الجمهورية الجزائرية الديمقراطية الشعبية
PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
وزارة التعليم العالي و البحث العلمي
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
جامعة عمّار ثليجي بالأغواط
UNIVERSITY OF AMAR TELIDJI LAGHOUAT

كلية العلوم
FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

MASTER Thesis

Domain : Mathematics and Computer Science
Field : Computer Science
Option : Networks, Systems and Distributed Applications

By:
BENAROUS Abderrazak

THEME

Steganography using Augmented Reality

Publicly defended on 10-06-2024 in front of the jury composed of:

Dr. Laradj CHELLAMA	*M.C.(B)*	*President*
Dr. Lahcen Mohammed BENSAAD	*M.C.(B)*	*Examiner*
Dr. Narjes HAMINI	*M.C.(B)*	*Examiner*
Dr. Leila BENAROUS	*M.C.(B)*	*Supervisor*

N ……… Academic year 2023-2024

Dedication

To my treasured parents,

My cherished brother,

My precious sisters,

My gorgeous nieces and nephews

Acknowledgement

First and foremost, I would like to immensely and infinitively thank Allah the great for blessing me, helping me and for giving me the ability to elaborate this work.

I deliver my ultimate gratitude to my parents. I am thankful to you for everything, I would not be who I am nor be where I am, without you. Thank you so much for your unlimited support and unending love, for your immense care, for your unceasing encouragements and patience.

I also thank my brother and sisters for being my pillar of support, for their cheers, and understanding.

I would like to thank my supervisor for her dedication to work, for her fruitful discussions, for her encouragements and support, for her open mind and flexibility, for all the guidance and advises, and for her patience and efforts, for her persistence in bringing the best out of me and honing my skills. It was an honor to work under your supervision.

Particular thanks the honorable jury members for the efforts and time spent reading the document, and for their constructive comments aiming to enhance its quality.

I am thankful to all the great persons and inspiring individuals who have helped me, your existence made a difference. I am appreciative for your positivity, your encouragements, your continuous support and your kind help.

ملخص

يتميز هذا العصر بتقدم كبير في التكنولوجيا، وخاصة في المجالات المتعلقة بالسرية، الحماية، الواقع المعزز والأنظمة الذكية. وفي عملنا، قمنا بدمج هذه التقنيات الأربع في عالم الواقع المعزز. لقد قمنا بتطوير تطبيق ذو دورين أساسيين: الدور الأول يخدم جميع المستخدمين، مما يسمح لهم باكتشاف وعرض أوصاف الأشياء المختلفة بناءً على البيانات التي يتم جلبها من قاعدة البيانات. الدور الثاني موجه الى فئة المستخدمين المهتمين بالاتصالات السرية و تبادل المعلومات بشكل خفي و محمي، حيث يتيح هذا الدور إخفاء المعلومات السرية والكشف عنها. لقد قمنا بتطوير التطبيق باستخدام Unity Game Engine من اجل تطوير تطبيق يعمل بالواقع المعزز. بالنسبة للذكاء الاصطناعي، تم استخدام نموذج YOLOv8 لتدريب نموذجنا بشكل مخصص على قائمة مكونة من 10 أشياء من اجل هذا النموذج. بالنسبة لإخفاء المعلومات والتشفير، استخدمنا خوارزمية LSB وخوارزمية AES.

الكلمات المفتاحية: إخفاء المعلومات، التشفير، الصور، الذكاء الاصطناعي، التعلم العميق ،الإخفاء والحماية، الواقع المعزز.

Abstract

This era is marked by a significant advancement of technology, particularly in areas related to secrecy, confidentiality, augmented reality and intelligent systems. In our work, we mixed these four technologies into the realm of augmented reality. We developed an application with two primary roles: Role 1 serves all users, allowing them to detect and display descriptions of various objects based on data fetched from a database. Role 2, intended users interested in covert communication, enables the hiding and revealing of secret information. We elaborated the application using Unity Game Engine for AR. For AI, YOLOv8 pretrained model was used to custom-train our model on a prototype list of 10 objects. For steganography and cryptography, we employed the LSB algorithm and the AES algorithm, respectively.

Key-words: steganography, cryptography, images, artificial intelligence, deep learning, concealment, protection, augmented reality.

Résumé

Cette époque est marquée par une avancée significative de la technologie, en particulier dans les domaines liés au secret, à la confidentialité, à la réalité augmentée et aux systèmes intelligents. Dans notre travail, nous avons combiné ces quatre technologies dans le domaine de la réalité augmentée. Nous avons développé une application avec deux rôles principaux : le premier rôle sert tous les utilisateurs, leur permettant de détecter et d'afficher des descriptions de divers objets en se basant sur des données récupérées d'une base de données. Le deuxième rôle, destiné aux utilisateurs intéressés par la communication secrète, permet de cacher et de révéler des informations secrètes. Nous avons élaboré l'application en utilisant le moteur de jeu Unity pour la réalité augmentée. Pour l'intelligence artificielle, le modèle pré-entraîné YOLOv8 a été utilisé pour entraîner notre modèle sur une liste prototype de 10 objets. Pour la stéganographie et la cryptographie, nous avons employé respectivement l'algorithme LSB et l'algorithme AES.

Mots-clés : stéganographie, cryptographie, images, intelligence artificielle, apprentissage profond, dissimulation, protection, réalité augmentée.

Table of content

List of Abbreviations

AES	Advanced Encryption Standard
AI	Artificial Intelligence
App	Application
AR	Augmented Reality
CIS	Coverless Image Steganography
CNN	Convolutional Neural Network
Colab	Collaboratory
CRT	Cathode Ray Tube
CSP	Cross Stage Partial
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DWT	Discrete wavelet Transform
GAN	Generative adversarial network
HHO	Harris Hawks optimization
HHO-IWT	Harris Hawks optimization-integer wavelet transform
JPEG	Joint Photographic Experts Group
LSB	Least Significant Bit
mAP	mean Average Precision
MR	Mixed Reality
PNG	Portable Network Graphics
PROCAM	Projector and Camera
SIFT	Scale Invariant Feature Transform
SQL	Structured Query Language
starGAN	Star Generative Adversarial Network
VR	Virtual Reality
XR	Extended Reality
YOLO	You Only Look Once

List of Figures

List of Tables

General Introduction

General Introduction

Nowadays, we are witnessing a fascinating technological progress. In recent days, two pillars are standing tall: augmented reality and artificial intelligence. Augmented reality, a domain where the real-world melds seamlessly with the digital world, while artificial intelligence encompasses techniques for problem-solving intelligently mimicking human brain.

In the secrecy domain, steganography emerges as a concealment art, embedding secret information within the innocuous everyday objects to make it undetectable. Cryptography, its counterpart, encrypts the secret to make unreadable permitting access only to those with the decryption key.

In this work, we embarked on a journey, combining these secrecy and artificial intelligence techniques with the augmented reality to develop an application with dual roles. Role 1 destined to everyone, offering descriptions to detected objects. Role 2, reserved for those who want a covert communication, enabling a concealed mean of exchanging secret information in the expanded reality.

The forthcoming chapters explains our development journey. In the creation process, Unity Game Engine companied by Vuforia SDK were used for augmented reality's techniques. AI, embodied by the YOLOv8 precept to custom-train our AI model on a prototype set comprising ten objects. Steganography was implemented with the LSB algorithm, while cryptography was with AES algorithm.

In chapter one, we have an overview of AR history, showing its evolution from its nascent beginnings to its current applications across different sectors. Next, we clarify the difference between mixed reality, augmented reality, and virtual reality. Finally, we highlight its future trends including the integration of AI.

In chapter two, we provide an exploration of steganography, demonstrating its history, examining its modern mediums, studying its use cases and applications, and focusing on the integration of AI for steganography.

In chapter three, we explain our novel contribution that applies steganography in the third dimension, in the AR environment extending the 2D computer world with the real world. We depict the backend development process in detail.

In chapter four, we illustrate the frontend part of the application. Beginning by explaining the views of this application, then explaining how each view was designed and created.

At the end, an Auto-Evaluation section is proposed, where we self-evaluate strengths and weaknesses of our project and highlight released aims from the set of initially drawn objectives. Additionally, we identify areas for improvement and outline potential avenues for future development. This critical self-assessment provides valuable insights to guide our ongoing efforts to improve and refine the application, ensuring its continued relevance and success.

CHAPTER 01
Augmented Reality

1.1 Introduction

Humanity has witnessed an evolving era of technology in the last decades, from learning how to deal with computers, then using the internet, to delving to the flourishing artificial intelligence applications. While people were fascinated by the electronic world that they named cyber world, virtual world, connect world and its provided services. Scientist got eager to coin the word with better implementation such as the virtual reality. In virtual reality, users enter completely virtual world disconnecting from reality. In that virtual environment, they are taken by the set-up, the adventures, and the tasks they do there isolating themselves from reality. Subsequently, the necessity to expend real world with the virtual one led to the appearance of augmented reality (AR) concept. In this chapter we have an overview of AR history, tracing its evolution from its nascent beginnings to its current applications across diverse sectors. Next, we clarify the difference between mixed reality, augmented reality, and virtual reality. Finally, we highlight its future trends including the integration of Artificial Intelligence and Internet of Things. This convergence promises to unlock new frontiers in AR technology, enhancing its capabilities and expanding its utility in various domains.

1.2 Augmented Reality History

AR was first introduced with Pepper's ghost in 1892 which involved setting up a stage with two distinct rooms one visible to the audience, and the other is hidden either placed on the side or below. A plate of glass is placed at the edge of the stage, angled so that it mirrors the view of the second room towards the audience. When the second room is illuminated, its image (the ghost in the room) is projected on the glass and appears to them as illustrated in Figure **1.1**. This staged setup although said to be the initial usage of augmented reality, however, it is more like nowadays hologram then like the AR we know today, which is thought of as overlaying images, one is virtual, and the other is a real-world view, done through devices such as goggles or tablets [1].

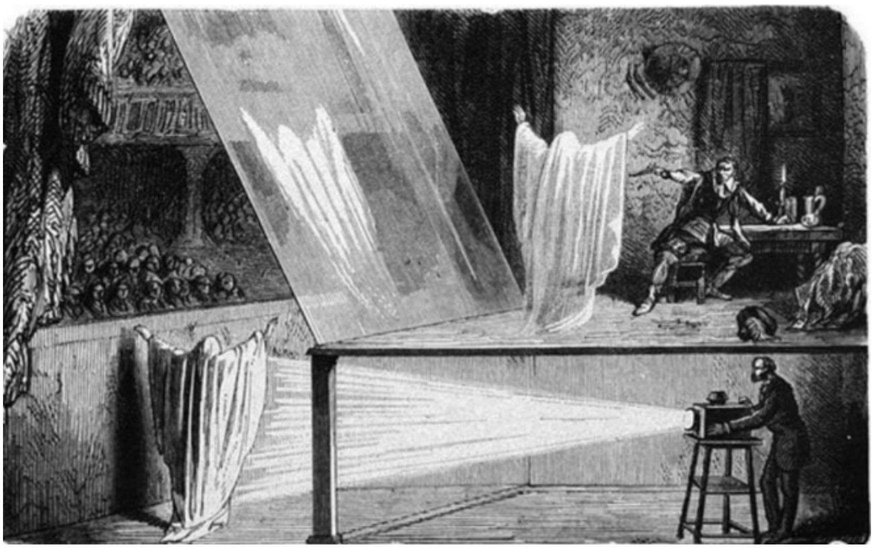

Figure 1. 1. Pepper's Ghost setup *[1]*

The English scientists are the ones who attempted the first electronic implementation of AR when they integrated an airborne RADAR navigation system in a WWII de Havilland Mosquito night fighter (a British twin-engined, multirole combat aircraft, introduced during the Second World War). In 1971, South Africa and the United States made significant advancements in helmet-mounted gun-sight technology. South Africa was the first air force to operationalize the use of helmet gun-sights, while Honeywell Inc. finalized the development of a visor projected display/tracking system for the US Air Force. This system featured a Cathode Ray Tube (CRT) mounted on the back of the helmet generating the display image, which was then relayed to the front via a coherent fiber optical bundle (Figure **1.2**) [1]. In 1974, The first appearance of Visual Target Acquisition System which consists of helmet-mounted vision system in US Navy fighter aircraft.

Figure 1. 2. Prototype AR helmet circa 1971 *[1]*

However, AR was not restricted to army use, its application extended beyond that. The entertainment industry quickly recognized its potential and raised people's interest in it and led to its expansion and the widespread adoption of AR technology. In 1994, The first use of AR in entertainment sector when Julie Martin created an AR theater production "Dancing in Cyberspace" [1]. In 1996, appearance of the first outdoor mobile AR system by Professor Steven Feinberg at Columbia University. In the same year, Sony released a head-mounted display called "Glasstron" see Figure 1.3.

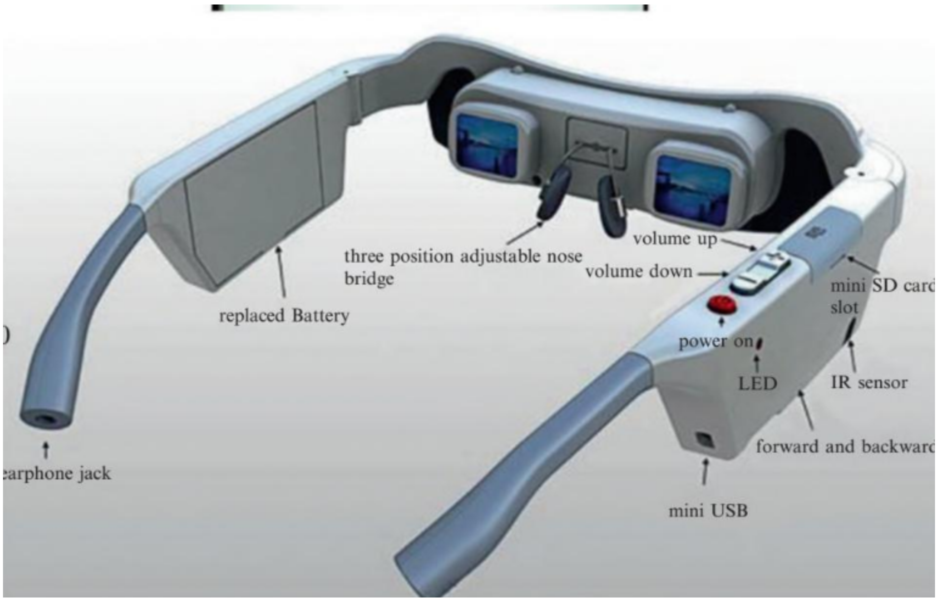

Figure 1. 3. Sony Glasstron device *[1]*

By late 1990's, AR has become a point of interest to the scientific community, conferences and workshops were held to discuss its progress and evolution. In 1997, Ronald Azuma established its three main elements which were combining real and virtual, interactivity in real time, and, recording in 3D [2]. Combining real and virtual means that AR blends the real-world environment with computer-generated virtual objects allowing users to interact with both of them. Interactivity in real time means that AR systems provide real-time interaction, where virtual objects respond to changes in the real-world environment in real-time. Recording in 3D means that virtual objects appear anchored or aligned correctly within the user's view of the physical world.

In 2004, Mathias Mohring, Christian Lessig and Oliver Bimber managed to put the first AR system on a consumer cell phone [3]. Then, in 2010, Microsoft launches project Baraboo, to develop mixed reality smart-glasses. Subsequently, in 2013, car manufacturers began using AR to replace traditional vehicle service manuals. One example of this technology is the

Volkswagen MARTA application (Mobile Augmented Reality Assistance), provides service technicians with detailed information about their vehicles, including diagrams and step-by-step instructions for performing maintenance tasks, while the Audi AR application uses the iPhone camera to provide details on 300 elements about the vehicle from windscreen wipers to the oil cap [4]. The real milestone that made the buzz, was the release of Metaverse in 2021 by Meta (formerly Facebook). Metaverse expanded social networks to the virtual environment. It is defined as a virtual reality space where users can interact with each other in a computer-generated environment. Recently, (2024), the Apple Vision Pro by Apple is a mixed reality headset was the talk of town. It features ultra-high-resolution displays that deliver more pixels than a 4K TV for each eye, allowing users to watch their favorite content on a 100-foot screen [5].

In Figure 1.4 significant AR milestones are depicted, spanning from the 1970s to the year 2024. These milestones mark key advancements, innovations, and developments that have shaped the evolution of augmented reality technology over the decades.

Figure 1. 4. Historical Overview of AR

1.3 AR current applications

AR usage emerged to several fields after it was limited to the military field. Now it is touching various aspects of our lives with its wide range of applications. AR can be used to enhance our shopping experiences by visualizing in 3D the products and virtually allowing you to try clothes and shoes to see how it looks on you and check if it fits you without the need to wear them. Moreover, it has revolutionized the education system by including 3d visualizations with explanations, animations, simulations, and emulations, one of its flourishing applications are medical applications including emulation of human body and ability to perform virtual surgeries, this application is useful to medical students and doctors in their trainings. Another interesting example is an AR application that showcases the chemical reactions in an interesting way that facilitate its understanding and memorization. Also, its usage for geography to introduce countries and their cultures, etc. Additionally, it can turn libraries to be interactive, where books come to life with 3D illustrations and immersive storytelling. Apart from its usage in educational and commercial fields, AR is useful at personal utility term as well, such as using it to do your daily chores such as organization, cleaning, cooking, etc. AR guides you step by step into cooking delicious recipes, turning you from novice into culinary maestro, it can give you indications on how to fold clothes, sets reminders on when to water your plant or where to place it to grow better, etc. The possibilities are endless and as the technology continues to develop, so do the applications, with new innovations emerging every day. Thanks to its potential to mix real world objects with virtual ones, AR is catching the attention of everyone.

In a world where everyone is fascinated by the possibilities of AR, there is s no doubt that the demand for innovative applications will only continue to grow. After all, in a field as dynamic and promising as AR, the lone limit is our imagination. Below are some examples of the industries where AR is already being used:

- **Education:** It is used to create interactive learning experiences, such as virtual field trips, 3D models, Chemistry formula and immersive simulations. For example*: ITCraft: Pop Up Books (see Figure 1.5).*
- **Healthcare:** It is used for medical training, surgical planning, and patient education. For example: AR Anatomy provides a 3D animated anatomical AR systems with diverse pathological conditions [6] *(see Figure 1.6).*
- **Navigation:** It is used to provide turn-by-turn directions, highlighting points of interest and providing information about the surrounding area. For example: Mercedes-Benz Augmented Reality Navigation [7] and Google Maps *(see Figure 1.7)*.
- **Real Estate:** It is used to create virtual tours of properties, allowing customers to explore a property without physically visiting it. For example: realar [8] *(see Figure 1.8).*
- **Entertainment:** It is used to create immersive experiences for movies, TV shows, and video games. For example: AR concerts: immersive music experiences *(see Figure 1.9).*

Figure 1. 5. Educational AR app *[9]*

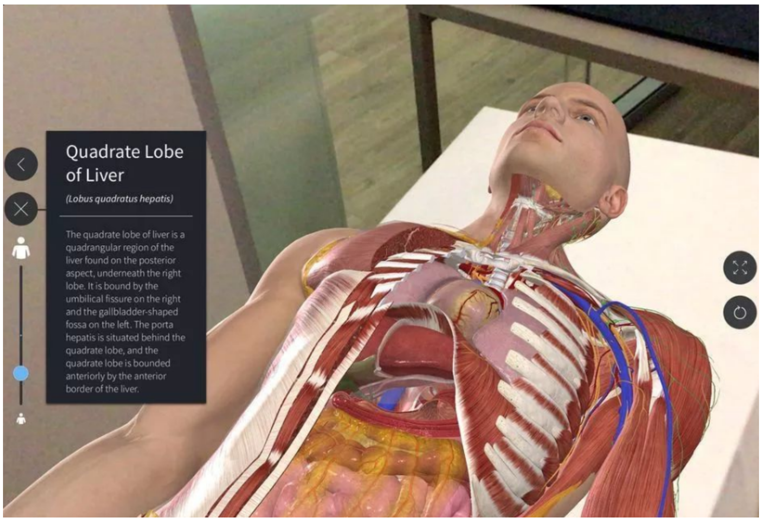

Figure 1. 6. Health care AR app *[10]*

Figure 1. 7. Google Maps AR feature *[11]*

Figure 1. 8. Example of real estate app *[12]*

Figure 1. 9. Pokémon-go app *[13]*

1.4 Virtual reality vs Mixed Reality vs Augmented Reality

As augmented reality, virtual reality and mixed reality applications are getting more popular and widely commercialized, people are consuming their application without necessarily knowing the difference between these three technologies, often mistaking them as one. Indeed, it may be confusing to differentiate between these three terms from a consumer point of view. But, from technical point of view, it is fundamental to do so, therefore, in this section, we explain these technologies and illustrate their use cases with examples to enable you to distinguish between them.

Starting by ***augmented reality*** which indicates that we are extending real things with virtual one. To formally define it, it is overlaying digitally created content on top of the real world [14]. It enables the user to interact with both the physical world and digital elements. Some of AR examples in real life are:

- **Pokémon-go:** It is an AR mobile game, developed and published by Niantic in collaboration with The Pokémon Company and Nintendo for Android and iOS devices. It was revolutionary game that made the headlines during its release time with 232 million curious players recorded that year. The game idea is to collect Pokémons hidden in real locations, players who installed the game in their phone, use the camera from the game app to localize Pokémon and catch it first. [15]
- **IKEA Place app:** an augmented reality application that lets people confidently visualize home accessories addition to your home and do interior designs [16].
- **Google maps**: it is including a reality mode in which arrows are added to guide you to your destination. After selecting this mode, you only have to direct your phone to one of the key landmarks surrounding your such as a marketplace, a museum, or a gym, then the application would recognize your location and starts drawing arrows on road that you can see them virtually from your app indicating where you should go and when to make turns until you reach your destination.
- **Google lens**: Google lens initially is an application that allows you to search google by images, it evolved later to allow object recognition and now it even includes augmented reality part. The AR is used from phone applications (iOS, Android) to do real time translations of boards, signs, etc. it combines artificial intelligence to recognize the texts from images and translate them and AR to display it real time to users [17].

In other hand, ***virtual reality (VR)*** is the experience conducted in a completely virtual environment. VR headsets are used to create a simulated environment and helps individual to immerse into it to experience an entirely different reality. Since it is completely virtual, the user is isolated from real world, and VR cannot be used unless the user has the headset or VR glasses as these are considered the entry keys to this virtual world. The VR is famous in game industry which creates a virtual world for gamers with virtual characters, strong story, music, and attractive environment design. Gamers like to be fully engaged in the games they play, and dislike being distracted, thus, VR, is the magical solution that let them immerse in their games and fully enjoy it. It is also used in education field to provide knowledge in an appealing manner to students. Few examples of VR products oculus quest Nintendo, Sony PlayStation VR, Lenovo Mirage (see Figure 1.10). Some of VR applications in real life are:

- **classVR:** it is a new concept in education sector which uses VR to raise the engagement and knowledge retention for students. A VR headset is used, occupied with user-friendly interface, gesture control and embedded educational resources.

- **Meta Horizon Worlds:** Meta Horizon Worlds is an online virtual reality game developed and published by Meta Platforms. On this multi-player virtual platform, players move and interact with each other in various worlds that host events, games, and social activities.

Figure 1. 10. VR products

Lastly, ***Mixed reality*** merges the user's physical environment and digitally created content, where both environments can coexist and interact with each other [14]. It mixes between the virtual reality and the augmented reality. To use the mixed reality applications, the use of headset or glasses is mandatory, yet, instead of isolating you from the real world like VR, it extends your virtual world with your real surroundings, for example, you may see your virtual characters of your favorite game roaming your room and sitting on your rolling chair. The best example of it in real life is Microsoft HoloLens, an untethered mixed reality headset that's designed to help you solve real business problems today using intelligent apps [18]. Figure 1.11 helps you visually distinguish between these three concepts. Table 1.1 resumes the keys differences between these concepts in terms of environment they function in, the nature of interactions you can have while using these apps, the transparency level to real world, level of immersion of users, type of applications and popular applications.

Table 1. 1. AR vs MR vs VR key differentiating features summarized.

Feature	**Augmented Reality**	**Mixed Reality**	**Virtual Reality**
Environment	Real-world environment	Real-world and virtual environment blend	Fully immersive virtual environment
Interaction	Interaction with both real and virtual objects	Interaction with both real and virtual objects	Interaction limited to virtual environment
Transparency	Partially transparent, overlays digital content onto real world	Virtual objects appear anchored in real world	Fully immersive, no view of real world
Immersion	Less immersive than VR	Varies from partial to full immersion	Fully immersive
Applications	Education, training, gaming, navigation	Training, design, gaming, entertainment	Gaming, simulations, virtual experiences
Examples	Pokémon Go, Snapchat filters, Heads-up displays (HUDs)	Microsoft HoloLens, Magic Leap, Google Glass	Oculus Rift, HTC Vive, PlayStation VR

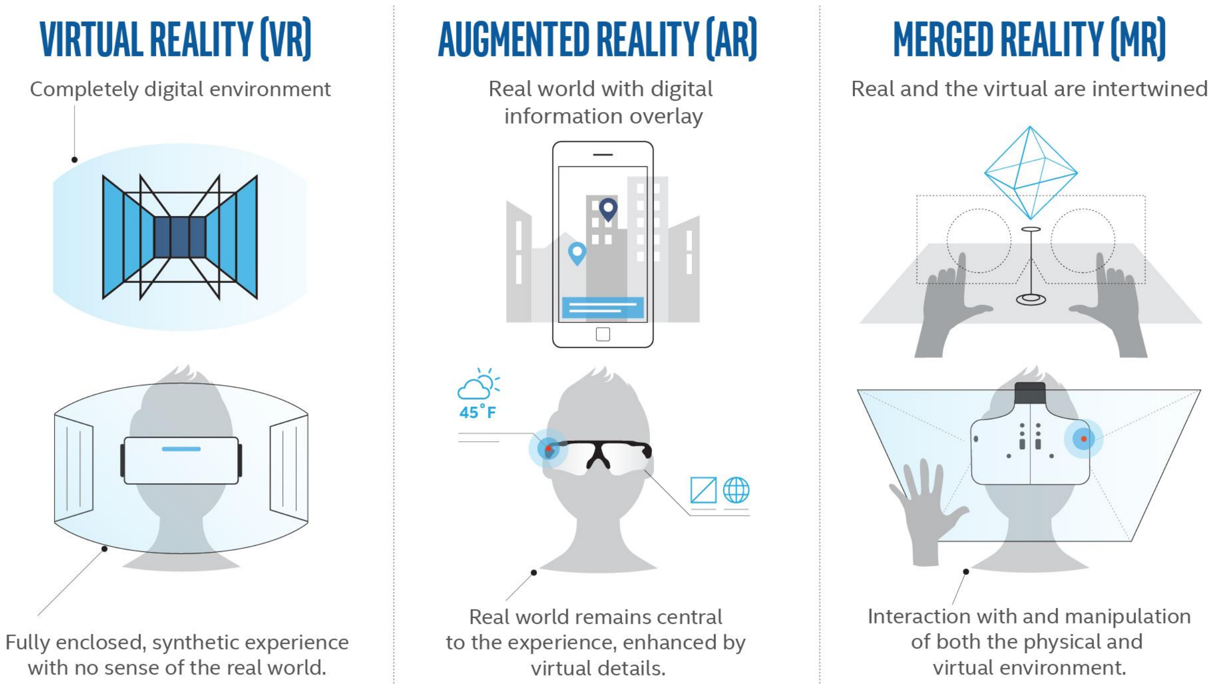

Figure 1. 11. MR vs AR vs VR *[19]*

1.5 Future trends

The future of AR promises different possibilities and trends that are to define its evolving and adoption. Some of the key future trends of AR are:

- **Widespread across fields of industry:** One of the expected things about AR is that it will be integrated into various sectors beyond what is already existing.
- **Suitable wearable devices:** more advanced AR wearable devices are to be created, such as smart glasses and AR headsets as we recently noticed the release of Apple Vision Pro which is heavy although it gives great experience, lighter versions are anticipated to be released.
- **3D mapping and spatial computing**: advancements in spatial computing and 3D mapping technology are expected with better engagement with the real world. Resulting in more realistic and precise AR experiences with accurate object detection and interactive physical surface.
- **AR cloud**: one of AR challenges is the complexity of crafting and disseminating AR content, beyond the known constraints such as sluggish mobile data speeds and device capabilities. AR cloud aims to make AR more accessible to a broader audience [20]. AR Cloud is a virtual replica of the real world that provides developers with a foundation to make AR experiences without starting from scratch. For example, while walking on the street, you can use your smartphone camera to observe your environment. An AR

application like Google Lens could then employ digital overlays to facilitate targeted searches, display information about nearby businesses, and offer crowdsourced insights like restaurant reviews [20].

- **Social AR Experiences:** AR may widely be used to enrich social communications. Social media platforms and messaging apps may include AR features, allowing users to share AR stickers, filters, and realistic-looking avatars in their conversations and posts which enables more natural AR experiences.
- **Integration with Artificial Intelligence (AI) and Internet of things (IoT):** AR applications could be empowered by AI and IoT to enhance their capabilities. Through AI algorithms, AR experiences will become more intelligent and context-aware, while IoT devices provide real-time data to enrich the user's interaction with its environment.
- **AR in Autonomous Vehicles:** AR interfaces may elevate the driving experience by offering navigation guidance and displaying real-time information about the road condition, potential congestion, and ideal shortcuts.

CHAPTER 02
Steganography

2.1 Introduction

Steganography which means hiding messages within ordinary objects, is an old new concept with a history stretching back to ancient civilizations. It has been, evolving with technology emergence by changing its techniques and cover objects. Nowadays, in our digital smart era, governed by advanced technologies and artificial intelligence applications, steganography finds itself at the threshold of new possibilities. Throughout history, steganography has been employed between parties to communicate covertly. From ancient Greece to the Roman Empire, genius methods were invented using everyday objects, preparing the foundation for today's digital steganographic techniques.

Fast forward to the modern age, steganography uses digital files such as image, audio, video files, and even text documents as carriers for hidden messages in form of digital files offering fertile ground for covert communication. Nonetheless, it is with the evolution of AI that opened new horizons for steganography. Not only did it offer more possibilities for improving the concealment of information, with algorithms refining the embedding process to ensure higher embedded data and securer covertness. But also, it made more sophisticated steganalysis tools, capable of detecting even the well-hidden messages.

In this chapter, we provide an in-depth exploration of steganography, tracing its historical roots, examining its modern mediums, studying its use cases and applications, and shedding light on the integration of AI for steganography.

2.2 Steganography History

Steganography which means "covered writing" is originally Greek term composed of **stegano** which means hidden and **graphy** standing for writing [21]. Although it has the same purpose as cryptography which is to transmit messages secretly, it differs from it by the fact it hides the existence of the message and sends it through implicit means, while the latter sends explicitly but in an unreadable manner. To explain it with an easier example, suppose that you are using a paper and pen to send a secret message to your friend, if you hide the message in a particular drawing or write it with invisible ink, you have used steganography to covertly exchange it. But, if we suppose that you and your colleague have priorly agreed on a certain code to use and when you use that coding mechanism to encrypt your message hi into (7H), that is cryptography. your other colleagues may see the code but will not be able to decode the message, while in steganography, they will not even notice the message. In history, this was known as the prisoner’s problem where two prisoners want to plan their escape from prison despite being monitored by the guardian. They used a secret method to exchange messages to remain undetectable so that potential monitors do not suspect that they are planning their escape [22].

In ancient times, the Greek historian Herodotus described how ancient Greeks used to shave slaves' heads, tattoo messages onto their scalps and send them as messengers when their hair grew back, effectively hiding the message until its intended recipient revealed it (see Figure 2.1). In addition to tattooing messages onto shaved heads, another technique of steganography consisted of scratching messages onto wooden tablets and then waxing them. The recipient, upon receiving the tablet, could then remove away the wax to reveal the hidden message underneath. This method provided a discreet means of communication, as the message remained secret until intentionally uncovered by the recipient. The Romans employed similar methods to.

The Chinese during the Yuan dynasty found a clever way to resist Mongolian rule through steganography. The leaders of that time were unhappy at being obliged to submit to foreign rules, created a way to coordinating the rebellion without arousing suspicion. Knowing that the moon festival was approaching, they ordered the making of special cake called “moon cakes” hidden inside each cake was a message with the outline of the attack. On the night of the festival, the cakes were distributed, the messages received, and the rebels successfully attacked the Mongolian government. What followed was the establishment of the Ming dynasty. Nowadays, moon cakes are eaten to commemorate this legend [23]. The use of these varied techniques demonstrates how ancient civilizations were creative, genius and full of resources in developing secret communication methods for various purposes, including military strategies, espionage, and diplomatic correspondence.

Figure 2. 1. tattooing messages onto the scalps of messengers *[24]*

During World War II, steganography was a crucial tool in hiding sensitive information. Various techniques were employed to hide messages from prying eyes. Microdots, for example, reduced documents to tiny dots barely visible to the naked eye yet containing valuable intelligence when magnified (see Figure 2.2). Invisible inks provided another means of covert exchange of information, using substances that can only be revealed under specific conditions. Even innocuous items like knitting patterns served as a way for conveying vital information to operatives in the field.

These stories from history emphasize the desire for secrecy and the quest for knowledge are timeless pursuits, woven into the fabric of human history. From the streets of Rome to the fields of battle, steganography was not just a means of communication, it was a weapon, wielded by those who understood the power of information in a world where every whisper could change the course of history.

Fast forward to the modern era, where the digital revolution has transformed communication. Steganography has transitioned into the digital world, finding new applications in the age of technology. Digital steganography involves hiding messages within digital media formats such as images, audio files, and video files. In addition to its traditional roles in espionage and military communications, it now serves diverse other purposes. For example, it is used in digital watermarking to protect intellectual property and enforce copyright laws.

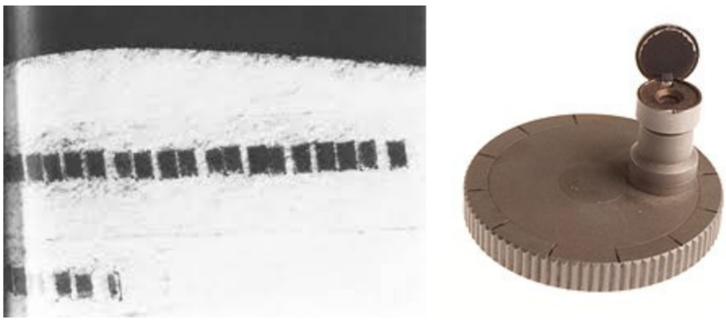

Figure 2. 2. Microdots technique *[25]*

2.3 Modern Steganography

Nowadays, steganography has taken on new forms. Sending an ordinary photo to a friend may have slyer purposed then innocently exchanging updates. A Photo may embed secret message within it that could contain any information. Similarly, trending songs that is listened by everyone may include more than just a melody and lyrics, while you and others are obsessed by its rhythms, it may carry a hidden message that only intended parties know about and can extract. Even text documents or network traffic can be used to conceal messages in plain sight.

Social media platforms are also a place where steganography can be used. Picture sharing sites, comments sections and even hashtags can be used as covert channels for communication. Cryptocurrencies, with their decentralized nature, offer yet another avenue for steganographic communication. Within the blockchain transactions are hidden messages, exchanged without the need for a central authority.

Videos offer a larger data capacity compared to images or audio files, making them suitable for hiding larger messages. Steganographic techniques for videos include concealing data within frames or modifying specific aspects of the video encoding process. Even text documents can be used by employing techniques like whitespace manipulation, font manipulation, or modifying the spacing between letters and words.

Network traffic can be a medium for steganographic communication. Network steganography involves embedding secret messages within network packets or protocols. Imagine sending an email or browsing the web, unaware that inside the packets of data traversing the network lies a hidden message. One method involves modifying the timing or size of network packets to encode information. Another technique is payload-based steganography, where data is embedded within the payload of network packets. By altering the contents of packets or inserting additional data, a covert message can be transmitted alongside legitimate network traffic.

These methods demonstrate how even the most normal activities, such as sending an email or browsing a website, can become means for covert communication. Thanks to the ubiquitous digital communication, network steganography offers a powerful tool for embedding information in real time.

2.4 Steganography use cases and applications

Steganography applications extend across different fields where secrecy and security are essential. In the world of espionage, where every word can be a matter of life or death, steganography provides invisibility for these words. A picture or an innocent-looking audio file, each can carry hidden secrets embedded within its pixels or sound waves. These covert messages, undetectable to the eye/ear, hold crucial information that could change the course of history. It enables military and intelligence agencies to operate with stealth and precision in the digital warfare. Industrial espionage is no exception, here steganography is used to protect the industrial secrets such as the designs, procedures, or systems.

Digital watermarking is another application of steganography serving to protecting intellectual rights like how artists sign their works. Watermark can be visible on the work like having logo imprinted in images, videos or adding the artist's name within the audios. It can also be invisible, like digital fingerprint, uniquely identifying the work that can be detected using specific tools, this is similar to how copyrights infringement is detected online when you post an audio file without buying its royalties. The website you are uploading the file to, can automatically detect the presence of copyrighted content.

Being frightened by the fact that criminals can use steganography to covertly exchange messages and/or to hide their cyber and physical activities/ crimes. Steganalysis tools are developed to be employed by forensic investigators to uncover evidence of wrongdoing and bring criminals to justice. By analyzing the digital fingerprints embedded within digital files, investigators can disclose the secrets put within, revealing criminal activities and illicit operations.

steganography can be used to hide metadata within digital files. Metadata is the silent companion that provides context and information about the content it comes with. It can contain geotagging information (coordinates that pinpoint the exact location where the photo was taken), it is like adding a digital GPS tag to your memories. Similarly, in documents and electronic files, it can embed timestamps, authorship details, or other data to provide context and traceability, it is like leaving a digital footprint, marking each file with a timestamp that records when it was created or modified, or attributing authorship to the person responsible for its creation.

Steganography could be used jointly with cryptography. Cryptography encodes the message, and steganography cover it within innocent-looking digital objects. By combining cryptography with steganography, we create a double layer of security that makes it exponentially more challenging for adversaries to intercept or decrypt the message. It is a powerful strategy that ensures your sensitive information remains safe and secure, even in the face of determined adversaries [26].

2.5 Image and video steganography techniques

Image and video steganography techniques involve hiding secret information within images or video files. These techniques use the characteristics of visual media to cover data in a way that is undetectable by the human eye (see Figure 2.3). Some of these techniques are:

- **Least Significant Bit (LSB) substitution** is one of the most used techniques in image steganography due to its simplicity. It belongs to spatial domain techniques that consists of operating directly on the pixel values of the image to embed the secret message without affecting its visual physical properties. Since the alterations are made to the least significant bits, they are imperceptible to the human eye, thus, the resulting stego image appears visually identical to the original cover image. Yet, the embedded message remains encoded within the image data, accessible only to those with the knowledge of the substitution pattern and the ability to extract the embedded information. It is noteworthy to mention that this technique is not resilient to image processing such as compression, applying filters and/or cropping the image, because these processing would delete the embedded data. An illustration of LSB insertion is given in Figure 2.4 [27]. The algorithm is illustrated in Table 2.1.

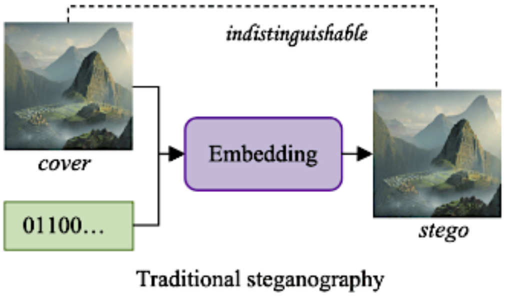

Figure 2. 3. Traditional Steganography *[28]*

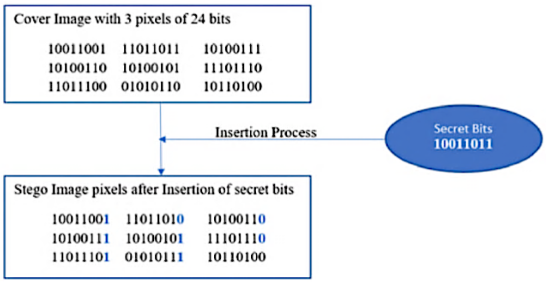

Figure 2. 4. Primary LSB insertion *[27]*.

Table 2. 1. Steganography by LSB *[29]*

The hiding algorithm

Input: the cover image (C) and a secret message (m) to hide.
Output: stego image (S)
for i = 1 to l(c) do
$S_i \leftarrow C_i$ /* Creation of the stego file*/
end for
for i = 1 to l(m) do
calculate the index j_i where to save the i^{th} bit of the message
$Sj_i \leftarrow Cj_i \rightleftharpoons mi$
end for

The extraction algorithm

for i = 1 to l(m) do
calculate the index ji where the i^{th} bit of the message is saved
$m_i \leftarrow LSB(Sj_i)$
end for

- **Palette-based Techniques** manipulates the images to embed hidden information. This could involve rearranging the colors in the palette, altering their shades and saturation levels, or even modifying the color indices themselves to encode the secret message. A color palette-based image steganography algorithm for the iterated Julia-set fractal image is proposed. A color palette is created and sorted based on the illumination values of the pixels. The palette is scanned for embedding the secret bit, and the palette index is modified based on the secret bit's value [30]. The results of image embedding process are illustrated in (Figure 2.5).

- **Transform and wavelet domain techniques:** “Transform domain” techniques have an advantage over “spatial domain” techniques because they hide information in areas that are least exposed to image compression, cropping and processing which are, as we mentioned earlier, the disadvantages of spatial domain techniques such as LSB. Some transform domain techniques work independently of the image format, and they can outperform lossless and lossy format conversions [27].

  Discrete cosines transform (DCT) is one of domain transform techniques, changes the spatial domain image to the frequency domain. It separates the image into spectral sub-bands while respecting its visual quality, i.e. high, medium and low frequency components. In DCT-based steganography techniques, DCT coefficients of the carrier image are obtained. The secret data is inserted into the carrier image for DCT coefficients lower than a certain threshold value (see Table 2.2). To avoid any visual distortion, the integration of secret data is avoided for DCT coefficient values equal to 0 [31].

Figure 2. 5. Results of image embedding process ((a) cover image (b) secret image (c) stego-image) [18].

Table 2. 2. Steganography in DCT [29]

The hiding algorithm
Step 1: Read the cover image.
Step 2: Read the secret message and convert it to binary.
Step 3: The cover image is divided into blocks of 8 × 8 pixels.
Step 4: Scanning the image from left to right, top to bottom, subtract 128 from each block of pixels.
Step 5: Apply DCT to each block.
Step 6: Each block is compressed by the quantization matrix.
Step 7: Read the LSB of each DC coefficient and replace it with a bit of the secret message.
Step 8: Write the stego image.
The extraction algorithm
Step 1: Read the stego image.
Step 2: The stego image is divided into 8 × 8 blocks of pixels.
Step 3: Swiping from left to right, top to bottom subtract 128 in each block of pixels.
Step 4: Apply DCT to each block.
Step 5: Each block is compressed by the quantization matrix.
Step 6: Read the LSB of each DCT coefficient.

Another technique of transform domain techniques is discrete wavelet transform (DWT). Wavelets are special functions which (in a form analogous to sine and cosine in Fourier analysis) are used as basic functions to represent signals. The simplest DWT is Haar-DWT.

DWT breaks down the image into four sub-sections bands: low-low (LL), low–high (LH), high-low (HL), and high-high (HH) (see figure 2.6). LL represents the low-frequency signal which is the approximation of the original image. LH and HL are intermediate frequency signals. And HH is a high frequency signal. LH, HL, and HH show changes in information in the image [32]. The LL part contains the most important characteristics. If the information is hidden in this part of the image, it may resist compression and other manipulations. The algorithm is illustrated in table 2.3 [29].

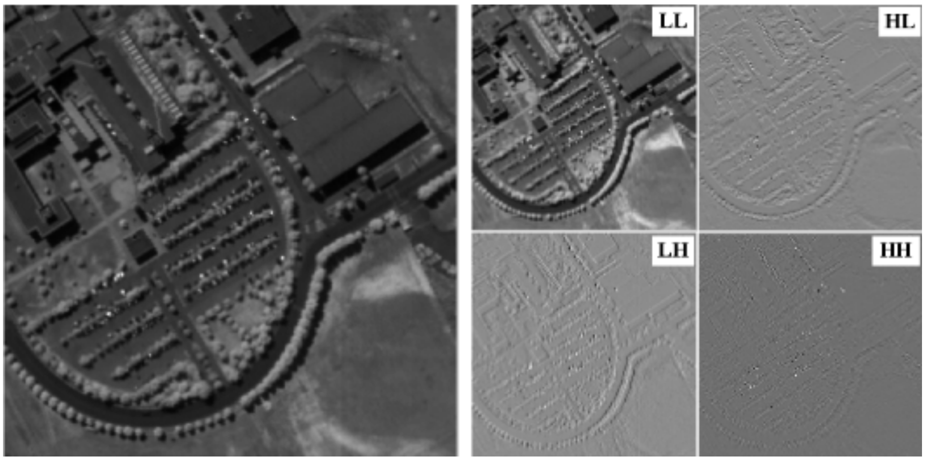

Figure 2. 6. (left side) the original image, cropped from the 100th band of Pavia University data, (right side) The 4 sub-bands, Application of the DWT algorithm *[33]*

Table 2. 3. Steganography in DWT *[29]*

The hiding algorithm
Step 1: Read the cover image.
Step 2: Read the secret message and convert it to binary.
Step 3: Apply DWT and decompose the image into four sub-bands.
Step 4: Insert the secret message by changing the LSBs of each sub-band
Step 5: Generate the stego image.
The extraction algorithm
Step 1: Read the stego image.
Step 2: Calculate the inverse DWT.
Step 3: Retrieve the message.

Let us summarize the information discussed about image steganography in Table 2.4.

Table 2. 4. Comparison between image steganography techniques *[29]*

	Spatial Domain		**Transform Domain**	
	LSB	**Palette-based**	**DCT**	**DWT**
Imperceptibility	high *	High	high	high
Robustness	Weak	High*	high	high
Capacity	high	Weak	Weak	Weak
*: depends on cover image				

We extend the concepts of image steganography to the dynamic video files, where every frame and every pixel can hold a secret data.

- **LSB substitution for videos:** unlike static images, videos offer ability for hiding messages over time. With temporal LSB substitution, the secret message is spread across multiple frames, the hidden information can unfold as the video progresses [34].
- **Motion Vector Modification:** motion vectors are referred as digital fingerprints which comprise of the translation of objects between sequential frames. In the video, every frame holds the continuing moving objects in it. Those motion vectors are hidden in each frame (camouflaged arrows that tell the direction and scale of movement for each element). The altering of such vectors, by properly tweaking the direction and velocity, allows us to code a secret message into a video [35].
- The secret message can be embedded in the sound signal of the video. Voice streams can also be manipulated to hide data by embedding secret message as signals with an amplitude or frequency that is undetectable to human ear, and that blends seamlessly with the original soundtrack.

2.6 Steganography and AI

AI can help in innovating new steganographic methodologies. For instance, researchers thinking about using generative adversarial networks (GANs) to tailor cover images adept at hiding secret data [28] or employing natural language processing (NLP) models to embed covert messages within textual content. Some of the works deployed AI techniques for steganography are mentioned in what follows:

Authors of [36] showed how deep neural networks are used to revisit the regular-singular (RS) method and demonstrates its modernization through neuralization. Incorporating adversarial learning helped the RS method to automatically capture the regularity of natural images, replacing handcrafted discrimination functions. Specifically, GANs are trained to predict bit-planes carrying hidden information. A synthetic image generated by GANs guides data embedding and image recovery. The experimental results indicated significant improvements over normal RS implementations.

Authors of [37] explained a new method named Harris Hawks optimization-integer wavelet transform (HHO-IWT) for covert communication and secure data transmission within the IoT environment, leveraging digital image steganography. This method hides secret data into cover images using the Harris Hawks optimization (HHO) metaheuristic optimization algorithm, efficiently selecting image pixels for concealing secret bits within integer wavelet transforms.

Authors of [38] proposed a novel Cover Image Steganography (CIS) approach based on image selection and Star Generative Adversarial Network (StarGAN) to overcome size limitations of existing CIS methods. The secret information is divided into two parts. First, a traditional CIS method based on image mapping and the scale invariant feature transform (SIFT) feature selects a natural image from a database to the first part of the secret information, while an image mapping is established between the remaining secret information and facial attributes. SIFT algorithm is used to generate robust and distinctive key points and descriptors from images, which can then be hashed to create unique, compact representations for efficient image retrieval, matching, and indexing. In part II, StarGAN generates a high-quality stego image based on the mapping relationship.

Authors of [39] leveraged Deep Convolutional GAN (DCGAN) to embed secret images within other images without altering the image size nor raising suspicious. To do so, they used the DCGAN encoder to hide the secret image within a cover image and used the decoder to extract the secret image from the resulting stego image.

2.7 Steganography and Augmented Reality

The authors of [40] proposed a novel approach to information hiding using AR combined with deep learning. The proposed AR-based architecture employs a secret-key matching policy where secret messages are mapped to objects, images, or coordinates that act as keys. This approach avoids direct transmission of secret messages and enhances flexibility by using deep learning models to extend mapping varieties.

Brandon Sloane recently registered patent [41] about making an AR platform for steganography to ensure secure exchange of secrets. The AR application captures real-time visual feed and hides messages within collected images. These messages are to be read only by authorized persons from authorized devices with privileges to read it. To achieve that, the inventor suggested using device-based authentication, user-based AR authentication system and access control mechanisms. Upon receiving the real-time visual feed, the system verifies if the user's device is authorized to access these messages, then uses AR authentication to identify the user followed by verifying their rights to read the secret. When the authentication and authorization are done, the system decrypts the image to reveal the message.

Authors of [42] investigated the exchange of steganography files in virtual reality environment would be possible and undetectable by forensic analysis. To do so, they used Unreal engine to create a virtual reality environment, then they used existing steganography tools such as OurSecret, OpenPuff and DeepSound to create stego files. Subsequently, they copied the stego files into the virtual environment and packaged the VR app with the stego files. Their experience proved that the use VR apps to exchange stego files would go successfully undetected.

2.8 Conclusion

In conclusion, steganography is timeless art form and science that persists and adapts across civilizations and technological epochs, that makes us understand that the essence of embedding information within innocuous objects remains as relevant as ever. Applications of steganography are multidisciplinary, from military operations to digital forensics, from copyright protection, privacy preservation to secrecy insurance. Steganography's versatility as a tool for covert communication and data security highlights its significance in an era defined by information exchange and digital connectivity. Moreover, the integration of AI and AR introduced new concepts, presenting opportunities for innovation and challenges for detection. In the next chapter, we introduce you to our proposed steganography technique in the era of augmented reality emerging applications which spread with meta and Apple vision. Our proposal mixes AR, AI, and steganography to enable your exchange secret in innovative way in mixed reality futuristic environment.

CHAPTER 03 Backend Development

3.1 Introduction

In the previous chapters, we discussed both AR and steganography concepts. We explored their history, current applications and use cases and how AI can affect each of them and its transformative potential that it brings to these fields. With this foundational knowledge, we are ready to delve into the backend development process of our AR application.

In this chapter, we explain our novel contribution that applies steganography is the third dimension, in the AR environment extending the 2D computer world with the real world. We start, by illustrating the steps involved in setting up the Unity work environment. Next, we depict the dataset preparation. Followed by explaining the training of our AI model using Google Colab, optimizing its performance, and ensuring it is ready for integrating within our desktop AR app. Lastly, we explain the employed encryption and steganography methods.

3.2 Contribution Explanation

With the emergence of AR, VR and XR technologies especially with the spread of AR applications and games, VR headsets and after the revolutionary release of metaverse and apple vision. They opened new horizon for research especially if combined with AI. Most importantly, they elevated questions about the means to secure information in such environment, would the traditional methods be enough? shall they be used as they are, or shall we start to think out of the 2D frame already? Sharing the burden and curiosity of researchers, we thought of a secure way to exchange data by relying on the foundational knowledge that we have related to AI, encryption, and steganography, but this time not in 2D world but in 3D world. This is all what our Stego-AR app is about, harnessing the potential of AR to offer a more secure and dynamic solution by hiding secret text files, within real objects in an innocuous AR app that is equipped with custom trained AI model for object detection. Our Stego-AR app redefines the way the information is concealed.

Our Stego-AR app has two roles, the first role is an AR application that uses the camera and expands the reality by giving descriptions about the objects it detects. This role is given to normal users, the second role is activated to the users aiming to hide and exchange data. It is a secret functionality that our application offers to those concerned about security. The user uses the camera to detect an object, activates the secret functionality to hide a text file in that object, types a password that will be used to encrypt the secret text. The recipient needs to do similar steps in reverse order, they should activate the secret functionality to be able to unhide a message, then, they need to detect the object in which the secret is hidden into, use the correct password and reads the secret text. For extra security measures, the secret text is destroyed as soon as read. We assume that the sender and the receiver agree thorough other means on the object and the passphrase to use, such as over phone. The mechanism upon which they agree on this information is user dependent and is out of the scope of this work. See Figure 3.1 for the process of development of this application, Figure 3.2 for the application building flow and Figure 3.3 for the illustration of the functioning of our application with its two modes.

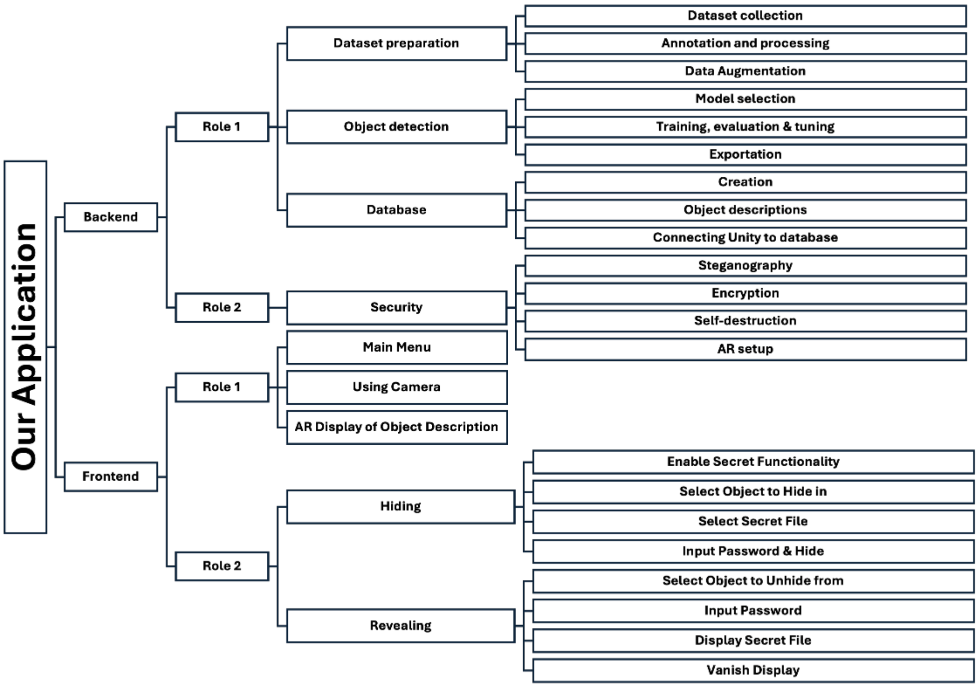

Figure 3. 1 The process of development of Our Stego-AR app

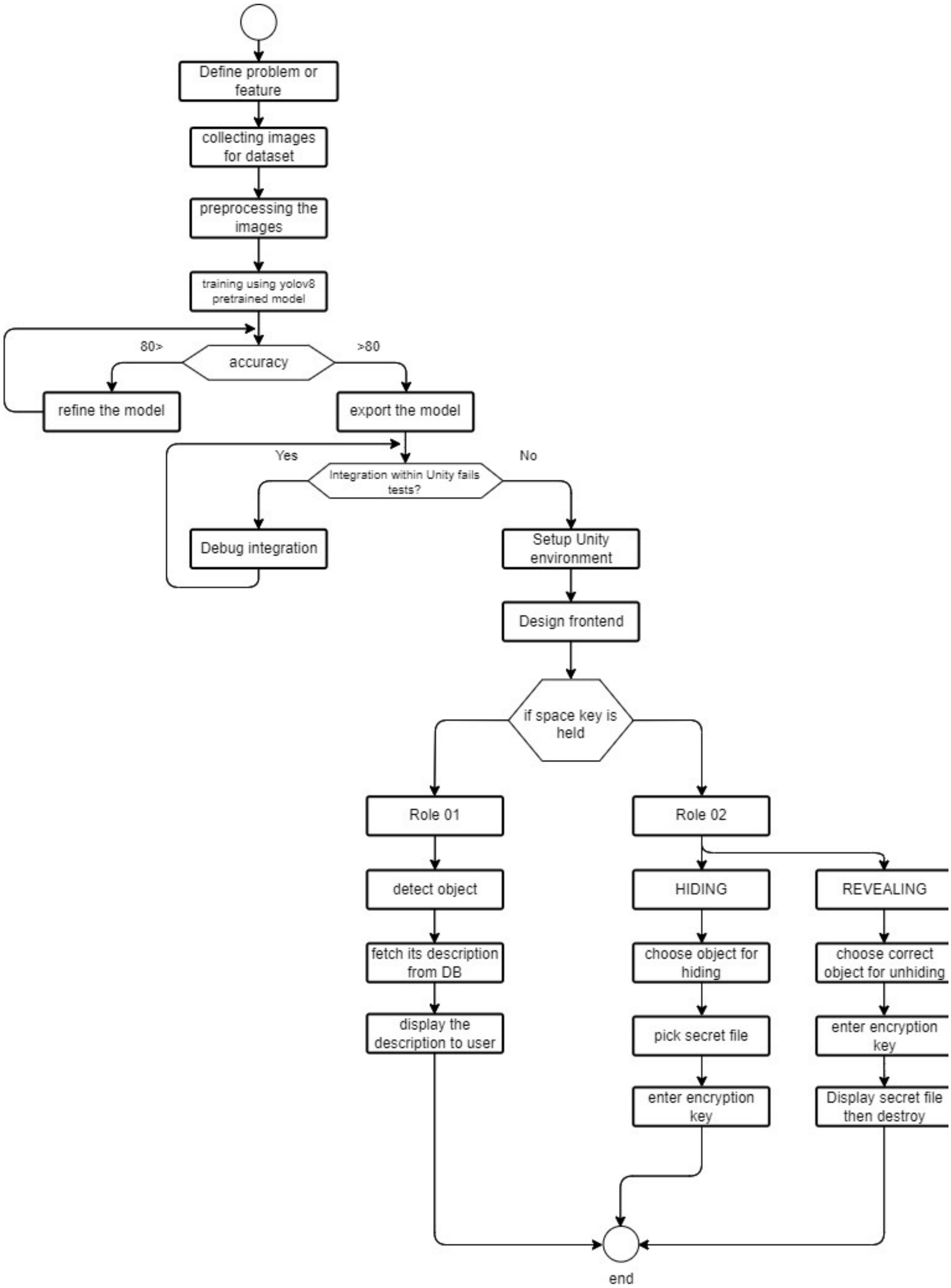

Figure 3. 2 Organigram of application building flow

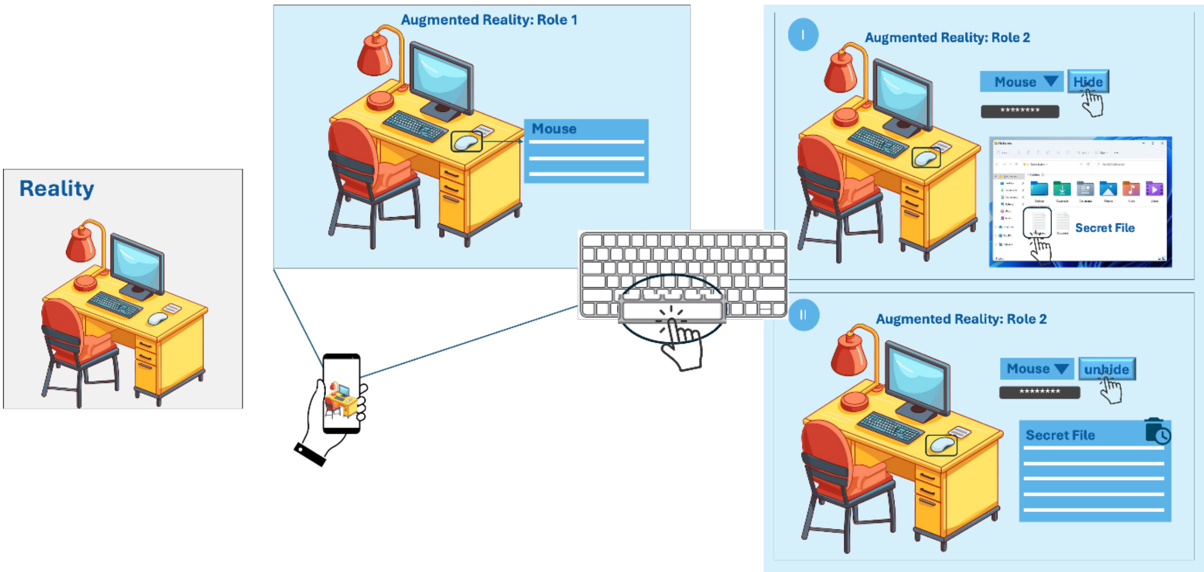

Figure 3. 3 Functioning of Our Stego-AR app

Our application is composed of a backend part which we describe in this chapter and a front-end part which we explain in the next chapter. As we have explained above, we hide in object, as such, the first step was to be able to detect the objects, therefore, the backend of this Stego-AR app used object detection model we shortened that as AI-Part, then our application has to be able to extend the cyber world to the real world, for that we needed to work on augmenting the reality and we call that phase the AR-part. Lastly, our application essential purpose is to hide data, which we further strengthened it using encryption, we call that part the Security-part. In what follows, we explain each of the previously mentioned parts in details. Starting by AR-Part, then AI-Part and lastly, Security-Part.

3.3 AR-Part: Setting Up AR Environment using Unity

Before we see the development process, it is essential to set up the Unity environment to support backend integration. This includes installing necessary plugins which we will be mentioning in coming paragraphs, setting up project dependencies, and configuring the development environment to handle backend services effectively.

Originally, Unity is multi-platform game engine created by Unity Technologies. It was initially announced in June 2005 at Apple Worldwide Developers Conference as a Mac OS X game engine. It supports a wide range of platforms, including desktop, mobile, console, and virtual reality environments [43].

Knowing that we use object detection models from within it later, we made sure to install the latest stable version of unity to ensure its functioning and to avoid compatibility problems with plugins versions we use. Subsequently, we structured our created project as follows:

- **Script folder**: it contains C# scripts that are used to ensure UI interactivity and functionality, and to ensure smooth integration of AI model into Unity.
- **Animation folder:** it contains animation controllers for all buttons.
- **Graphics folder:** it contains UI elements such as background sprite (sprites are 2D graphic objects used for characters, props, projectiles and other elements of 2D gameplay [44]) and application's logo. All of which are used to create the frontend which we demonstrate in the next chapter.
- **SQL folder:** it contains C#-MySQL driver for connection to database.

Following that, we installed the Sentis plugin (beta program) which is a framework for importing and running third-party AI models through the Unity Runtime on user devices [45], in other words, it is a neural network inference library for Unity. The version used in developing our AR application is 1.3.0 pre-release.

We also setup our database in this phase, the database is currently composed of one table that stores the 10 selected objects and their corresponding brief description. It is hosted locally using XAMPP server and interrogated directly from our Stego-AR app using SQL queries from within specified C# script. We opt for creating a database, and not using JSON or csv files, because we are working currently on the server side and the app will be hosted on the web and user subscription option will be offered which means that there will be more than one table to work with in the future.

3.4 AI-Part: Object detection

In this part, we explain the model we selected for the object detection part, the dataset collection and preparation phase, followed by model training phase, then the last phase related to saving it, importing it, and using it from Unity.

3.4.1 YoloV8 model

YOLO (You Only Look Once) is a family of compound-scaled object detection, image segmentation and vision AI tasks models launched by Ultralytics and trained on multiple datasets such as the COCO2017 dataset. It exports to ONNX, CoreML and TFLite [46]. The

8th version of YOLO has a higher mean Average Precision (mAP) and faster inference time than previous versions. YOLOv8 has five pretrained models (n, s, m, l, x). For our experimentation, we used YOLOv8s for its lightness and fast inference, it is worth mentioning that as we write this thesis, YOLOv9 is released, but we did not try it, yet. YOLOv8n (nano) is the smallest model variant, designed for extremely lightweight applications where computational resources are minimal, and inference speed is critical. YOLOv8s (small) offers a balance between lightness and performance. It is faster than the larger models, making it suitable for real-time applications while still providing a higher accuracy than the 'nano' variant. YOLOv8m (medium) offers a further improved accuracy over the 'small' variant, with a moderate increase in model size and computational requirements. YOLOv8l (large) is designed for scenarios where higher accuracy is crucial, even if it comes at the cost of increased computational resources and slower inference times compared to the smaller models. YOLOv8x (extra-large) is the largest and most powerful variant, offering the highest accuracy among all YOLOv8 models. It requires significant computational power and is typically used in scenarios where maximum detection performance is paramount, and resource constraints are less of a concern. The total number of layers in all YOLOv8 variants is estimated to exceed 250 layers [48].

YOLOv8 model architecture is shown in Figure 3.4. It consists of backbone, neck, and head. The backbone is composed of the Cross Stage Partial (CSP) architecture, which splits the feature map into two parts with convolution operations and concatenation. Additionally, it incorporates the C2f module, consisting of 2 ConvModules (Conv-BN-SiLU) and n DarknetBottleNecks connected through Split and Concat. The model also reduces the number of blocks to 3, 6, 6, 3 across four stages and includes the Spatial Pyramid Pooling - Fast (SPPF) module in Stage 4 to enhance inference speed. The backbone's role is to improve learning ability and reduce computational cost.

In the Neck section, in general, deeper neural networks capture more feature information, enhancing dense prediction quality. However, excessive depth reduces object localization accuracy, and information loss from excessive convolution operations will affect smaller objects. To mitigate this, architectures like Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) are essential for multi-scale feature fusion.

The Head section is composed of a decoupled head, where the classification and detection heads are separated, unlike the coupled head in YOLOv5. The model retains only the classification and regression branches, deleting the objectiveness branch. Additionally, it employs an anchor-free approach, identifying the center of the object and estimating the distance to the bounding box, rather than using numerous anchors to determine offsets. Its role is to enhance object detection accuracy by separating tasks [47].

This enhanced model architecture and optimized algorithms and backbone network contribute to quicker and more efficient object detection. Enabling YOLOv8 to process a higher number of frames per unit of time, making it highly suitable for real-time applications. Additionally, the mean average precision (MAP) scores of YOLOv8 have shown improvement compared to YOLOv7.

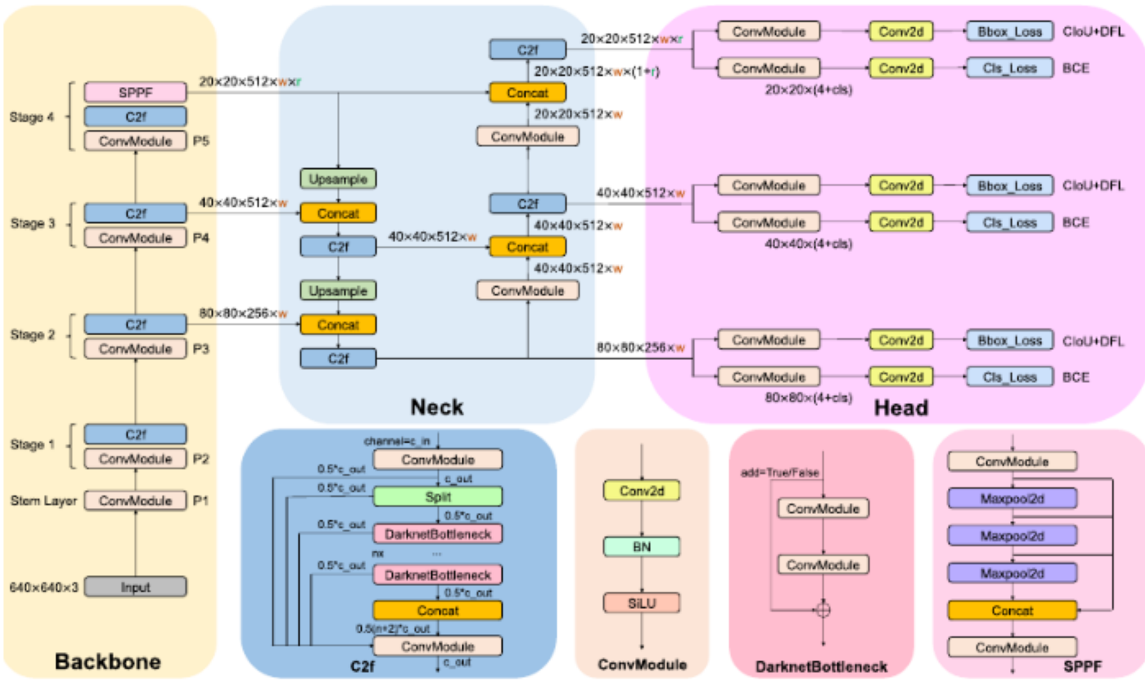

Figure 3. 4 Detailed illustration of YOLOv8 model architecture *[47]*.

3.4.2 Dataset Preparation

The innocuous part of the application is its ability to recognize a list of objects and display their description overlayed on the real scene. That is what normal user will be using the app for. Initially, while we tested the Unity setup, we used Yolov8s pretrained model which was trained on Coco dataset. At this phase, we wanted to confirm that object detection can occur successfully from within unity. In matter of fact, we intended to keep using that model, because its mAP50-95 was acceptable of 44.9 [48] and it recognizes 80 objects. However, upon further observations of the objects it detects which were mainly animals, fruits, vegetable, transportation means, etc. we had to think of retraining YOLOv8s on dataset of our selection. The dataset contained random objects that did not match the scope of the prototype we are making. We wanted an application that describes digital related devices, first because it relates to our field of study, second because these objects can be found in the classrooms where the defense would be held to facilitate the demonstration of our app. These were our motivations; they were made for the sake of this prototype. However, it is important to highlight that in real deployment of the application, the more the objects it recognizes, the better, the more varied, the merrier as it would mean having larger client base. We recapped the object names of Coco dataset by category in Table 3.1.

Table 3. 1. Coco 2017 dataset objects per category

Category	**Objects**
Transportation	Bicycle, motorcycle, car, bus, truck, train, airplane, boat, fire hydrant, stop sign, traffic light, parking meter
Animals	bird, dog, cat, sheep, cow, horse, zebra, giraffe, bear, elephant
Accessories	Tie, Backpack, handbag, suitcase, umbrella,
Sports	Frisbee, kite, skateboard, skis, snowboard, surfboard, tennis racket, baseball bat, baseball glove, sports ball
Food	banana, apple, orange, donut, cake, broccoli, carrot, hot dog, sandwich, pizza
Furniture	Chair, couch, potted plant, bed, dining table, Toilet, sink, Bench
Household items	book, clock, vase, scissors, teddy bear, toothbrush, bottle, wine glass, cup, fork, knife, spoon, bowl
Electronics	TV, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, hair dryer, refrigerator

As mentioned earlier, in this prototype, we decided to select 10 digital devices, which are computer mouse, keyboard, smartwatch, laptop, Wireless earbuds, phone charger, smart phone, video projector, hard disk and USB flash drive. The images of our dataset were self-collected from the internet, we resume the number of images per each object for training, testing, and validating in Table 3.2.

Table 3. 2. Number of images of our collected dataset per object per split

Objects	**Computer mouse**	**Keyboard**	**Smart watch**	**Laptop**	**Wireless earbuds**	**Phone charger**	**Smart phone**	**Video projector**	**Hard disk**	**USB Flash drive**
Train	100	304	113	193	369	190	162	199	366	109
Valid	30	84	21	60	113	40	72	47	107	31
Test	14	51	18	20	53	8	32	15	60	11

The next step was to label the data and specify the objects' locations (surround it with a box) also to add its annotation or identity. For this purpose, we used Roboflow, a framework to label, train, and deploy computer vision solutions [49], to annotate the images. Furthermore, since the original dataset had a total of 2992 images with an average of 299 picture per class, we assumed from previous experiences that it was not enough to get accurate detection results, so we decided to apply data augmentation techniques that provide us with the ability to generate additional pictures from our original dataset to extend with it. These images are generated by applying random and realistic transform operations such as image rotation, saturation, zooming, etc. on our original images [50]. The training dataset was extended to 6541 images. Roboflow was also used for this phase, and it automatically labeled the generated data based on the original dataset. Moreover, we used it to unify the size of pictures to 640x640. It offers multiple options to export the dataset, we used the option that allows us to use it directly from Google Drive as illustrated in the appendix. We resume in Table 3.3 the dataset size per class after the data augmentation for the training, validation, and testing splits.

Table 3. 3. Number of images of our collected dataset per object per split after Augmentation

Objects	**Computer mouse**	**Keyboard**	**Smart watch**	**Laptop**	**Wireless earbuds**	**Phone charger**	**Smart phone**	**Video projector**	**Hard disk**	**USB Flash drive**
Train	290	869	237	520	968	324	434	555	1052	405
Valid	30	84	21	60	113	40	72	47	107	31
Test	14	51	18	20	53	8	32	15	60	11

3.4.3 Training with YOLOv8s

For the training, we seeded the dataset composed of 10 classes and 6.5K images and trained using YOLOv8s pretrained model for 350 epochs with one ".yaml" file that contains classes and path to its data as illustrated in the appendix. To use Googl Colab for the training, we had to setup few things first, we need to connect it with our Google Drive using the script as illustrated in the appendix. Then, we installed YOLOv8 dependencies in our project folder as illustrated in the appendix.

3.4.4 Training Results

As we mentioned in section 3.4.3, the training was done using the YOLOv8s pretrained model. Because of Google Colab resource allocation restrictions, the training session was not done continuously on one-go and was divided on multiple rounds. Each round's duration depended on the remaining available time for using the resources, leading to training sessions of varying lengths: 15 epochs, 40 epochs, and sometimes 70 epochs. Each new round used the weight "best.pt" of the previous round as input, this was possible thanks to the *transfer training* technique which was allowed by YOLOv8. Once the training is finished the results are saved in runs/train folder.

The evaluation metrics calculated during the training and validation phases by YOLOv8 are resumed and saved at the end of each training. The metrics include:

- **Recall** which is the ratio between the number of positive samples correctly classified as positive to the total number of positive samples. It measures the model's ability to detect actual positive samples. The higher the recall, the more positive samples detected [51].

$$Recall = \frac{True_{\ positive}}{True_{\ positive} + False_{negative}} \qquad (1)$$

- **Precision** which is calculated as the ratio between the number of positive samples correctly classified to the total number of samples classified as positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive [51].

$$Precision = \frac{True_{positive}}{True_{positive} + False_{positve}} \qquad (2)$$

- **Confusion Matrix** is a tool that evaluates the performance of a classification model by tabulating true positive, true negative, false positive, and false negative predictions. It compares the actual target values with those predicted by the machine learning model [52].
- **Mean Average Precision** (mAP) is the mean of the average precision scores for each class in the dataset. It gives a single number that summarizes the performance of the model across all classes. Referring to Figure 3.5, we can notice that mAP is calculated with the use of these four metrics (confusion matrix, precision, recall and intersection over union) [53].

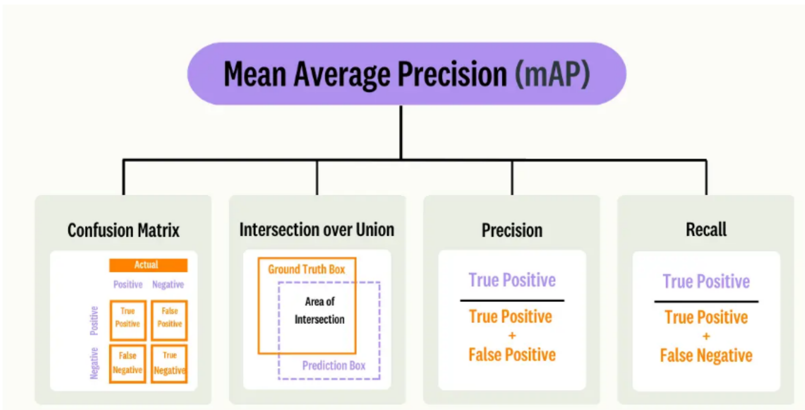

Figure 3. 5. Mean Average Precision (mAP)

From these metrics, we can know whether our model did well in training or not. In our case, we obtained good accuracy from the metrics. The results of the last round of 40 epochs of the training are shown in Figures 3.6, 3.7, 3.8 and 3.9.

The normalized confusion matrix scales the values in the confusion matrix to reflect proportions instead of raw counts. This makes it easier to compare performance across classes. Referring to Figure 3.6 (a), we can see that the diagonal has high values. These values represent the proportion of correct predictions for each class. Higher values indicate better performance for that class. The rest of the matrix is almost clean with low value indicating few misclassifications cases.

The F1 score is the mean of precision and recall, providing a single metric to evaluate the balance between precision and recall. The peak of the F1-confidence curve indicates the confidence threshold that provides the best balance between precision and recall. This is often the threshold we might want to use for classifying detections as positive which is in our case **0.80** (refer to Figure 3.6 (b)). An increased confidence threshold leads to higher precision but lower recall. This means fewer false positives but more false negatives. A decreased confidence threshold leads to higher recall but lower precision. This means fewer false positives but more false negatives.

A high precision is noticed at high confidence thresholds in Figure 3.7 (c) indicating that the model is very accurate when it is most confident about its predictions. This means that false positives are rare when the model is certain.

The curve closer to the top-right corner in Figure 3.7 (d) indicates better performance, with high precision and high recall at **0.82**. A high recall is maintained across a wide range of thresholds in Figure 3.8 indicating a strong model. As shown in Figure 3.9 the mAP of all classes is between **0.77** and **0.82**. The train/box_loss and train/cls_loss measure the error in predicted bounding boxes and classification accuracy, respectively, during training. Similarly, val/box_loss and val/cls_loss provide these metrics for the validation dataset, indicating how well the model generalizes to unseen data. Box losses focus on the alignment of predicted and ground truth bounding boxes, while classification losses address the model's accuracy in identifying object types.

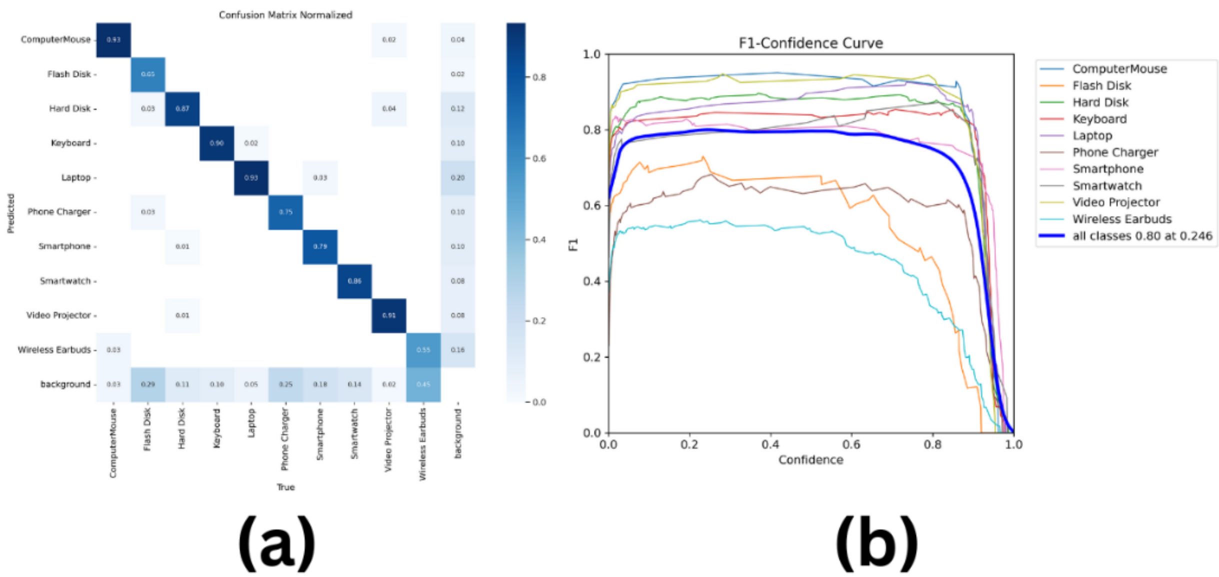

Figure 3. 6. Results: (a)Confusion Matrix Normalized, (b) F1-Confidence Curve

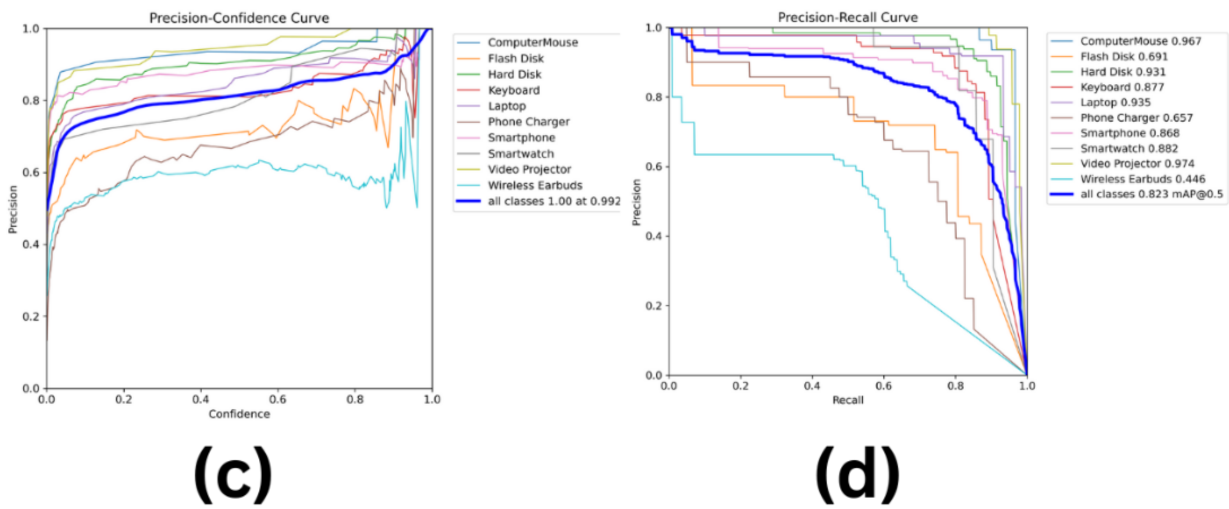

Figure 3. 7. Results: (c) Precision-Confidence Curve, (d) Precision-Recall Curve

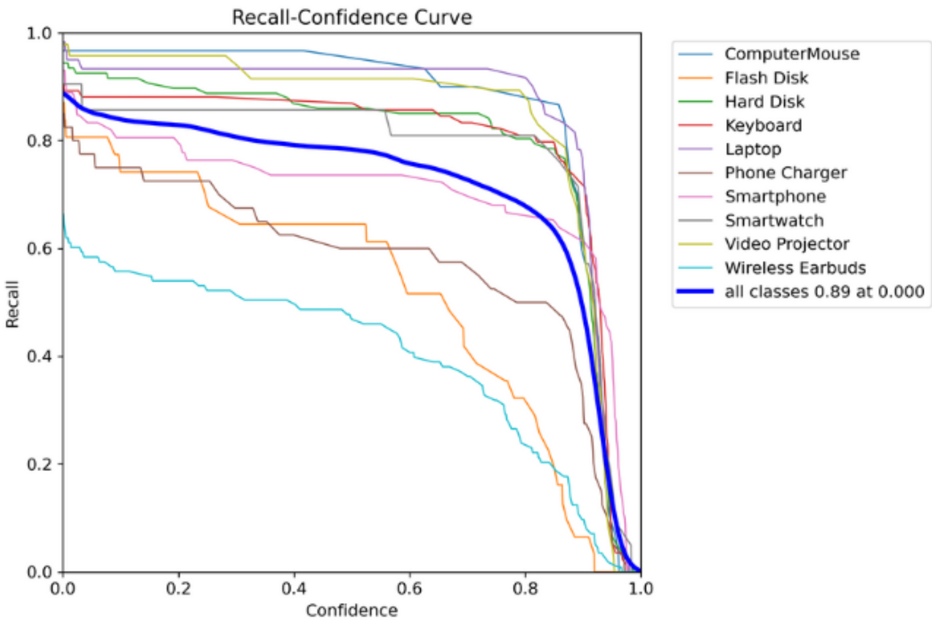

Figure 3. 8. Recall-Confidence Curve

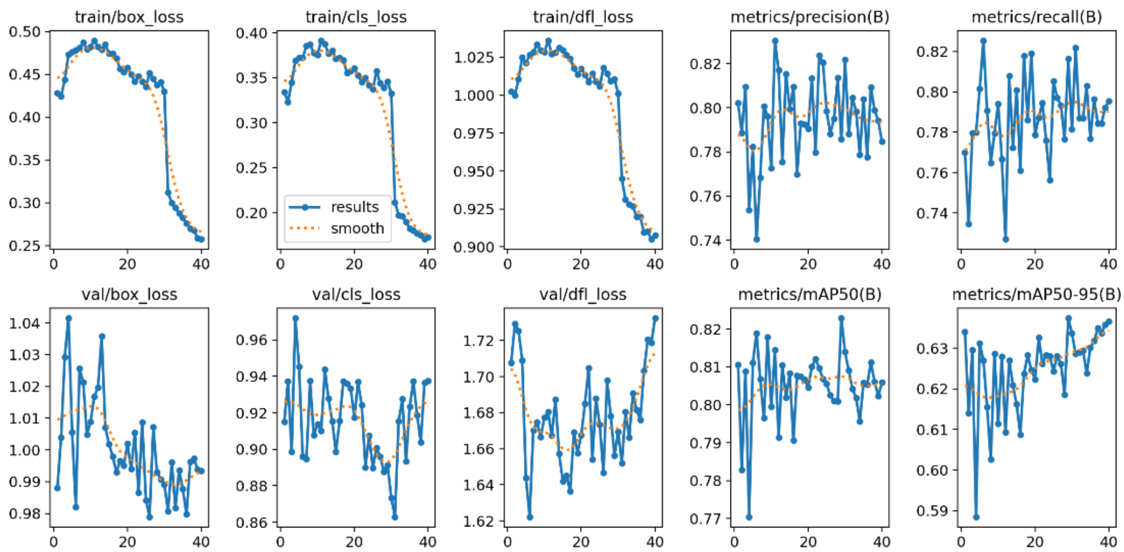

Figure 3. 9. Summarized Results of the last 40 epochs

3.4.5 Using the model in Unity

After we were able to conduct a successful training experience and got satisfied with the model good accuracy (which is around 80%), we had passed the model to our Unity application. We used Sentis framework which is a neural network inference library for Unity. It is in a "pre-release" state in an open beta program available to all Unity users via the package manager in Unity Editor [54]. It accepts Open Neural Network Exchange (ONNX) format, we needed to export our custom saved model in Colab to this ONNX format. Noting that ONNX is an opensource format that aims to provide a common format to ensure interoperability between different deep learning frameworks [55]

To integrate the model in Unity we need to import it in assets folder in our project, then we had to serialize it to “StreamingAssets” using the inspector window (it opens by clicking on the model) and then click on the button "Serialize to StreamingAssets" to obtain a file of “.sentis” format. We had to do the previous step for various reasons such as, firstly, it is validated in Unity, secondly, to save disk space in our project, lastly, for faster loading times [56].

We made a specific C# script file for the real time inference that manages the following functions:

- Opening the webcam.
- Loading the model.
- Setting up the input by capturing video frames at 30 fps and then feeding them to the model for inference.
- Saving the results in a list that composes the elements of the drop-down component whose role will be explained in the next chapter.
- Fetching from the database the detected object’s description.

3.5 Security-Part: Steganography and encryption

We have discussed earlier the backend from the augmented reality aspect which represents the first role of AR app, in this section we are going to see the security aspect of our Stego-AR app. The second role ensures the secret file remains hidden except to intended recipient. To achieve this, we have opted to joint steganography with cryptography techniques for adding extra layer of security.

For steganography, we have chosen the LSB algorithm for several reasons including its simplicity, ease of implementation and its ability to effectively embed secret files within stego-images. The LSB algorithm was implemented in C# making it compatible with our unity project. Currently, the app accepts only “.txt” files as secret files because we currently are interested in small messages being secretly sent in this prototype. Once selected, the secret file is parsed and then embedded into the cover image according to the object selected from the list of objects detected by our custom trained AI model.

To enforce the security, we combined steganography with encryption, the secret file is first encrypted then hidden. In this version, we required the user to input 16 characters password that was used as the encryption/decryption key, in future amelioration, the secret key would be generated from the passwords regardless of its length, thus, we will alleviate the long password condition to facilitate the app usage. For the encryption purpose, we used the Advanced Encryption Standard (AES) algorithm with a key of 128 bits and implemented it in C# because it is a symmetric encryption algorithm known for its security and efficiency. Rijndael is the original name of the AES, it is a cipher that encrypts and decrypts data in fixed-size blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits (see Figure 3.10). First, the plaintext message is divided into fixed-size blocks of 128 bits. Then, a secret key is used to derive a series of unique round keys to enhance security. Each data block goes through multiple rounds of substitution, transposition and mixing using the round keys to transform it into ciphertext. The number of rounds depends on the key length (10 rounds for 128-bit, 12 for 192-bit, 14 for 256-bit). For the decryption, the ciphertext undergoes the inverse transformations using the same secret key to obtain the plaintext [57]. Figure 3.10 depicts the encryption and decryption processes.

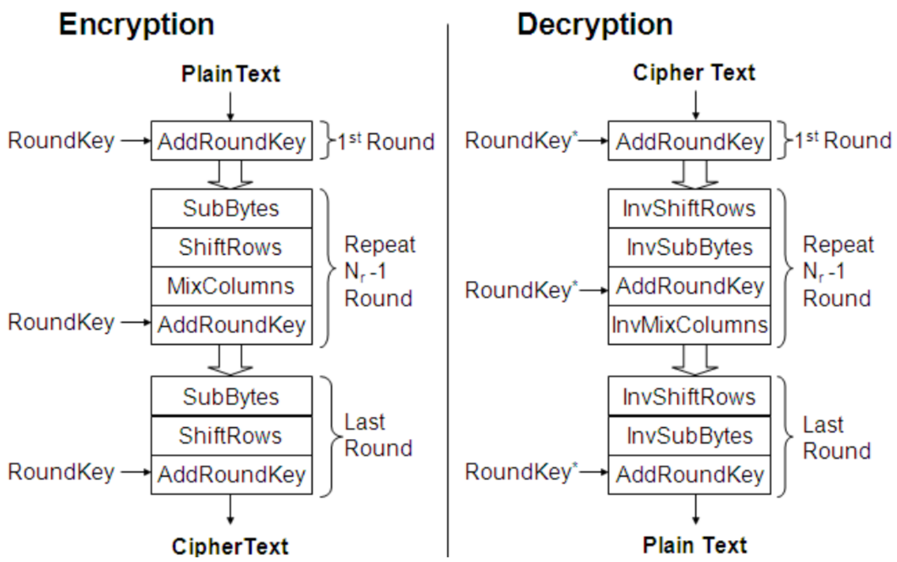

Figure 3. 10. AES encryption/decryption *[58]*

For further security enhancement, the passwords used in encryption decryption are stored hashed. For the password hashing we used SHA256. The SHA-256 algorithm 256 is designed to be a one-way function that processes input data through a series of mathematical operations to create a unique digital fingerprint of the data. This hash value acts as a digital fingerprint, uniquely representing the original data. Making it practically impossible to reverse-engineer the original input from its hash output [59].

3.6 Conclusion

In this chapter, we have explained the setup of the Unity work environment and illustrated how we prepared the dataset, trained the AI model using Google Colab, optimized its performance, and saved it. We have also seen the security techniques used and how they were integrated in our application. In the next chapter, we describe the frontend building process of our application and incorporated our model into it to use it in AR.

Chapter 04 Frontend development

4.1 Introduction

In the previous chapter, we provided a comprehensive overview of the development process for the backend of our application. We explained our contribution, outlined the setup procedure for the Unity environment, and we have seen the dataset preparation and seeding to train our custom AI model based on YOLOv8 pretrained model. We discussed the training results and described the integration process of the model into our unity Stego-AR application. Additionally, we explained the security part of the application and its deployed techniques both steganography and cryptography.

In this chapter, we depict the frontend part of the application. Starting by explaining the views of this application, then explaining how each view was designed and created. Additionally, we illustrate the interfaces used for steganography and encryption techniques.

4.2 Elaborating the Frontend

Before we explain the elaboration process of the frontend, let us explore first the views of our Stego-AR application. The application comprises two scenes, referred to as "main menu" and "AppScene" within Unity. The "main menu" scene serves as a welcoming interface for users, to guide them on how the app is used and showcases its functionalities using AR technology developed with the use of Vuforia SDK. The second scene is core of the application, it includes both roles, role 1 for AR object description and role 2 for Steganography in which we employed object detection, AR, steganography, and encryption techniques. Figure 4.1 illustrate the "main menu" scene, both with and without AR. While Figure 4.2 represents the "AppScene".

Figure 4. 1. The Main Menu scene

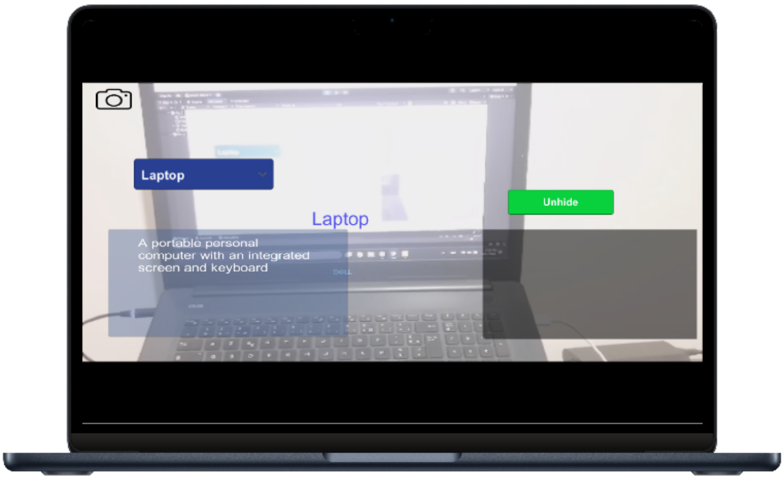

Figure 4. 2. The AppScene view

To create the main menu scene, we used Vuforia from unity. To set up Vuforia with Unity, we first download Vuforia SDK. After that we imported it into our Unity project. Next, we generated a License key from the Vuforia Developer Portal website and added it to our Vuforia Configuration file. Upon doing so, Vuforia elements would be present in unity.

Then, we had to set up the image target from which we extend the application to the real world to add the augmented reality functionalities. To do so, we had to prepare the image in a particular way, first we created as our target image a QR code of our application's name. In the real-world elaboration version, this QR code could be generated from the link of the user's space in our hosted AR app server. Subsequently, we uploaded the QR code image to Vuforia database, then processed it to get its augmentation score. Noting that only if the augmentation score exceeds 3 stars out of 5, that a picture could be used as an image target. Our QR code image received a 5-stars rating (refer to Figure 4.3). It worths mentioning that Image Targets in Unity have certain restrictions. These include:

- Image Targets can be created using JPG or PNG images in RGB or grayscale format.
- The size of the input images must be 2.25 MB or less.
- The images must have a minimum width of 320 pixels

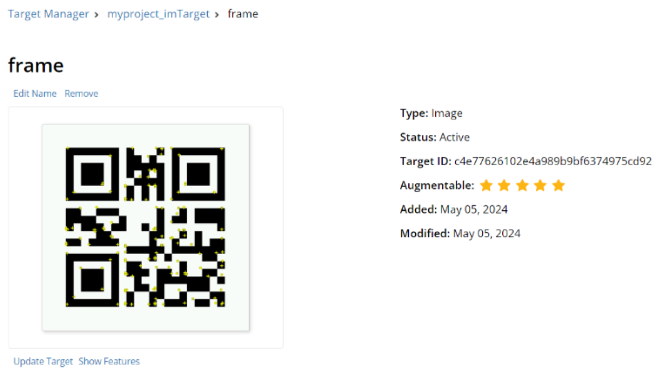

Figure 4. 3. The augmentation score and features of the target in Vuforia database

Then, from Unity project, within the inspector of the ImageTarget GameObject, we selected the imported database that we have downloaded from our workspace in Vuforia Developer Website. This database contains our target image and then choose the QR code target. Once finished, we can create any GameObject as a child of this *ImageTarget* and it gets displayed when detecting the target in real-world scene.

Now that we have created the augmented environment, we can add it elements. As seen in Figure 4.1, our augmented environment contains a virtual start button, a video explaining the functioning of our application, an avatar and information about the developer (me) and its picture (mine).

To create a virtual button, we navigate to the Image Target Behaviour tab within the Image Target. Under the Advanced Options, we select "Add Virtual Button". This action will generate a blue-colored virtual button within the game scene. We adjust the button's position and size to suit the target image. In the Inspector panel, under Virtual Button Behaviour, we can fine-tune sensitivity settings as required (we set it to “High” for better experience). This option facilitates tailored interactions for our augmented reality application. We attached a C# script to the virtual button, ensuring its functionality of loading the "AppScene" once a hand passes over the virtual area. The script, as illustrated in the appendix, contains the necessary logic to trigger the transition to the "AppScene" upon detecting hand movement within the specified region.

To incorporate the illustrative video, we create a Plane game object as a child of ImageTarget. We then attach a video player component to it. Next, we import the video in our assets folder. Finally, we drag and drop the video file into the video player component attached to the Plane GameObject. To ensure that the video sound plays only when the object is detected and not when the app runs, we uncheck the option "Play on Awake" for the video component. We choose to put illustrative video instead of textual guide for two reasons. The first reason is that videos are more effective in conveying app functionality compared to readme files or other forms of documentation. The second reason is that we aimed to show the power of AR technology and inject an element of fun into the video to engage users and prevent monotony. The video is created using an online animation video creator "Animaker [60]". This tool provides you with the ability to create your own cartoon character with multiple gestures, add text and elements and export the video.

For the avatar standing and rotating, we custom-created it to look like me. We used for that an online avatar creator for games called "Ready Player Me [61]". Then, we downloaded Ready Player Me SDK for Unity and import it into our project. After that, we generated an API key on the Ready Player Me website and copy this API key into the avatar loader in Unity. Later, we created a script to make the avatar rotate at regular intervals.

4.2.1. Role 01: AR Object Description

Role 01 focuses on real-time object detection using the custom-trained AI model. We already explained the training part in the backend, now we explore the frontend that uses it for mainly in "AppScene" view. The user points the camera towards an object, if the model recognizes the object, then its description is fetched from the database and displayed in the left part of the screen (refer to Figure 4.4).

Figure 4. 4. Role 01: Detecting object and displaying its description

Role 02

In role 02 depicted in Figure 4.5, which focuses on security, we delve into the frontend design of the "AppScene" and its integration with the backend logic discussed earlier. To enter this mode, the user holds the space button from the keyboard. By doing so, instead of displaying the objects description only, a drop list containing all the detected objects from the captured scene is displayed to the user. The objects detected in real time from the camera captured scene. Upon selecting an object to hide in, two new buttons are displayed to the user, the first is “pick file” the second is “hide”. The “Pick File” allows the user to pick the secret file from his/her internal storage. To enhance the design of the button, we decided to incorporate an animation effect. When the user hovers over the button, it scales up to a larger size, and when the user moves the cursor away, the button returns to its initial size. Upon selecting the file, a password field appears to the user to encrypt the secret file with an indication that the password should be composed of 16 characters. Upon clicking the “Hide” button, the encrypted secret file is hidden.

To reveal the secret file, first the user has to enter the second role (steganography mode), then, selects the same object used to hide the secret. We mean well the object by its type not necessarily the exact same picture of the object. Upon doing so, the "Unhide" button is displayed, this button also incorporates the same design concept and animation as the previous ones. When the unhide button is clicked, the user is asked to provide the password to decrypt the file, the same password used to encrypt the file because as we have explained in the previous chapter we are using AES symmetric key encryption method. After which, the secret message is displayed in clear text in a display area. Each line of the file is displayed using a typing animation effect sequentially. Only one line is visible at a time, and there is no option to revisit the content once displayed, as it vanishes afterward, because the secret file is destroyed once read as explained in the backend section.

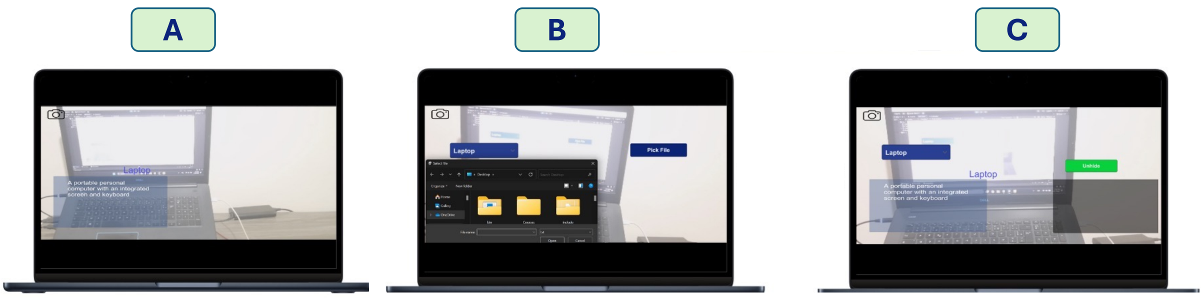

Figure 4. 5. (A) Detected objects added to Drop Down component, **(B)** Picking the secret file, **(C)** Unhide process

4.3 Conclusion

In this chapter, we depicted the frontend of our application, including its roles and its interaction with the backend part. First, we explained the “MainMenu” scene and its AR functionalities that was achieved with the use of Vuforia engine. Additionally, we explored the “AppScene” with its both roles. In role 01, we designed an innocent-looking object detection operation. We created an area for displaying the description of detected objects, providing users with useful information about the objects they scan using the application's AR features. In role 02, we added extra design features to enhance the user experience. These features include a "pick file" button, an "unhide" button, a drop-down component for the detected objects, and an area to display the secret file content upon decrypting.

General Conclusion

General Conclusion

The main objective of our work is to create an application with dual functionality. Firstly, it deploys a custom-trained AI model for detecting a predefined list of 10 objects. This prototype employs AR technology to display the detected objects along with their corresponding descriptions fetched from a database. Secondly, our application offers a covert communication mean by allowing users to conceal and retrieve secret files in augmented reality environment.

To establish our application, we studied augmented reality, its historical background, prevalent techniques, current applications, and future trends. Subsequently, we reviewed information security techniques such as steganography and cryptography. Next, we implemented the AES algorithm for encryption and the LSB algorithm for steganography. After that we created our AI-powered AR-based steganography application, initially conceived as a standard object detection AI and AR app but reinforced with clandestine communication capabilities.

In our future work, we intend to further train the model and refine its parameters in order to optimize its accuracy. Additionally, we aim to improve the mechanism for revealing (extracting) hidden content. Furthermore, we aim to host the application on a server and provide users with subscription feature. This would enable users to share QR codes indicating the logical space where the recipient will be able to access the hidden information. Also, we are aiming to incorporate GANs for a coverless image steganography, enhancing security standards and replace the classical LSB algorithm. Finally, we intend to link our application to generative AI APIs to get automatic detailed description of detected objects.

The Auto Evaluation Grid

Task/objective	**State**	**Details and remarks**
☒: Achieved. ☐: Not achieved		
A historical review of the augmented reality systems.	☒	
Explain different types of extended reality (XR)	☒	
Overview of steganography techniques	☒	
Evolution of modern steganography	☒	
Explaining steganography techniques	☒	
AI-based Steganography techniques	☒	
Using image steganography	☒	In the current version, we used LSB due to its simplicity, a better AI-based coverless solution to be implemented in the future.
Using encryption	☒	In the current version, we used AES for its quick calculation and robustness. In the future, we intent to test public key cryptography as well.
Creating AR application for digital devices descriptions	☒	
Using the AR application for stealth communication of secrets	☒	
Creating multi-user AR application	☐	Not added in the current version due to time limit, to be added in its amelioration.
Hosting the application in server	☐	Not added in the current version due to time limit, to be added in its amelioration.
Using the QR code to exchange secret	☐	Not added in the current version due to time limit, to be added in its amelioration.
Generalizing the AR application to give descriptions to multi-category objects	☐	Not added in the current version due to time limit, to be added in its amelioration.
Successfully and securely exchanging the secrets using Stego-AR	☒	
Performance evaluation: accuracy	☒	
Hiding multi files	☐	Not added in the current version due to time limit, to be added in its amelioration.
Hiding secrets of multi-type	☐	Not added in the current version due to time limit, to be added in its amelioration.
Protecting against reverse engineering.	☐	Not added in the current version due to time limit, to be added in its amelioration.
Using generative AI APIs for automatic detailed object description.	☐	Not added in the current version due to time limit, to be added in its amelioration.

Appen

dix

The dataset structure in order to train the YOLOv8 model:

```
--images
        --train (training images folder .jpg, .png…)
        --valid (validation images folder .jpg, .png…)
--labels
        --train (folder contains .txt files of annotation for train images)
        --valid (folder contains .txt files of annotation for train images)
```

The command used for training:

```
model = YOLO('yolov8s.pt)
results=model.train(data="/content/drive/MyDrive/arapp02/data.yaml",  epochs=350)
```

Importing Dataset from Roboflow to Google Drive:

```
!curl -L "https://app.roboflow.com/ds/mwcBPANHy8?key=Yourkeyhere" > roboflow.zip;
#BENAROUS ABDERRAZAK
!unzip roboflow.zip
!rm roboflow.zip
```

The content of "data.yaml" file:

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 10
names: ['ComputerMouse', 'Flash Disk', 'Hard Disk', 'Keyboard', 'Laptop', 'Phone Charger', 'Smartphone',
'Smartwatch', 'Video Projector', 'Wireless Earbuds']
```

Script used to link Google Colab to Google Drive	Installing YOLOv8 dependencies

`#BENAROUS ABDERRAZAK M2`
`from google.colab import drive`
`drive.mount('/content/drive')`
	`#BENAROUS ABDERRAZAK M2`
	`# clone YOLOv8 repository`
	`!git clone https://github.com/ultralytics/ultralytics`
	`!pip install -e ultralytics`

The script attached to the virtual button to trigger the transition to the "AppScene":

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using Vuforia;
using UnityEngine.SceneManagement; // added for SceneManager

public class virtuaButton : MonoBehaviour
{
    VirtualButtonBehaviour vStartButton;
    // Start is called before the first frame update
    void Start()
    {
        vStartButton = GetComponentInChildren<VirtualButtonBehaviour>();
        //there is no onclick not like UI so w need to register the function ourselves
        vStartButton.RegisterOnButtonPressed(onButtonPressed);
    }
    void onButtonPressed(VirtualButtonBehaviour vb)
    {
        SceneManager.LoadScene("AppScene");
    }
}
```

References

[1] Peddie, Jon. *Augmented reality: Where we will all live*. Vol. 349. Cham: Springer, 2017.

[2] Azuma, Ronald T. "A survey of augmented reality." *Presence: teleoperators & virtual environments* 6.4 (1997): 355-385.

[3] Mohring, Mathias, Christian Lessig, and Oliver Bimber. "Video see-through AR on consumer cell-phones." *Third ieee and acm international symposium on mixed and augmented reality*. IEEE, 2004.

[4] [Online]. Available: https://www.engadget.com/2013-08-12-audi-a3-ekurzinfo-ar-app.html. [Accessed 17th March 2024].

[5] Apple, [Online]. Available: https://www.apple.com/newsroom/topics/apple-vision-pro/. [Accessed 17 03 2024].

[6] [Online]. Available: https://catchar.io/augmented-reality-apps-lenses-projects/anatomy. [Accessed 11 04 2024].

[7] [Online]. Available: https://www.mbscottsdale.com/blog/how-to-use-mercedes-benz-augmented-reality-navigation/. [Accessed 17 03 2024].

[8] [Online]. Available: https://www.realar.com/. [Accessed 17 03 2024].

[9] Snehaal Dhruv, "Medium," [Online]. Available: https://medium.com/superfanstudio/ar-across-industries-education-8b80a70686d5. [Accessed 01 05 2024].

[10] "Complete Anatomy," [Online]. Available: https://catchar.io/augmented-reality-apps-lenses-projects/complete-anatomy-2019. [Accessed 01 05 2024].

[11] Frederic Lardinois, 01 10 2020. [Online]. Available: https://techcrunch.com/2020/10/01/google-maps-gets-improved-live-view-ar-directions/. [Accessed 01 05 2024].

[12] [Online]. Available: https://genesisaugmented.com/augmented-reality-real-estate/. [Accessed 01 05 2024].

[13] Kris Naudus, 12 07 2016. [Online]. Available: https://www.engadget.com/2016-07-12-pokemon-go-augmented-reality.html?. [Accessed 01 05 2024].

[14] Unity, [Online]. Available: https://unity.com/how-to/xr-glossary. [Accessed 17 03 2024].

[15] "Pokemon Go Statistics," [Online]. Available: https://www.businessofapps.com/data/pokemon-go-statistics/. [Accessed 13 April 2024].

[16] [Online]. Available: https://www.ikea.com/global/en/newsroom/innovation/ikea-launches-ikea-place-a-new-app-that-allows-people-to-virtually-place-furniture-in-their-home-170912/. [Accessed 09 04 2024].

[17] Abner Li, "Google lens translate camera," 05 12 2022. [Online]. Available: https://9to5google.com/2022/12/05/google-lens-translate-camera/. [Accessed 13 April 2024].

[18] [Online]. Available: https://www.microsoft.com/en-us/hololens. [Accessed 09 04 2024].

[19] [Online]. Available: https://www.toptal.com/designers/ui/augmented-reality-vs-virtual-reality-vs-mixed-reality. [Accessed 13 04 2024].

[20] [Online]. Available: https://www.verizon.com/about/blog/what-is-ar-cloud. [Accessed 03 22 2024].

[21] Sajedi, Hedieh, ed. *Recent advances in Steganography*. BoD–Books on Demand, 2012.

[22] Fridrich, Jessica. *Steganography in digital media: principles, algorithms, and applications*. Cambridge university press, 2010.

[23] Kleiman, Dave. *The official CHFI study guide (exam 312-49): for computer hacking forensic investigator*. Elsevier, 2011.

[24] [Online]. Available: https://medium.com/@.Qubit/what-is-steganography-92c9a611456b. [Accessed 30 04 2024].

[25] [Online]. Available: https://medium.com/@arpitbhayani/internals-of-image-steganography-b0d1d60425bf. [Accessed 30 04 2024].

[26] Dubey, Richa, Apurva Saxena, and Sunita Gond, "An innovative data security techniques using cryptography and steganographic techniques.," *International Journal of Computer Science and Information Technologies,* pp. 2175-2182, 2015.

[27] Korhan Cengiz, Mangesh Manikrao Ghonge, Renjith Ravi, Sabyasachi Pramanik, Multidisciplinary Approach to Modern Digital Steganography, United States: IGI Global, 2021.

[28] Li, Li, et al. "Image Steganography and Style Transformation Based on Generative Adversarial Network." *Mathematics* 12.4 (2024): 615.

[29] BENAROUS Leila, "Etudes comparatives d'outils de stéganographie et d'outils de stéganalyse: Application aux images et aux vidéos," Laghouat, 2015.

[30] Patel, Hetal N., Dipanjali R. Khant, and Darshana Prajapati. "Design of a color palette based image steganography algorithm for fractal images." *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2017.

[31] Hardik Patel, Preeti Dave, "Steganography Technique Based on DCT Coefficients," *International Journal of Engineering Research and Applications,* vol. 2, no. 11, pp. 713-717, 2012.

[32] Jihad Maulana Akbar and De Rosal Ignatius Moses Setiadi, "Joint method using Akamatsu and discrete wavelet transform for image restoration," *Applied Computing and Informatics,* vol. 19, no. 3/4, pp. 226-238, 2023.

[33] Yang, Jingxiang, Yong-Qiang Zhao, Jonathan Cheung-Wai Chan, and Liang Xiao, "A Multi-Scale Wavelet 3D-CNN for Hyperspectral Image Super-Resolution," *Remote Sens,* 2019.

[34] Banupriya, R., J. Deepa, and S. Suganthi. "Video steganography using LSB algorithm for security application." *International Journal of Mechanical Engineering and Technology* 10.1 (2019): 203-211.

[35] Kasra Rezagholipour,Mohammad Eshghi, "Video Steganography Algorithm based on motion vector of moving object," in *Eighth International Conference on Information and Knowledge Technology (IKT)*, Hamedan, Iran, 2016.

[36] Chang, Ching-Chun, "Adversarial Learning for Invertible Steganography," *IEEE Access,* vol. 8, pp. 198425-198435, 2020.

[37] Hassaballah, M., et al. "A novel image steganography method for industrial internet of things security." *IEEE Transactions on Industrial Informatics* 17.11 (2021): 7743-7751.

[38] Chen, Xianyi and Zhang, Zhentian and Qiu, Anqi and Xia, Zhihua and Xiong, Neal N, "Novel Coverless Steganography Method Based on Image Selection and StarGAN," *IEEE Transactions on Network Science and Engineering,* vol. 9, no. 1, pp. 219-230, 2022.

[39] METABIS Youcef, "Sécurité Des Images: Stéganographie et Cryptographie," LAGHOUAT, 2023.

[40] Chuanlong Li, Xingming Sun, Yuqian Li., " Information hiding based on Augmented Reality," *Mathematical Biosciences and Engineering,* vol. 16, no. 5, pp. 4777-4787, 2019.

[41] Sloane, Brandon. "System for implementing steganography-based augmented reality platform." U.S. Patent No. 11,551,426. 10 Jan. 2023.

[42] Stuart Wilson, "Unreal Steganography: Using A VR Application As A Steganography Carrier," Sheffield Hallam University, Forensic Focus, Sheffield, UK, 16th July 2019.

[43] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Unity_(game_engine). [Accessed 19 04 2024].

[44] Unity, [Online]. Available: https://docs.unity3d.com/ScriptReference/Sprite.html#:~:text=Sprites%20are%202D%20graphic%20objects,used%20for%20a%20specific%20Sprite.. [Accessed 21 04 2024].

[45] Unity, "sentis," [Online]. Available: https://unity.com/products/sentis. [Accessed 21 04 2024].

[46] [Online]. Available: https://docs.ultralytics.com/. [Accessed 23 04 2024].

[47] [Online]. Available: https://github.com/ultralytics/ultralytics/issues/8831. [Accessed 18 06 2024].

[48] Ju, Rui-Yang & Cai, Weiming., "Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm.," *Scientific Reports,* vol. 13, pp. 10.1038/s41598-023-47460-7. , 2023.

[49] [Online]. Available: https://docs.ultralytics.com/datasets/detect/coco/#coco-pretrained-models. [Accessed 28 05 2024].

[50] Roboflow, [Online]. Available: https://roboflow.com/. [Accessed 23 04 2024].

[51] Haba, Duc. *Data Augmentation with Python: Enhance deep learning accuracy with data augmentation methods for image, text, audio, and tabular data*. Packt Publishing Ltd, 2023.

[52] BENAROUS Abderrazak, "Talk2me: A Real-Time Sign Language Translator," Laghouat, 2022.

[53] [Online]. Available: https://encord.com/glossary/confusion-matrix/. [Accessed 25 05 2024].

[54] Nisha Arya Ahmed. [Online]. Available: https://kili-technology.com/data-labeling/machine-learning/mean-average-precision-map-a-complete-guide. [Accessed 24 05 2024].

[55] Unity, [Online]. Available: https://docs.unity3d.com/Packages/com.unity.sentis@1.4/manual/index.html. [Accessed 04 05 2024].

[56] [Online]. Available: https://onnx.ai/get-started.html. [Accessed 24 04 2024].

[57] Unity, [Online]. Available: https://docs.unity3d.com/Packages/com.unity.sentis@1.3/manual/serialize-a-model.html. [Accessed 24 04 2024].

[58] "pandasecurity," [Online]. Available: https://www.pandasecurity.com/en/mediacenter/what-is-aes-encryption/. [Accessed 20 05 2024].

[59] Telagarapu, Prabhakar, Birendra Biswal, and Vijaya Santhi Guntuku. "Design and analysis of multimedia communication system." *2011 Third International Conference on Advanced Computing*. IEEE, 2011.

[60] "Javatpoint," [Online]. Available: https://www.javatpoint.com/what-is-sha-256-algorithm. [Accessed 20 05 2024].

[61] Animaker, [Online]. Available: https://www.animaker.com/. [Accessed 10 05 2024].

[62] Ready Player Me, [Online]. Available: https://readyplayer.me/. [Accessed 10 05 2024].

[63] Unity Technologies, [Online]. Available: https://unity.com/unity/features/arfoundation. [Accessed 25 03 2024].

[64] Sharma, Sudhanshi & Kumar, Umesh, "Review of Transform Domain Techniques for Image Steganography.," *International Journal of Science and Research (IJSR),* 2013.

[65] QIN, Hui & Sasao, Tsutomu & Iguchi, Yukihiro., "A design of AES encryption circuit with 128-bit keys using look-up table ring on FPGA," *IEICE Transactions on Information and Systems,* vol. E89D, 2006.