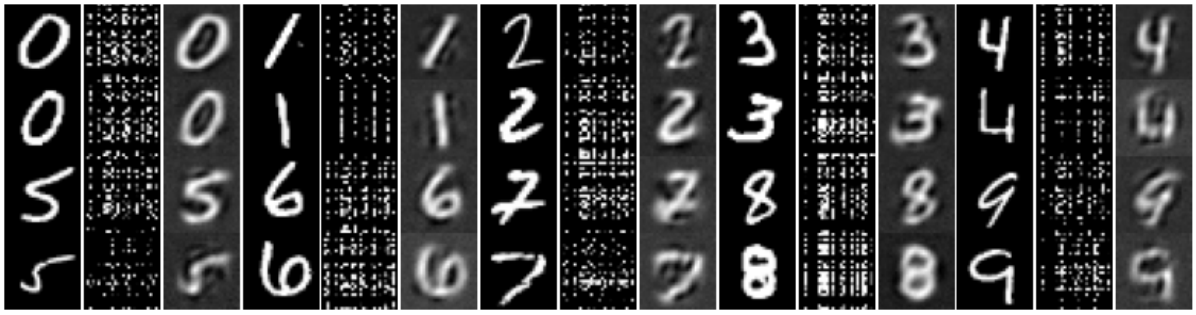


Figure 6: Unpermuting images from MNIST data set using an auto-encoder model trained for only 20 epochs

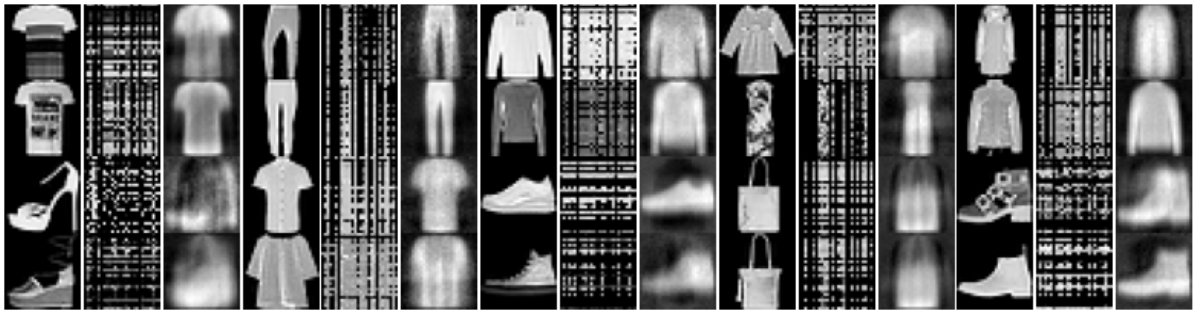


Figure 7: Unpermuting images from FashionMNIST data set using an auto-encoder model trained for only 20 epochs

Our models exhibited varied levels of success in unpermuting different types of images. While MNIST and FashionMNIST data sets were relatively easier for the models to unpermute, CIFAR-10 presented a greater difficulty. Surprisingly, the models trained on MNIST and FashionMNIST images were able to successfully unpermute each other; however, they struggled with images from CIFAR-10. We hypothesize that the sparsity of the image matrix in CIFAR-10 images likely contributed to the models' difficulty in learning and unpermuting them. For more details and visualizations of our experiments, please refer to the appendix section of our paper.

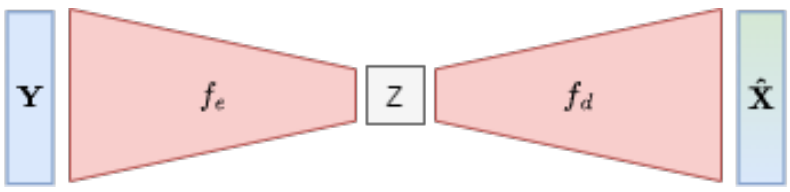


Figure 8: An Auto-encoder model, that uses a lower-dimensional latent representation z to encode the input data, which is then decoded back to reconstruct the original input.

3.3 Classifier and Generator

By dividing the previously mentioned auto-encoder model into two distinct models, a classifier and a generator, we were able to generate images from integer values, corresponding to the classes of the images. The two models were trained separately, which gave us more flexibility and control over the generated images. The classifier model is used to determine the class of the image, while the generator model is responsible for generating the images based on the input values. Overall, this approach proved to be effective in generating images of given classes from integer-valueed class labels.

The Classifier model is trained on the permuted images to classify them, while the Generator model takes the output values of the Classifier model, which are integer values, as input and is trained to construct an image matrix that is later plotted as an image. The Generator model uses mean squared error as a loss function, which takes a randomly selected image that corresponds to the label of the original image. This approach allowed us to generate new images by feeding integer values into the Generator model, enabling us to explore the space of possible images. The use of the Classifier model as an intermediate step ensures that the generated images belong to the same class as the original image, while the use of mean squared error as the loss function ensures that the generated images resemble the original image. The results of this approach can be seen in Figure 9.

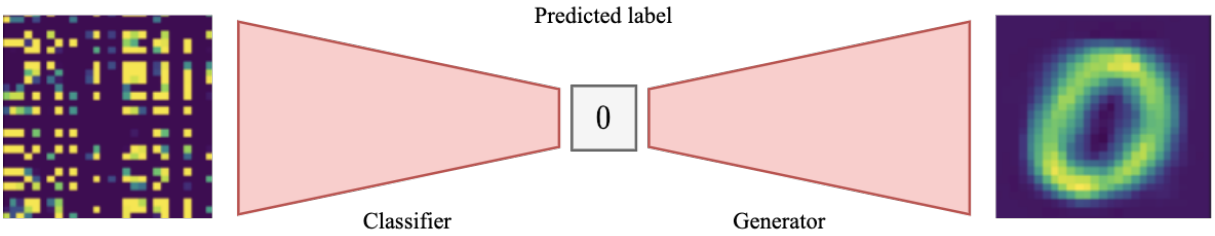


Figure 9: Generating images from integer values using our classifier and generator model

Now that we have successfully developed models that can classify, unpermute and generate images by learning their distributions, it is worth noting that all these tasks were possible due to the availability of original images that could be used for learning to reconstruct and unpermute the permuted images in a supervised learning setting. In the next chapter of the paper, we will explore different techniques for unpermuting images without any prior information about the original images or labels. This presents a more challenging and realistic scenario, where we will need to rely on unsupervised learning techniques to infer the structure of the data and its underlying patterns. This would help to address the real-world problem of dealing with permuted images where we do not have access to the original images or their corresponding labels.

4 Unsupervised

We are now focused on addressing the challenge of unpermuting images in the absence of their labels. This is the next stage of our work, where we aim to develop a solution for unpermuting images without having access to their original or unpermuted versions. This presents a new challenge, as the models we developed earlier were trained on the original images to unpermute them. Without prior knowledge of the original images, we need to develop new techniques to unpermute the images in a completely unsupervised manner. This will require novel approaches and creative thinking to overcome this new challenge. We will discuss our proposed methods and the results in the next section of the paper.

A crucial challenge in addressing the issue of unpermuting an image in an unsupervised learning setting is determining the appropriate loss metric for training our model. To address this, we employed a Cycle Consistency loss [9]. This loss metric is designed to ensure that our generator model can generate an image and then reconstruct it back to its original form as closely as possible. In other words, we aim to ensure that the generated image is both realistic and accurate, and that the process of generating and reconstructing it does not introduce any unexpected changes. Our use of Cycle Consistency loss enables

our model to learn the underlying structure of the image, and to use this knowledge to unpermute images in the absence of the original images.

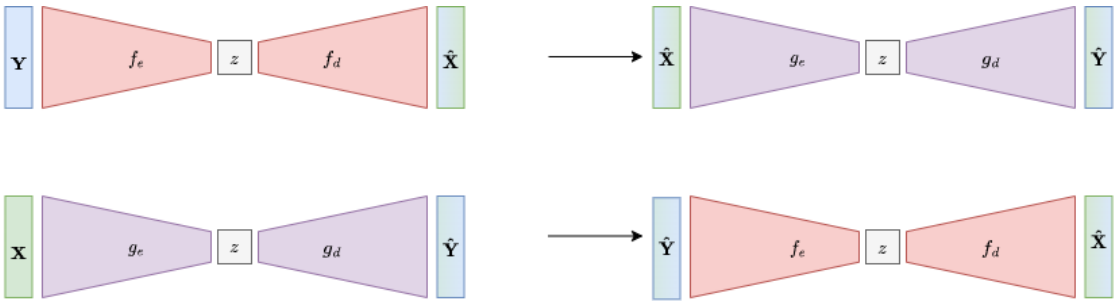


Figure 10: During the training cycle, an unpaired permuted image $\mathbf{Y}$ is processed through the model f and used as input for the model g. Likewise, an unpaired original image, which is denoted as $\mathbf{X}$, is fed into the model g and its output is passed through the model f. Moreover, f_e and g_e represent the encoder, while f_d and g_d represent the decoder sections of the auto-encoder models.

This experiment involved training two auto-encoders simultaneously using the Cycle Consistency loss function outlined in (6) and (7). The two models, denoted as g and f, were trained to process an original image and an unpaired permuted image, respectively. The model g was given the original image as input, and its output was passed to f. The resulting output of f was then compared to the original image using the mean squared error (MSE) loss, denoted as L_o in Equation (6). Conversely, the model f was given the permuted image as input, and its output was passed to g. The output of g was compared to the permuted image using the MSE loss, denoted as L_p in Equation (7).

$$L_o = E_{x \sim X}(\|f(g(x)) - x\|_2^2) \tag{6}$$

$$L_p = E_{y \sim Y}(\|g(f(y)) - y\|_2^2) \tag{7}$$

The total loss function, denoted as L in Equation (8), was defined as the sum of the two MSE losses. This total loss function was then used for back-propagation to update both models.

$$L = L_o + L_p \tag{8}$$

Once both models were trained, permuted images were passed through f to obtain the unpermuted images.

In our experiments, we trained models to unpermute images from both MNIST and FashionMNIST data sets. As shown in Figure 11, the models were able to successfully match the distributions of the permuted and the original images which resulted in unpermuting of the images to their corresponding correct classes. However, for FashionMNIST images, the results as shown in Figure 12, suggest that the models learned to map the distribution of permuted images to original images but it struggled to match them to their actual classes. The reason why certain unpermuted images in a given class may not exactly match their original counterparts is because, when it comes to the FashionMNIST data set, the models have not fully learned the correct mapping. In contrast, for the MNIST data set, the models have learned the correct mapping to a much greater extent. Therefore, in the case of FashionMNIST, many images are either being misclassified into the wrong class or are being transformed into slightly different images within the same class, rather than being an exact replica of the original image.

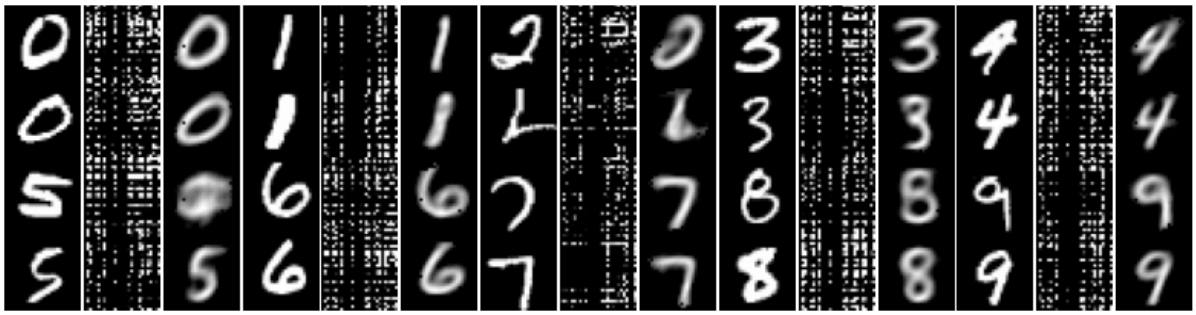


Figure 11: unpaired MNIST data unpermuted using 2 auto-encoders

To evaluate the accuracy of our models, we employed a CNN model with batch normalization layers that we developed during the supervised stage of our research. Using this model, we assessed the number of permuted images that were correctly unpermuted to their respective class. For the MNIST unpermuted images, the resulting accuracy was 69.89%. This metric provided us with valuable insights into the performance of our models and allowed us to further optimize our approach to improve accuracy.

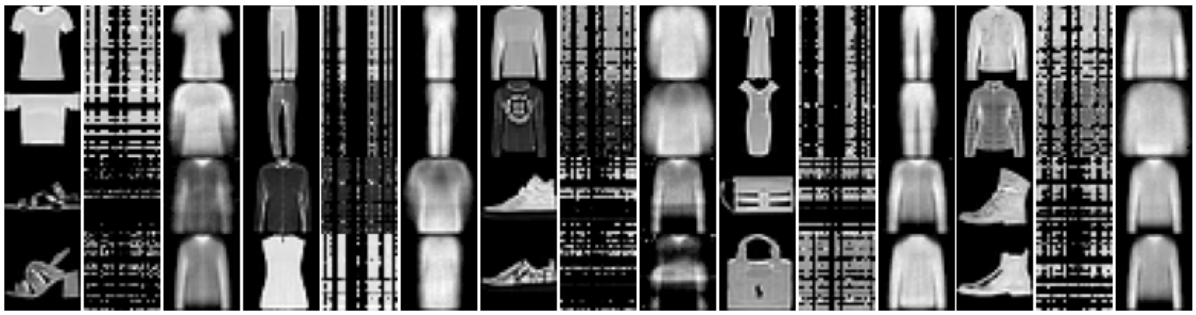


Figure 12: unpaired FashionMNIST data unpermuted using 2 auto-encoders

5 Discussion

In our study, we were able to match the distributions of permuted images with their original counterparts using our models, which effectively addressed the issue of unpermuting images in a supervised setting. Through experimenting with different data sets we trained single-layer models and auto-encoders and found that increasing the number of training epochs resulted in better outcomes. However, we observed that our models trained on the MNIST data set had limited success in unpermuting images beyond MNIST. This was due to the sparse nature of MNIST images, where only a few pixels have different values, making it difficult for the network to unpermute the entire image. In contrast, our models trained on the CIFAR10 data set were able to unpermute any image.

We also explored the unsupervised setting and were able to unpermute MNIST images. However, we noticed that our models were not able to learn without the addition of a dropout layer. When we attempted to apply the same model to CIFAR10 images, the results were not satisfactory. We propose that the use of models with more layers or convolutional neural networks could lead to better performance in unpermuting CIFAR-10 images. Further details on our experiments, including visualizations, can be found in the appendix section of this paper.

Acknowledgement: This work was supervised by Professor *Magda Gregorova*, from *Faculty of Computer Science and Business Information Systems,Technische Hochschule Würzburg-Schweinfurt. This paper have benefited by the English and grammatical assistance offered by ChatGPT,OpenAI*

References

[1] Li Deng. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.

[2] I. J. Good. "Rational Decisions". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 14.1 (1952), pp. 107–114.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press, 2016.

[4] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015.

[5] Alex Krizhevsky and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. 0. Toronto, Ontario: University of Toronto, 2009.

[6] F. Rosenblatt. *The Perceptron, a Perceiving and Recognizing Automaton Project Para*. Report: Cornell Aeronautical Laboratory. Cornell Aeronautical Laboratory, 1957.

[7] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958.

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. Aug. 28, 2017.

[9] Jun-Yan Zhu et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2223–2232.

Muhammad Zakriya Shah Sarwar
Technische Hochschule Würzburg-Schweinfurt
FIW
Sanderheinrichsleitenweg 20 97074 Würzburg
Germany
E-mail: *muhammadzakriya.shahsarwar@study.thws.de*

Nehmiya Shikur
Technische Hochschule Würzburg-Schweinfurt
FIW
Sanderheinrichsleitenweg 20 97074 Würzburg
Germany
E-mail: *nehmiya.shikur@study.thws.de*

A Appendix

In this section, we evaluated the generalization performance of our SLP models trained to unpermute images from various data sets. We tested the models on data sets they were not trained on to assess their ability to perform on unseen data. The objective was to verify if our models could handle new data sets beyond their original training data.

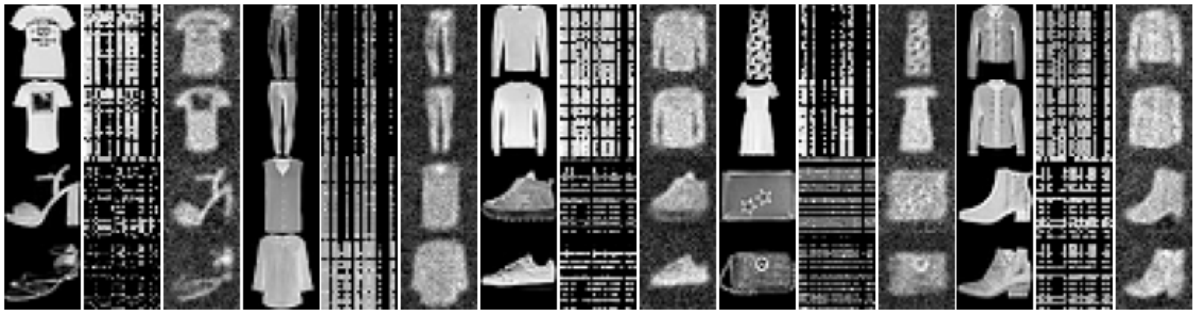


Figure 13: The result of applying an SLP model trained on the MNIST data set to images from the FashionMNIST data set. The output displays the model's performance in unpermuting the FashionMNIST images.

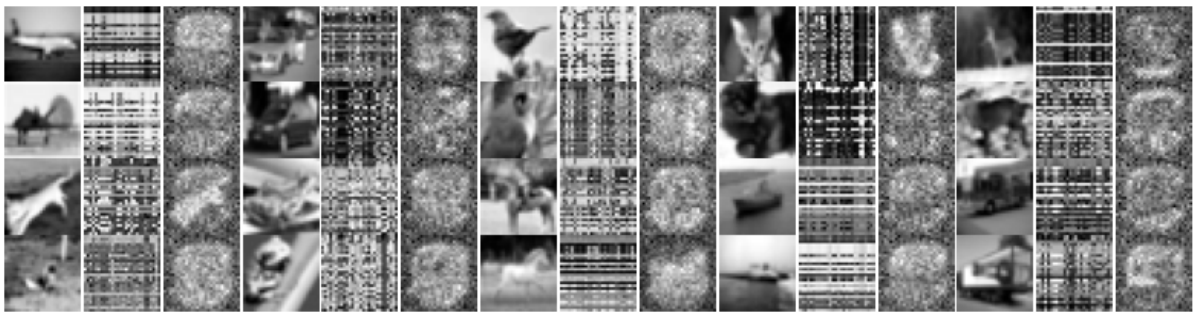


Figure 14: The SLP model trained on MNIST dataset was tested on images from CIFAR-10 dataset, but failed to unpermute them, indicating that the model cannot generalize well to other types of images.

Figure 15: The generated output from the SLP model, trained on the FashionMNIST data set, for the images from the MNIST data set demonstrates the generalization ability of the model. This indicates that the model is capable of unpermuting images from data sets that it has not been trained on, which further validates its effectiveness.

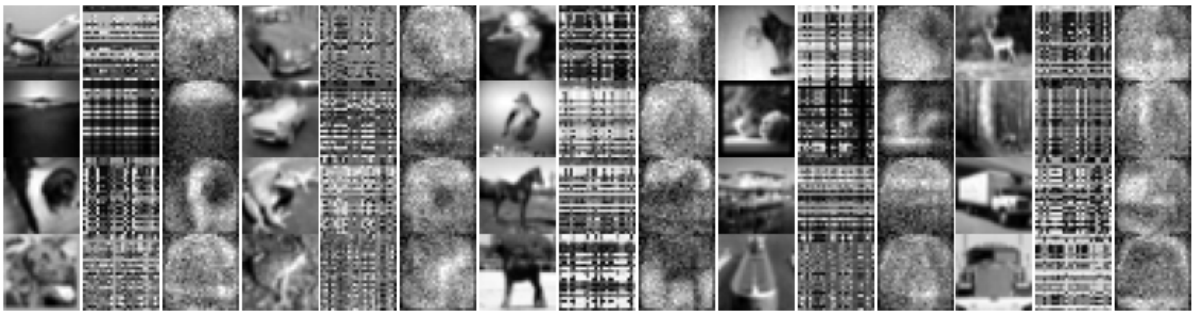


Figure 16: The SLP model trained on FashionMNIST data set was also tested on CIFAR-10 data set and it showed better results compared to the SLP model trained on the MNIST data set when unpermuting the images.

Figure 17: The SLP model trained on CIFAR-10 data set demonstrated a high level of proficiency in unpermuting images from the MNIST data set, achieving good results.

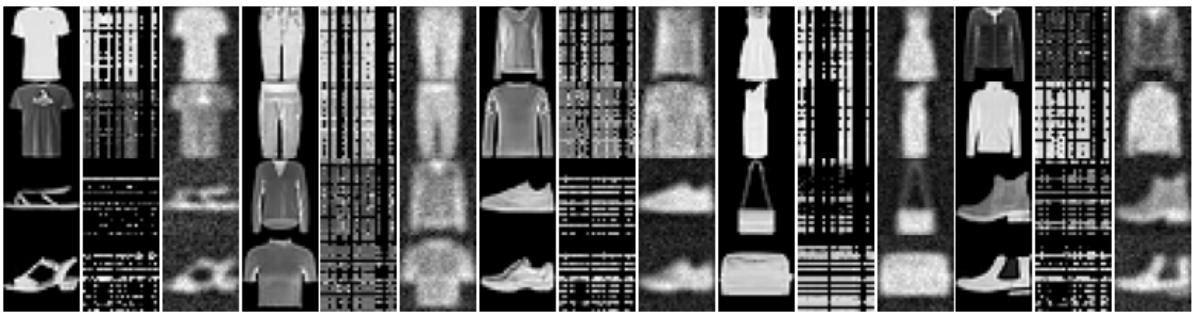


Figure 18: The output of the SLP model trained on the CIFAR-10 data set for images from FashionMNIST data set was very encouraging, as it was able to successfully unpermute the images with high accuracy. This indicates that the model has a good generalization ability and can be applied to various data sets.

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

GEditor

Matei Cristian Steavu

Abstract

GEditor is an app made with the purpose to improve the efficiency of a 3D printer. The app does this by modifying the GCode of an object generated by a slicer such as Ultimaker Cura or Prusaslicer with encoded functions such as speeding up certain layers of filament. The programming language is C# and the UI is made by using Windows Forms in Visual Studio.

1 Introduction

The purpose of this application is to make a 3D printer more efficient and to reduce the time it takes it to print a 3D object. This is achieved by editing the generated GCode, the programming language that controls a 3D printer. The app can be used by anything that uses GCode, yet, it is mainly made for 3D printers. The theme of the application is to improve a process that usually takes a lot of time.

I first had this ideea when I got acces to a 3D printer. There were layers that I could print faster than others without a cost in quality, yet, since the object also had layers that had to be printed at a slower speed, that meant that I had to print the entire object with slow speed, which significantly raised the print time of an object. I first tried to find something embedded into the slicers, the applications that generate the GCode for a 3D object, yet I did not find anything. This is when I started working on my application. I chose to use Visual Studio and Windows Forms as an interface because I already had prior experience in using both technologies together.

What is Visual Studio and what is Windows Forms? Visual Studio is a flexible programming environment that can be used for countless projects and which supports most programming languages. In Visual Studio, a user can create anything from console applications all the way to video games and websites. Windows Forms is a type of project in Visual Studio that allows the user to create every type of application for any type of Windows. Additionally, it already comes with the usual buttons associated with a Windows application, namely the 3 buttons from the upper right corner. Another advantage is the fact that it already has preintegrated elements such as buttons or text boxes. Modifying those is very simple with certain parameters.

2 Compatibility and GCode

First, one should understand what exactly is a slicer. A slicer is a software application in which a user can open a 3D object either as an .stl file, .obj file or others. After the user opens the object, the slicer has more settings, namely, for the 3D printer, filament (material), and settings for the object itself (Fig. 1). After the user selected all the settings that he wants, the slicer generates the GCode that will control the 3D printer is such a way that it will print the opened object.

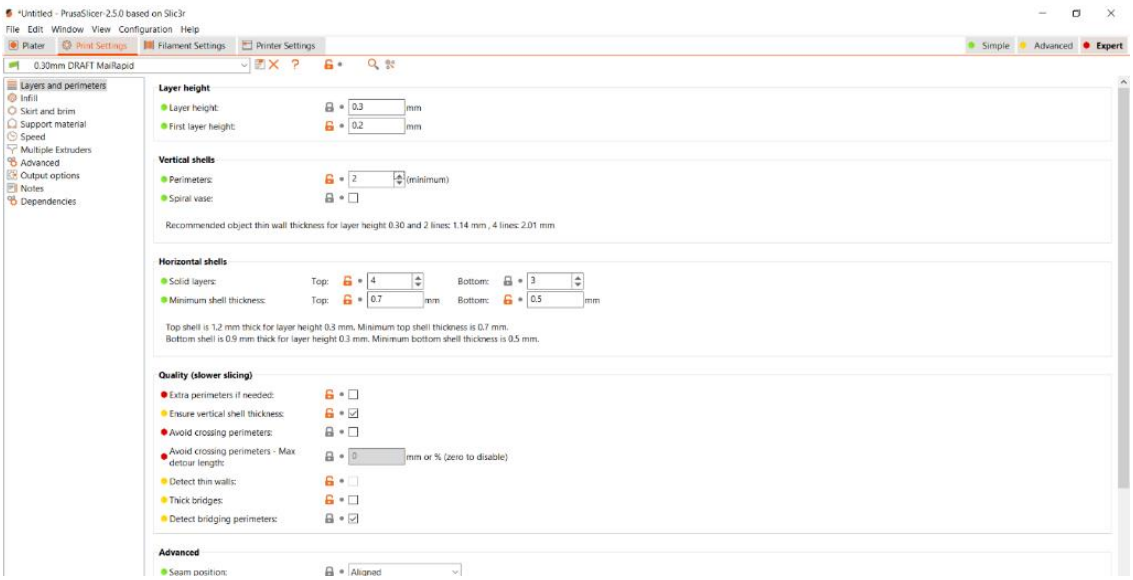


Fig. 1 PrusaSlicer Settings

How does this generation work? An object has a certain height, the slicer takes this height and splits it up in multiple layers of filament with a certain layer height. Afterwards, the slicer takes each layer seperately, creates the GCode of the perimeter for this layer. Then, it creates the GCode for supports (Fig. 2) at that certain layer height, so that plastic is not printed mid-air at future layers, and then it generates the GCode for the filling pattern of the perimeter. For example, if someone wants to print a cube, the slicer will split that cube into a lot of layers and then it will take the first layer, generate a GCode command for one of its corners, then the second one, then the third one, the forth one and then back to the first.Then it generates the GCode for the infill pattern and then it will give a G1 command to raise the “Z” axis so that another layer can start.

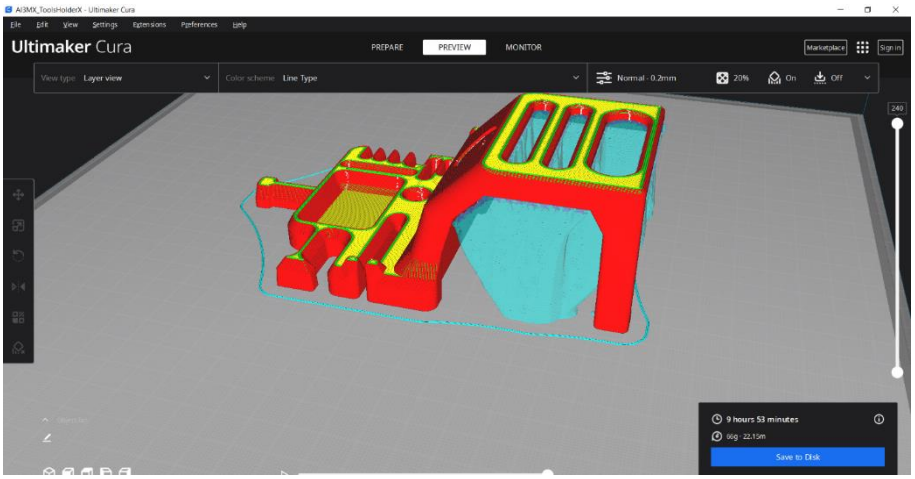


Fig. 2. Object with supports

Because of the fact that every slicer generates GCode a bit differently I have to know how each of them works. Possible differences might be utlizing different coordinate systems. One might use the local coordinate system and another might use the global coordinate system or one might use extra parameters for their GCode commands. The difference between those 2 coordinate systems beeing the fact that that the local coordinate system takes the coordinates of the next command from where the print head is at that particular moment and the global coordinate system takes the coordinates of the next commands from where the center of the print bed is. My app is made as a post processing application after generating the GCode with Ultimaker Cura or PrusaSlicer. Since PrusaSlicer generates the GCode more conveniently for editing it, all the code segments will be explained for PrusaSlicer.

In order to understand what the application really does it is also necessary to understand some GCode commands. A first command, and certainly the most important one is:

```
G1 X10 Y10 Z10.32 E12.6 F1200
```

The G1 command represents a movement in straight line towards the coordinates X, Y and Z that are found as parameters. The coordinates are given in milimeters. The “E” parameter is some kind of 4th axis and represents the milimeters of filament that the 3D printer extrudes while performing the command. The “F” parameter is the one that represents the speed at which the command is made and is measured in milimeters/minute. For a simple movement, without extruding filament one should use the command G0 with the same X, Y, Z and F parameters.

For both of these commands parameters can be omitted, needing at least 1 parameter. This allows us to just set an “F” parameter for the speed of certain commands or the parameter “Z” for a certain layer. This also reduces the file size of the generated GCode.

Other important commands are those for controlling the temperature of the print bed or the print head and those that wait for those to heat up. In table 1 [1] there are commands together with their explanation.

`M140 S60`	This command sets the temperature of the print bed at 60 degrees Celsius.
`M104 S205`	This command sets the temperature of the print head at 205 degrees Celsius.

M190	This command pauses the execution of the next commands until the print bed heats up at the set temperature.
M109	This command pauses the execution of the next commands until the print head heats up at the set temperature.

Table 1. - GCode commands

3 The design and interface of the application

The application is made in Visual Studio using the Windows Forms project template. I used those because I am already familiar with them. In the next figure one can see how the application looks like and what its functions are. It is important to know that, if the user checks more than one fuction at the same time, they will be done from top to bottom.

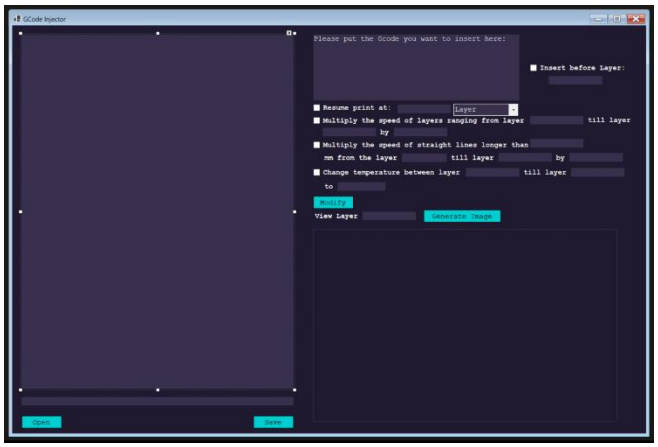


Fig. 3. Application interface

I have chosen those functions through own experience and by thinking what else might be useful for another 3D printer type, such as a direct drive one. For example, the function for modifying the temperature is useful for flexible filaments, those are not compatible with all kinds of 3D printers, while an owner can print flexible filaments on a bowden printer it is not the easiest. Whereas, the function for inserting additional GCode can be used on most 3D printers.

4 How it works

When it comes to how it works, I will describe some important code sequences, written in C# [2, 3] and I will motivate why I implemented those.

Because GCode is a programming language without repetitions, it is necessary to read all commands. Adding this over the fact that a GCode file can have over 12 MB the use of memory by the program has to be minimised. Thus, it does not work to create a copy of the file and just ommit or add GCode commands, the program has to directly work on the text box that contains the opened file. This also allows us to save the modifications it has made.

The first function is the one that inserts GCode before a certain layer. This function is useful for 3D printers that print with only one filament, because the user can plan pauses for filament change, even make the 3D printer unload the filament throught GCode. This would usually be done manually. Lower is the code sequence used for inserting the text from the text box named “Whattoinsert” before a layer with the number lwi. Using the StreamReader class the code can read every line from the file and search for the G1 commands with the “Z” parameter signifying that it is at another layer. Because of the fact that every function needs to read all the lines, from now on, I will only show the code from the “while” statement block.

```
StreamReader sr = new StreamReader(FileName.Text);
String line;
int index = 0;
curlayer = 0;
while ((line = sr.ReadLine()) != null)
{
       if (line.Contains("G1") && line.Contains("Z"))
       {
           curlayer++;
       }
       if (curlayer == lwi)
       {
           Display.Text = Display.Text.Insert(index, Whattoinsert.Text + "\n");
           r.ReadToEnd();
       }
}
```

Another important utility that the application has is that the user can increase or decrease the speed of layers if he wants to, something that is not possible or hard, if an user only uses a slicer. Since speed plays an important role in print times, modifying it efficiently can reduce print time a lot. The code sequence is the next one. In this sequence the code extracts the speed of a certain comand in the variable named “speed”. If it is between the coresponding layers, and, if it is not too high, in my case 5000 mm/min, the speed is raised or decreased by multiplying it with the variable “sm”. Afterwards, the value of the “F” parameter is deleted in that command and then the calculated one is inserted.

```
if (line.Contains("F") && line.Contains("G1") && curlayer >= ssl && curlayer <= sel)
{
       aux = line.Substring(line.IndexOf("F") + 1);
       if (aux.IndexOf(" ") != -1)
              aux = aux.Remove(aux.IndexOf(" "));
       speed = Convert.ToInt32(aux);
      if (speed <= 5000)
       {
              speed = Convert.ToInt32(speed * sm);
              Display.Text = Display.Text.Remove(index, line.Length);
              line = line.Remove(line.IndexOf("F") + 1, aux.Length);
              line = line.Insert(line.IndexOf("F") + 1, speed.ToString());
              Display.Text = Display.Text.Insert(index, line);
       }
}
if (curlayer > sel)
       sr.ReadToEnd();
```

The function that modifies the temperature between certain layers is useful for 3D printers that can print with flexible filament or with a filament that modifies its volume depending on the printing

temperature such as lightweight PLA. How it works? It saves the command in which the temperature of the print head is set; when it gets to the layers that need the temperature changed, it inserts a new command and then reinserts the saved one after the layers are gone. Unfortunately, this means that all the "E" parameters for the following commands have to be recalculated, because by printing the flexible filament with a higher temperature the volume of the filament changes due to a controlled process named foaming, where gas bubbles from inside the filament or the filament itself foams up in a controlled manner. This also allows the flexible filament to be softer or harder. It is important to mention that not all filaments experience foaming, in fact, most of them do not.

```
if (saved == 0 && line.Contains("M104 S"))
{
	save = line;
	saved = 1;
}
if (line.Contains("G1") && line.Contains("Z"))
	curlayer++;
if (inserted == 0 && curlayer == tsl)
{
	Display.Text = Display.Text.Insert(index, "M104 S" + tv.ToString() + " ;" +
"\n");
	inserted = 1;
	index = index + ("M104 S" + tv.ToString() + " ;" + "\n").Length;
}
if (inserted == 1 && curlayer == tel)
{
	Display.Text = Display.Text.Insert(index, save + "\n");
	index = index + (save + "\n").Length;
	inserted = 2;
}
index = index + line.Length + 1;
```

The last utility that I will be presenting is the function that allows the GCode to start from a certain layer or a certain height in milimeters. Whether the 3D printer has an error or the print object fails after 12 hours or the filament runs out overnight, nobody wants to throw away an object that is half printed or maybe even 90% printed and have to wait again for that object to print. Usually, if something like this happens, those who know how to use GCode can save the object by printing what is missing and then sticking the two parts together. But what will those, that do not know how to utilize GCode or those who do not want to glue the two parts together, do? This is what my function is made for. The user can give it a certain layer number or a certain height in milimeters and the code will cut the GCode so that everything is printed over the object. The user can find the height at which the print has failed by placing the print head over the failed object and then giving my application the height on the Z axis. Of course, for this to work, the print object has to still be stuck to the print bed after the printing has failed.

```
if (line.Contains("G1") && line.Contains("Z"))
{
	curlayer++;
	aux = line.Substring(line.IndexOf("Z") + 1);
	aux = aux.Remove(aux.IndexOf(" "));
	if (aux[0] == '.')
		aux = "0" + aux;
	prev = actual;
	previndex = newindex;
```

```
        newindex = index;
        actual = Convert.ToDouble(aux);
}
if (type == "Layer" && curlayer == Math.Floor(ntr))
{
Display.Text = "G0 Z" + (actual + 10).ToString() + " F750\n" +
Display.Text.Substring(index);
        break;
}
else
if (type == "mm")
        if (Math.Abs(ntr - prev) < Math.Abs(ntr - actual) && prev <= ntr && actual
        >= ntr)
        {
                Display.Text = Display.Text.Substring(previndex);
                Display.Text = Display.Text.Insert(0, ("G0 Z" + (ntr + 10).ToString()
        + " F750\n"));
                break;
}
else
        if (Math.Abs(ntr - prev) >= Math.Abs(ntr - actual) && prev <= ntr && actual
>= ntr)
        {
                Display.Text = Display.Text.Substring(newindex);
                Display.Text = Display.Text.Insert(0, ("G0 Z" + (ntr + 10).ToString()
        + " F750\n"));
              break;
        }
index = index + line.Length + 1;
```

The next function of the app is too long for me to be able to share the code sequence, but I will explain how it works. This function increases or decreases the speed of straight lines longer than a certain number of milimeters. I implemented this function because long, straight lines can be printed faster than multiple short ones. This is because of the fact that the print head does not have to counter the inertia from changing the direction in which it is going. The function works as follows: the code extracts the X and Y coordinates from a command, using these coordinates and those from the previous command the app calculates the distance using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

Afterwards, the code checks whether the lenght “d” is greater than or equal to the minimum length that was read. If it is, a comand to speed up or slow down the next command is inserted. Besides this, the code keeps a variable for the current speed so as not to insert more commands than needed.

The other function is the function to see the movement of the print head when it prints a certain layer (Fig. 4). This function works the same as the last one when it comes to extracting coordinates. Then, the code just draws the lines with the extracted coordinates, by creating “PointF” variables and them using the “DrawLine” function from the Graphics class.

```
if (line.Contains("G1") && line.Contains("Z")){
        curlayer++;
}
if(line.Contains("G1") && line.Contains("X") && line.Contains("Y") && curlayer ==
ltv){
```

```
        aux = line.Substring(line.IndexOf("X") + 1);
        if (aux.IndexOf(" ") != -1)
               aux = aux.Remove(aux.IndexOf(" "));
        prevx = x;
        x = Convert.ToDouble(aux);
        aux = line.Substring(line.IndexOf("Y") + 1);
        if (aux.IndexOf(" ") != -1)
               aux = aux.Remove(aux.IndexOf(" "));
        prevy = y;
        y = Convert.ToDouble(aux);
PointF pt1 = new PointF(Convert.ToSingle(prevx), Convert.ToSingle(prevy));
        PointF pt2 = new PointF(Convert.ToSingle(x), Convert.ToSingle(y));
        g.DrawLine(redPen, pt1, pt2);
        index = index + line.Length + 1;
}
```

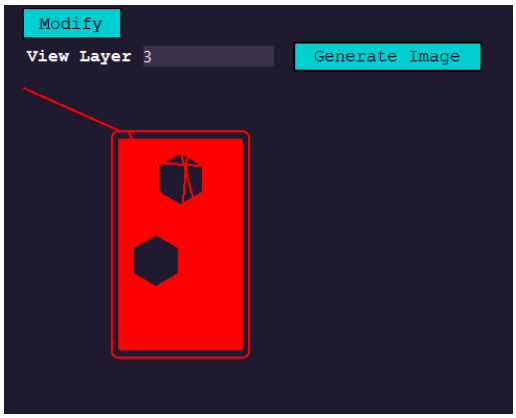


Fig. 4 Drawing a certain layer

After the user makes the modifications that he wanted to the model, all he has to do is save the contents of the text box as a GCode file, because it contains all the modified GCode.

5 Conclusions

Because this application is more of a proof that editing GCode can improve the quality of a 3D printer, and that not only the slicer that one uses or the hardware that one uses matter, the application can have a lot of improvements.

A first improvement would be making the app compatible with as many slicers as possible, so as not to be restrained to using PrusaSlicer or Ultimaker Cura. A second improvement could be trying to publish the application so that 3D printer owers can download it and give me useful feedback. Aditionally, the efficiency of this application could be improved drastically by embedding it with its own slicer so as not to have to do any recalculations.

Acknowledgement: This work was supervised by Professor Steavu Nicolae and Steavu Cristina Elena from “Doamna Stanca” National College.

References

[1] Website for GCode commands, https://marlinfw.org/docs/gcode/G000-G001.html –
[2] Website for C# help, https://learn.microsoft.com/en-us/dotnet/api/system.drawing.graphics.clear?view=windowsdesktop-7.0
[3] Saving and opening files in C#, https://www.youtube.com/@christopherlong8332/videos -

Matei-Cristian STEAVU
"Radu Negru" National College, Făgăraș
Mathematics-informatics intensive English
ROMANIA
E-mail: mateisteavu@yahoo.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Contextual based supervised text correction with index-based tokenization

Marin-Eusebiu Șerban

Abstract

The aim of this article is to propose a supervised learning algorithm, that can recognises unsuitable words (e.g., incomplete, mistyped) in a given text input and correct them based on contextual information. The algorithm takes the input text and builds the training set for the supervised learning process. We propose an indexing-based tokenisation approach, used first for the training dataset and then for the text we want to process. The predictive capabilities of our algorithm depend on the contextual relationship between the training dataset and the input text. We have also developed a general method for generating a learning dataset starting from raw text. We implemented the proposed algorithm using Python and TesorFlow and validated it. The reported results are promising.

1 Introduction

Digital searches are an essential component of today's information-driven society. With an increasing dependency on precise database search results, the demand for improved and precise text processing and correction techniques has grown significantly. The purpose of this article is to present a novel algorithmic technique for improving the efficiency and accuracy of full-text searches.

Current digital search systems generally rely on direct text input, which frequently leads to potential misunderstandings as a result of typos, fragments of words, or contextually wrong word placements. The focus on speed over precision has aggravated the situation.

In this paper we are presenting an algorithm developed for supervised learning scenarios. This algorithm takes text input, creates a training dataset, then tokenizes it using an indexing-based technique. The contextual relationship between the training dataset and the input text determines the model's predicted accuracy. The most important findings, with the main objective of boosting the accuracy of database full-text search results, indicate:

- The suggested method can process and auto-correct text inputs, improving the precision and relevance of search results.
- The study explains how to create a learning dataset from raw text.
- The suggested technique was created using Python and TensorFlow.
- Further testing supported the method's efficacy, leading to future paths for refining full-text searches.

Following an examination of the technique's basic concepts and development, we will take a look at its larger implications and prospective applications in the following sections. Furthermore,

the next sections will provide a comprehensive examination of its effectiveness and potential areas for improvement. In Section 2 will be presented the proposed architecture, discussing its benefits, limitations, and implementation methods. The final section will present our conclusions and potential future developments.

2 Proposed Architecture

To accomplish the objectives mentioned in Section 1, a model focusing on text processing needs to be developed before it can be used in database searches. The current state of search solutions and the resulting limitations are the essence of the problem. Various technologies, such as ElasticSearch, Algolia, and Meilisearch, have emerged to meet these demands. However, an in-depth analysis reveals that each system has significant trade-offs:

- ElasticSearch is well-known, it relies on a limited free trial framework and inevitable experiences charges after the trial, making it unsuitable for long-term, unlimited usage.
- Algolia represent a free alternative to ElasticSearch, bases its zero-cost strategy on a limited number of requests, after which costs are charged.
- Meilisearch, even the most open-source solution, has its own set of limitations. As it is still in the early stages of development, its capacity for complete business activities is unreliable. Furthermore, because it is executing on the same system as the primary project, it consumes a substantial amount of resources. This higher spending could put a pressure on startups or businesses with limited resources.

After identifying these limitations, our recommended approach attempts to cut costs and improve operations. In contrast to external API-based solutions like Algolia and ElasticSearch, or resource-intensive local alternatives like MeiliSearch, our method would work within the server architecture. This integrated process not only improves speed and responsiveness, but it also guarantees a level of consistency and performance scalability that external or concurrent running solutions cannot.

The main objective of the paper is on the creation and implementation of such a model. For our model to be truly effective, it has to:

- Perform Corrections: Before any data is sent to the search database, the model must autocorrect textual inputs to avoid spelling errors during searches. This phase is critical for ensuring that search results are accurate and relevant to the user's query.
- Tokenization: To allow the model to execute rapid and accurate query refactoring, the input data needs to be tokenized. This method employs a tokenization mechanism that separates the text into manageable parts and recognizes contextual relationships, leading to a more natural search experience.
- Contextual Use: The model is intended for use in systems where precise, fast, and resource-efficient search capabilities are important. This methodology can adapt and deliver whether it's an e-commerce platform where customers need to find things quickly or a research database where precision is essential.

2.1 Tokenization

Tokenization [4] is a fundamental process in Natural Language Processing (NLP) and is essential for this model. Tokenization, also is the process of turning continuous text into separate tokens, generally words or sub-words. While there are several approaches for tokenizing data, this method optimizes for both efficiency and relevancy, particularly in search contexts.

Raw Data Preparation: The dataset used is built on raw text. This data typically remains stored as a '.txt' file in the format showed in Figure 1.

```
Universitatea Lucian Blaga din Sibiu Romania Facultatea de Medicină
ULBS Romania Facultatea de Medicină
ULB Sibiu Romania Facultatea de Medicină
ULB Sibiu Romania Medicină
ULBS Sibiu Romania Facultatea de Medicină
ULBS Sibiu Romania Medicină
Universitatea Lucian Blaga din Sibiu Romania Medicină
Universitatea ULBS Sibiu Romania Facultatea de Medicină
Universitatea Lucian Blaga Sibiu Medicina
Universitatea Lucian Blaga Sibiu Facultatea de Medicină
Universitatea din Sibiu Lucian Blaga Facultatea de Medicină
Sibiu ULBS Facultatea de Medicină
```

Fig. 1: Raw text

Latinization: As an initial phase, the tokenizer reads each line of text and latinizes it, replacing diacritics with equivalent English letters, shown in Figure 2. This procedure increases the data's generality, making it less resistant to popular tokenization techniques.

```
def read_file(self):
    with open(self.file, 'r') as f:
        for line in f:
            self.phrases.append(self.latinize(line.lower().replace('\n', '')))
```

Figure 2: Read data method

Punctuation Handling: Punctuation has various implications in various contexts. This is addressed differently in this scenario, it is only eliminated if it follows words and not if it stands alone as a single token. The 'string.punctuation' method, which recognises common punctuation, is used by default. Figure 3 illustrates the process of transformation.

```
1  def remove_punctuation(self):
2      self.phrases = [''.join(c for c in phrase if c not in self.punctuation or c == '-')
   for phrase in self.phrases]
```

Figure 3: Remove punctuation from input data

Generating Input-Output Pairs: The essential tokenization procedure occurs here. Sentences are separated into individual words using the function named 'word_tokenize' from the popular NLTK package. Once recognized, these words are used to generate expected 'sub-words' that are paired with the original words to generate input-output pairs. One of the tokenization's distinguishing features is how it handle words in relation to their position in the sentence and the presence of 'sub-words'. To create a briefer expression, combine prior words and 'sub-words' of both the target word and the term before it. This reduced statement is evaluated by the algorithm, which ensures that all words up to (and including) the target term have lengths greater than one, while ensuring all other words have a minimum length of two. If these criteria are not met, the loop moves on to the next possibility. Valid 'reduced sentences' are added to the 'self.input_data' list, while phrases corresponding to the target word's prefix length are added to the 'self.output_data' list. This complex approach, presented in Figure 4, contributes to the versatility and precision of the model's search capability.

```
def tokenize(self):
    for sentence in self.phrases:
        words = word_tokenize(sentence)
        self.all_words.extend(words)
        for target_word_index, target_word in enumerate(words) :
            for i in range(len(target_word), 0, -1):
                self.all_words.extend([target_word [:i]] if target_word[:i] not in self.all_words else [])
                self.splitted.append(target_word [:i]) if target_word[:1] not in self.splitted else None if i == len(target_word) else None
                for j in range(1, len(words (target_word_index - 1))+1) if target_word_index > 0 else [0]:
                    reduced sentence = (
                        " ".join(words[:target_word index - 1]) + " " + words[target_word_index - 1][:j] + " " + target_word[:i]
                        if target_word_index > 0 else target_word[:1])

                    if any(len(word) < 2 for word in reduced_sentence.split()) and len(reduced_sentence.split()) >= 2 and all(len(word) > 1 for word in words[:target_word_index + 1]):
                        continue

                    if not (target_word_index == len(words) - 1 and i == len(target_word)):
                        self.input_data.append(reduced_sentence.strip())
                        if i == len(target_word) :
                            output_sentence = " ".join(words[:(target_word_index + 2)])
                        else:
                            output_sentence = " "•join(words[:(target_word_index+1)])
                        self.output_data.append(output_sentence)
```

Figure 4: Tokenize method

Although systems such as SpaCy, Stanford NLP, and TextBlob provide tokenization capabilities, this approach stands distinguished due to its nested loop structure, which is optimised for search relevance. Unlike generic tokenization, which focuses just on text segmentation, this approach prioritises tokens that can increase search accuracy and construct input-output pairs that reflect the phrase's context. The only downside to this method is the tokenizer's tendency to generate an excessive amount of data for a modest training set, attempting to capture every possible combination, whether of entire words or fragments.

After feeding the dataset through the tokenizer, the resulting dataset will appear like Figure 5.

Input sentence	Output sentence
universitatea	universitatea lucian
universitate	universitatea
universitat	universitatea
universita	universitatea
universit	universitatea
...	...
eam bucur ac	eam bucuresti academia
eam bucure ac	eam bucuresti academia
eam bucures ac	eam bucuresti academia
eam bucurest ac	eam bucuresti academia
eam bucuresti ac	eam bucuresti academia

Figure 5: The input-output pairs after tokenization

Vocabulary Creation: After the tokenization process is finished, a list with the different terms recognised through the entire process is created. These words are indexed to provide a bidirectional mapping, allowing for rapid data processing during model training. Figure 6 illustrates an example of this strategy.

```
def words2idx(self):
    self.vocab = sorted(set(self.all_words))
    self.vocab_size = len(self.vocab)
    self.word2idx = {word: i+1 for i, word in enumerate(self.vocab)}

def idx2words(self):
    self.idx2word = {i+1: word for i, word in enumerate(self.vocab)}

def convert2idx(self):
    for sentence in self.input_data:
        self.X.append([self.word2idx[word] for word in sentence.split()])
    for sentence in self.output_data:
        self.Y.append([self.word2idx[word] for word in sentence.split()])
```

Figure 6: Index based conversion

Post-tokenization transformation: When the tokenization procedures are completed, our dataset experiences a significant transformation. The data is transformed from its starting state in Figure 1 to its final state in Figure 7, where padding is used to optimise model feeding.

```
1  print("X_train: ", X_train[:1])
2  print("Y_train: ", Y_train[:1])

X_train:  [[94 49  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]]
Y_train:  [[94 55  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0]]
```

Figure 7: Index based transformation

2.2 Model Architecture

The architecture is made up of a pair of bidirectional layers, which are enhanced by a dropout layer intended to reduce the chance of overfitting. Long Short-Term Memory (LSTM) [3] cells have been involved into the first bidirectional layer. These cells give the model the ability to understand sequences both forward and backward.

The second layer then includes a Gated Recurrent Unit (GRU) [2]. The GRU additionally improves the model's overall efficiency, but it also makes it especially effective at dealing with medium-length textual inputs.

The model's last component is implemented as a TimeDistributed layer. This layer wraps around the previous one, ensuring that the dense (fully-connected) layer is consistently and independently applied to each time slice of the sequence. This individual setup enables the model to predict individual words inside a given time step while preserving consistent weight distribution across predictions. It essentially allows the model to make discrete predictions for each word in the input sequence rather than producing a single forecast for the entire sequence of words.

The model was implemented using TensorFlow.

Figure 8 illustrates the full model's architecture.

Layer (type)	Output Shape	Param #
input_4 (InputLayer)	[(None, None)]	0
embedding_3 (Embedding)	(None, None, 256)	53504
bidirectional_6 (Bidirectional)	(None, None, 1024)	3149824
dropout_3 (Dropout)	(None, None, 1024)	0
bidirectional_7 (Bidirectional)	(None, None, 1024)	4724736
dense_6 (Dense)	(None, None, 209)	214225
time_distributed_3 (TimeDistributed)	(None, None, 209)	43890

Figure 8: Model structure

2.3 Model Fitting

Throughout the training process, the model performed well. It achieved a 99% basic training and validation accuracy in just three epochs, with a loss value of 0.4%. These metrics indicate the model's high efficacy and predictability in performing the task for which it was intended.

Training and validation curves, are presented in Figure 9 and Figure 10.

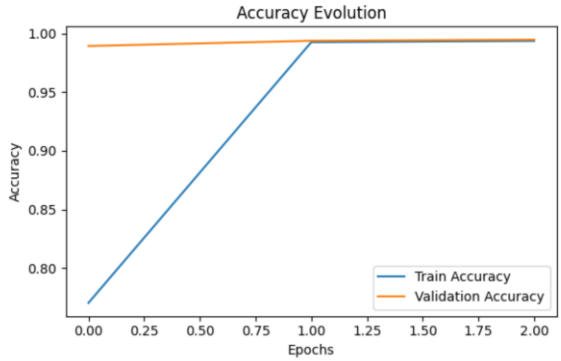


Figure 9: Evolution of Training and validation accuracy

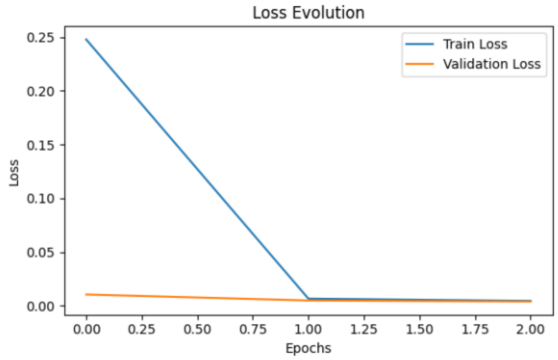


Figure 10: Evolution of training and validation Loss

2.4 Model prediction

While the model has a high level of precision, it is critical to note that it does not guarantee perfect predictions in all situations. Its strength, like many other models developed using machine learning, resides in detecting patterns in the data it's been trained on. Its basic feature relies around making predictions that are contextually relevant.

The model's primary goal is to keep contextual relevance. In specific cases, the model displays remarkable accuracy in evaluating and decoding input. In contrast, in different contexts, its major goal is to guarantee that the predictions are compatible with the current context, rather than being exact. As an example:

- Figure 11 illustrates an instance where the model makes the most accurate prediction given the current context.
- Figure 12 illustrates how the model adapts and refactors its prediction based on the given context. In this test, I referenced "facultatea de stiinte medicale", but since this faculty does not exist, the model attempts to preserve the context's integrity..
- Figure 13 demonstrates a scenario in which the model responses to a misspelt word, demonstrating its capacity to handle inadequate inputs.

```
Input phrase: sib ul fac de sti so
1/1 [==============================] - 0s 31ms/step
Output phrase: sibiu ulbs facultatea de stiinte socio-umane
```

Figure 11: Best prediction

```
Input phrase: sib ul fac de sti medi
1/1 [==============================] - 0s 40ms/step
Output phrase: sibiu ulbs facultatea de stiinte socio-umane
```

Figure 12: Prediction made on context

```
Input phrase: universita lucian blaga facultate de stint
1/1 [==============================] - 0s 36ms/step
Output phrase: universitatea lucian blaga facultatea de stiinte
```

Figure 13: Phrase correction with misspelled word

2.5 Spell Correction using Levenshtein Distance

One of the most important ability of the presented model consists in repairing misspelt words. This capability is reached using the Levenshtein Distance algorithm.

The Levenshtein Distance [5] computes the number of shifts (insertions, deletions, or substitutions) required when converting one string to another. Using this technique, the algorithm can quickly scan the model's existing vocabulary and identify the term that is most similar to the misspelt input, requiring the fewest modifications.

In practice, if a user submits an expression that does not directly match one of the model's entries, the Levenshtein Distance will determine the closest, most likely alternative. This ensures that, even in the presence of typographical errors or uncommon spellings, the model can deliver a relevant and contextually suitable prediction, boosting the user experience even further.

3 Conclusions and Future Developments

In our research, the tokenization process plays an important role in the model's training. Currently, despite the model's well performance in recognizing patterns, the tokenization approach creates a moderate dataset into a significantly larger training set by attempting to generate pairs between words in all their possible forms. While this methodology appears effective now, there are potential areas for improvement to explore in the future, such as:

- Tokenization Efficiency: While the current tokenization procedure is effective, it can be improved. Decreased data redundancy and improved tokenization techniques will allow the model to handle and analyses larger datasets without impacting prediction quality.
- Self-Correction: One of the characteristics of a mature model is its capacity to recognize and correct errors on its own. The goal for future iterations is to provide the model with a greater sense of self-awareness allowing it to learn and adjust from its mistakes without the need for external intervention.
- Enhanced Prediction: The predictive capabilities of the model can be expanded to anticipate the next words, particularly when the terminal word in an input sequence is completely provided. This feature has the potential to greatly improve user experience by recommending contextually relevant follow-ups.

Acknowledgement: This work was supervised by Professor Dr. Dana Simian, from Faculty of Science, University "Lucian Blaga" of Sibiu.

References

[1] K. Chakradhar,K. Sai Kiran,K.Shanmukh, K. Sharath Kumar,K. Dinesh Sagar, *Next Word Prediction Using Deep Learning*, 2022

[2] Cho, K., et al., *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. arXiv preprint arXiv:1406.1078, 2014

[3] Hochreiter, S. and J. Schmidhuber, *Long short-term memory. Neural computation*, 1997. 9(8): p. 1735-1780

[4] Lukas Michelbacher, *Multi-Word Tokenization for Natural Language Processing* , 2013

[5] Rishin Haldar and Debajyoti Mukhopadhyay, *Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach*, 2011

Marin-Eusebiu SERBAN
Lucian Blaga University of Sibiu, Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: eusebiu.serban@ulbsibiu.ro

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

From Turing Complete game to my own computer

Dennis-Vlăduț Teglas

Abstract

In this paper, we discuss an application called Turing Complete, which helped me discover a passion for computer systems architecture. At its core, this application is a game that teaches you how to build a computing machine starting from a single logic gate (NAND) and then program it to solve various computational problems. Furthermore, it has a tutorial in which you learn what the basic architecture of a processor looks like, as well as the different components it consists of. Once the tutorial is completed, you are free to improve the created computing machine in any way you like, even simulating an entire computer.

I present here the computer I built using this application, starting with an overview of the architecture that connects the components, followed by an explanation of what each component does, and finally, a demonstration of a program written in machine code that the computer can understand. I will also try to show a part of the tutorial I went through, along with a few challenging examples it provides. This approach will highlight the educational aspect it can offer a student who wants to understand what is behind any device they use and perhaps discover a passion in this field.

Compared with other applications where you can create and simulate a logic circuit, this one has a much friendlier and more colourful interface. It follows to be attractive for those who have no idea how a computer works, as well as those who have some ideas but don't know how to put them together. Furthermore, the application has a rather large and increasing community, with any user being supported to publish the project they have done that might be useful to someone else.

1 Introduction

In today's technology-driven world, understanding the fundamentals of computer architecture and programming has become increasingly important. Turing Complete [1] is a game that explains computer science concepts by using gamification as a unique and engaging way for students and self-learners to explore these complex topics and by providing an interactive learning environment.

Apart from presenting my own Turing Complete project, this paper aims to highlight the importance of gamification in educational applications, namely applying attractive methods in teaching various fields to students. Mainly, Turing Complete is a computer science game which offers enthusiasts of computing systems architecture an easy way to create a computer from scratch using the tutorial that guides you in what you should do to achieve an operational computer.

Once the tutorial is completed, the game offers a sandbox option where you can use any already created components or create new ones to increase the computer functionality. There are also specific

components, such as those which take keyboard inputs, allowing you to connect a keyboard to the created computer. Additionally, some components facilitate the reading from a hard disk, reading time, or even connecting to the internet.

The next section will describe a part of the Turing Complete game tutorial I went through, along with a few examples of challenges it presents. I also aim to highlight the educational aspect this game can offer a student who wants to understand what is behind any device they use and perhaps discover a passion in this field. As the main personal contribution, I present here the computer I built using this application: start with an overview of the architecture that connects the components, followed by an explanation of what each component does, and finally, present a demonstration of a program written in assembly machine code that my computer can understand.

Considering other similar applications for designing and simulating logic circuits like Logisim [2], Turing Complete and its resulting projects have a much friendlier and more colourful interface. These are significant features for any application intended not only for those without any idea about how a computer works, but also for those who have some ideas and don't know how to put them together. Furthermore, the application has a rather large community since anyone can publish any work they have done that might be useful to someone else.

2 The tutorial and its challenges

The Turing Complete game tutorial is like a teacher, guiding us throughout the concept of an operational computer, telling us the steps to follow but allowing us to experiment and learn from mistakes to understand better how things work.

2.1 First levels - where it all begins

In the first level (Fig. 1), you proceed with the easy introductory task by completing the truth table of the NAND logic gate and experimenting with what happens to the output when you change the inputs.

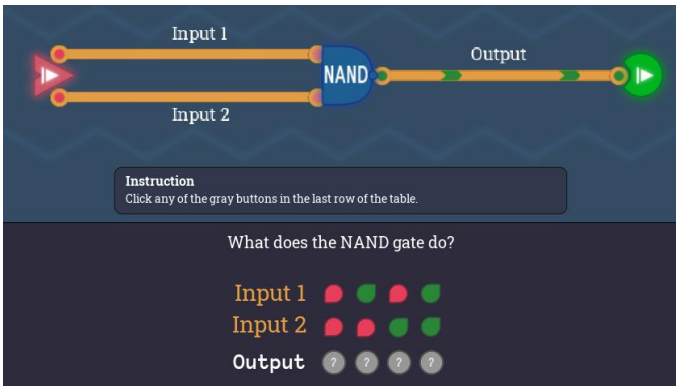


Fig. 1: Tutorial first level

The following levels are dedicated to creating all the logic gates, which are the foundations we will use further on. These are NAND, NOT, AND, NOR, OR, XOR, and XNOR gates. To guide us, the game shows the name of the gate we need to obtain and its truth table, and we need to use the logic gates built earlier to achieve it. For example, Fig. 2 represents the circuit for the eXclusive OR gate, XOR.

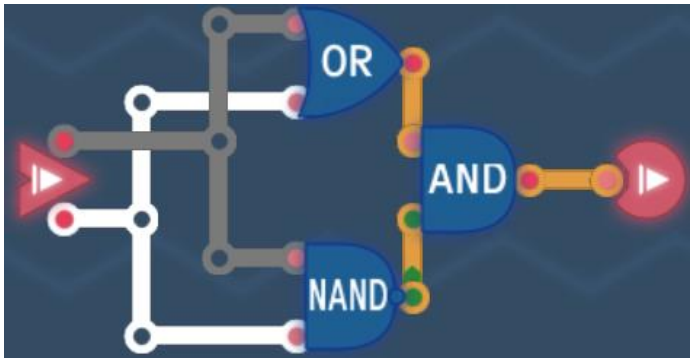


Fig. 2: XOR gate

2.2 Tutorial structure

The tutorial is then divided into two parts, the arithmetic part and the memory part. In the arithmetic module, we learn how a natural number is binary represented, how to design basic circuits like the adder or the decoder, and how to represent negative numbers. In the memory part, we learn how to save information for an extended time. Fig. 3 shows the circuit that forms the basis of the memory in the computer I made. It saves a bit when the input is 1 and always outputs the stored value.

Fig. 3: Memory binary cell

After that, the tutorial guides us in combining the created components to make a first architecture. Even if it is simple, it represents an impressive milestone for any beginner. After completing the hardware, we are ready to learn how to program the computer we made, to solve some tasks that challenge our minds. Once we understand both the hardware and software aspects, the game offers us greater freedom from now on, allowing us to choose the names of commands and what they do, and even modify the architecture in any way we want. The tutorial becomes more than a guide to improving our computer structure and functionality.

2.3 Some of the most challenging tutorial levels and how to solve them

This section presents some tutorial levels that surprised me with their uniqueness compared to other learning levels. They are relevant both from educational and programming points of view since they confront the user with some of the most critical computer architecture problems, such as memory space, managing specific signals for different tasks or correctly addressing memory.

Little box. In the *little box* tutorial level, we have to create a component that can save and load four numbers using the registers already designed one level before. The challenge is that we don't have as much available space as we want, but this is the only limited-space level in the game.

The maze. At this level, we need to create a program that guides a robot through *a maze* without knowing what the maze looks like, the only input consisting in what the robot sees in front of it (a wall

or free space). One way to solve the level is to rotate the robot to the left every time it moves, and then while the robot is looking at a wall, we will rotate it to the right. When the robot looks at free space, we will command it to move forward once. This level marks the end of the first architecture and the beginning of a new architecture capable of many more interesting things.

Water world - the last level. At this level, a *water world* comes into the scene. We need to calculate how much water fits in a landscape consisting of 16 ground-level measurements. To complete this level, it is not necessarily required to have the stack implemented because it can be done iteratively. We don't necessarily need to use recursive functions, too, but we need to be able to represent a vector in which we store each height, which means we need to be able to access RAM using addresses.

3 A Classroom Companion

In the context of higher education, Turing Complete can serve as a valuable educational resource. Next, I will discuss several ways in which this game can help students learn about computers in universities in the future.

3.1 Practical labs

Within the realm of university education, Turing Complete emerges as an invaluable educational companion. The game transcends the conventional pedagogical approach by not merely explicating theoretical concepts but by rendering them tangible through an interactive environment. Consequently, students can apply theoretical knowledge in a practical setting, constructing their own computers and programming them to surmount challenges presented in the tutorial. It is akin to operationalizing theory, thereby rendering the educational experience significantly more engaging.

Throughout the process of creating my own computer, I've faced several obstacles. One of them was figuring out how to store vectors in memory to complete the final level of the tutorial. After a day of digging into information on how vectors are represented in memory, not only did I manage to add this function to my computer, but I also gained a much better understanding of what's happening behind the scenes when we use vectors and pointers in programming languages like C/C++.

So, another advantage I see in using the Turing Complete game in universities would be the possibility of asking direct questions to the professor, which is much better than searching for information on your own.

3.2 Team projects

Why learn in isolation when the potential for robust collaboration exists? Turing Complete is not solely a game; it acts as a facilitator of teamwork. Through team projects, students can be assigned specific components, ranging from decoders to arithmetic-logic units. Subsequently, armed with these seamlessly functioning components, the team collectively applies their intellect to integrate all facets. This hands-on experience fosters collaboration and enhances team-building skills.

The project plays a crucial role in bridging the gap between theoretical knowledge and practical application. As students engage in the creation of a functional computer, they not only grasp theoretical concepts but also gain practical insights into how these concepts manifest in real-world scenarios. This bridge between theory and practice enhances the educational experience, making it more holistic and applicable.

3.3 Learning through gamification

The adage of "learning is fun" finds resonance in the realm of gamified education. Research [3] indicates that gamified learning can accelerate information assimilation. Turing Complete injects an element of exhilaration into the learning process, rendering it more efficient and enjoyable. Therefore, not only do participants acquire knowledge, but they do so with a sense of enjoyment.

The report also investigates the impact of gamification on students based on their personality traits. The study reveals that the effects of gamification vary depending on the specific traits of users, underscoring the importance of personalizing the educational experience.

3.4 The Impact of Gamification in the Learning Process

Within the realm of higher education, utilizing Turing Complete as an educational resource becomes relevant not only by elucidating theoretical concepts but also by providing an interactive experience. Gamification transcends conventional pedagogical approaches, turning theoretical concepts into tangible entities through an interactive environment.

Students not only apply theoretical knowledge in a practical setting, constructing their own computers and programming them to overcome challenges presented in the tutorial, but also benefit from an operational experience of theory. This approach assimilates theory in a practical manner, making the educational experience significantly more engaging.

Therefore, introducing gamification into the learning process brings an element of excitement to education, making the accumulation of knowledge not only more efficient but also more enjoyable. Students not only learn but do so with a sense of joy and involvement, and this aspect can have positive consequences on their performance and motivation in the learning process.

4 What I Built

This section focuses on what I managed to achieve in this application without having any prior knowledge of what a computer system architecture is or what a program in assembly language looks like. The only thing I knew was how to program in a high-level language like C++, so I came to appreciate more the ease of writing code in such a language.

4.1 Description of the architecture used

The following figure (Fig. 4) represents the basic architecture as a foundation for the entire computer I built.

In this figure, the red wire represents a 64-bit bus used to specify to the computer which component should be activated and which operation should be performed. The orange and green wires represent a 64-bit bus each, specifying the memory addresses to be used in the operation indicated by the opcode. The pink wire represents a 64-bit bus that moves the result of the operation to the memory address specified by the blue wire.

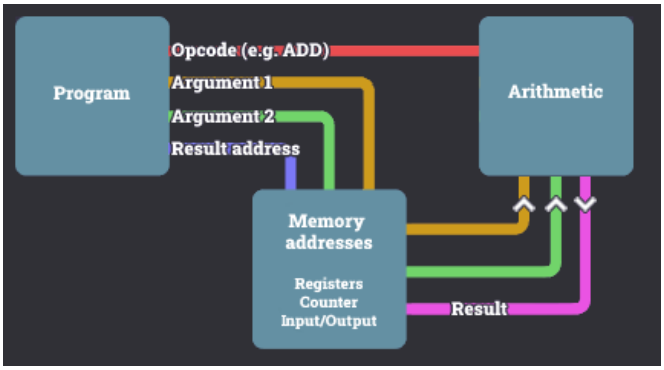


Fig. 4: My computer architecture

4.2 Description of the components

I will describe here the main components of my computer, everything being developed step by step and level by level in the Turing Complete framework, the address decoder, the arithmetic-logic unit ALU, the condition unit, the computer's stack, and the registers.

In Fig. 5, the address decoder is represented. It receives three bits through which it determines where the arguments for processing will be taken from and where the result will be sent to. Although it has only four outputs represented by just two bits, I added a third bit to have the possibility to add other memory areas in the future. The last input bit is for disabling the component when I send a number from the program directly to the arithmetic unit.

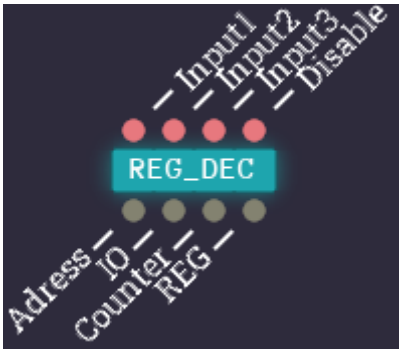


Fig. 5: Address decoder

Fig. 6 represents the arithmetic-logic unit, ALU, where all calculations are made, from additions and subtractions to multiplications, divisions, modulo, and bitwise operations between the two inputs. There are two 64-bit outputs because if I multiply two 64-bit numbers, the result will be represented on 128 bits.

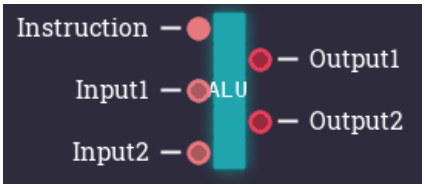


Fig. 6: Arithmetic Logic Unit (ALU)

Fig. 7 is the condition unit. It checks how Input1 relates to Input2. The condition, for example less than, greater than, equal, and so on, is transmitted through the Opcode. If the two inputs meet the condition, the output pin will be on. Otherwise, it will be off.

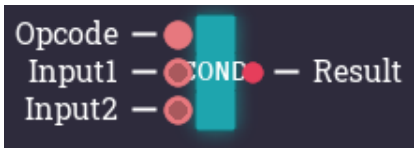


Fig. 7: Condition unit

Fig. 8 is the computer's stack, which allows the use of recursive functions because program positions can be stored so functions know where they need to return. Even function values can be placed on the stack before a function is called. This way, the program avoids modification when returning from that function.

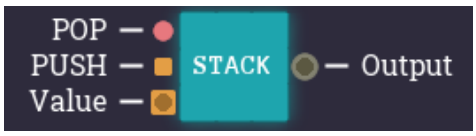


Fig. 8: Computer stack

Fig. 9 represents the computer's registers. At this development level of the project, the registers component consists of 256 registers. It allows the program to access two registers simultaneously and even to save into a register a value that will be applied in the next tick. This component is essential for the architecture I am using.

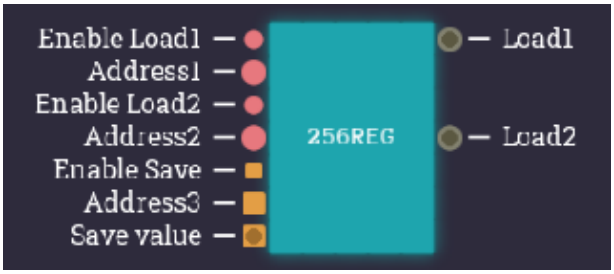


Fig. 9: Registers

4.3 Assembly and an example of a program

The game gives the user the possibility of using a component called "Program", which is similar to a RAM block, but its values can only be modified by the user. The only input of this module is a memory address to go to and output the respective address and the following three addresses. Based on the name of the component, we can understand that it stores the program we want the computer to execute.

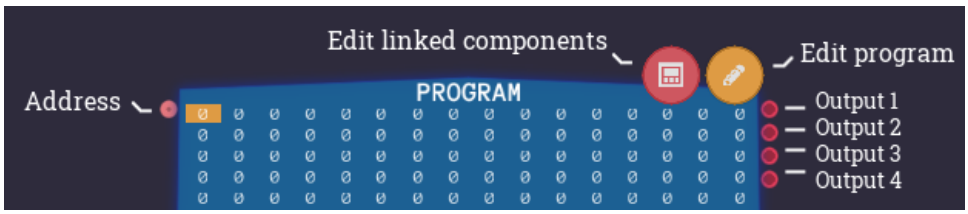


Fig. 10: Memory stored program

Fig. 10 represents this component, where the <Edit linked components> button is for debugging. Once the <Edit program> button is pressed, the selected program is displayed in a window (Fig. 11) by the code it contains. Moreover, a list of created commands, and four buttons to simulate ticks for line by line executing the program are also depicted in that window.

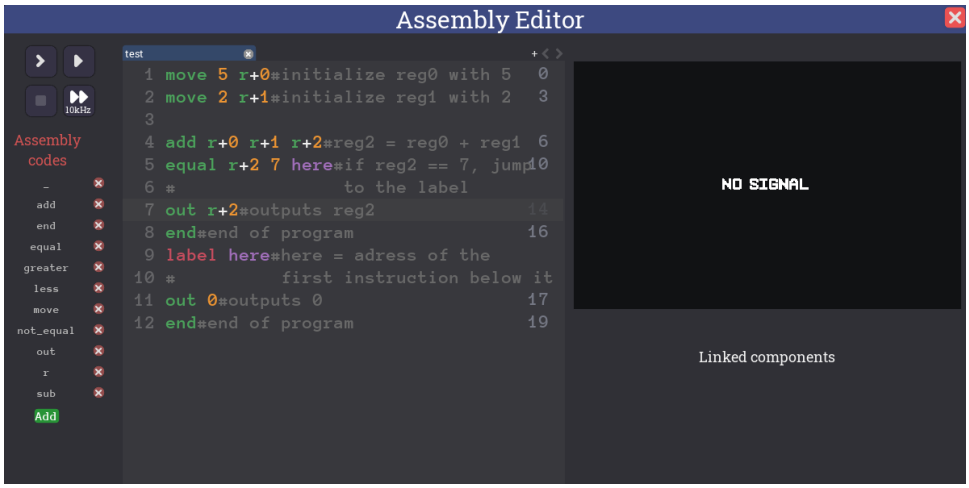


Fig. 11: Assembly editor

By connecting all of these modules representing a basic computer architecture, the assembly editor could describe specific programs for this computer. Moving forward, as an example, I would like to present a program that calculates the n^{th} number in the Fibonacci sequence, written in the assembly language for the Turing Complete computer I have created:

```
goto _ _ main
label fib
        eoli 0 1 if
        push _ 0 _
        subi 0 1 0
        call _ _ fib
        pop _ _ 2
        pop _ _ 0
        push _ 2 _
        subi 0 2 0
        call _ _ fib
        pop _ _ 3
        pop _ _ 2
        add 2 3 1
        pop _ _ 4
        push _ 1 _
```

```
        push _ 4 _
        return _ _ _
label if
        pop _ _ 4
        push _ 0 _
        push _ 4 _
        return _ _ _
label main
        movei _ 20 0
        call _ _ fib
        pop _ _ 1
        ramin 1 0 _
```

5 Turing Complete Community

Another relevant aspect of the Turing Complete application game is its community, where members can publish their projects regardless of complexity. Furthermore, they can contribute with even separate programs that can reduce the time required for writing assembly code or other quality of life improvements.

Using other people's creations, we can learn what another type of architecture looks like and see how another person approaches a particular component. If we find something interesting or more efficient than the components we created, we can use that component in our computer. If it lacks a certain functionality we need, we can modify it to be compatible with our architecture.

From the educational point of view, it is important to mention that this application community offers community-made guides as valuable learning support in computer architecture. As an example, I can mention a community-created tutorial where you can learn Karnaugh Veitch diagrams [4], which is helpful for designing a circuit that uses as few logic gates as possible. Once again, this appears as an essential educational feature for teaching Boolean algebra in terms of simplifying logic expressions and representing logic functions by circuits.

Another interesting example is a guide that shows how to finish the maze level using only two components, and one of them is a delay line [5].

The framework allows people to develop internal projects, fully available to the community members only. Such an extended and exciting example represented in the following Fig. 12 is a computer that can run Tetris [6]. Even if it is a game from the 1980 years, it is still a very challenging subject in the gamification industry and particularly in the computer architecture educational context [7].

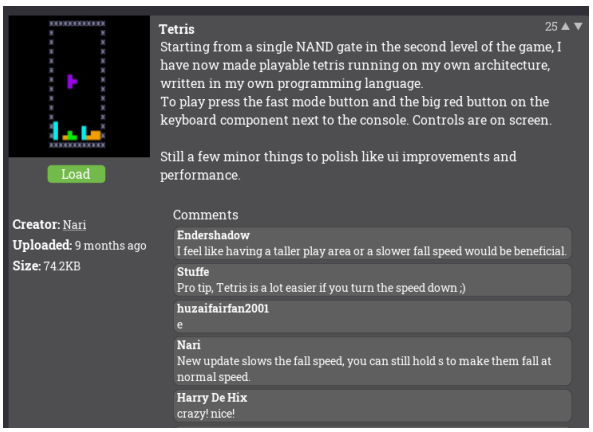


Fig. 12: Tetris project details

6 Conclusion

Regardless of the user's background in hardware and software, Turing Complete offers a unique and enjoyable approach to understanding computer systems. In addition, the Turing Complete game can significantly help students and self-learners find their passion for computer system architecture through its unique way of presenting the material and allowing you to experiment and learn from mistakes. I remember the evening I found this game online at around 8 pm, not even realizing when it was already 2 am, and I had to go to high school at 8 am, tired but excited.

This personal project not only brought a functional computer to life but also provided me with fertile ground to deeply understand essential concepts in computer architecture. These personal results demonstrate that a gamified project can have significant consequences for the learning process and understanding of complex concepts.

Until now, I managed to create a computer capable of performing all the basic operations, recursive functions and accessing the RAM by address. Using all of these capabilities, my computer can accomplish any task in a moderate amount of time. In the future, I would like to add more advanced components, such as the ability to use the keyboard, files from the host computer, read the time, and even connect to the internet. I also plan to build a small GPU to display text and images on the screen.

In conclusion, Turing Complete is a highly valuable educational tool that teaches users computer architecture and assembly language programming through engaging modern methods like gameplay, step-by-step learning, learning by doing, try and error and, not least, motivating challenges.

Acknowledgement: This work was supervised by Professor Anca Vasilescu, from *Transilvania* University of Brașov, Romania.

References

[1] ***, *Turing Complete Game*, available online at https://turingcomplete.game/
[2] Carl Burgh, *Logisim – a graphical tool for designing and simulating logic circuits*, available online at http://www.cburch.com/logisim/
[3] Smiderle, R., Rigo, S.J., Marques, L.B. et al. The impact of gamification on students' learning, engagement and behavior based on their personality traits. Smart Learn. Environ. 7, 3 (2020). https://doi.org/10.1186/s40561-019-0098-x
[4] ***, Turing Complete – *Karnaugh Veitch diagrams* community-created guide, https://steamcommunity.com/sharedfiles/filedetails/?id=2828368407
[5] ***. Turing Complete – Finish *the maze level* with two components, https://steamcommunity.com/sharedfiles/filedetails/?id=2842294429
[6] ***, Tetris – the puzzle video game, https://en.wikipedia.org/wiki/Tetris
[7] Noam Nisan, Shimon Schocken, *The official website of Nand to Tetris courses*, https://www.nand2tetris.org/license

Dennis-Vlăduț TEGLAS
Transilvania University of Brașov
Faculty of Mathematics and Computer Science
Brașov, B-dul Iuliu Maniu nr 50
ROMANIA
E-mail: teglasdennis@yahoo.com

Seventh International Conference on Applied Informatics
Imagination, Creativity, Design, Development
ICDD 2023, May 11-13
Sibiu, Romania

Using Nudging to Influence Sustainable Product Selection in an Online Supermarket under the Factor of Time Pressure

Leonie Zech

Abstract

The advancing climate change confronts society with new problems. For this reason, new solutions must be created to reduce climate change. Past studies have revealed that nudging can influence people in making healthier or more sustainable food choices in different environments. This study examines the effect of nudges on sustainable food choices under the factor time pressure, a system-1-condition. This was realized with the help of a realistic online supermarket. The more sustainable alternatives were highlighted with a nudge. A simplification nudge and a social norms nudge were used for this purpose. Participants had to add four products of different product categories to the shopping cart. Each of the four products was available in four different versions in the online store, of which two product options were marked with a nudge and two of the product options were presented without a nudge. The first group of participants had to do the shopping without any time constraints. The comparison group had to do the shopping under time pressure. It was found that the nudging effect was not influenced by time pressure, what indicates that the effect of nudging is not increased under a system-1-condition.

1 Introduction

Environmental risks are seen as the greatest future threats to planet earth [16]. Environmental risks include, among others, extreme weather and the failure of environmental protection measures [16]. People have numerous opportunities to make their everyday lives more sustainable in various areas and make their contribution to environmental protection measures. One area where a lot of carbon dioxide can be saved is grocery shopping. For example, buying one kilogram of strawberries in Germany in winter (not seasonal) leaves a carbon footprint of 3.4 kg CO_2-eq (kg CO_2-equivalents / kg food) [9]. In comparison, one kilogram of seasonal strawberries from the region leaves a carbon footprint of only 0.3 kg CO_2-eq [9]. This shows that buying regional and seasonal products can save pollutant emissions, which is partly because of shorter transport routes [15]. By buying organic products, consumers can also prevent pesticide contamination of groundwater [15]. In the process of digitalization, groceries can more and more be purchased online. Everyone is familiar with the situation when they have to buy something in a hurry at the supermarket just before closing time. This form of time pressure can also be found in

online shopping, for example, when you have to keep to a deadline so that delivery or collection takes place at the desired time. Often the customer is confronted with many different product options from one product category. Nudges can be used to guide people in the direction of the better product option [12], for example the more sustainable alternative. Nudging originates from the dual process theory of human decision making, which implies that people make decisions either quickly and intuitively (system 1) or reflectively and slowly (system 2) [8,12,13].

With this background, the opportunity was identified to test whether nudges in an online supermarket contribute to more sustainable products being chosen under time pressure. The research question is whether the use of time pressure increases the effect of nudging compared to no time pressure. In order to investigate this, a split test was performed with two groups using a realistic online supermarket. Group 1 had to do their shopping without any time constraints and Group 2 had to do their shopping under time constraints. The paper begins with the necessary theoretical background for the study. It shows how nudging works and describes the dual process theory. The next chapter describes the structure and process of the research. In the following chapter, the results of the study are presented and critically examined. The final chapter concludes the paper with a conclusion about the gained knowledge and provides information on future research approaches.

2 Literature Review

Previous research has confirmed that the use of nudging can have a positive influence on the dietary behavior of people [14,1,3,4,5]. This includes choosing healthier or more sustainable groceries. A research from 2021 examined the effect of nudging in a virtual supermarket using VR glasses [3]. The study tested whether nudging generally increases healthy food choices and whether there is a difference between the group without and with time pressure [3]. It was discovered that there was no difference between the two groups [3]. Time pressure as a system-1- condition made no difference in the choice of products marked with a nudge [3]. Another study also found that the effect of a nudge did not differ under system-1 and system-2-conditions [13].

From the previous research, it could be concluded that there is still a need for research regarding online grocery stores and the effect of nudging under a system-1-condition. This paper will specifically examine whether the use of nudging differentially influences the decisions of users regarding the choice of the more sustainable alternative in the context of online grocery shopping with or without time pressure.

3 Theoretical Background

3.1 Nudges

Nudging means giving a little push in a certain decision direction [12]. Nudges are measures that influence human behavior in a certain direction, but without pushing the human to make that decision [11,12]. Basically, there is no neutral design and even small details can draw attention and influence human behavior [12]. The goal of nudges is to guide people in a certain direction and make life easier or safer. A big advantage of nudging compared to prohibitions is that they do not force people to do anything. Nudges have great potential for companies, because on the one

hand they cost little and on the other hand economic or, as in this case, sustainable goals can be achieved more easily [11]. Existing research has already shown that the use of nudging can have an impact on the dietary behavior and can have a positive effect on the number of healthy or sustainable foods purchased [14,1,3,4,5].

3.2 System 1 and System 2

The concept of nudging comes from the dual process theory of human behavior and decision making, which states that people make decisions based on two different systems [13,8,12]. A differentiation is made between system 1, which works automatically and quickly, and system 2, which is responsible for the exhausting mental tasks [8]. These are two fictitious systems that represent two different types of thinking and cannot be localized directly in the brain [8]. It is assumed that up to 95 percent of purchasing behavior is controlled by system 1[6]. System 1 is used in particular when the customer has little time or has a low level of involvement with the product [6].

It is generally believed that nudges work better under system-1-conditions [13]. The effectiveness of system-1-conditions on nudging can be tested by applying time pressure [2], because it simulates fast and intuitive thinking [8]. A study from the year 2020 found that the effectiveness of nudges is not dependent on a system-1-condition, but can work just as well with a system-2- condition, for which participants were explicitly instructed to think longer about their decision [13]. This result was also confirmed by another study that examined shopping behavior in a virtual reality supermarket [3]. It was found that there was no difference in the effect of nudging on the purchase of healthy products when people shopped with or without time pressure and concluded that a system-1-condition is not required to make nudges more effective in healthy food choices [3]. These results do not confirm the approach that nudging works better under a system-

1-condition by tending to make faster and more unconscious decisions [3]. The following study will analyze whether the presence of time pressure, a system-1-condition [2], increases the choice of sustainable foods labeled with a nudge in an online supermarket.

4 Method and Study Design

4.1 Participants

A total of 62 subjects participated in the study. Two had to be excluded because they did not follow the instructions correctly. Therefore, the final sample size is 60. Of this total, 25 participants were male and 35 were female. 52 of the participants were between 18-29 years old and the remaining were older.

4.2 Procedure

The study was completed either on a desktop PC in a laboratory or remotely on a desktop. The purchasing process was recorded for the following evaluation. After the purchase, the participants were instructed to fill out a survey, which was not recorded. Because the study took place in Germany, the study was conducted in German. The participation was voluntary.

The sample was divided into two groups. The first group started without time pressure. This group was instructed in the following way: “I would like to ask you to put four products in the shopping

cart, which you can read from the shopping list that is in front of you. These products are 500 ml of chocolate ice cream, 500 ml of dishwashing liquid, 6 l of non-carbonated water and 1 kg of apples. Within these product types you can choose any item. Once all four products are in the cart, you are done shopping and can proceed to the final survey." The second group got the same instructions as group 1. But additionally, the second group was told that they were under time pressure because they wanted to get their groceries delivered today and the deadline for that was about to expire. A timer of 2 minutes 26 seconds was presented to each candidate from group 2. This value was calculated as the average value of group 1. In addition to the running timer, group
2 was told to hurry up, but still not to choose just any product, but what reflects their real shopping behavior. All participants (group 1 and group 2) were shown the same store.

4.3 The Use of Nudges

There are many different nudges, such as default rules, reminders or disclosure [11]. For this work, the simplification nudge and the social norms nudge were used [11]. Ideas or projects often have less success than they could have because they are too complex [11]. The simplification nudge should help to make ideas or programs easy to understand and more intuitive [11]. For this reason, the simplification nudge (Fig. 1) was used in this study by labeling the sustainable product with a sustainability label so that it is recognizable to the buyer immediately. One of the most effective nudges is the social norms nudge, which aims to show people that other people also show the behavior [11]. In the study, the sustainable product was labeled with a popularity label (Fig. 1) so that it is directly recognizable to potential buyers that previous buyers liked the product.

Fig. 1 Sustainability label (simplification nudge) and popularity label (social norms nudge)

4.4 The Online Supermarket

The online supermarket was built with the wix.com website builder. The pictures of the products used in the supermarket were taken from the website rewe.de. The detailed product view had the same information as the overview page. There were no additional information or descriptions about the products added in the detailed view. To model a real supermarket, participants were shown other products in addition to the products on their shopping list. In total, the supermarket had 7 product categories and a total of 60 products. The store includes both branded products and store-brands. The sustainable products were marked with two different nudges: the simplification nudge and the social norms nudge [11]. They were always in the same place for both Group 1 and Group 2. These nudge types were chosen because they are easy to implement in an online supermarket.

There were four different versions of each of the four products. They differed, for example, in brand and price. Of these four versions, one was labeled with a sustainability label (simplification nudge), one with a popularity label (social norms nudge), and two were not specifically labeled (Fig. 2). The first product on the shopping list was chocolate ice cream. In this series of four products,

the first product was labeled with a sustainability label, the second product with a popularity label, and the following two products without a label. The second product on the shopping list was dishwashing liquid. In this series of four products, the first product was marked with a popularity label, the second product with a sustainability label and the following two products without a label. The third product on the shopping list was non-carbonated water. In this row of four products, the first two products were without a label, the third product was marked with a sustainability label, and the fourth product was marked with a popularity label. The last product on the shopping list were apples. In this row of four products, the first two products were without a label, the third product was marked with a popularity label and the fourth product with a sustainability label. These products were chosen because the effect of nudges was to be measured with different product types.

Fig. 2 Example of the product options
Source: Rewe.de [17,18,19,20]

4.5 Measures

Product choice: To investigate whether time pressure increases the selection of nudged products, it was calculated how many products selected by participants, were nudge products. The number could range from zero to four.

Time pressure: In order to be able to calculate how much time pressure the second group should have so that they could still complete the task, all times were recorded by the first group and then the average value (2 minutes 26 seconds) was calculated.

Shopping behavior: After the test, the shopping behavior of the subjects was asked with the following two questions in an online survey (1) When you buy food, how much do you pay attention to the sustainability of the product you buy? (2) How strongly do you pay attention to the price of the purchased product when buying food? These questions could be answered on a

Likert Scale with the answer options: either strongly, rather strongly, undecided, less strongly and not at all strongly.

5 Results

Product selection. The main question of the study was if there is a difference between people without and with time pressure in terms of their product selection. The independent variables group without time pressure and group with time pressure were compared on their choice of products marked with a nudge (dependent variable).

To answer this question, a t-test for independent samples was performed (Fig. 3). For this purpose, Levene's test for variance homogeneity was first conducted. This revealed that there was variance homogeneity ($p = 0.584 > 0.05$), so a t-test for equal variances was performed. The two-tailed t-test revealed $p = 0.146 > 0.05$, which means that no significant difference between the two groups could be proven. For this reason, the null hypothesis is accepted, which states that there is no difference between the group without time pressure ($m = 1.9$, $sd = 1.13$) and the group with time pressure ($m = 1.5$, $sd = 0.97$) in the choice of products marked with a nudge.

	Levene's Test for Equality of Variances		t-test			
					Significance	
	F	Sig.	t	df	One-Sided p	Two-Sided p
Products marked with nudge	,304	,584	1,472	58	,073	,146

Fig. 3 Result of the t-test for the choice of products marked with a nudge

In addition, the effect of the simplification nudge and the social norms nudge was tested separately. It was first started with the question if there is a difference between people without and with time pressure regarding the choice of the products marked with a simplification nudge (Fig. 4). In this case, there is also variance homogeneity ($p = 0.361 > 0.05$). The two-sided t-test showed $p = 0.350 > 0.05$. No significant difference between the groups could be detected here either. For this reason, the null hypothesis is accepted, which states that there is no difference between the group without time pressure ($m = 0.9$, $sd = 1.06$) and the group with time pressure ($m = 0.67$, $sd = 0.84$) in the choice of the products marked with a simplification nudge.

		Levene's Test for Equality of Variances		t-test			
						Significance	
		F	Sig.	t	df	One-Sided p	Two-Sided p
Simplification Nudge	Equal variances assumed	,849	,361	,942	58	,175	,350

Fig. 4 Result of the t-test for the choice of products marked with a simplification nudge

Finally, it was checked whether there is a difference between persons without and with time pressure regarding the choice of the products marked with a social norms nudge (Fig. 5). There is no variance homogeneity here ($p = 0.019 < 0.05$). The two-tailed t-test showed $p = 0.442 > 0.05$. Again, no significant difference between the groups could be detected. For this reason, the null hypothesis is accepted, which states that there is no difference between the group without time

pressure (m = 1.0, sd = 0.98) and the group with time pressure (m = 0.83, sd = 0.65) in the choice of products marked with a social norms nudge.

		Levene's Test for Equality of Variances		t-test			
						Significance	
		F	Sig.	t	df	One-Sided p	Two-Sided p
Social Norms Nudge	Equal variances assumed	5,814	,019	,776	58	,221	,441
	Equal variances not assumed			,776	50,200	,221	,442

Fig. 5 Result of the t-test for the choice of products marked with social norms nudge

Estimation of sustainable purchasing behavior. After the test task, the participants filled out a questionnaire. In it, they were asked to state how strongly they pay attention to buying sustainable products in their everyday lives. This question could be answered either strongly, rather strongly, undecided, less strongly, and not at all strongly. The purpose of this question was to find out if there is a difference between people with the assessment of strongly (answer options either strongly and rather strongly summarized) or less strongly (answer options less strongly and not at all strongly summarized) paying attention to sustainability in groceries in their choice of sustainable products. For this purpose, the independent variables of strong attention to sustainability and less attention to sustainability were compared with the dependent variable, the number of selected nudge products (Fig. 6). There is no variance homogeneity here ($p = 0.010 < 0.05$). The two-tailed t-test showed $p = 0.011 < 0.05$. A significant difference between the groups could be detected. For this reason, the null hypothesis is rejected. There is a difference between individuals with the estimation of strongly (m = 2.29 sd = 1.33) or less strongly (m = 1.2, sd = 0.42) paying attention to sustainability in the choice of sustainable products. This and a look at the collected data shows that people who said they pay attention to sustainability when buying groceries also bought more sustainable and therefore more nudged products. People who said they pay no attention to sustainability when buying groceries also bought less sustainable and therefore less nudged products.

		Levene's Test for Equality of Variances		t-test			
						Significance	
		F	Sig.	t	df	One-Sided p	Two-Sided p
Products marked with nudge	Equal variances assumed	8,066	,010	2,487	22	,010	,021
	Equal variances not assumed			2,867	16,464	,005	,011

Fig. 6 Result of the t-test for sustainable purchasing behavior

6 Discussion and further research

The main question this paper aimed to answer was whether the effect of nudges differs without and with time pressure. It was found that there was no difference in the effect of nudges in the two different groups. The result that the effect of nudging is not changed under system-1-conditions confirms the results of past studies [3,13]. Furthermore, no significant differences in the effect of the simplification nudge and the social norms nudge could be found in the groups without and with time pressure either. In summary, the results show that the nudge effect does not depend on whether individuals are under no or high time pressure. In this study, a system-1-condition was

simulated by using time pressure. The influence of a system-2-condition on the effectiveness of nudges was not investigated. The group without time pressure would have had more time for system-2-considerations, but the participants were not explicitly instructed to consider their purchase decisions carefully and for this reason the system-2-conditions were not tested [3]. For future studies, two groups, one group under a system-1-condition and the other group under a system-2-condition could be compared in the choice of products marked with a nudge in an online grocery store. It would also be interesting to check whether certain nudges work better or worse with different products. It could be the case that the social norms nudge had not worked so well because these are everyday products that everyone buys on a regular basis. It has been found in the past that the social norms nudge works better when there is ambiguity about what the correct behavior would be [7]. For this reason, it would be a good idea to test it with products that the test persons do not buy regularly and that are more complicated.

A study has found that nudges have a better impact when the nudged option already matches personal preferences anyway, or when the person still has doubts about their choice and has a hard time to decide [10]. If the options being used do not basically match the intentions of a subject, they are more likely to be ineffective [10]. This was also confirmed in this study. In the test, people who said they already pay attention to buying sustainable food in their everyday life also increasingly chose the sustainable products labeled with a nudge. People who said they were less likely to buy sustainable food in their everyday life also chose less sustainable with a nudge labeled products. At this point it must be mentioned that the sample size was very low due to the available resources. This result should be validated with a larger sample size. In addition, during the study, it was increasingly noticeable through the comments of the participants that they often chose products based on preferred brands, what confirms the assumption that the chosen products match personal preferences. To confirm this assumption further studies can explore the effect of nudges in increasing product choice when people already have a preference in that direction or the effect of nudges when people do not have a preference in that direction. Furthermore, an eye tracker can also be used to check the extent to which the nudges are being noticed at all.

The small number of participants of 60 is a limiting factor of this study. As well as that only four different product types were tested. To achieve more valid results, in a future study the process of conducting the study should be automated with the help of an application. With an application, a larger number of subjects and products can be tested over a longer period of time. The application should be able to track the shopping behavior of users in the online store and evaluate the data automatically in a dashboard with regard to the product choice.

Acknowledgement: This work was supervised by Professor *Karsten Huffstadt*, from *Technical University of Applied Sciences Würzburg-Schweinfurt*.

References

[1] Arno, Anneliese and Thomas, Steve, *The efficacy of nudge theory strategies in influencing adult dietary behaviour: a systematic review and meta-analysis*. BMC Public Health, 16, 1, 676, 2016.

[2] Bago, Bence and De Neys, Wim, *Fast logic?: Examining the time course assumption of dual process theory*. Cognition 158, 90–109, 2017.

[3] Blom, Stephanie S. A. H.; Gillebaart, Marle; De Boer, Femke; Van der Laan, Nynk; De Ridder, Denise T. D., *Under pressure: Nudging increases healthy food choice in a virtual reality supermarket, irrespective of system 1 reasoning*. Appetite 160, 105116, 2021.
[4] Broers, Valérie J. V.; De Breucker, Céline; Van den Broucke, Stephan; Luminet, Olivier, *A systematic review and meta-analysis of the effectiveness of nudging to increase fruit and vegetable choice*. European Journal of Public Health 27, 5, 912–920, 2017.
[5] Hansen, Pelle G; Schilling, Mathilde and Malthesen, Mia S., *Nudging healthy and sustainable food choices: three randomized controlled field experiments using a vegetarian lunch-default as a normative signal*. Journal of Public Health 43, 2, 392–397, 2021.
[6] Häusel, Hans-Georg, *Neuromarketing: Erkenntnisse der Hirnforschung für Markenführung, Werbung und Verkauf*. 2. Edition, Freiburg München, Haufe-Lexware, 2012.
[7] Higgs, Suzanne, *Social norms and their influence on eating behaviours*. Appetite 86, 38–44, 2015.
[8] Kahneman, Daniel, *Schnelles Denken, langsames Denken*. 10. Edition, München, Penguin Verlag, 2011.
[9] Reinhardt, Guido; Gärtner, Sven and Wagner, Tobias, *Ökologische Fußabdrücke von Lebensmitteln und Gerichten in Deutschland*. Institut für Energie- und Umweltforschung Heidelberg, 2020.
[10] De Ridder, Denise, Kroese, Floor and Van Gestel, Laurens, *Nudgeability: Mapping Conditions of Susceptibility to Nudge Influence*. Perspectives on Psychological Science 17, 2, 346–359, 2022.
[11] Sunstein, Cass R., *Nudging: A Very Short Guide*. Journal of Consumer Policy 37, 4, 583–588, 2014.
[12] Thaler, Richard and Sunstein, Cass R., *Nudge - Wie man kluge Entscheidungen anstößt*. 6. Edition, Berlin Ullstein, 2016.
[13] Van Gestel, L. C.; Adriaanse, M. A. and De Ridder, D. T. D., *Do nudges make use of automatic processing? Unraveling the effects of a default nudge under type 1 and type 2 processing*. Comprehensive Results in Social Psychology 1–21, 2020.
[14] Van Gestel, L. C.; Kroese, F. M. and De Ridder, D. T. D., *Nudging at the checkout counter – A longitudinal study of the effect of a food repositioning nudge on healthy food choice*. Psychology & Health 33, 6, 800–809, 2018.
[15] Von Koerber, Karl and Kretschmer, Jürgen, *Zukunftsfähige Ernährung Gesundheits-, Umwelt-, Wirtschafts- und Sozialverträglichkeit im Lebensmittelbereich*. Zeitschrift für Ernährungsökologie 1(1), 39-46, 2000.
[16] World Economic Forum, *The global risks report 2022: 17th edition: insight report*. Geneva: World economic forum, 2022.
[17] Fig. 2: Rewe.de: *Example of the product options*. https://shop.rewe.de/p/rewe-regional-apfel-rot-1kg/1907470, accessed on 08.11.2022.
[18] Fig. 2: Rewe.de: *Example of the product options*. https://shop.rewe.de/p/rewe-bio-apfel-rot-netz/1907545, accessed on 08.11.2022.
[19] Fig. 2: Rewe.de: *Example of the product options*. https://shop.rewe.de/p/apfel-gala-1kg-schale/8179366, accessed on 08.11.2022.
[20] Fig. 2: Rewe.de: *Example of the product options*. https://shop.rewe.de/p/kanzi-apfel-nicoter-1kg/8146226, accessed on 08.11.2022.

Leonie ZECH
Technical University of Applied Sciences
Würzburg Schweinfurt
Faculty of Computer Science and Business
Information Systems
Münzstraße 12, 97070, Würzburg
GERMANY
E-mail: leonie.zech@study.thws.de

LIST OF AUTHORS

Bozhidar ATANASOV
University of Ruse "Angel Kanchev"
Natural Sciences and Education
8 Studentska str., POB 7017, Ruse
BULGARIA
E-mail: s216271@stud.uni-ruse.bg

Galina ATANASOVA
University of Ruse "Angel Kanchev"
Natural Sciences and Education
8 Studentska str., POB 7017, Ruse
BULGARIA
E-mail: gatanasova@uni-ruse.bg

Ștefănel-Alexandru BANU
Vasile Alecsandri University of Bacău
Faculty of Science
Calea Mărăşeşti, nr. 157, Bacău, 600115
ROMANIA
E-mail: stefanelbanu@yahoo.com

Marius BENKERT
Technical University Würzburg-Schweinfurt
CAIRO, 97082, Würzburg
GERMANY
E-mail: marius.benkert@study.thws.de

Chandu BHAIRAPU
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: chandu.bhairapu@study.thws.de

Cristian Nicolae CÎNDEA
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: cristi21122001@gmail.com

Sorin Ionuț CONEA
Vasile Alecsandri University of Bacău
Department of Mathematics& Informatics
Calea Mărăşeşti, nr. 157, Bacău, 600115
ROMANIA
E-mail: stefanelbanu@yahoo.com

Zeynep Dilan DAŞKIN
Atılım University
Department of Mechatronics Engineering
Kizilcasar Mahallesi, 1184.
Cad No:13, 06830 Incek,
Golbasi/Ankara
TURKEY
E-mail: daskin.zeynepdilan@gmail.com

Stephanie FÎȘCĂ
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: stephanie.fisca@ulbsibiu.ro

David-Stelian HOKA
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: davidstelian.hoka@ulbsibiu.ro

Felix HUSAC
ROMANIA
E-mail: husacfelix@gmail.com

Augusta IONAȘ
Samuel von Brukenthal National College
Mathematics-Informatics
Piața Albert Huet 5, Sibiu
ROMANIA
E-mail: ionasaugusta@gmail.com

Corina Minodora IORDACHE
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: corinaminodora.iordache@ulbsibiu.ro

Indrasena Reddy KACHANA
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: sivasaikumarreddy.kachana@study.thws.de

Maria-Denisa KISS
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: mariadenisa.kiss@ulbsibiu.ro

Diana Christina MANCAS
Ovidius University
Mathematics and Computer Science Department
Bd. Mamaia 123, Constanta
ROMANIA
E-mail: diana.mancas@365.univ-ovidius.ro

Matei-Ioan MARIN
Gheorghe Lazăr National College
1-3 Gheorghe Lazăr str, Sibiu, 500035
ROMANIA
E-mail: matei.i.marin@gmail.com

Peter MISHEV
University of Ruse “Angel Kanchev”
Natural Sciences and Education
8 Studentska str., POB 7017, Ruse
BULGARIA
E-mail: s206252@stud.uni-ruse.bg

Fatima MOHAMED
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: fatima.mohamed@study.thws.de

Rareș-Gabriel MUȘEA
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: raresgabriel.musea@ulbsibiu.ro

Christopher NEEB
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: christopher.neeb@study.thws.de

Vladislav NIKOLOV
University of Ruse “Angel Kunchev”
Informatics and Information Technologies
Ruse, 17 Studentska street,
BULGARIA
E-mail: s206253@stud.uni-ruse.bg

Victor Iulian ŐTVEȘ
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: victorotves@yahoo.com

Phuoc Nguyen PHAM
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: phuocnguyen.pham@student.fhws.de

Gabriel Claudiu RADU
"Titu Maiorescu" National College Aiud
Mathematics-informatics intensive English
ROMÂNIA
E-mail: radugabriel796@gmail.com

Lennard ROSE
Technical University Würzburg-Schweinfurt
CAIRO, 97082, Würzburg
GERMANY
E-mail: lennard.rose@study.thws.de

Mahesh SARAVANAN
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: mahesh.saravanan@study.thws.de

Muhammad Zakriya Shah SARWAR
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: muhammadzakriya.shahsarwar@study.thws.de

Jochen SCHMIDT
Technical University Würzburg-Schweinfurt
CAIRO, 97082, Würzburg
GERMANY
E-mail: jochen.schmidt.1@study.thws.de

Marin-Eusebiu ȘERBAN
Lucian Blaga University of Sibiu
Faculty of Science
Department of Mathematics and Informatics
5-7 Dr. Ratiu Str, Sibiu 550012
ROMANIA
E-mail: eusebiu.serban@ulbsibiu.ro

Nehmiya SHIKUR
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: nehmiya.shikur@study.thws.de

Kameliya SHOYLEKOVA
University of Ruse “Angel Kanchev”
Natural Sciences and Education
8 Studentska str., POB 7017, Ruse
BULGARIA
E-mail: kshoylekova@uni-ruse.bg

Matei-Cristian STEAVU
"Radu Negru" National College, Făgăraș
Mathematics-informatics intensive English
ROMANIA
E-mail: mateisteavu@yahoo.com

Dennis-Vlăduț TEGLAS
Transilvania University of Brașov
Faculty of Mathematics and Computer Science
50 Iuliu Maniu Boulevard, Brașov
ROMANIA
E-mail: teglasdennis@yahoo.com

Julian TILLY
Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)
Faculty of Computer Science and Business Information Systems
Sanderheinrichsleitenweg 20, 97074 Würzburg
GERMANY
E-mail: julian.tilly@study.thws.de

George Bogdan TRÎMBIȚAȘ
"Titu Maiorescu" National College Aiud
Mathematics-informatics intensive English
ROMÂNIA
E-mail: trimbitasb@gmail.com

Leonie ZECH
Technical University of Applied Sciences
Würzburg Schweinfurt
Faculty of Computer Science and Business
Information Systems
Münzstraße 12, 97070, Würzburg
GERMANY
E-mail: leonie.zech@study.thws.de

Organized with support of Romanian Ministry of National Education

Romanian Ministry of National Education

SPONSORS (in alphabetical order)

Asociatia Sibiu IT

AUSY Technologies Romania

ASTRA National Museum Complex

Asociația BIT

CodexWorks technologies

***Ardeleana Academy* Foundation**

GESS Engineering

Global Solutions for Development

NXP

NTT Data

PAN FOOD

Omeron Technologies, Romania

ProIT

ROPARDO

Top Tech

Wenglor