People's Democratic Republic of Algeria.
Ministry of Higher Education and Scientific Research
University of 8 May 1945 - Guelma -
Faculty of Mathematics, Computer Science and Science of Matter
Department of Computer Science

Master Thesis

Speciality: Computer Science

Option: Science and Technology of Information and Communication

Theme :

Enhancing Campus Security: Face Detection and Automatic Recognition System for Guelma University

---

Presented by : *Bourdima Youssef*

Jury members:

— **President** : Dr.Mehenaoui Zohra
— **Supervisor** : Pr.Farou Brahim
— **Examiner** : Pr. Kouahla Mohamed Nadjib

June 2024

Acknowledgments

Alhamdulillah that I was finally able to get here, and that I was able to achieve one of my goals in life. First of all, I would like to thank Allah for giving me the strength, knowledge, ability and opportunity to undertake this research study, persevere and complete it satisfactorily.

I would like to deeply thank Mr. ***Farou Brahim****, my supervisor, for having supervised and directed this work with great scientific rigor, his availability, his advice and the confidence that he gave me to carry out this work.*

I would like to express my deep gratitude to Mr. ***Kouahla Mohamed Nadjib*** *and M.* ***Mehenaoui Zohra*** *for the honor they did me by accepting the responsibility of examining this work and participating in the defense jury.*

I owe a lot to our head of department Dr ***Zineddine Kouahla****, who gave me the benefit of his immense experience and encouraged me to progress in my work and changed my mindset during my years of study in this university. I thank him very much for his help and express my deep gratitude to him.*

I thank all the professors of the computer science department of the University of May 8, 1945 of Guelma, I was lucky to have respected teachers, some of them even gave me more than just information , their advice has been instructive at several stages of my career.

I would like to give a special thanks to my colleagues ***Zied****,* ***Safir*** *For being part of this wonderful journey that lasted 5 years. We did many things every day. We studied and succeeded. We laughed and complained. We ate and drank. We criticized many things inside and outside the university. We watched and analyzed many football matches. We really did many good things together. I think this journey would have been boring and bad without you. We made these five years pass by in the blink of an eye. I will never forget the memories we had.*

I would also like to thank my extended family, my friends, my colleagues at the University of May 8, 1945 Guelma and the entire class of 2024 in the computer science department.

The best for the end ! A big thank you to my family for being by my side throughout my school career, and for all the support and love you have given me since my childhood, may God give you health, happiness , language life and make sure I never let you down.

Dedications

To dear mother, my strength,my hero the eternal support throughout my life and this educational journey, and I am infinitely grateful to have you by my side. God bless you dear mom.

*To my dearest brothers **Wail** and **Fateh** my strength and my joy, those who want to see me always happy and successful, and share all my sorrows and my joys, may God bless you my dear ones.*

*To all my family **Bourdima** and **Hadri**.*

*To my favorite player, **Leo Messi**, who has entertained me a lot since I was little, who celebrates his birthday today and who concluded football on December 18, 2022. I love you, Leo, you are the greatest.*

*To my close friends **Houssem**, **Anis**.*

To my colleagues at the University who I spent a lot of time with. I will never forget the memories of playing "Loup garou" and the moments of laughter and arguments with some of you. How I always exposed you and was the best player in the game. Those were truly wonderful days.

To everyone who has made me happy with a word or an act, I thank you from the bottom of my heart.

الملخص:

نقدم في هذه الأطروحة الأساليب المختلفة لاكتشاف الوجوه واستخراج الميزات، بالإضافة إلى طرق التعرف على الوجه.

تقدم الدراسة نظام التعرف على الوجه في الوقت الحقيقي المصمم لتعزيز الأمن في نقاط الوصول المؤسسية. يمكن للنظام التعرف على الوجوه في الوقت الفعلي حتى في حالة وجود العديد من الوجوه في وقت واحد. ويتم تحقيق ذلك من خلال استخدام رؤية الكمبيوتر المتقدمة وخوارزميات التعلم العميق التي تكتشف الأشخاص أثناء دخولهم إلى المؤسسة. ومن خلال مقارنة الوجوه المحددة بقاعدة بيانات أنشأها أفراد الأمن بالفعل، يمكن للنظام تنبيههم على الفور بشأن أي شخص غير مصرح به أو غير معروف يحاول الدخول. كما يقوم النظام بتسجيل أوقات الدخول والخروج للأفراد المعترف بهم. لتحقيق اكتشاف أفضل للوجه، تركز دراستنا على تقنية اكتشاف الوجه المستندة إلى الشبكة العصبية التلافيفية المتتالية متعددة المهام أولاً،ثم سيتم بعد ذلك إخضاع مخرجات اكتشاف الوجه لاستخراج الميزات باستخدام انسبشن شبكة الرواسب النسخة الاولى لتحديد الهوية. أظهر النظام أداءً جيدًا في المراقبة وتحديد الهوية بدقة عالية في الوقت الفعلي.

كلمات مفتاحية: التعرف على الوجه، اكتشاف الوجه، استخراج الميزات، في الوقت الحقيقي، نظام الحضور الآلي.

Abstract

In this thesis we present the different approaches to face detection and feature extraction, as well as approaches to facial recognition.
The study introduces a real-time facial recognition system designed for enhancing security in institutional access points. The system can recognize faces in real time even when there are many faces at once. This is achieved through the use of advanced computer vision and deep learning algorithms that detect people as they walk into an institution. By comparing the identified faces with a database already established by security personnel, the system can immediately alert them about any unauthorized or unknown person trying to enter. The system also records the entry and exit times of recognized individuals. To achieve better face detection, our study focuses on MTCNN based face detection technique. First, the output of face detection will then be subjected to feature extraction using Inception-ResNet-v1 for identification. the system demonstrated good performance in high accuracy real-time monitoring and identification.

Keywords: Face recognition, Face detection, feature extraction , real-time, Automated Attendance System.

Résumé

Dans ce mémoire, nous présentons les différentes approches de détection de visage et d'extraction de caractéristiques, ainsi que les approches de reconnaissance faciale.
L'étude présente un système de reconnaissance faciale en temps réel conçu pour améliorer la sécurité des points d'accès institutionnels. Le système peut reconnaître les visages en temps réel, même s'il y en a plusieurs à la fois. Ceci est réalisé grâce à l'utilisation d'algorithmes avancés de vision par ordinateur et d'apprentissage profond qui détectent les personnes lorsqu'elles entrent dans une institution. En comparant les visages identifiés avec une base de données déjà établie par le personnel de sécurité, le système peut immédiatement les alerter de toute personne non autorisée ou inconnue tentant d'entrer. Le système enregistre également les heures d'entrée et de sortie des personnes reconnues. Pour obtenir une meilleure détection des visages, notre étude se concentre sur la technique de détection des visages basée sur MTCNN. Tout d'abord, la sortie de la détection de visage sera ensuite soumise à une extraction de caractéristiques à l'aide d'Inception-ResNet-v1 pour l'identification. le système a démontré de bonnes performances en matière de surveillance et d'identification en temps réel de haute précision.

Mots clés: Reconnaissance faciale, détection de visage, extraction de caractéristiques, Système de présence automatisé.

Contents

List of Tables

List of Figures

General Introduction

In recent years, computer vision has emerged as a transformative technology with wide-ranging applications in various fields, including surveillance and security. Computer vision encompasses the automated extraction, analysis, and interpretation of visual information from images or video data. By leveraging advanced algorithms and machine learning techniques, computer vision systems can perform tasks such as object detection, tracking, and recognition with remarkable accuracy and efficiency.

In the context of surveillance systems, computer vision plays a pivotal role in enhancing situational awareness and enabling proactive threat detection. By analyzing live video feeds from surveillance cameras, computer vision algorithms can detect suspicious activities, identify unauthorized individuals, and alert security personnel in real-time. Furthermore, computer vision enables automated monitoring of restricted areas, perimeter surveillance, and crowd management, thereby augmenting the capabilities of traditional security measures.

Facial recognition technology represents a significant advancement in the field of surveillance and security. By leveraging sophisticated algorithms, facial recognition systems can accurately identify individuals based on their unique facial features. This technology has applications in various domains, including law enforcement, border control, and access control.

Universities face a constant challenge in balancing security, convenience, and efficient management of a large student and faculty population. Traditional access control systems using ID cards can be cumbersome and prone to loss or misuse. This project proposes a novel solution using a real-time facial recognition system for university entry and attendance management.

Our objective is to develop a system that leverages advancements in facial recognition technology to create a seamless and secure experience. The system will be equipped to:

— **Identify Students and Faculty in Real-Time:** Upon entering the university, the system will utilize facial recognition to identify authorized individuals (students and faculty) against a pre-enrolled database.

— **Stranger Detection and Alerting:** The system will be able to distinguish between authorized personnel and unknown individuals. In the case of an unrecognized face, the system will trigger an alert to security personnel.

— **Automated Attendance Marking:** The system can be integrated with existing university databases to automatically mark attendance for students and faculty upon identification.

To validate the proposed approach, a facial recognition system was implemented. This dissertation is organized into three main chapters to address our problem:

Chapter 1 presents the technologies used in biometric systems for person identification. It also provides an overview of the techniques for measuring their performance. A critical realm of university campus security is deleved, spotlighting the integration of surveillance systems, notably those employing facial recognition technology.

Chapter 2 explores the evolution of methodologies and techniques employed in face detection and recognition, spanning from traditional approaches to state-of-the-art deep learning-based methods.

Chapter 3 explains the proposed approach which describes the facial recognition. The main objectives, general structure and functional structure are all presented.

Chapter 4 discusses the development and implementation of a real-time facial recognition system and present all the results.

At the end, we will end our research with a general conclusion with proposals that will improve our system in the near future.

Chapter 1

Biometrics and Computer Vision

1.1 Introduction

In this chapter, we explore facial recognition, a widely utilized biometric modality. We begin by defining biometrics and discussing its applications. Key concepts and foundational definitions related to biometrics and its various technologies are examined. These technologies leverage unique individual characteristics that cannot be lost, stolen, or duplicated, unlike traditional methods such as passwords or magnetic cards. Biometric traits are distinct to each person, making it highly unlikely for others to replicate these features. Consequently, biometric technologies are regarded as exceptionally secure. Biometrics is increasingly recognized as a superior alternative to traditional methods, effectively addressing their limitations.

1.2 Biometrics

1.2.1 Definiton

The Greek words "bio" (meaning life) and "metric" (meaning measurement) combine to form the phrase "biometrics." Biometrics, in general, refers to technologies that quantify and examine an individual's distinctive qualities. A person can be identified using one of two conventional methods: either by possession—such as an identification document, a key, a badge, etc.—or by knowledge—such as a password that verifies the identity of the person desiring to enter a location, an account, a computer, etc. Security may be attained by using these identifying techniques. Security can be attained through the usage of these identifying techniques. Both scenarios have certain drawbacks, though: the user may forget their password or let someone else know what it is in the first instance, and the identity document may be misplaced or stolen in the second[BK17]. An alternative to the two earlier identification techniques is biometrics[BK17]. It involves recognizing an individual based on their physical attributes or their behavioural traits, with the benefit that they are:

- Universal :(found in all individuals).
- Unique : (allows one individual to be distinguished from another).
- Permanent : (allows evolution over time).
- Recordable : (collects an individual's traits with that individual's consent).
- Measurable : (allows comparison in the future).

1.2.2 Architecture

A biometric system always consists of two modules at minimum: the recognition module and the learning module. The adaption module is the third (optional) module. The system will gather one or more biometric measurements during the learning process, which will be applied to the creation of an individual model. When recognizing, this reference model will be used as a benchmark. The adaptability moduled allows the model to be reassessed after every use. [PD02]

1. **Learning module:**

   During the learning process, the biometric characteristic is first measured by a sensor; this is referred to as acquisition or capture. This is because the signal contains information that is useless for recognition, and only the relevant parameters are extracted. It should be noted that sensor quality can greatly influence system performance.[PD02]

2. **Recognition module:**

   Similar to the learning phase, the biometric feature is assessed during recognition, and a set of parameters is extracted. The properties of the sensor to be utilized should be as similar as possible to those of the sensor used in the learning phase. It is usually required to apply a number of additional pre-processing steps to limit performance loss if the attributes of the two sensors differ too much. Depending on the system's operating mode, the following stage of recognition is either identification or verification.

   When in identification mode, the user's identity must be surmised by the system. As so, it provides a response to queries like "Who am I? Under this mode, the system makes a comparison between the observed signal and the various models that are present in the database (1:n issue). When we discuss identification, we often presume that the issue is resolved and that every system user has a model stored in the database.

The system must respond to inquiries like "Am I really the person I claim to be?" while in verification mode. When a user suggests an identity to the system, it is the system's responsibility to confirm that the suggested identity belongs to the person. To solve the 1:1 problem, all that is required is to compare the signal with just one model from the database. Since we anticipate that someone who is an impostor and does not have a model in the database could nevertheless want to be recognized, we refer to model checking as an open problem. Thus, identification and verification are two distinct issues. When a database has hundreds or even millions of identities, identification can be a difficult operation, especially if the system is subject to "real-time" limitations. These challenges are comparable to those faced, for instance, by systems that index multimedia documents.[PD02]

3. **Adaptation module:**

   During the learning phase, the biometric system often captures only a few instances of a given attribute, in order to limit user inconvenience. If a user is identified by the recognition module, the parameters extracted from the signal are then used to re-estimate its model. In general, the adaptation rate depends on the recognition module's degree of confidence in the user's identity. Of course, unsupervised adaptation can be problematic in the event of errors on the part of the recognition module. [PD02]

1.2.3 Performance evaluation

Three factors primarily determine an identification system's performance: accuracy, efficiency (or how quickly it operates), and the amount of data that must be kept for each person. We will focus on the first factor in this section. Identification and verification are two distinct modes of operation, as we have seen previously. As a result, they call for various precision standards, which we shall discuss in the next sections [PD02].

1.2.4 Biometric modalities

Biometrics divides into three technical categories: biological analysis, which includes testing on blood, DNA, urine, and other bodily fluids. The second is behavioral analysis, which focuses on a person's signature dynamics, such as how they move or use a keyboard. The last type of study is called morphological analysis, and it covers things like voice, facial traits, fingerprints, hand shapes, ocular vein patterns, and more. [ASS20]

Biological analysis:

DNA: The direct outcome of advancements in molecular biology, DNA fingerprinting is an incredibly accurate means of identification [ASS20]. Existing in the body's cells, it is unique to each person and enables a person to be positively identified from a little piece of skin, a drop of saliva, or a speck of blood.[ASS20]

Behavioral analysis:

Handwriting (signature): Each person writes in a different way. We are able to define an identifying model based on an individual's signature. In numerous nations, the signature serves as a legal or administrative component, either to validate an individual's genuineness or to perplex them in the presence of previously signed documentation.[ASS20]

Morphological (physiological) analysis:

This method relies on identifying specific physiological characteristics that are both permanent and unique to each individual. The iris of the eye, hand form, fingerprints, facial traits, and other characteristics fall within this group.[ASS20]

Face : The most often used biometric is the face. It is still the most widely accepted since it is consistent with how people interact visually. Several facial features, such as the lips, nose, eyes, and cheeks, are taken out of a picture or video and subjected to geometric analysis (distance between various locations, positions, forms, etc.). The issue with this technique is the potential disruptions that could affect the face (make-up, bad lighting, spectacles or a beard, odd expressions, aging, etc.).[ASS20]

1.3 University campus and surveillance systems

1.3.1 University campus

In an era characterized by technological advancement and heightened security concerns, ensuring safety within educational institutions has become a paramount priority. Universities, as hubs of learning and innovation, host diverse populations, including students, faculty staff, and visitors. With this diversity comes the challenge of effectively monitoring and managing campus security.

University campuses are susceptible to various security threats, ranging from unauthorized access and theft to more serious concerns such as violence and terrorism. Traditional security measures, such as security personnel and access control systems, play a crucial role in mitigating these risks. However, the dynamic nature of modern threats necessitates innovative solutions that leverage cutting-edge technology to enhance campus security.

1.3.2 Traditional security problems

In university campuses where facial recognition systems are not utilized, several security and surveillance challenges persist, impacting access control and overall safety. Here are some key issues:

1. **Manual Monitoring:**

   Without facial recognition technology, universities rely on manual monitoring through security cameras, which can be labor-intensive and prone to human error. This manual approach may not provide real-time alerts or efficient identification of security threats, potentially compromising campus safety.[AS20].

2. **Ineffective Access Control:**

   Universities may find it difficult to effectively manage entrance points, monitor access to facilities, and guarantee the security of campus premises in the absence of facial recognition technology. It's possible that manual access control techniques won't be as successful in strengthening security measures overall and preventing unwanted entry.[AS20].

1.3.3 Importance of Camera Placement

Camera placement is absolutely critical for the success of a real-time facial recognition system in a university setting. The main elements influencing the effectiveness of camera facial recognition are listed below: [W1]

Figure 1.1 – The entry point of the University of 8 May 1945 - Guelma -

— **Strategic Locations :** Setting up the facial recognition cameras at choke points, like doorways or confined spaces with lots of people standing around, will usually yield the best results. (e.g. bus stops).

— **Face Image Size:** The quantity of pixels that make up a face image is known as its size.

— **Video Resolution:** Video resolution is the width and height expressed in pixels for a given video.

— **Lighting Conditions:** Uneven or poor lighting can significantly reduce accuracy. Position the camera to avoid glare, shadows, or excessive backlight.

— **Distance to Subject:** The separation between the camera and the interesting subject.

— **Sharpness:** The degree to which pixels do not blur together and edges stay sharp is known as sharpness.

— **Depth of Field:** Depth of Field is the separation between the closest and farthest objects that a camera can simultaneously focus on.

1.4 Computer vision and face recognition

1.4.1 Definition of computer vision

Computer vision is a branch of artificial intelligence that enables computers to extract information from images, videos, and other visual inputs, allowing them to observe and understand the world around them. It involves automating tasks by interpreting and understanding the content of digital images or video streams. . Computer vision encompasses various functionalities beyond image analysis, such as optical character recognition (OCR), facial recognition, and iris recognition, each serving unique purposes[W2]

Facial recognition, a subset of computer vision, involves recognizing or verifying a person's identity by analyzing their facial features in photos, videos, or real-time using facial recognition systems. This technology is widely used in security systems, biometrics, and robotics, among other applications. Implementing computer vision in face detection involves training neural networks to detect human face landmarks and separate faces from other objects in images. This process utilizes techniques like object detection, segmentation, and image tagging to interpret and analyze images for decision-making. The combination of computer vision and facial recognition technologies offers vast potential for automation, data extraction, and transformative advancements in multiple industries[W3].

1.4.2 Challenges in Face Recognition

The field of facial recognition has advanced significantly. Nonetheless, in the field of computer vision, it is a very difficult task that is not at all simple. The accuracy of face recognition is affected by a variety of factors, including age group, lighting, expression variations, and pose variation.[VGKT22].

— Lighting : Another name for the illumination effect is the lighting effect. Different backgrounds and environments, such as day and night, have an impact on illumination. It could result in areas of the region of interest that are overly bright or dark, which would seriously impair facial recognition accuracy. [VGKT22].

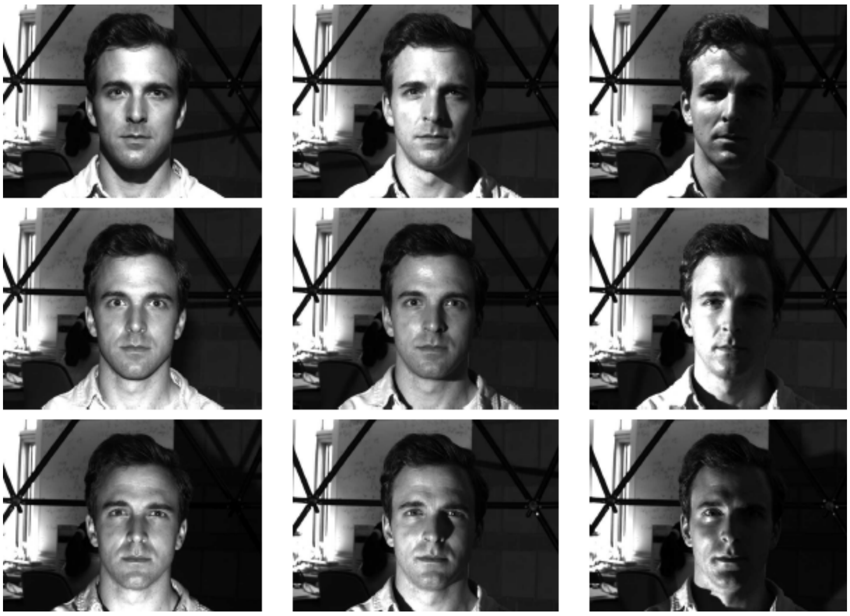

Figure 1.2 – Example of a face of the same person undergoing a change brightness whose angle and azimuth of the source are variable. [Buy11].

— Pose invariant : The accuracy of the face recognition system will be impacted by any slight deviation in head posture. There are three degrees of freedom for the head pose: roll, pitch, and yaw. The head uses these three motions when it moves. As a result, the eyesight and view angle significantly alter the facial pose, which ultimately leads to low accuracy. [VGKT22].

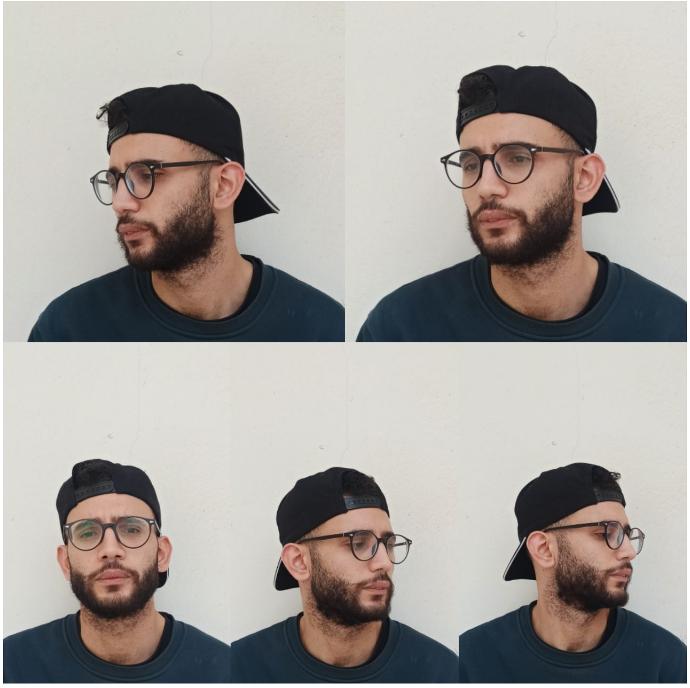

Figure 1.3 – Example of a face of the same person undergoing a change brightness whose angle and azimuth of the source are variable.

— Expression Change : Important features for facial recognition are produced by the internal relationships between the mouth, nose, eyes, and chin, among other facial parts. Therefore, any alteration in facial expression during the registration or verification process could have an immediate impact on the face verification system's functionality. [VGKT22].

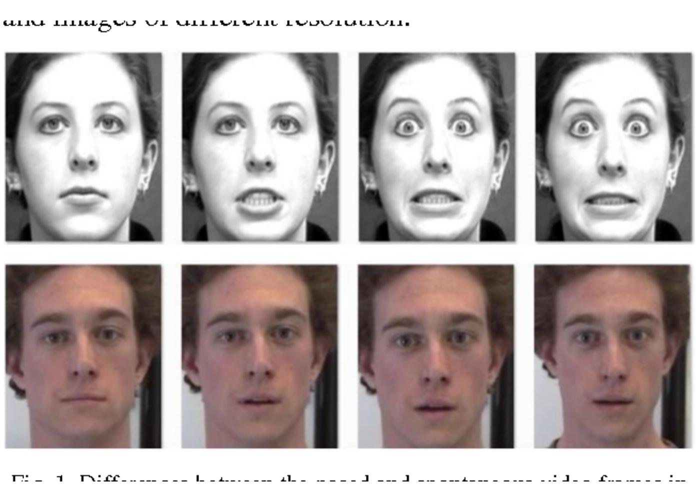

Figure 1.4 – Diffrence of face expressions [NC17].

— Age Variation : Age is a crucial consideration when developing a high-quality facial recognition system. The face is made up of muscles, tissues, and skin. Each person experiences changes in their facial muscles and skin as they age. Therefore, it's critical to regularly update each user's database on time. [VGKT22].

Figure 1.5 – Some of examples faces from the dataset FGNET with real age labels. [ZL18].

1.5 Conclusion

In this chapter, we explored the technologies utilized in biometric systems for personal identification and provided an in-depth examination of the methods for assessing their performance. We delved into the crucial domain of university campus security, emphasizing the integration of surveillance systems, particularly those employing facial recognition technology. Starting with an analysis of the security challenges facing contemporary educational institutions, we evaluated the shortcomings of traditional security measures and highlighted the urgent need for innovative solutions. Our investigation into facial recognition technology revealed its potential as a revolutionary tool for enhancing campus security, offering a proactive approach to threat detection and access control. Despite the complexities associated with its implementation, we are convinced of the effectiveness of facial recognition systems.

Chapter 2

Face Detection and Recognition Methods

2.1 Introduction

In recent years, face detection and recognition have emerged as fundamental tasks in computer vision with widespread applications ranging from security and surveillance to human-computer interaction and digital entertainment. The ability to automatically detect and recognize faces in images and videos has revolutionized various domains, enabling advancements in security systems, personalized user experiences, and forensic analysis, among others.

The aim of this chapter is to clarify the notion of face detection and face recognition, and the various existing techniques are explained. At the end of the chapter, a few works on face detection and recognition are presented.

2.2 Face detection

2.2.1 Definition of Face detection

The majority of face-related applications, including face identification, face tracking, facial emotion recognition, facial landmark detection, and others, begin with face detection. The goal of face detection is to detect faces in photos and provide the bounding boxes that correspond to each face's spatial location.[LJD24]

2.2.2 Importance of Face detection

Prior to using any recognition algorithm, it is necessary to find faces in a scene for biometric systems that employ faces as nonintrusive input modules. In order to react appropriately, an intelligent vision-based user interface should be able to determine the user's attention focus, or what they are looking at. In order to simplify subsequent processing, faces must first be found and registered in order to reliably recognize facial features for applications like digital cosmetics. It is obvious that the success of any face-processing system depends heavily on face detection [Yan15].

The majority of detection algorithms accomplish this task by extracting certain characteristics (such holistic intensity patterns or local features) from a collection of training photos that were taken in an off-line environment while in a fixed stance (like an upright frontal pose). These photos are treated using standardization (i.e., zero mean unit variance) or histogram equalization to lessen the effects of light variation. These systems usually search the full image at every conceivable position and size to find faces based on the retrieved attributes.[Yan15]

2.2.3 Techniques of Face detection

Many algorithms have been proposed to learn their generic templates (e.g., eigenface and statistical distribution) or discriminant classifiers (e.g., neural networks, Fisher linear discriminant, sparse network of Winnows, decision tree, Bayes classifiers, support vector machines, and AdaBoost) because face detection can be primarily formulated as a pattern recognition problem[Yan15]. Face detection has been greatly enhanced since the Viola–Jones (V–J) detector [VJ01] was introduced in 2001. Handcrafted features, such as Haar-like features, have been replaced by end-to-end convolutional neural networks (CNNs) for superior feature extraction. Compared to ten years ago, face identification algorithms have significantly improved in speed and accuracy.[LJD24]

2.2.4 Classification of face detection techniques

Yang et al.[YKA02] divided face detection techniques into four groups. The foundation of knowledge-based approaches is a collection of rules that characterize faces. Certain features are found via feature invariant approaches that remain constant despite changes in viewing point or posture illumination. Template matching techniques make use of many templates that record the face's characteristics or description as a whole. By calculating the correlation between the incoming image and the stored templates, detection is carried out. A model is trained from a collection of training facial photos in appearance-based approaches. For detection, the learnt models are employed. [VP16]

Knowledge-based methods

According to this approach, guidelines generated from the researcher's understanding of human faces are used to construct face detection techniques. Generating elementary rules that characterize facial characteristics and their interrelationships is a straightforward task. For instance, a face with two symmetrical eyes, a nose, and a mouth is frequently seen in a picture. Relative locations and distances between features can be used to depict their connections. An input image's facial characteristics are first retrieved, and face candidates are then found using the coded rules. Usually, a verification procedure is used to lower false positives. [Cha20]

This method has limitations including the challenge of converting human knowledge into precise regulations.

Feature invariant approaches

Unlike the top-down, knowledge-based method, researchers have been attempting to identify face invariant traits for detection. The fundamental premise is founded on the fact that faces and things can be easily recognized by people in a variety of positions and lighting situations, indicating the existence of qualities or traits that are independent of these variations. A plethora of techniques have been put forth to identify facial traits and hence deduce the existence of a face. With edge detectors, facial characteristics including the lips, nose, eyes, eyebrows, and hairline are frequently extracted. A statistical model is constructed based on the retrieved characteristics in order to explain their correlations and confirm the presence of a face.[Cha20]

These feature-based methods have limitations in that lighting, noise, and occlusion may all significantly skew the picture features. Perceptual grouping algorithms become ineffective when faces have weaker feature boundaries and shadows create several strong edges.

Template matching methods

In template matching, a function either manually defines or parameterizes a standard face pattern, which is typically frontal. The facial contour, eyes, nose, and mouth correlation values with the standard patterns are separately calculated for an input image. The correlation values are used to establish if a face exists. One benefit of this strategy is its ease of implementation. However, because it is unable to handle variations in scale, position, and form, it has shown to be insufficient for face identification. To attain size and shape invariance, several subtemplates, including multiresolution and multiscale templates as well as deformable templates, have been suggested. An active robot vision system's face localization has been implemented using an extension of Sinha's technique. The improved template with 23 specified relations is seen in Fig under [Cha20].

Twelve confirming relations (dashed arrows) and eleven fundamental relations (solid arrows) are supplementary categories for these stated ties. In the illustration, each arrow represents a relationship, and its head designates the second area (i.e., the ratio's denominator). A face is localized if the number of crucial and confirming relations surpasses a threshold, and a relation is satisfied for face temple if the ratio between two areas is

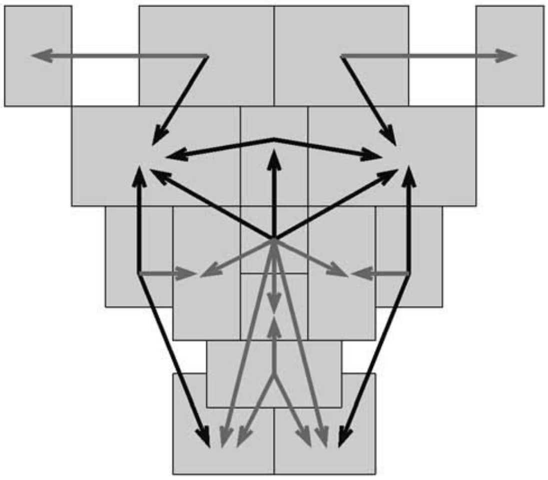

Figure 2.1 – A 14x16 pixel ratio template for face localization based on Sinha method. The template is composed of 16 regions (the gray boxes) and 23 relations (shown by arrows) [Cha20].

greater than a certain value.[Cha20]

Appearance-based methods

Unlike template matching, which involves using a collection of training photographs to capture the representational variability of face appearance, models, or templates, are learnt from the data. Then, these acquired models are applied to detection. The primary purpose of these techniques is face detection.[Cha20]

The "templates" in appearance-based approaches are learnt from instances in photographs, in contrast to template matching methods where templates are established by experts. Appearance-based approaches, for the most part, use machine learning and statistical analysis techniques to identify the pertinent features of both face and nonface photos. The acquired traits take the shape of discriminant functions or distribution models, which are then used to face detection. In the meanwhile, dimensionality reduction is often done to improve detection effectiveness and processing efficiency.[Cha20]

Another approach in appearance-based methods is to find a discriminant function (i.e., decision surface, separating hyperplane, threshold function) between face and nonface classes. Conventionally, image patterns are projected to a lower dimensional space and then a discriminant function is formed (usually based on distance metrics) for classification, or a nonlinear decision surface can be formed using multilayer neural networks. Recently, support vector machines and other kernel methods have been proposed. These methods implicitly project patterns to a higher dimensional space and then form a decision surface between the projected face and nonface patterns. [Cha20]

Eigenfaces Principal component analysis was used by Turk and Pentland for face detection and identification. Principal component analysis, is used on a training set of face images to produce Eigenpictures, often referred to as Eigenfaces, which span the face space, a subspace of the image space. Face images are grouped after being projected into the subspace. Training pictures without faces are similarly projected into the same subspace and grouped. Since the projection of non-face pictures seems substantially different from that of face images, images of faces do not alter significantly when projected onto face space. The distance between each point in the image and the face space is calculated in order to identify whether or not a face is present in the scene.[Cha20]

Neural networks Numerous pattern recognition challenges, including optical character recognition, object recognition, and autonomous robot driving, have seen the successful application of neural networks. Many neural network topologies have been developed for face identification because it may be viewed as a two class pattern recognition problem. The ability to train a system to capture the intricate class conditional density of face patterns is one benefit of utilizing neural networks for face detection. One disadvantage is that in order to achieve remarkable performance, the network architecture must be fine-tuned in many ways (number of layers, number of nodes, learning rates, etc.).[Cha20]

Support vector machine (SVM) Face detection has also been accomplished with SVMs [LSYC13]. SVMs can be used as a new paradigm for neural networks, polynomial functions, and radial basis function (RBF) classifiers. Structural risk minimization, the induction principle that drives SVMs, aims to reduce an upper bound on the predicted generalization error. The separating hyper plane of an SVM classifier, which is a linear classifier, is selected to minimize the predicted classification error of the test patterns that have not been observed. Their method operates around 30 times quicker than the system and has somewhat lower error rates based on two test sets of 10,000,000 test patterns of 19×19 pixels. In the wavelet domain, SVMs have also been utilized to recognize faces and pedestrians.[KKK19]

The AdaBoost-based face detector Faces may be recognized quite reliably in real time (more than 15 frames per second on 320×240 pictures using desktop computers) under partial occlusion, according to Viola and Jones' AdaBoost-based face detector[VJ01] . Although faces and pedestrians were represented using Haar wavelets, they suggested using Haar-like features instead, which may be calculated well with integral images. In order to encode the horizontal, vertical, and diagonal intensity information of face pictures at various positions and scales, the figure illustrates four different forms of Haar-like features. [Yan15]

Unlike most previous methods (such as neural networks and support vector machines) which utilize a single strong classifier, they used an ensemble of weak classifiers, each of which is built by thresholding a single Haar-like feature. The AdaBoost method is used to

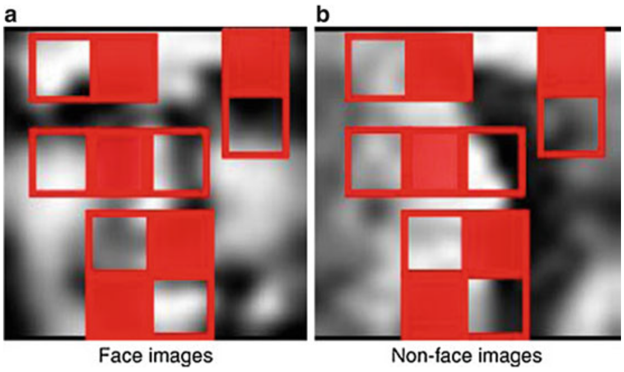

Figure 2.2 – The four different forms of Haar-like feature [Yan15].

choose and weight the weak classifiers. Regression applications and function approximation with gradient descent are two other perspectives from which boosting methods may be derived. Given the abundance of weak classifiers, they proposed a strategy based on a set of optimization criteria to order these classifiers into many cascades. The AdaBoost technique is used to train an ensemble of many weak classifiers inside each step. [Yan15]

The idea behind the cascade of classifiers is that most negative examples may be effectively filtered out by basic classifiers at an early stage, and only occurrences that resemble faces require the use of stronger classifiers at a later stage. containing relatively high detection and low false-positive rates, the final detector—a 38-layer cascade of classifiers containing 6,060 Haar-like features—showed outstanding real-time performance. Since then, a number of enhancements have been proposed to recognize faces in several perspectives with in-plane ration. The Intel OpenCV library has an implementation of the AdaBoost-based face detection. [Yan15]

Advantages and Disadvantages of Face recognition Techniques

Approach	**Advantages**	**Disadvantages**
Knowledge-based Methods	-Leveraging domain-specific information for accuracy - Tailored to specific applications or scenarios.	- Extensive manual effort required. - Limited by knowledge base accuracy
Feature Invariant Approaches	- Robust to pose variations. - Reliable face localization technique.	- Limited to predefined features. - High computational complexity required.

(continued on next page)

Approach	Advantages	Disadvantages
Template Matching Methods	-Simple and intuitive implementation - Direct comparison with predefined templates.	- Susceptible to variations in appearance. - Computationally intensive for large datasets.
Appearance-based Methods	-Statistical analysis for relevant characteristics. - Dimensionality reduction for computation efficiency.	- Sensitivity to image variations. - Requirement for substantial training data.

Table 2.1 – Advantages and Disadvantages of Face recognition Techniques.

Deep learning

CNNs, or Convolutional Neural Networks, represent a powerful form of artificial neural networks tailored for tasks involving grid-based data structures like images. They employ convolution in at least one hidden layer, enabling effective processing of image data. Convolutional layers within CNNs are controlled by various hyper-parameters, such as kernel scale, number of kernels, padding, and stride, which provide flexibility and adaptability to different tasks and datasets. Face detection and recognition in video sequences pose challenges due to varying qualities and complexities, necessitating feature extraction techniques capable of handling diverse scenarios. CNNs have revolutionized image recognition, with models like AlexNet showcasing their efficacy in pattern recognition tasks. These networks comprise layers including input, convolutional, activation function, pooling, and fully-connected layers, enabling automatic feature generation and combination for robust classification. However, convolution operations can be computationally intensive, particularly in extensive networks, impacting training times. Despite this, CNNs remain indispensable tools for image and video analysis, facilitating object detection and tracking in security applications.[SDM23]

The latest advancements in computer vision applications have generated significant interest in this field. We examined several contemporary machine learning methods for face recognition, as well as tracking and identifying individuals in both still images and videos

2.2.5 Benchmark databases

Numerous face detection methods exist, but limited focus has been on developing comprehensive image databases for research. The FERET database comprises monochrome images featuring frontal views and profiles, used for evaluating face recognition approaches.

Method	Authors	Year	Architecture	Training Set	Accuracy (%) ± SE
DeepFace	Taigman et al. [TYRW14]	2014	CNN-9	Facebook (4.4 M, 4 K) *	97.35 ± 0.25
DeepID	Sun et al. [SWT14a]	2014	CNN-9	CelebFaces + [SWT14b] (202 k, 10 k) *	97.45 ± 0.26
DeepID2	Sun et al. [SCWT14]	2014	CNN-9	CelebFaces+ (202 k, 10 k) *	99.15 ± 0.13
DeepID2+	Sun et al. [SWT15]	2014	CNN-9	WDRef [CCW+12] + CelebFaces+ (290 k, 12 k) *	99.47 ± 0.12
DeepID3	Sun et al. [SWT15]	2015	VGGNet	WDRef + CelebFaces+ (290 k, 12 k) *	99.53 ± 0.10
FaceNet	Schroff et al. [SKP15]	2015	GoogleNet	Google (200 M, 8 M) *	99.63 ± 0.09
Web-Scale	Taigman et al. [TYRW15]	2015	CNN-9	Private Database (4.5 M, 55 K) *	98.37

Table 2.2 – A comparison of several deep face verification methods utilizing the Labeled Face in the Wild (LFW) database.

However, due to its uniform background, it's unsuitable for face detection benchmarking. Other databases like Turk and Pentland's, ATT Cambridge Laboratories', Harvard's, Yale's, M2VTS multimodal, UMIST, and Purdue AR offer varied features including head orientation, lighting conditions, facial expressions, and occlusions. These databases serve diverse research needs, facilitating experiments in face recognition under different scenarios. [Cha20]

2.3 Face recognition

One of the key subfields that is vital to biometric identification is facial recognition. A facial recognition system can match a person's face from a picture or a video, search through a database of faces, and attempt to authenticate the user. Applications for facial recognition systems include security and surveillance (i.e., tracking criminals, locating missing children, kinship verification), health care (i.e., patient medication, genetic disease detection), banking and retail (mobile users, KYC, customer verification), and other areas. [VGKT22].

2.3.1 Basics of Face Recognition System

Face recognition is used in many PC vision frameworks to identify images provided by symmetric structures that are commonly used. Applications for perception make use of symmetric images, while identification algorithms identify faces and concentrate on facial images that include facial features. As a result, it is more challenging and unpredictable than a single identification or recognition calculation. [VGKT22] The basic workflow of face recognition system is shown in the figure under

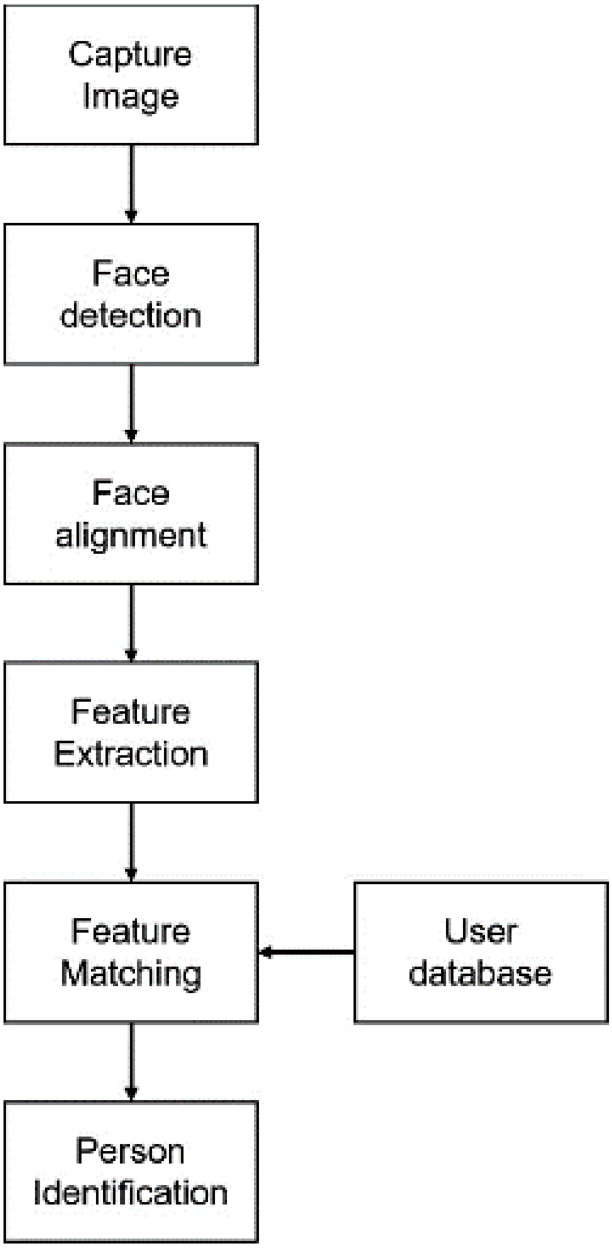

Figure 2.3 – The basic workflow of face recognition system [VGKT22].

1. **Capture image :**

   Taking a picture with the camera is the first step in the face recognition program. When dealing with videos, several frames are taken, and the standard workflow is supplemented with a face tracking stage.

2. **Face detection:**

   Face detection happens in the following stage. It might involve some preprocessing techniques like noise reduction and background extraction, depending on the requirements.

3. **Face alignment:**

   The next action is to align the face. One of the key components of a reliable face recognition system is face alignment. It addresses the precise localization and normalization of facial features, including the mouth, nose, eyes, and eyelids. A higher level of facial recognition accuracy is facilitated by the geometric relationship between the different parts of the face.

4. **Feature extraction:**

   The feature extraction process then begins. The process of collecting different useful facial features or data that can be used to represent a particular user in a database is known as feature extraction. We extract low- and middle-level features for global and local feature-based methods, and high-level and complex features for deep learning-based approaches.

5. **Feature matching:**

   A query face from the user database is matched using this gathered feature information in the feature matching stage.

6. **Person identification:**

   Lastly, we give the query face a user label based on the feature matching outcomes.

2.3.2 Techniques of Face recognition

Since then, several research communities, including those in computer vision, image processing, artificial intelligence, and machine learning, have made face recognition a priority during the past three decades. Techniques for facial recognition come from a wide range of scientific domains, which is why it is challenging to define a precise boundary that would classify these methods in a uniform manner. Furthermore, it is challenging to classify these methods in conventional branches for feature representation or classification due to the use of hybrid models. Nonetheless, we summarize and present face recognition approaches as a high-level, distinct categorization based on recent literature. The categorical distribution of face recognition techniques is displayed in the figure.

Local Approach :

Model-based local face recognition techniques rely on processing distinct facial picture areas independently. A priori knowledge of facial morphology forms the basis of the models employed. This typically involves local facial feature extraction or detection. A method for automatically extracting a set of 35 geometric features from a face image is proposed by Brunelli and Poggio [BP93]. To accomplish recognition, these feature sets are then compared two by two using the Mahalanobis distance. [Bel13]

Yu et al.[YLH$^+$19] introduced a model aimed at local representation-based facial recognition. They utilized aligned downsampling of local binary pattern features extracted from

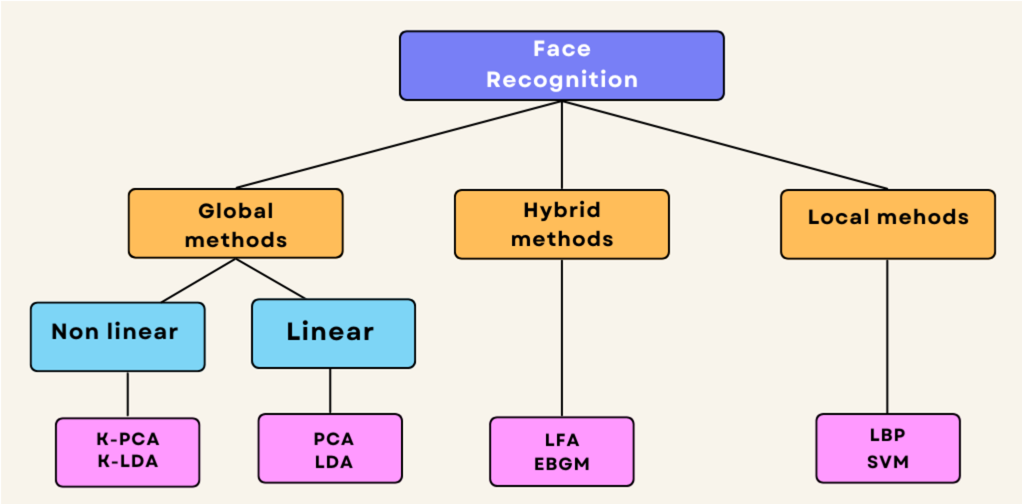

Figure 2.4 – Classification of the main algorithms used in facial recognition.

frontal facial images for classification purposes. Subsequently, recognition tasks were executed using an enhanced robust sparse coding algorithm. An et al.[ADH+19] proposed a facial alignment framework known as Adaptive Pose Alignment (APA), implemented to enhance the efficiency of facial recognition and analysis tasks in data processing. It has been noted that the APA methodology not only diminishes intra-class variation but also effectively eliminates noise introduced during the alignment process. [Mou20]

The great advantage of local face recognition methods is that they can easily model the variations in pose, illumination and expression that a face can undergo. However, they often require the manual placement of numerous points of interest for good accuracy, and are therefore cumbersome to implement.[Bel13]

Global Approach :

Global methods employ well-known statistical analysis techniques on the entire facial picture. Generally speaking, the goal is to project the previously vectorized facial input picture into a lower-dimensional space, where it is thought that recognition will be simpler. Frequently, the projection is made to include only those characteristics that are significant and sufficiently discriminating to allow for individual identification. Global techniques have the benefit of being quick to implement, since their computations rely on very basic matrix operations. But since they view the face as a whole, they are perceptive to position, lighting, and emotion.[Bel13]

Hybrid Approach :

Combining local and global approaches leads to hybrid methodologies. They integrate both global feature extraction and local feature detection. In the end, these strategies aim to take use of the benefits of both kinds of approaches. One example of a hybrid approach to face recognition is the proposed method by Kodinariya, which uses Principal Component Analysis (PCA) and Independent Component Analysis (ICA) with score-based fusion. The results of the experiment demonstrate that the proposed hybrid face recognition technique, which uses a score-based strategy as a combiner/fusion process, is more accurate than each of the independent approaches.[Kod14]

Table 2.3 – Summary of hybrid approach reviews

No.	Method	Database	Performance
1 [PSP23]	Support Vector Machine (SVM), Convolutional Neural Network (CNN), and Principal Component Analysis (PCA)	was not explicitly mentioned	nearly 98%
2 [EU19]	VGG, deep learning	Brazilian FEI and LFW	For a 3/4 portion of the face image, Brazilian FEI database is 100% SVM-Wo, 76% to 99% for SVM-Wo
3 [BCL23]	HOG, SIFT, CNN	ORL database	accuracy rate of up to 100%

2.3.3 Methods of Face recognition

There have been several academics working in the wide field of facial recognition, each with their own unique perspective and set of algorithms. In this section, we go over a few local and global feature-based techniques in this section. We also go over a few new deep learning models that have been developed for facial recognition.

Eigen Faces and Fisher Face Methodology:

It is difficult to compute the lower Hessenberg matrices under the specified conditions. Eigen faces and the Fisher Face Methodology provided a way to determine common eigenvalues of matrices that function on the lower Hessenberg matrix reduction approach. As a result, a companion matrix is proposed in this view to overcome computational problem complexity. A straightforward and effective technique for facial recognition may be suggested, which is based on the companion matrix and common eigenvalues approach. Eigenface techniques are mostly utilized for feature extraction, where PCA is combined with a facial characteristic using the K-Nearest Neighbor algorithm. PCA is used in the implementation of eigenfaces.[SSTS21]

PCA:

The foundation of Principal Component Analysis (PCA) is multi-variant data analysis, which is derived from projection techniques. PCA is employed as a feature extraction technique in face recognition. Prior to any picture analysis, the data collection is preprocessed using this method. It eliminates noise and repetitive information, saves the necessary properties of picture-based data, and, in some situations, allows for the reduction of image dimensions. It also speeds up processing and saves money and time. These days, Statistical PCA is utilized to increase recognition rates and is simple to compute; nonetheless, it depends on linear assumptions and scale variation approaches[SSTS21].

LDA:

This method is comparable to the discriminating analysis used by fishermen. Images, even those with local elements, have been identified using it. The way these characteristics work is as a pixel value. They are divided into three categories: recognized color, texture, and form aspects. The functionality that will be applied to the linear vector separation is defined by it. In the picture, they also employed a comparable element. These techniques have been applied to face identification and recognition in order to maximize both intra-class variance and class dispersion.[NFTT20]

SVM:

Support vector machines (SVMs) are used for various classification problems, much like neural networks. SVM is a significant learning-based technique that works well for creating classifiers for facial recognition issues, as seen the Local feature detector is used to extract the main features. The facial features are used to train an SVM classifier based on the feature attributes. Numerous studies have used this type of hybrid technique, such as binary tree with SVM and ICA with SVM. Before classifying using SVM, a variety of methods, including PCA and LDA, were used to conduct feature extraction and reduction. SVM offers quicker training and requires less computing than ANN. Still, the precision is not very high. [VGKT22]

MLP:

An example of a classifier using a layered neural network topology is the multilayer perceptron (MLP). The input layer, hidden layers, and output layer are the three different kinds of layers that make up an MLP. Each layer of the MLP, as seen, is made up of many SLPs, which, in contrast to the perceptron in Fig , may have several outputs. The number of units in the input layer is equal to the size of the data vector since each perceptron in the input layer has a single input, which corresponds to a single component of the data vector. Every unit in the network has a unique output bias, and in Figure , every link between units has a distinct weight. [NFTT20]

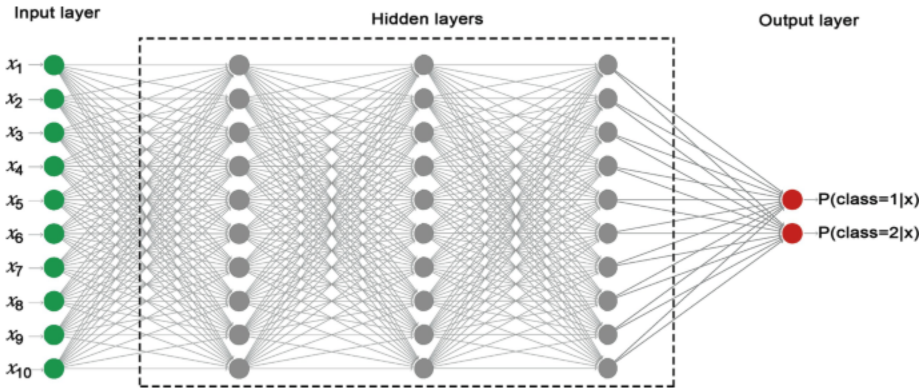

Figure 2.5 – Example of multilayer perceptron (MLP) [NFTT20].

Convolutional Neural Networks (CNN):

Convolutional neural networks (CNNs) are one kind of neural network that works very well with picture data, or more broadly, any sort of data that can be naturally organized in a two-dimensional grid. As seen in Fig, CNNs are composed of units in each layer that act as filters in response to patches in the visual field. The same patch may be covered by a bank of filters, with each filter in the bank extracting unique data. The replication of this bank structure results in a collection of all banks combined that together form a "sliding window" that overlays patches throughout the picture, referred to as receptive fields. Furthermore, it is possible to blend outputs from several patches in the subsequent layer, enabling CNNs to efficiently scan the picture at various resolutions to detect both minute details and larger-scale characteristics. Further details on this process are provided in the sections that follow[NFTT20].

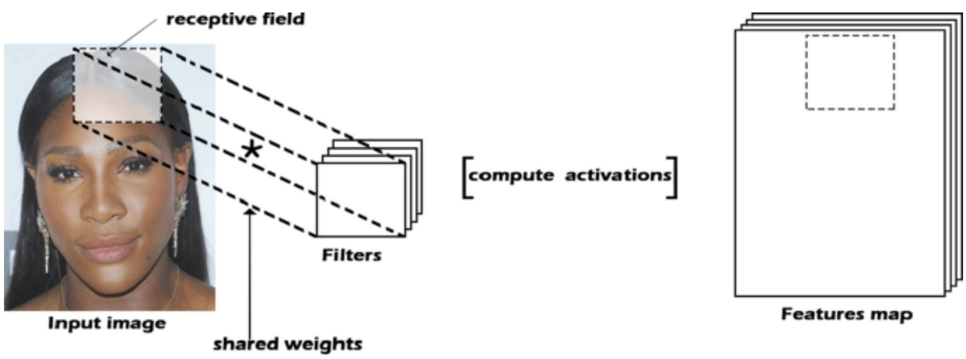

Figure 2.6 – Convolutions of an input image with a bank of four filters, yielding four feature maps. [NFTT20].

Thanks to the benefits of weight sharing, pooling, and local receptive field, CNN performs well on a number of image modification tasks, including rotation, scaling, and translation. The remainder of this section provides a quick introduction of CNN's preliminary for the purpose of completeness.

- **Convolutional process:**

  One type of mathematical procedure that is frequently utilized in image processing is convolution. Convolutional output may be categorized into three modes: Full, Same, and Valid. These modes can be applied in many contexts. For instance, Full mode is frequently used in back propagation to achieve the ideal weights, while Valid mode is typically utilized for forward propagation to simplify picture feature extraction. [LSX20]

  The convolution procedure performs edge zeroing for the input picture, whereby the size of the convolution kernel determines the layer amount of the edge. In order to guarantee that the convolution kernel and the input image's components may be weighted and summated in a sequential manner, edge zeroing is used. As seen in Figure , where the kernel is really rotated 180 degrees around the center, the convolution kernel should also be twisted around and flipped up and down.[LSX20]

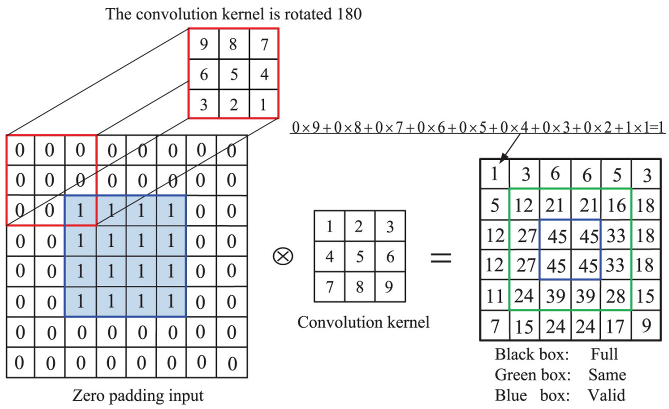

Figure 2.7 – Convolutional process. [LSX20].

  Convolution in CNNs works differently than regular neural networks. It uses parameter sharing and sparse multiplication to efficiently process data without needing a full connection for every input neuron. In simpler terms, it extracts features like other neural networks, but in a more efficient way.[LSX20]

- **Receptive field:**

  Convolutional Neural Networks (CNNs) use receptive fields, which limit connections between neurons to a local area of the input data (like a 10x10 pixel region of an image) instead of connecting every neuron to all the input data. This significantly reduces the number of connections and makes CNNs more efficient to train.[LSX20]

- **Pooling:**

  The primary function of the pooling layers, which are often positioned after the convolutional layers, is to compress the convolutional layers' output feature data. The enhanced output results after the pooling layer can lessen the chance that the neural network would overfit. Additionally, the image feature may be further extracted by pooling operations without affecting the picture's information gathering. [LSX20]

The majority of research has focused on deep convolutional networks since 2014. The remarkable success of CNN architectures[GWK+15] in the ImageNet competition [DDS+09], including ResNet [HZRS16], GoogLeNet [SLJ+15], VGGNet [SZ14], AlexNet [KSH12], and others, served as the impetus for this work. As a particular application of object recognition, facial recognition may be based on well-designed frameworks for generic object identification. Several effective CNN models for face recognition are briefly discussed in this section. Due to the field's fast progress, benchmark accuracies fluctuate significantly from year to year as a result of ongoing model revisions.[NFTT20]

Comparative Analysis of Deep Learning Models with Benchmark Datasets:

Here, the outcomes of many experiments on the simulated face recognition datasets are emphasized for deep networks . Using DeepID1, DeepID2, DeepID2+, DeepID3, DeepFace, Face++, FaceNet, and Baidu models, the tests are run on the simulated data. Utilizing the simulated dataset, the models are trained. The LFW and YTF datasets are used to assess the deep models. Since the identities in these two datasets differ from those in the simulated dataset, they are regarded as benchmark datasets.[Cha20]

The comparative examination of the deep learning models with reference to the LWF and YTF benchmark datasets is highlighted in this paragraph. The outcomes of deep face recognition models on the LWF and YTF datasets are displayed in Table 2.3. These datasets serve as a baseline for the outcomes shown in the preceding table. Table 2.3 shows that for the LWF and YTF datasets, DeepID1, DeepID2, DeepID2+, and DeepID3 reach an accuracy of more than 99". For these two datasets, DeepFace and Face++ achieve an accuracy of more than 97". For both datasets, FaceNet and Baidu attain accuracy rates of more than 98". Included are the relevant training times as well.[Cha20] [YKA02]

Table 2.4 – Deep face recognition techniques [Cha20]

Deep face recognition techniques	**Accuracy % (LWF)**	**Accuracy % (YTF)**	**Training time (min)**
DeepID1	99.27	99.24	46
DeepID2	99.34	99.37	48
DeepID2+	99.34	99.37	50
DeepID3	99.34	99.39	54
DeepFace	97.69	97.68	58
Face++	97.70	97.69	186
FaceNet	98.75	98.77	246
Baidu	98.89	98.86	406

2.4 Conclusion

In conclusion, the field of face detection and recognition has witnessed remarkable progress and innovation, driven by advancements in computer vision, machine learning, and deep learning technologies. Throughout this chapter, we have explored the evolution of methodologies and techniques employed in face detection and recognition, spanning from traditional approaches to state-of-the-art deep learning-based methods.

Chapter 3

System Conception and Implementation

3.1 Introduction

This real-time facial recognition system will be deployed at institutional entry points, using surveillance cameras to identify individuals entering the premises. The main objective is to improve security by identifying authorized individuals and detecting any unauthorized access attempts.

In addition to security, the system will also provide an attendance tracking system, automatically recording the entries and exits of staff or students. This will simplify and automate attendance management, while ensuring the accuracy of recorded data.

To ensure easy and convenient operation, the system will have an intuitive user interface. This will allow users to effectively manage and monitor the system, accessing presence information and receiving real-time alerts when needed.

3.2 System Objectives

The proposed face recognition system have many objectives:

- Enhance security at entry points by accurately identifying and verifying individuals entering an institution, preventing unauthorized access and identifying potential security threats. It implements real-time facial recognition to provide immediate feedback and ensures smooth entry and exit.
- Detect multiple faces simultaneously, allowing for efficient handling of scenarios with multiple individuals entering simultaneously.
- Identity verification and alerts, flagging unauthorized access attempts and notifying security personnel.
- Automates attendance tracking, reducing manual effort and errors.
- Data is stored securely in a database, ensuring data integrity and confidentiality.

- A user-friendly interface is developed for administrators and end-users, facilitating seamless interaction.
- Scalable and adaptable to different institutions and entry point setups, ensuring easy expansion or modification.
- Achieve high accuracy in facial recognition and ensure reliable system performance under various conditions, minimizing false positives and negatives, ensuring dependable operation even in challenging conditions.

3.3 System Architecture

The following figure illustrates the overall architecture of our system :

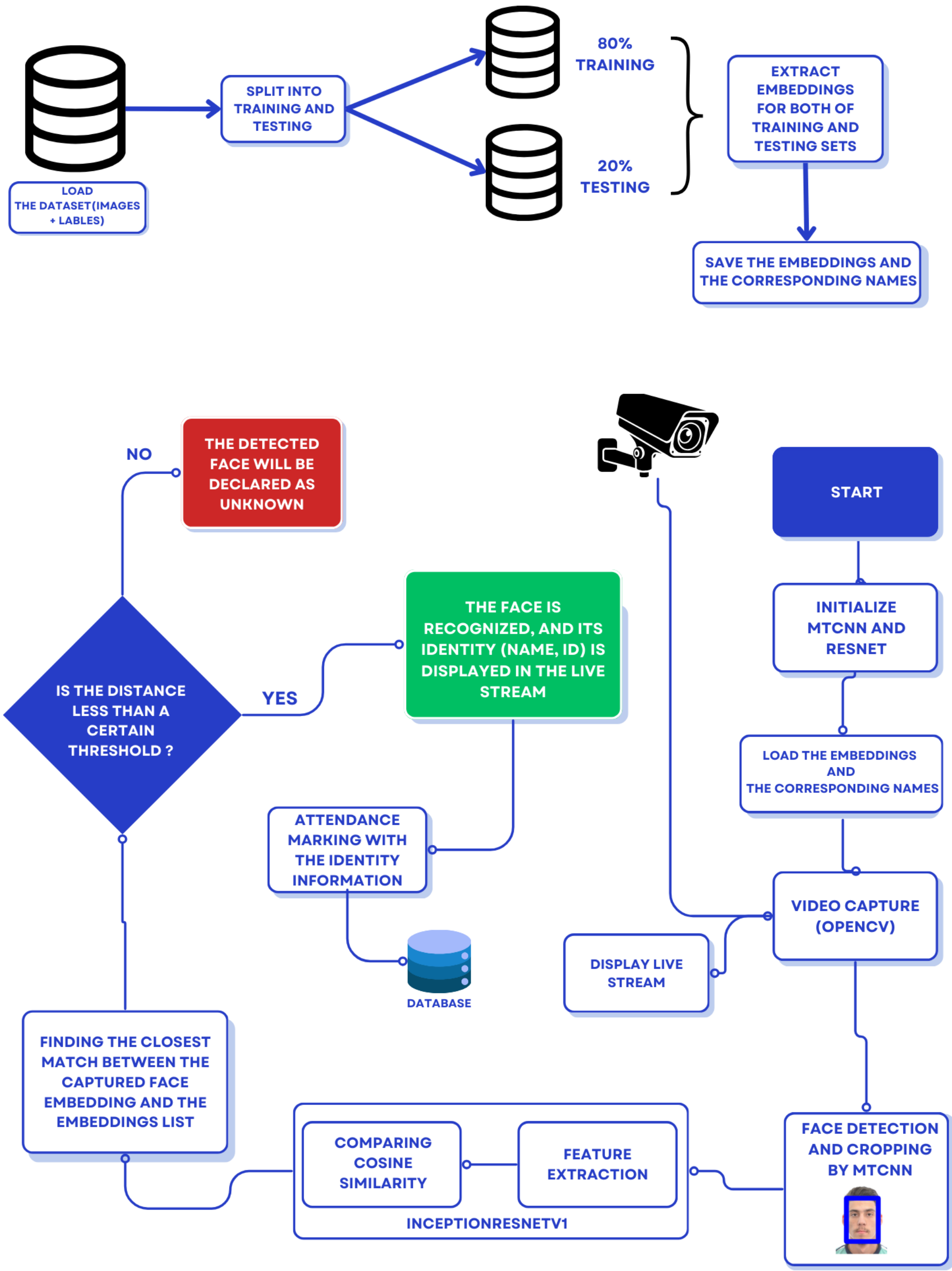

Figure 3.1 – the overall architecture of our system

3.3.1 Data Preprocessing

Data augmentation:

One of data augmentation techniques is applying the geometric transformations to improve the model's robustness to variations in object size, camera movements, and position changes. These transformations include flipping, scaling, translation, rotation, shear, and photometric transformations. Flipping introduces viewpoint invariance, scaling helps the model cope with variations in object size, and translation introduces slight shifts.

Rotation enhances robustness to rotations during image capture, while shear introduces distortion effects. Photometric transformations modify the intensity or color distribution of pixels, such as brightness adjustment and Gaussian blur, to simulate changes in lighting conditions.

Image preprocessing:

Image preprocessing typically involves preparing images before they are fed into a machine learning model. In our system, preprocessing includes loading images, detecting faces, aligning them, and resizing them.

— **Resizing to a Fixed Size (160x160 pixels):** The InceptionResnetV1 model uses a fixed size of 160x160 pixels for input images, allowing for efficient processing and standardized learning of pixel relationships. This consistency helps the model identify features like eyes, nose, and mouth more effectively, regardless of the original image dimensions, making it a valuable tool for deep learning models.

— **Alignment of Facial Landmarks:** Facial landmark alignment is a process used in machine learning to align faces in real-world images, aiming to reduce the impact of pose variations on feature extraction. MTCNN detects facial landmarks like eyes, nose, and mouth in the image, estimating the rotation and translation needed to align the face with a reference template. The image is then transformed using techniques like affine transformations. This ensures the model can focus on actual facial features rather than their positions in the image.

— **Normalization of Pixel Values:** Deep learning models perform better when input data is normalized to a specific range, reducing the influence of variations in image lighting and camera settings. Common normalization techniques include Min-Max Normalization, which scales pixel values to a range between 0 and 1, and Z-Score Normalization, which subtracts the mean value of the image and divides by the standard deviation. Normalization ensures that all pixels contribute equally to the model's learning process, leading to more robust and generalizable feature representations. During post-processing, pixel values are typically normalized from their original range to a new range, such as 0 to 1, -1 to 1, or 0 and standard deviation.

3.3.2 Image/Video Capture:

The system uses a camera to capture frames (images) or a video stream. This often involves the OpenCV library for interfacing with cameras and acquiring frames.

The Image/Video Capture stage is the first and crucial step in our surveillance system. Here's how it works:

1. **Camera Interfacing:** A real-time video footage will be captured by the system using the camera. For example, this could be a webcam, CCTV camera, or other digital cameras that our system can interface with.
2. **Frame Acquisition:** When the camera captures video, we essentially have individual images ("frames") displayed one after another at very high speed. The system needs to acquire these frames individually for further processing. This is where the OpenCV library comes in.

In this project, we'll use OpenCV to interface with the camera and get images from the video. An output from this module becomes an input to face detection and recognition stages of our system[AS20].

3.3.3 Face Detection(MTCNN):

This stage employs a Multi-task Cascaded Convolutional Network (MTCNN) to identify potential faces in the captured frame.

"MTCNN stands for Multi-task Cascaded Convolutional Networks. It is a framework proposed for joint face detection and alignment tasks. The architecture includes a cascaded structure with three stages of deep convolutional networks designed to predict face locations and facial landmarks in a coarse-to-fine manner. Additionally, MTCNN incorporates an online hard sample mining strategy to further enhance performance in practice" [ZZLQ16]

The proposed deep cascaded multi-task framework enhances face detection and alignment in unconstrained environments by leveraging the inherent correlation between these two tasks. Here are the key ways in which the framework improves performance:

— **Multi-task:** By jointly training the network for both face detection and alignment tasks, the model can learn shared representations that benefit both tasks. This shared learning approach helps in capturing the complex relationships between facial features (like eyes, nose, mouth) and improves overall performance.[ZZLQ16]

— **Cascaded Architecture:** The framework utilizes a cascaded architecture with three stages of carefully designed deep convolutional networks. This coarse-to-fine approach allows for progressively refining the face detection and landmark localization, leading to more accurate results.[ZZLQ16]

— **Real-Time Performance:** Despite the complexity of the deep learning model, the framework maintains real-time performance, making it suitable for practical[ZZLQ16] applications where speed is crucial.[ZZLQ16]

The key components of the cascaded architecture with three stages of deep convolutional networks in the MTCNN framework are as follows:

- **Stage 1 - Proposal Network (P-Net):**
  The first stage utilizes a fully convolutional network known as the Proposal Network. P-Net is responsible for generating candidate facial windows and their bounding box regression vectors. It calibrates the candidate windows based on the estimated bounding box regression vectors. Non-maximum suppression (NMS) is applied to merge highly overlapped candidates. [ZZLQ16].
- **Stage 2 - Refine Network (R-Net):**
  In the second stage, all candidates from Stage 1 are input to another CNN called the Refine Network. R-Net further rejects a large number of false candidates, performs calibration with bounding box regression, and conducts NMS to refine the results. [ZZLQ16].
- **Stage 3: O-Net (Output Network):**
  The third stage is similar to the second stage but aims to identify face regions with more supervision. The network in this stage outputs the positions of five facial landmarks. The Output Network (O-Net) produces final bounding box and facial landmarks position.

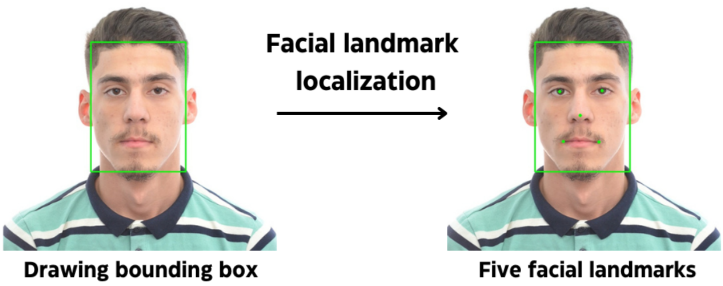

Figure 3.2 – the final bounding box and facial landmarks position

[ZZLQ16].

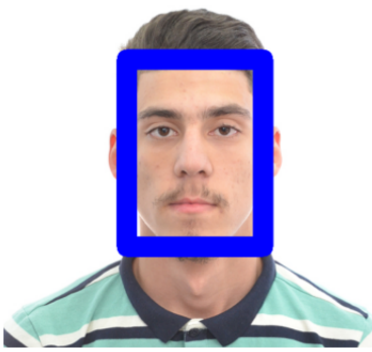

Figure 3.3 – Example of a detected face using MTCNN

These three stages work in a cascaded manner, with each stage refining the results from the previous stage. The cascaded architecture allows for a gradual improvement in face detection and alignment accuracy, leading to superior performance in challenging scenarios

3.3.4 Face recognition using Facenet

A) Overview

FaceNet is a system developed by Google that can learn a direct mapping from face images to a compact Euclidean space such that the distances between the points in this space correspond to a measure of face similarity. It optimizes these embeddings via deep convolu-tional networks and a triplet-based loss function so that they represent tasks such as face recognition, verification, clustering easily and can be used as feature vectors. Furthermore, it takes only 128 bytes per face on representation for which it achieves state-of-the-art performance in face recog- nition at very low computational cost[SKP15] .The difference of FaceNet from previous approaches of face recognition is summarized in the following points:

— **Direct Optimization of Embedding:** FaceNet directly learns a mapping from face images to a compact Euclidean space where distances correspond to face similarity. This is in contrast to previous approaches that used intermediate bottleneck layers for optimization. [SKP15]

— **Triplet-Based Loss Function:** FaceNet uses a triplet-based loss function based on Large Margin Nearest Neighbor (LMNN). The triplets consist of two matching face thumbnails and a non-matching face thumbnail, aiming to separate positive pairs from negative pairs by a distance margin. This approach is different from using pairs of positives and negatives in other loss functions.[SKP15]

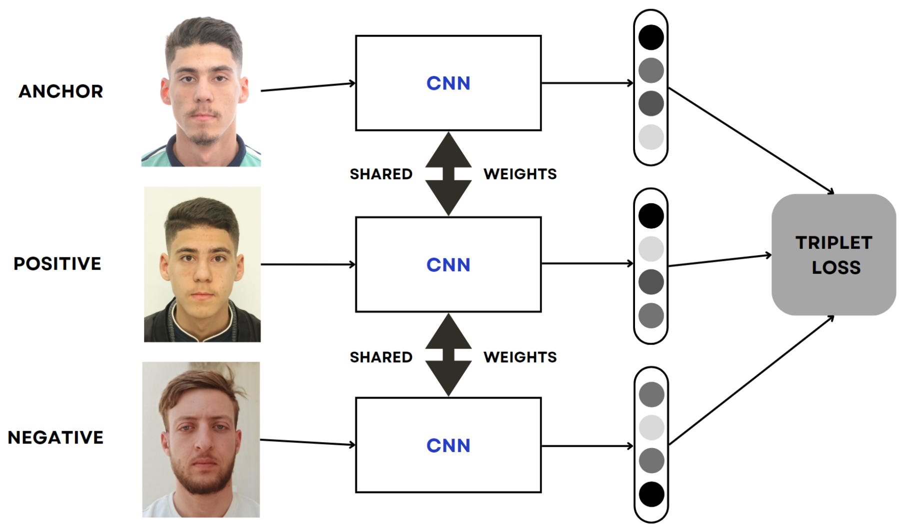

Figure 3.4 – Triplet-Based Loss Function [SKP15]

— **Deep Convolutional Networks:** FaceNet utilizes deep convolutional networks, specifically the Zeiler et Fergus [ZF13] and Inception models, which have been successful in computer vision tasks. These networks reduce the number of parameters and FLOPS required for comparable performance.[SKP15]

— **Online Triplet Mining:** FaceNet introduces a novel online negative exemplar mining strategy to ensure increasing difficulty of triplets as the network trains. This strategy, inspired by curriculum learning, helps improve performance[SKP15].

— **Efficiency:** FaceNet achieves state-of-the-art face recognition performance using only 128-bytes per face, showcasing much greater representational efficiency compared to previous methods.[SKP15]

Facenet makes use of the Inception ResNet architecture, which blends residual connections with the Inception module. Because of its design, the network is able to learn and produce numeric embeddings, which are representations of faces. The distinct characteristics of a person's face are represented by these high-dimensional vector embeddings. Face recognition systems may distinguish between two faces that belong to the same person by exploiting cosine similarity.

B) Inception-ResNet-v1:

Definition: Combining the advantages of ResNets with Inception networks, the Inception-ResNet V1 is a convolutional neural network (CNN). It was created to enhance learning models' speed, accuracy, and error-reduction capabilities1. In the 2016 study [SIVA16], Inception-Resnet v1 was first presented. Its goal was to increase picture categorization tasks' accuracy.

Architecture: Inception-ResNet-v1 is composed of several Inception-ResNet blocks. These are a blend of the ResNet and Inception methods. There are three types of these blocks, which include Inception-ResNet-v1-A, B and v1-C. Every Inception-ResNet block is made from multiple branches that employ distinct sizes of convolutional kernels to process input data. This assists the model in capturing features at various scales and resolutions.[SIVA16]

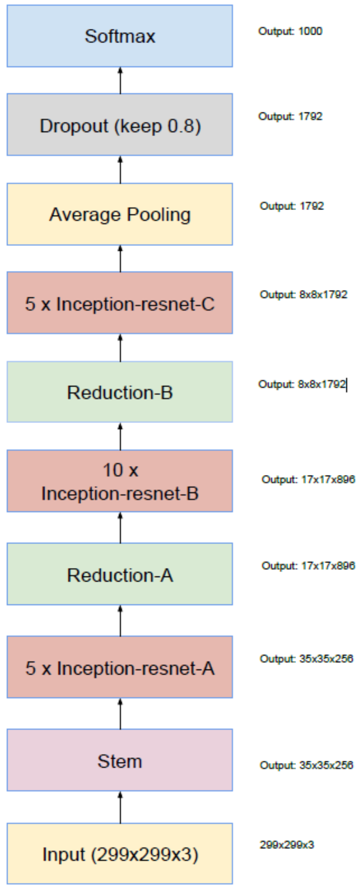

Figure 3.5 – Schema for Inception-ResNet-v1 [SIVA16]

Component	**Output**	**Description**
Input Layer (299x299x3)		The input layer accepts images of size 299x299 pixels with 3 color channels (RGB).
Stem	35x35x256	The stem block consists of initial convolutional layers and max-pooling operations designed to reduce the spatial dimensions of the input image while increasing the depth (number of feature maps).

(continued on the next page)

Component	**Output**	**Description**
5 x Inception-ResNet-A	35x35x256	This section consists of five Inception-ResNet-A blocks, each of which is designed to efficiently capture spatial hierarchies in the data using a combination of 1x1, 3x3, and 5x5 convolutions with residual connections.
Reduction-A	17x17x896	The Reduction-A block reduces the spatial dimensions from 35x35 to 17x17 while significantly increasing the depth to 896. This is typically done using a combination of convolutions and pooling layers.
10 x Inception-ResNet-B	17x17x896	This section includes ten Inception-ResNet-B blocks. These blocks are similar to Inception-ResNet-A but designed to work with the higher depth of 896 feature maps.
Reduction-B	8x8x1792	The Reduction-B block further reduces the spatial dimensions to 8x8 while increasing the depth to 1792. This further compacts the spatial information into deeper features.
5 x Inception-ResNet-C	8x8x1792	The Inception-ResNet-C blocks process the feature maps with the same spatial dimensions (8x8) but maintain a high depth, enabling the network to capture complex patterns and features.

(continued on the next page)

Component	**Output**	**Description**
Average Pooling	1792	An average pooling layer is applied to reduce the spatial dimensions to 1x1 while preserving the depth of 1792. This effectively aggregates the spatial information across the entire feature map.
Dropout (keep 0.8)	1792	Dropout is applied with a keep probability of 0.8. This helps prevent overfitting by randomly dropping a fraction of the neurons during training.
Softmax	1000	The final softmax layer outputs a probability distribution over 1000 classes, typically used for classification tasks.

Table 3.1 – Descriptions of Inception Resnet-V1 Architecture

The stem of the Inception-ResNet-v1 network: This is the initial set of layers, also referred to as the "stem" of the architecture. These layers are used before the Inception blocks in the architecture.

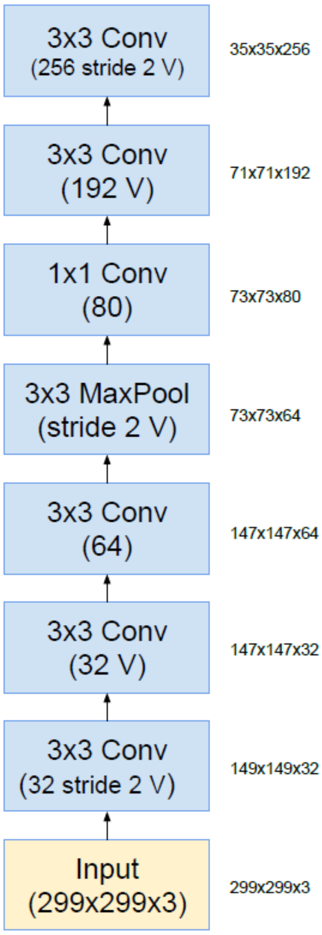

Figure 3.6 – Schema for stem of Inception-ResNet-v1 [SIVA16]

1. **Input Layer:** The input layer is sized at 299x299x3, indicating an input image with dimensions 299 by 299 pixels and 3 color channels (likely RGB).
2. **Convolutional Layers:** There are several convolutional layers, each labeled with the filter size and number of filters (e.g., "3x3 Conv (32 stride 2 V)"). The numbers in parentheses indicate the number of filters in each layer, and "stride 2" suggests that the first convolutional layer uses a stride of two.
3. **Max Pooling Layers:** There are max pooling layers described as "3x3 MaxPool (stride 2 V)". These layers are used to reduce the spatial dimensions of the output volume.

On the right side of every rectangle is a dimension showing how each operation affects the size and depth of the data moving through it. As an instance, after numerous layers from an initial input size 299x299x3, it becomes 35x35x192.

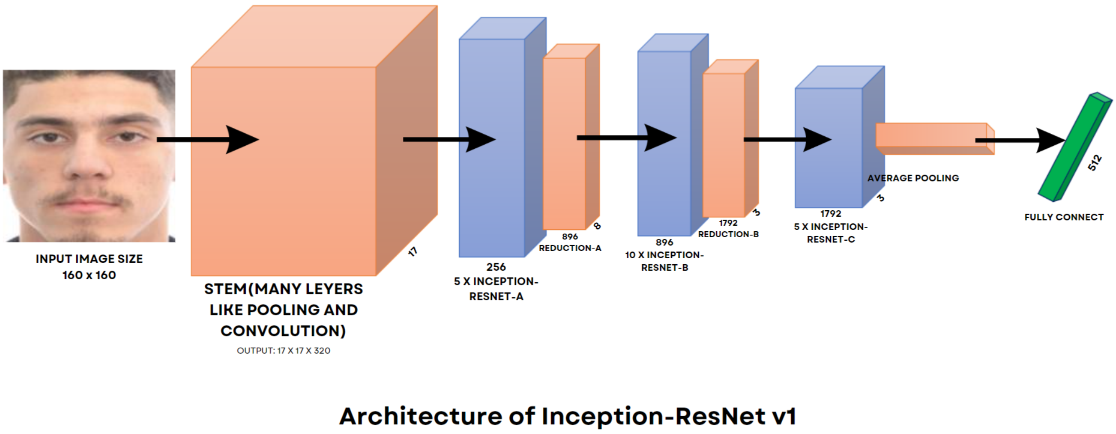

Figure 3.7 – Architecture of Inception-ResNet-v1

The steps of the embedding extraction process:

1. **Input Image:** The input to the network is an image of size 160x160 pixels.
2. **Stem:** The initial layers of the network, referred to as the "Stem," consist of several convolutional and pooling layers. These layers are responsible for reducing the spatial dimensions of the input image while increasing the depth (number of channels) to 320. The output from the Stem is a feature map of size 17x17x320.
3. **Inception-ResNet-A Module:** The next stage consists of 5 Inception-ResNet-A modules. These modules incorporate both inception structures and residual connections to enhance learning efficiency and feature extraction.
4. **Reduction-A Module:** Following the Inception-ResNet-A modules is the Reduction-A module. This module reduces the spatial dimensions and increases the depth of the feature maps to 896 channels. The primary purpose of the Reduction-A module is to downsample the feature maps efficiently.
5. **Inception-ResNet-B Module:** After the Reduction-A module, there are 10 Inception-ResNet-B modules. These modules continue to extract complex features from the input data using inception blocks combined with residual connections.
6. **Reduction-B Module:** The Reduction-B module further reduces the spatial dimensions of the feature maps, increasing the depth to 1792 channels. This step is crucial for handling deeper layers while maintaining computational efficiency.
7. **Inception-ResNet-C Module:**

   Next, 5 Inception-ResNet-C modules process the data. These modules operate similarly to the previous inception-residual blocks, continuing to refine and extract high-level features from the input data.
8. **Average Pooling:** The feature maps are then subjected to an average pooling operation, which reduces each feature map to a single value by averaging all the values in each feature map. This results in a 1792-dimensional vector.

9. **Fully Connected Layer:** The final layer is a fully connected layer with 512 units. This layer combines all the extracted features to make the final prediction or classification.
10. **Output Embeddings:** The final feature maps are then flattened and passed through a fully connected layer to produce a fixed-size embedding vector. This embedding vector represents the facial features and is used for facial recognition and classification.

C) Reasons for Choosing Inception Resnet-V1:

- **High Accuracy:** Inception Resnet-V1 was chosen for facial recognition applications because to its track record of obtaining high accuracy in picture recognition tasks.
- **Robust Performance:** Even in intricate and congested settings, the architecture and design of the model allow for robust performance in processing and evaluating facial data taken by security cameras.
- **Efficiency:** Inception Resnet-V1 provides fast comparisons against databases of recognized threats or individuals of interest, enabling real-time identification of individuals.

D) Identification Process:

The system uses MTCNN to detect faces in video capture, and the InceptionResnetV1 model processes a detected face into a high-dimensional feature vector called face embedding. This vector identifies unique properties of the person's face and allows comparison against others, enhancing the system's facial detection capabilities.

- **Embedding Comparison:** The system then compares the newly extracted face embedding with the embeddings stored in its database. This comparison is done by calculating the Euclidean distance (or another suitable distance metric) between the new embedding and each of the database embeddings. The smaller the distance, the higher the similarity between the two embeddings.
- **Recognition by Thresholding:** It uses a predetermined threshold value to determine if the nearest match indicates a successful recognition. The maximum distance at which a face is considered as being recognized is what this threshold represents.
- **Decision Making:**
  - If the captured face embedding's distance from the nearest matching one in the database goes below the threshold:
    - The face is considered as recognized.
    - It picks out and brings up the identity (name, ID) of that closest match from the database.

- If the distance is greater than or equal to the threshold:
  - The system considers it an unknown face.
  - No identification or attendance marking can be done.

3.3.5 Attendance Marking Process

1. **Determining Category:** The identity information is the basis for determining the category that individuals fall in, i.e. student or worker. In this system, categorization is used to distinguish different types of users.
2. **Recording Attendance:** Once an individual's identity and class are verified, attendance is marked on their behalf. The recording takes place by inserting a record into a dedicated database table. Usually included in the record are: name, ID number, status of studentship, and time of attendance in the form of a timestamp.
3. **Updating Database:** New attendance records have to be updated in the SQLite database so they can be stored permanently for reporting or verification purposes.
4. **User Interface Updating:** The user interface gets updated with the new attendance record as well as displaying the name and ID of recognized individuals on screen and confirming that attendance has been marked.
5. **Real-Time Monitoring:** This system continuously monitors the video feed. Whenever a face is detected, it re-runs the face detection-recognition-attendance marking process. This helps monitor attending students' movements by identifying who has arrived at school.

3.4 Conclusion

This chapter started with presenting the system objectives then discusses the development and implementation of a real-time facial recognition system, focusing on its objectives, architecture, and practical steps. The system aims for enhanced security, real-time processing, and automated attendance tracking. The architecture is designed for efficient data preprocessing, accurate image and video capture, and robust face detection and recognition. MTCNN and InceptionResnetV1 are used for precise detection and reliable recognition. The implementation phase highlights the critical tools and libraries used, and the results show the system's accuracy and effectiveness. However, limitations related to dataset size and computational capacity were identified. The project successfully achieved its core objectives, providing a functional and user-friendly interface for security and attendance management in various institutions.

Chapter 4

Implementation, Tests and Results

4.1 Introduction

In this chapter, we outline the implementation and testing of biometric technologies for person identification within university campus security systems, particularly focusing on surveillance systems that utilize facial recognition technology. Our implementation section details the integration process of facial recognition technology, highlighting its potential as a transformative tool for enhancing campus security through proactive threat detection and access control. The results section presents the findings from our rigorous testing, demonstrating the effectiveness of facial recognition systems despite the complexities of their deployment.

4.2 Description of the Libraries and Tools Used

4.2.1 Hardware environment

The development of our application is carried out by computer with the following characteristics:

- ✓ **Operating System (OS):** Windows 10 64-bit
- ✓ **Processor:** Intel® Core™ i7-4710HQ CPU @ 2.50GHz × 4
- ✓ **Hard disk:** 512 GB SSD
- ✓ **RAM:** 8 GiB
- ✓ **Graphics card (or Video card):** NVIDIA GeForce GTX 860M

4.2.2 Software Environment

- **Python:** Python is an object-oriented, interpreted high-level programming language that is particularly appealing for creating applications quickly and for usage as a collage language or scripting language to join together pre-existing components. Python has an easy-to-learn syntax. Because it places an emphasis on readability, maintenance costs are decreased. Python's support for packages and modules promotes code reuse and program modularity.. In our work, we made use of the version **the version 3.6.9**.[W4]

- **Jupyter Notebook:** Using the robust open-source web tool Jupyter Notebook, users can create and share documents with narrative text, live code, and graphics. It is extensively used for interactive data exploration, algorithm prototyping, and result communication in the scientific computing and data science sectors.[W5]
- **OpenCV:** OpenCV (Open Source Computer Vision Library) is an open-source library highly optimized for real-time applications, including video processing for face recognition. It supports cross-platform interfaces in C++, Python, and Java, making it accessible on various operating systems like Linux, MacOS, Windows, iOS, and Android. OpenCV is equipped with tools and algorithms that facilitate real-time image and video analysis, object identification, and machine learning capabilities [W6].
- **NumPy:** NumPy, which stands for Numerical Python, is an open-source Python library frequently used for numerical computations in science and engineering. It offers a robust N-dimensional array object together with an extensive set of methods to easily operate on these arrays. For scientific computing in Python, NumPy is a must-have. It provides capabilities for random number generation, Fourier transforms, and linear algebra, which are very helpful in real-time face recognition applications when managing big datasets and computations are required [W7][W8][W9].
- **PyTorch:** PyTorch is a deep learning optimized tensor library that runs on both CPUs and GPUs. Because of its adaptability and simplicity in creating and refining neural network models—a crucial component for applications such as real-time facial recognition—it has gained special favor in the academic community [W10].
- **FaceNet-PyTorch:** A Python package called FaceNet-PyTorch offers pre-trained models for PyTorch-based face detection and identification. It consists of the following essential elements [W11]:
  - Multitask Cascaded Convolutional Networks (MTCNNs): A deep learning-based face identification model, MTCNN can identify and align faces in both photos and videos. It's known for its efficiency and precision.
  - InceptionResnetV1: FaceNet-PyTorch's primary face recognition model is InceptionResnetV1, based on the InceptionResnetV1 architecture. The model can be used for face identification and verification applications and has been pre-trained on large face datasets like VGGFace2 and CASIA-Webface.

- **Pandas:** For the Python programming language, Pandas is an open-source, BSD-licensed library that offers fast, user-friendly data structures and data analysis capabilities. It is extensively utilized in data analysis and modification, which is important when working with datasets for real-time face recognition applications [W12].
- **Pillow (Python Imaging Library):**The Pillow (Python Imaging Library) is a library that gives your Python interpreter additional image processing features. This covers the ability to access, work with, and save a wide variety of picture file types. It is helpful for preparing photos in face recognition projects before feeding them into a model for inference or training [W13].
- **Imgaug:** A package called imgaug is used in machine learning studies to enhance images. It assists you in transforming photos to reflect their appearance under various settings, which can enhance the generalization and resilience of your machine learning models. It includes several different augmentation techniques, like cropping, flipping, blurring, changing the color, and more [W14].
- **Datetime:** Python comes with a built-in module called the datetime library that offers classes for manipulating dates and times. It is a frequently utilized, strong, and adaptable tool for many different applications, such as real-time facial recognition projects. Numerous classes are available for representing and working with dates, times, and time intervals in Python thanks to the datetime package [W15]. The primary courses are:
  - `datetime`: Indicates a certain day and time.
  - `date`: Indicates a certain date.
  - `time`: Indicates a precise moment in time.
- **SQLite:** A lightweight, open-source relational database management system called SQLite enables you to organize and manage data. It is made to be simple to use and to interface with other libraries and tools [W16][W17]. The characteristics include:
  - Self-contained database: Easy to distribute and implement, suitable for embedded platforms or limited resources.
  - No separate server process: Easy to use without setup or configuration.
  - Transactions: Supports transactions, providing atomic and consistent database operations.
  - SQL subsets: Supports various subsets of SQL, including SELECT, INSERT, UPDATE, and DELETE statements.
- **PyQt5:** PyQt5 is a Python package that offers a comprehensive collection of tools for creating graphical user interfaces (GUIs). It is known for its simplicity and compatibility with various Python frameworks and packages [W18][W19]. Here are some key features of PyQt5 for GUI development:

- Extensive Widget and Tool Selection: PyQt5 provides a large variety of widgets and tools for building GUIs, including buttons, labels, text boxes, and more.
- Cross-Platform Compatibility: PyQt5 programs can run on various operating systems like Windows, macOS, and Linux.
- Integration with Diverse Libraries: PyQt5 seamlessly integrates with various Python frameworks and libraries, including OpenCV and SQLite.

4.3 Steps of realisation

4.3.1 Dataset

The data collected contains 12 identities (folders named by person's name). Each folder contains images numbering between 10 and 20. The faces in the images vary in location and poses .

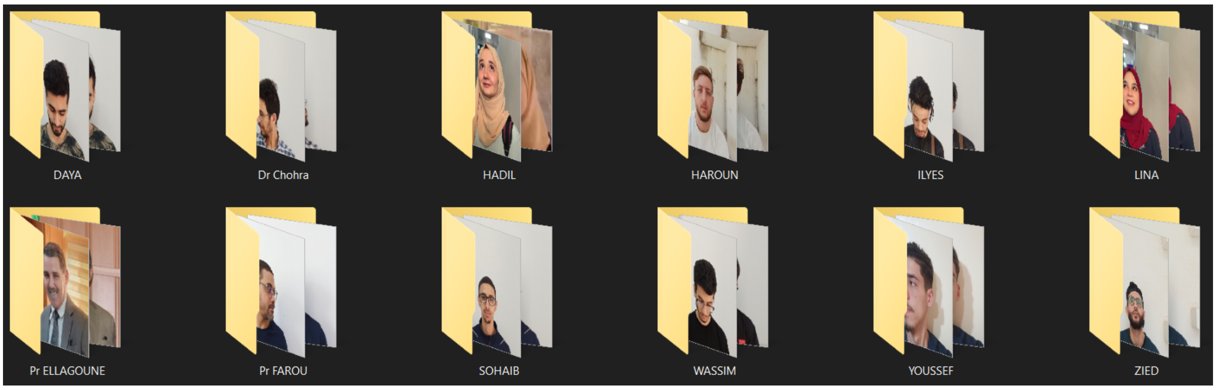

Figure 4.1 – The folders of the dataset

But the quality and amount of training data play a critical role in determining the model's performance in the field of machine learning, especially in computer vision applications that include picture categorization or identification. A basic problem occurs when there is insufficient training data, which can cause overfitting. A model becomes overfit when it becomes too adept at recognizing the unique patterns and quirks of the training set, which makes it less able to generalize to new samples. To address this challenge and improve the generalization capabilities of the model, we employed data augmentation techniques on our dataset.

4.3.2 Data augmentation

Geometric Transformations:

These transformations modify the spatial relationship between pixels in the image. The code employs:

— **Flipping:** Random horizontal flips (applying iaa.Fliplr) introduce viewpoint invariance, as objects in the real world can be viewed from different angles.

— **Scaling:** Random scaling (scale="x": (0.8, 1.2), "y": (0.8, 1.2)) helps the model cope with variations in object size within the image frame.

— **Translation:** Random translations introduce slight shifts, simulating potential camera movements or object position changes.

— **Rotation:** Random rotations (rotate=(-20, 20)) enhance robustness to rotations that might occur during image capture.

— **Shear:** Shearing (shear=(-16, 16)) introduces a distortion effect, forcing the model to learn features that are resistant to such variations.

Photometric Transformations:

These transformations modify the intensity or color distribution of pixels. The code utilizes:

— **Brightness Adjustment:** Random variations in brightness (iaa.Multiply((0.8, 1.2))) simulate changes in lighting conditions.

— **Gaussian Blur:** A slight Gaussian blur (iaa.GaussianBlur(sigma=(0, 1.0))) can improve model robustness to noise and minor image imperfections.

Figure 4.2 – Applying the data augmentation to one of the images of the dataset

After applying the data augmentation on all of the images that we have in the collected dataset , the total number of images of the dataset is increased to 1061 images.

4.3.3 Transfer learning

In machine learning, a technique known as transfer learning involves using a model that has been trained on one task as the foundation for a model on a different task. When there is insufficient data available for the second task or when the tasks are comparable to each other, this strategy is especially helpful. Transfer learning, in its simplest form, is the process of using the information you get from addressing one problem to better handle another that is related to it. For example, you may quickly design high-performing models for complicated tasks like natural language processing (NLP) or computer vision by beginning with pre-trained models and building upon them.[W20][W21] Our system employs transfer learning using the InceptionResnetV1 model pre-trained on the VGGFace2 dataset.

With an average of 362.6 photos per subject, the dataset includes 3.31 million photographs from 9131 people. Pictures are taken from Google ImageSearch and vary greatly in terms of age, lighting, ethnicity, occupation, and stance (e.g. actors, sports, politicians).Three objectives guided the collection of the dataset: (i) having a high number of identities and photos for each identity; (ii) covering a wide range of posing, age, and ethnicity; and (iii) minimizing label noise. We go over the methodology used to get the dataset, namely the automatic and human filtering steps.[CSX+17]

Figure 4.3 – Examples of VGGFace2 templates. Left: three distinct position templates (grouped by row): frontal, three-quarter, and profile. Right: age templates (organized by row) for two subjects at different ages: young and mature.[CSX+17]

4.3.4 Model Initialization

MTCNN Parameters:

- image_size=160: The size to which detected faces are resized.
- margin=0: Margin around the detected face.
- min_face_size=20: Minimum size of the faces to be detected.
- thresholds=[0.6, 0.7, 0.7]: Detection thresholds for the three stages of the network.
- factor=0.709: Scale factor for the image pyramid.
- post_process=True: Whether to apply post-processing to the detected faces.
- device=device: The device to run the model on (GPU or CPU).

InceptionResnetV1:

- InceptionResnetV1 is a deep learning model used for extracting face embeddings. These embeddings are used for comparing and recognizing faces.

— It is pre-trained on the VGGFace2 dataset, which contains a large number of face images.

— The model is set to evaluation mode (eval()) to disable dropout and batch normalization, ensuring consistent results during inference.

4.3.5 Data Preparation

1. `Load:` The dataset used in this project is a collection of facial images stored in a directory structure where each sub-directory represents a different class (or person). The images are loaded using the ImageFolder class from the torchvision.datasets module. This class automatically assigns labels based on the sub-directory names.
2. `Split:` The dataset is then split into training and test sets using an 80-20 split. The random split function is used for this purpose, ensuring that the split is random..
3. `Handle:` DataLoaders are created for both the training and test sets to handle batching and parallel loading of data. A custom collate function is used to ensure proper handling of image batches.

4.3.6 Face Detection and Embedding Extraction

1. Iterate over batches of images and labels from the DataLoader.
2. Use MTCNN to detect and align faces. The `return_prob=True` flag returns the probability of detection for each face.
3. Check if faces are detected (`x_aligned` is not None). If faces are detected, they are added to the aligned list, and the corresponding labels are added to the names list.
4. Convert the list of aligned images to a tensor and move it to the appropriate device.
5. Pass the aligned images through InceptionResnetV1 to get embeddings.
6. Move the embeddings back to the CPU and detach them from the computational graph to prevent gradient computations.
7. Applying the Function on: Training Set: Extract embeddings for the training dataset. Test Set: Extract embeddings for the test dataset.

4.3.7 Training

A) Instantiation of the SVC:

a. `kernel='linear'`: This parameter specifies the kernel type to be used in the algorithm. A linear kernel means that the decision boundary is a hyperplane, which is a generalization of a line in 2D or a plane in 3D to higher dimensions. In our case, since the embeddings are high-dimensional, the decision boundary is a hyperplane in that space.

b. `probability=True`: This enables probability estimates. SVC does not directly support probability estimates; it needs to be fit with `decision_function_shape='ovr'` and then calibrated after fitting (using `CalibratedClassifierCV`). This is useful if you want to know the confidence of the predictions.

B) Training the SVC:

The training of SVC model is done calling (`clf.fit(train_embeddings, train_labels)`. The `train_embeddings` are the feature vectors extracted from the training images using the InceptionResnetV1 model. These embeddings are high-dimensional vectors that capture the facial features. The `train_labels` are the corresponding class labels (e.g., the identity of the person in the image).

During training, the SVC algorithm finds the hyperplane that best separates the different classes in the feature space. The goal is to maximize the margin between the classes, which is the distance between the hyperplane and the nearest data points (support vectors). This is done by solving a quadratic optimization problem that finds the optimal weights and bias for the hyperplane.

C) Saving the Embeddings

1. The embeddings and corresponding names of the identities are saved for later use to avoid repeating the training process every time.

D) Attendance Table

1. The attendance table has the following schema:
   (a) `name TEXT`: The name of the person.
   (b) `id TEXT`: The ID of the person.
   (c) `category TEXT`: The category of the person (e.g., student, staff).
   (d) `timestamp TEXT`: The timestamp of the attendance record.

4.3.8 Processing

A) FaceRecognitionApp Class:

1. This class handles the core functionalities of the application.
2. `init` method:
   (a) Initializes the device to run computations on (CPU, CUDA, or MPS if available).
   (b) Loads the pre-trained MTCNN (Multi-task Cascaded Convolutional Networks) model for face detection and the InceptionResnetV1 model for face recognition.

(c) Loads the saved embeddings and names data from a file (`Facenet Pytorch Finetuning embeddingsALLDATA.pt`).

(d) Reads identity data (names, IDs, categories) from a CSV file (`identity data.csv`) and performs checks for duplicate IDs.

(e) Creates a connection to a SQLite database (`attendance.db`) and creates a table named `attendance` to store attendance records (name, ID, category, timestamp).

(f) Initializes an empty dictionary to track the last recorded attendance time for each person.

(g) Sets up the user interface (UI) using Qt widgets.

(h) Initializes the webcam capture using OpenCV.

B) UI Functions:

1. `get device:` Checks for available computing devices and returns the appropriate one.
2. `init ui:` Creates the application window with the following elements:
   (a) Label for displaying the camera feed.
   (b) Table widget to display attendance records.
   (c) Start/Stop buttons for controlling the camera.
   (d) Display Attendance button to show all attendance records.
3. `start camera:` Starts a timer to continuously capture and process frames from the webcam.
4. `stop camera:` Stops the timer, effectively pausing the camera capture.

C) Processing Camera Frames (`update frame`):

1. Captures a frame from the webcam.
2. Converts the frame from BGR (OpenCV format) to RGB (PyTorch format).
3. Uses MTCNN to detect faces in the frame.
4. Loops through each detected face:
   (a) Extracts the facial region from the frame.
   (b) Generates a face embedding using the InceptionResnetV1 model.
   (c) Calculates the distance between the current face embedding and all the embeddings in the loaded dataset.
   (d) Identifies the person with the closest embedding (smallest distance) under a certain threshold (0.75 in this case).

(e) Retrieves the person's name, ID, and category from the identity dictionary.

(f) Draws a bounding box around the detected face and displays the name, ID, and distance on the frame.

(g) If the identified person is not already marked as present within the last minute and the distance is below the threshold:

i. Logs the attendance in the database table with the current timestamp.

ii. Updates the attendance table in the UI.

(h) Otherwise, the person is marked as unknown.

D) Displaying and Updating Attendance Records:

1. `display image:` Converts the processed frame back to BGR format and displays it on the label in the UI.
2. `log attendance:` Inserts a new attendance record into the database table.
3. `update table:` Adds a new row to the attendance table in the UI with the details of the logged attendance.
4. `display attendance:` Clears the attendance table and populates it with all records retrieved from the database.

E) Closing the Application (`closeEvent`):

1. Releases the webcam capture.
2. Closes the database connection.

4.4 Test and Results

4.4.1 Evaluation Metrics

A) Accuracy:

This is the most basic metric, representing the overall percentage of correctly classified faces. It's calculated as the number of true positives (correctly recognized faces) and true negatives (correctly rejected faces) divided by the total number of predictions.The accuracy is calculated using the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Accuracy provides a general idea of how well the system performs, but it can be misleading in imbalanced datasets. If there are significantly more faces from one class (e.g., most images are of you), the system can achieve high accuracy simply by always predicting that class, even if it performs poorly on less frequent classes.

B) Precision:

This metric focuses on the positive predictive value. It represents the proportion of faces predicted as a specific class that actually belong to that class. It's calculated as the number of true positives divided by the total number of positive predictions (including false positives).The Precision is calculated using the following formula:

$$\text{Precision} = \frac{TP}{TP + TN} \tag{4.2}$$

Precision is crucial for tasks where false positives are costly. For example, in a security application, a high precision ensures the system only identifies authorized individuals, minimizing false alarms.

C) Recall:

This metric focuses on how well the system identifies all positive cases. It represents the proportion of actual positive cases (faces belonging to a specific class) that are correctly identified by the system. It's calculated as the number of true positives divided by the total number of actual positive cases (including false negatives).The Recall is calculated using the following formula:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4.3}$$

Recall is important when it's critical to not miss any relevant faces. For example, in a missing person identification system, a high recall ensures the system has a good chance of recognizing the missing person's face if it's present in the database.

D) F1 Score:

This metric combines precision and recall into a single score, providing a balance between the two. It's calculated as the harmonic mean of precision and recall.The F1 Score is calculated using the following formula:

$$\text{Recall} = \frac{2.(Precision.Recall)}{Precision + Recall} \tag{4.4}$$

F1 score offers a more balanced view of the system's performance, considering both its ability to avoid false positives and its ability to identify all true positives. It's a valuable metric when both precision and recall are important.

E) Confusion Matrix:

1. **Role of Confusion Matrix:** A confusion matrix is a powerful visualization tool used to evaluate the performance of a classification model, including facial recognition systems. It provides a clear breakdown of how the model performed on each class, allowing you to identify areas for improvement.
2. **Structure of a Confusion Matrix:** The confusion matrix is a table with rows representing the actual class labels (ground truth) and columns representing the predicted class labels. Each cell contains the number of instances that fall into a specific combination of actual and predicted classes. Here's a breakdown of the key elements:
   (a) **True Positives (TP):** These are correctly classified instances. They fall on the diagonal of the matrix, where the actual class and predicted class match.
   (b) **False Positives (FP):** These are instances incorrectly classified as a particular class. They appear above the diagonal, where the predicted class differs from the actual class.
   (c) **False Negatives (FN):** These are instances that belong to a specific class but were predicted as a different class. They appear to the left of the diagonal.
   (d) **True Negatives (TN):** This category is typically not applicable in facial recognition tasks, as there are usually only positive classes (faces) and negative isn't explicitly predicted.

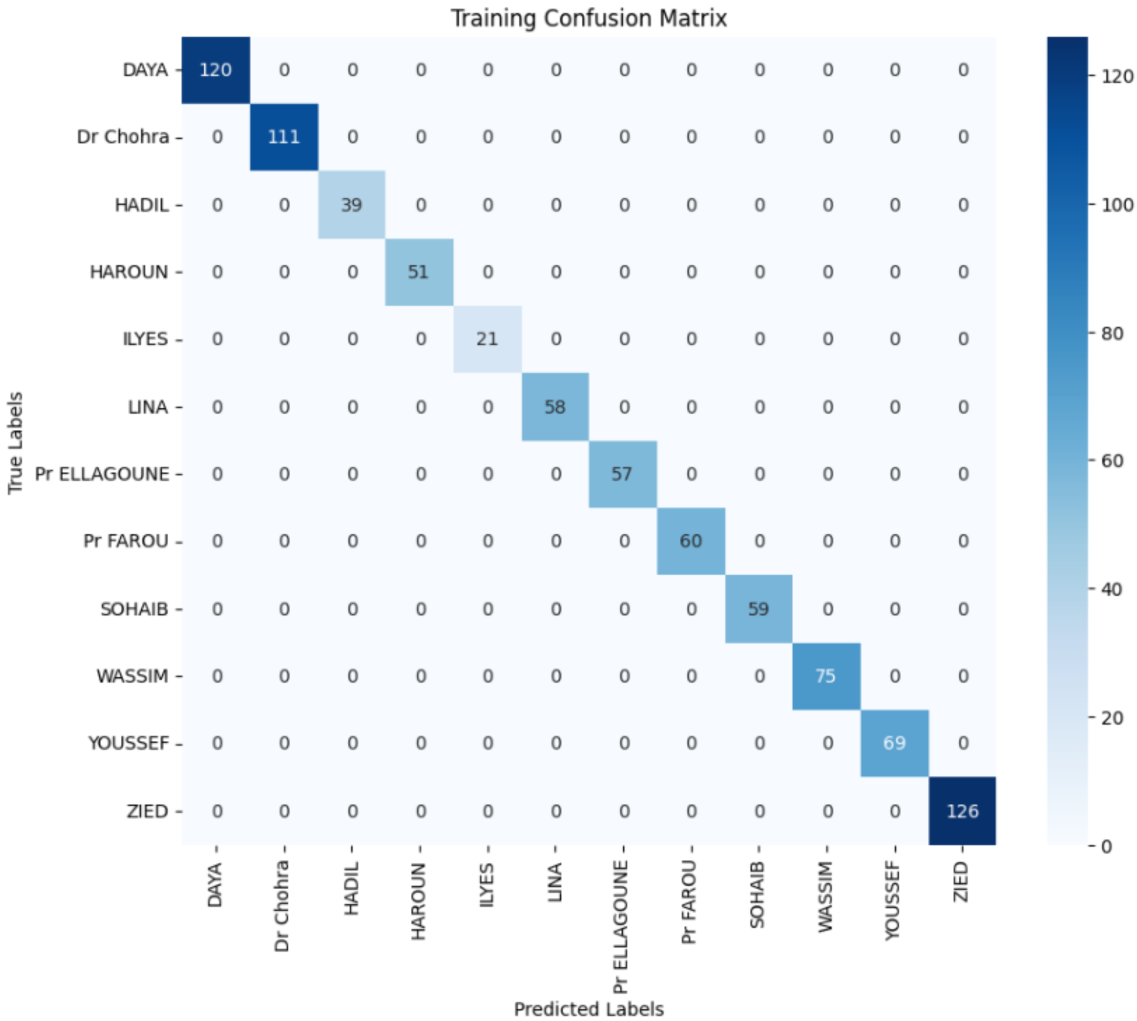

Figure 4.4 – The training set confusion matrix

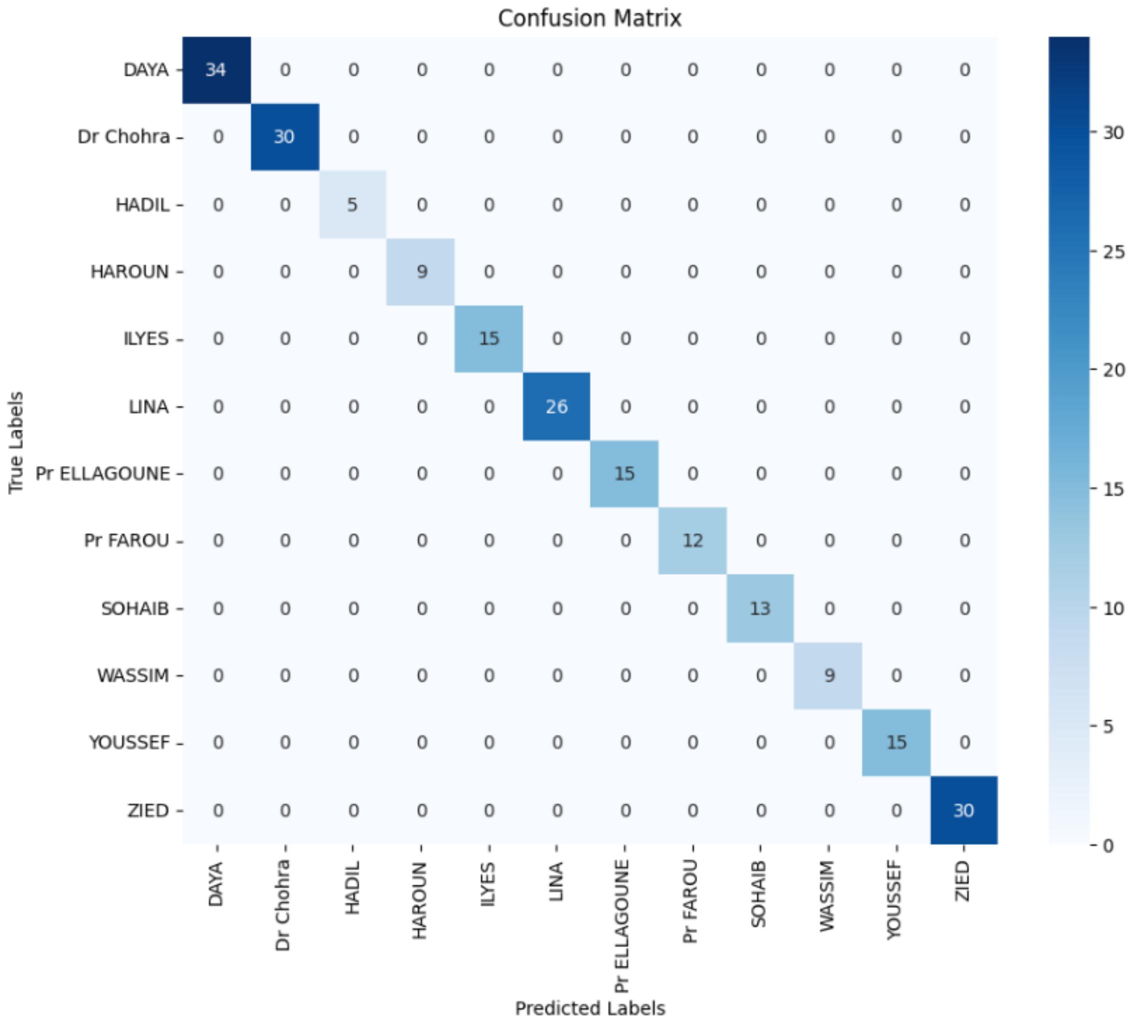

Figure 4.5 – The test set confusion matrix

4.4.2 Stratified K-Fold Cross-Validation:

1. **StratifiedKFold:** This ensures that each fold of the cross-validation has approximately the same percentage of samples of each class as the original dataset. This is particularly important in cases of imbalanced datasets.
2. **n_splits=5:** Specifies that the dataset will be split into 5 folds.
3. **shuffle=True:** Ensures that the data is shuffled before splitting into folds to enhance the randomness and robustness of the cross-validation.
4. **random_state=42:** Ensures reproducibility by setting a random seed.

4.4.3 Latency Metrics:

A) Definition:

1. Detection Time: Measure the time taken by the MTCNN model to detect faces in a frame.

2. Recognition Time: Measure the time taken by the InceptionResnetV1 model to recognize the detected faces.
3. Frame Rate: Calculate the number of frames processed per second.

B) Implementation:

To implement the latency metrics, we use the `time` module to measure the time taken for each operation.

1. Import `time` Module: `import time`
2. Add Variables for Timing:
   — `self.detection_times = []`
   — `self.recognition_times = []`
   — `self.frame_times = []`
3. Measure Detection and Recognition Times:
   — Use `time.time()` to record the start and end times for face detection and recognition.
   — Append the measured times to the respective lists.
4. Calculate Frame Rate:Measure the time taken to process each frame and calculate the frame rate by taking the reciprocal of the average frame time.
5. Display Metrics: Calculate and print the average detection time, recognition time, and frame rate in the `display_metrics` method.

C) Results:

Metric	Value	Description
Average Detection Time	0.1507 seconds	This metric measures the average time taken by the system to detect a face in a frame. A detection time of 0.1507 seconds is relatively fast and indicates that the system can quickly locate faces within images.
Average Recognition Time	0.1435 seconds	This metric measures the average time taken by the system to recognize a detected face. A recognition time of 0.1435 seconds is also quite fast and suggests that the system can quickly process and identify faces once they are detected.
Frame Rate	5.09 FPS	The frame rate indicates how many frames per second the system can process. A frame rate of 5.09 FPS is decent for many real-time applications but may be insufficient for high-speed requirements such as fast-moving surveillance or interactive systems.

Table 4.1 – System Performance Metrics

4.5 Real-life Experience

4.5.1 Qualitative Results

Now we will present the experience of the system in a real-life experience that took place in front of the entrance door of the Department of Computer Science at the University of 8 May 1945.

As we see here, the two apparent individuals were identified by their faces correctly. The bounding box is drawn around the both of faces, and their names,person-id,category are printed above that bounding box. The blue box is displayed for the campus workers. And a Green bounding box for Students. Also showing the ability to detect and recognize multiple faces at one-time.

Figure 4.6 – An example of the identification of a student and a worker in the entry door

Then, we notice that the system keeps knowledge of the student's identity(ZIED). But he can't recognize who the other person was, because his face was curled up to another side(Pose invariant).

Figure 4.7 – An example of a recognized face and unrecognized face

We continue, While maintaining knowledge of the student's identity, the system identified the identity of the person who had not been able to identify him in 2 different frames and two different face poses ,despite not looking at the camera. As usual The blue bounding box is drawn around the face of a worker and the name,person-id,category is printed above the the bounding box.

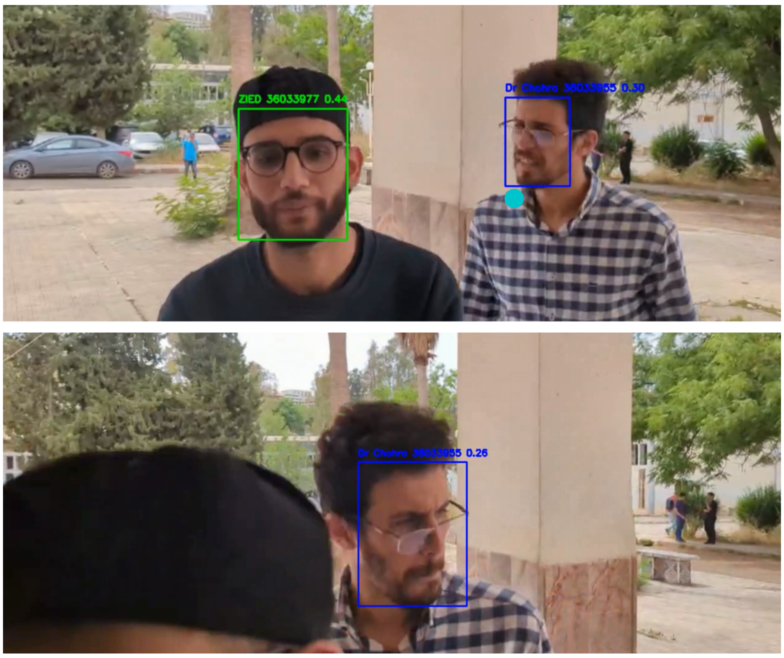

Figure 4.8 – 2 different frames showing the abilty of the system to recognize 2 faces

4.5.2 Interface Components

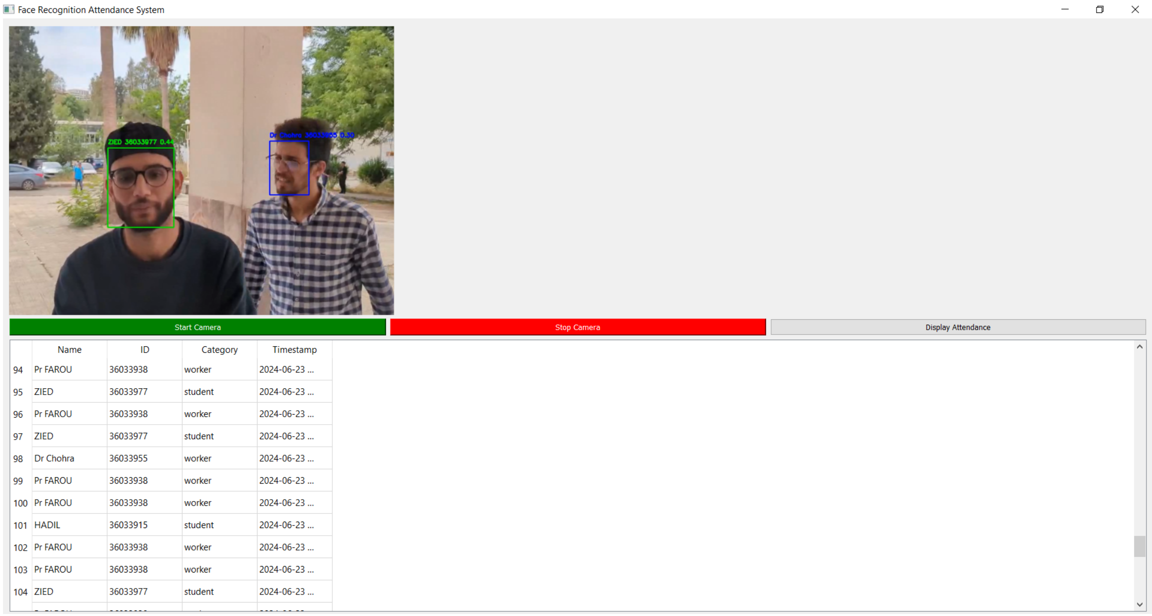

Figure 4.9 – The user interface the real-time face recognition

1. **Camera Feed Display**
   - **Description**: The left side of the interface shows a live video feed from a camera.
   - **Functionality**: This section displays the faces detected in real-time. Each face is enclosed in a colored rectangle, likely indicating the system's detection and recognition status. The bounding boxes around the faces include labels with the names "ZIED" and "Dr Chohra," along with some numerical values (presumably ID numbers and confidence scores).
2. **Control Buttons**
   - **Start Camera**: A green button to start the camera feed.
   - **Stop Camera**: A red button to stop the camera feed.
   - **Display Attendance**: A gray button to display the attendance records.
3. **Attendance Log Table**
   - **Columns**:
     - **Name**: Names of the individuals recognized by the system.
     - **ID**: ID numbers associated with each individual.
     - **Category**: Classification of individuals, such as "worker" or "student".
     - **Timestamp**: Date and time when the individual was recognized and their attendance was recorded.

— **Description**: This table shows a log of attendance entries. Each row represents a recognition event, listing the name, ID, category, and timestamp of the recognized individual.

4.5.3 Detailed Analysis

1. **Face Detection and Recognition**
   — **Detection**: Our system uses bounding boxes (green and blue rectangles) to indicate detected faces.
      — Green boxes is to indicate sudents faces that have been successfully recognized.
      — Green boxes is to indicate workers faces that have been successfully recognized.

   — **Recognition Labels**: The labels within the boxes display the recognized name, ID number, and a confidence score. This provides immediate visual feedback about the system's recognition capabilities.
2. **User Interface (UI)**
   — **User-Friendly Design**: The interface is simple and user-friendly, with clearly labeled buttons and a straightforward layout. This simplicity is important for ensuring that users with varying levels of technical expertise can operate the system efficiently.

   — **Real-Time Feedback**: The live video feed combined with real-time recognition labels provides instant feedback, which is crucial for security and monitoring purposes.
3. **Attendance Logging**
   — **Comprehensive Records**: The attendance log table keeps detailed records of each recognition event, including precise timestamps. This feature is essential for institutions that need accurate attendance records for security and administrative purposes.

   — **Data Categorization**: The inclusion of a "Category" column helps differentiate between different types of users (e.g., workers, students), which can be useful for generating specific reports or analyses.

4.5.4 Discussion

The results demonstrate that the face recognition system is highly accurate and efficient, with no errors in classification. The system's ability to detect and recognize faces quickly, coupled with its perfect accuracy metrics, suggests that it is well-suited for applications where reliability and speed are critical. However, the frame rate of 5.09 FPS may need to be improved for applications requiring very high-speed processing, such as real-time surveillance in high-traffic areas or interactive systems with rapid user interactions.

To further enhance the system, future work could focus on improving the frame rate to support higher-speed applications and exploring ways to maintain high performance under varying conditions, such as different lighting, angles, and facial expressions. Additionally, the system could be tested against a more diverse dataset to ensure robustness across different demographics and scenarios.

4.5.5 Limitations

- **Dataset Size and Overfitting:**
  - **Small Dataset:** A dataset with 1060 images can be considered small, especially for tasks like facial recognition that require a large amount of data to generalize well.
  - **Overfitting:** Overfitting occurs when a model learns the noise and details in the training data to an extent that it negatively impacts the model's performance on new data. This is common with small datasets where the model memorizes the training data rather than learning the underlying patterns.
- **Data Augmentation:**
  - **Impact:** While data augmentation helps in increasing the dataset size and introducing variability, it does not replace the need for more diverse real data. Augmented data is still derived from the original data, and the model may still not generalize well to completely new images.
  - **Limited Effectiveness:** Augmentation cannot fully simulate the variety found in real-world scenarios, such as different lighting conditions, occlusions, or variations in facial expressions that were not originally present in the dataset.
- **Resource Constraints:**
  - **Hardware Limitations:** Training deep learning models requires high-performance hardware. Without access to sufficient computational resources, training times become impractical, and memory limitations cause crashes.
  - **Feasibility of Training:** Due to hardware constraints, it's challenging to experiment with larger datasets or more complex models that could potentially mitigate overfitting.

In this study, we worked with a dataset of 1060 images, which is relatively small for facial recognition tasks. This small dataset size presents challenges in achieving a model that generalizes well to unseen data, as indicated by the perfect accuracy scores obtained during cross-validation. Despite employing data augmentation techniques to artificially increase the dataset size, the augmented data does not fully replicate the diversity found in real-world scenarios. Additionally, due to hardware limitations, training on larger datasets or more complex models is infeasible, often leading to system crashes or impractically long training times. Therefore, while the current results demonstrate high accuracy on the provided dataset, they may not accurately reflect the model's performance in a real-world setting. Future work should focus on expanding the dataset and leveraging more powerful computational resources to build a more robust and generalizable model.

4.6 Conclusion

The proposed face recognition system demonstrates exceptional accuracy and efficiency, making it ideal for applications where reliability and speed are paramount. However, the current frame rate of 5.09 FPS necessitates improvements for high-speed processing requirements. Future efforts should focus on enhancing the frame rate, ensuring high performance under varying conditions, and testing the system against a more diverse dataset to bolster robustness. The study's limitations, including a small dataset size and hardware constraints, underscore the need for more extensive data and computational resources to achieve a model that generalizes well to real-world scenarios. Addressing these issues will be crucial in developing a more robust and versatile face recognition system.

General Conclusion

The integration of real-time facial recognition technology into institutional security systems addresses critical challenges in monitoring and managing access, particularly as institutions seek to enhance security measures and streamline administrative tasks. The growing need for effective security solutions has led to the exploration of advanced technologies capable of providing accurate and efficient identification of individuals.

In this work, we tackled these challenges by developing a comprehensive real-time facial recognition system. This approach offers numerous benefits. It significantly enhances security by accurately identifying individuals in real-time and alerting authorities about unauthorized access. Additionally, it automates the attendance recording process, reducing manual errors and increasing operational efficiency. The system's user-friendly interface ensures accessibility for users with varying levels of technical expertise.

Research in the field of facial recognition has demonstrated that no single method can serve all applications effectively. Our system leverages advanced models like MTCNN for face detection and InceptionResnetV1 for face recognition, ensuring high accuracy and robustness even in complex scenarios. However, it is important to note that the effectiveness of facial recognition systems can be influenced by various factors, including the size and diversity of the dataset used for training.

While our system performs well, the limited dataset size presents a challenge, potentially leading to overfitting and reduced generalization to new data. Therefore, future work should focus on expanding the dataset and optimizing the system for larger-scale deployment. Additionally, integrating complementary features such as emotion detection and behavior analysis could provide a more holistic security solution.

In conclusion, all system objectives have been achieved. The real-time facial recognition system developed in this thesis provides a robust and effective solution for enhancing security and automating attendance management in institutional settings. As future work, we propose to:

- Expand the dataset to improve the model's generalization capabilities.
- Optimize the system to handle larger datasets and real-time processing require-

ments.

- Integrate additional features, such as emotion detection and behavior analysis, to provide a more comprehensive security solution.
- Ensure compliance with privacy regulations to protect individuals' data.

This approach is innovative and significantly improves the understanding and application of real-time facial recognition technology, offering valuable insights and practical recommendations for enhancing security and operational efficiency in various institutional contexts.

Bibliography

[ADH+19] Zhanfu An, Weihong Deng, Jiani Hu, Yaoyao Zhong, and Yuying Zhao. Apa: Adaptive pose alignment for pose-invariant face recognition. *IEEE Access*, 7:14653–14670, 2019.

[AS20] Mark Andrejevic and Neil Selwyn. Facial recognition technology in schools: Critical questions and concerns. *Learning, Media and Technology*, 45(2):115–128, 2020.

[ASS20] NADJETTE ASSADI. Mise au point d'une application de reconnaissance faciale. 2020.

[BCL23] Hicham Benradi, Ahmed Chater, and Abdelali Lasfar. A hybrid approach for face recognition using a convolutional neural network combined with feature extraction techniques. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 12:627–640, 06 2023.

[Bel13] Mébarka Belahcen. *Authentification et identification en biométrie*. PhD thesis, Université Mohamed Khider Biskra, 2013.

[BK17] Samiha BEGGARI and Khaoula KHAMRA. Système de reconnaissance de visage par un réseau de neurone convolutionnel (cnn). 2017.

[BP93] Roberto Brunelli and Tomaso Poggio. Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, 15(10):1042–1052, 1993.

[Buy11] Pierre Buyssens. Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux e_transactions. (fusion of different capture modes for face recognition applied to e_transactions). 2011.

[CCW+12] Dong Chen, Xudong Cao, Liwei Wang, Fang Wen, and Jian Sun. Bayesian face revisited: A joint formulation. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III 12*, pages 566–579. Springer, 2012.

[Cha20] Arindam Chaudhuri. Deep learning models for face recognition: A comparative analysis. *Deep biometrics*, pages 99–140, 2020.

[CSX+17] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74, 2017.

[DDS+09] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: a large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009*, pages 248–255. IEEE, Piscataway, 2009.

[EU19] Ali Elmahmudi and Hassan Ugail. Deep face recognition using imperfect facial data. *Future Generation Computer Systems*, 99, 04 2019.

[GWK+15] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, L. Wang, G. Wang, et al. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*, 2015.

[HZRS16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[KKK19] Ashu Kumar, Amandeep Kaur, and Munish Kumar. Face detection techniques: a review. *Artificial Intelligence Review*, 52:927–948, 2019.

[Kod14] Trupti M. Kodinariya. Hybrid approach to face recognition system using principle component and independent component with score based fusion process. *ArXiv*, abs/1401.0395, 2014.

[KSH12] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[LJD24] Stan Z. Li, Anil K. Jain, and Jiankang Deng, editors. *Handbook of Face Recognition*. Springer Cham, 3 edition, 2024. Published: 29 December 2023, Hardcover ISBN: 978-3-031-43566-9, Softcover ISBN: 978-3-031-43569-0.

[LSX20] Peng Lu, Baoye Song, and Lin Xu. Human face recognition based on convolutional neural network and augmented dataset. *Systems Science & Control Engineering*, 9:29 – 37, 2020.

[LSYC13] Mingxing Li, Weijun Su, Chongchong Yu, and Xiuxin Chen. Gait recognition bases on the compressed sensing. *2013 Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 407–410, 2013.

[Mou20] E. G. Moung. Face recognition state-of-the-art, enablers, challenges and solutions: A review. *International Journal of Advanced Trends in Computer Science and Engineering*, 9:96–105, 2020.

[NC17] Mina Navraan and Nasrollah Moghadam Charkari. Automatic facial emotion recognition method based on eye region changes. 2017.

[NFTT20] Arnauld Fountsop Nzegha, Jean Louis Kedieng Ebongue Fendji, Christopher Thron, and Clementin Djameni Tayou. Overview of deep learning in facial recognition. 2020.

[PD02] Florent Perronnin and Jean-Luc Dugelay. Introduction à la biométrie authentification des individus par traitement audio-vidéo. *Traitement du signal*, 19(4), 2002.

[PSP23] Ramesh Poonia, Debabrata Samanta, and P. Prabu. *Design and Implementation of Machine Learning-Based Hybrid Model for Face Recognition System*, pages 59–68. 03 2023.

[SCWT14] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *Advances in neural information processing systems*, 27, 2014.

[SDM23] Manu Shree, Amita Dev, and AK Mohapatra. Review on facial recognition system: Past, present, and future. In *Proceedings of International Conference on Data Science and Applications: ICDSA 2022, Volume 1*, pages 807–829. Springer, 2023.

[SIVA16] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *ArXiv*, abs/1602.07261, 2016.

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[SLJ+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[SSTS21] Prateek Singhal, Prabhat Kumar Srivastava, Arvind Kumar Tiwari, and Ratnesh Kumar Shukla. A survey: Approaches to facial detection and recognition with machine learning techniques. *Proceedings of Second Doctoral Symposium on Computational Intelligence*, 2021.

[SWT14a] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898, 2014.

[SWT14b] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1891–1898, 2014.

[SWT15] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deeply learned face representations are sparse, selective, and robust. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2892–2900, 2015.

[SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[TYRW14] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

[TYRW15] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Web-scale training for face identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2746–2754, 2015.

[VGKT22] Amit Verma, Aarti Goyal, Nitish Kumar, and Hitesh Tekchandani. Face recognition: a review and analysis. *Computational Intelligence in Data Mining: Proceedings of ICCIDM 2021*, pages 195–210, 2022.

[VJ01] Paul Viola and Michael Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. *Advances in neural information processing systems*, 14, 2001.

[VP16] MK Vidhyalakshmi and E Poovammal. A survey on face detection and person re-identification. In *Computational Intelligence in Data Mining—Volume 1: Proceedings of the International Conference on CIDM, 5-6 December 2015*, pages 283–292. Springer, 2016.

[Yan15] Ming-Hsuan Yang. *Face Detection*, pages 447–452. Springer US, Boston, MA, 2015.

[YKA02] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:34–58, 2002.

[YLH+19] Aihua Yu, Gang Li, Beiping Hou, Hongan Wang, and Gaoya Zhou. A novel framework for face recognition using robust local representation–based classification. *International Journal of Distributed Sensor Networks*, 15, 2019.

[ZF13] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *ArXiv*, abs/1311.2901, 2013.

[ZL18] Huiling Zhou and Kin-Man Lam. Age-invariant face recognition based on identity inference from appearance age. *Pattern Recognit.*, 76:191–202, 2018.

[ZZLQ16] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23:1499–1503, 2016.

Webography

[W1] , https://docs.real.com/guides/safrdocs/Camera_Best_Practices.html, Last accessed : 05/05/2024.

[W2] , https://deepomatic.com/blog/difference-between-computer-vision-and-image-recogn Last accessed : 05/05/2024.

[W3] , https://www.plugger.ai/blog/implementing-computer-vision-in-face-detection, Last accessed : 05/05/2024.

[W4] , https://www.python.org/doc/essays/blurb/, Last accessed : 24/06/2024.

[W5] , https://jupyter.org, Last accessed : 24/06/2024.

[W6] , https://opencv.org/, Last accessed: 19/06/2024.

[W7] , https://numpy.org/doc/stable/user/absolute_beginners.html, Last accessed: 19/06/2024.

[W8] , https://www.w3schools.com/python/numpy/numpy_intro.asp, Last accessed: 19/06/2024.

[W9] , https://www.programiz.com/python-programming/numpy/introduction, Last accessed: 19/06/2024.

[W10] , https://pytorch.org/docs/stable/index.html, Last accessed: 19/06/2024.

[W11] , https://pypi.org/project/facenet-pytorch/, Last accessed: 19/06/2024.

[W12] , https://pandas.pydata.org/pandas-docs/dev/index.html, Last accessed: 19/06/2024.

[W13] , https://pillow.readthedocs.io/en/stable/handbook/overview.html, Last accessed: 19/06/2024.

[W14] , https://imgaug.readthedocs.io/en/latest/, Last accessed: 19/06/2024.

[W15] , https://docs.python.org/3/library/datetime.html, Last accessed: 19/06/2024.

[W16] , https://www.sqlite.org/docs.html, Last accessed: 19/06/2024.

[W17] , https://www.w3schools.com/sqlite/default.asp, Last accessed: 19/06/2024.

[W18] , https://doc.qt.io/qt-5/pyqt5.html, Last accessed: 19/06/2024.

[W19] , https://realpython.com/python-gui-tutorial/, Last accessed: 19/06/2024.

[W20] , https://www.geeksforgeeks.org/ml-introduction-to-transfer-learning/, Last accessed: 19/06/2024.

[W21] , https://datascientest.com/transfer-learning, Last accessed: 19/06/2024.

START-UP ANNEX

Presentation of the Project

1. Project Idea (Proposed Solution)

- **Activity Area**: Security sector, with applications in institutions, workplaces, and educational facilities.
- **Project Concept**: Enhance security and streamline attendance management using real-time facial recognition at entry points.
- **Solution Description**:
  - Place surveillance cameras at entry points to capture faces in real time.
  - Recognize and verify identities against a database.
  - Generate alerts for unknown or unauthorized individuals.
  - Automatically record attendance and departures for employees and students.

2. Proposed Values

- **Enhanced Security**: Real-time monitoring and alert system for unauthorized entries.
- **Efficiency**: Automated attendance tracking reduces manual processes.
- **Accuracy**: High precision in identity verification.
- **User-Friendly Interface**: Easy-to-use system for administrators and users.

3. Working team

- Bourdima Youssef: has skills in Deep learning and app development, Design and User Interface Design.
- Farou Brahim: has skills in IT and project management.
- Bourdima Youssef's role is: Responsible for the development design of the App, integrates real-time monitoring, creating attractive and functional interfaces, ensures (UX/UI).
- Farou Brahim’s role: Lead the project.

4. Objectives of the Project

- **Short Term**: Develop and deploy the system in a pilot institution to test and refine the technology.
- **Medium Term**: Expand to multiple institutions, including workplaces and educational facilities.
- **Long Term**: Become a leading provider of real-time facial recognition security systems with a significant market share in the security and attendance management sector.

5. Project Delivery Schedule

		1m	2m	3m	4m	5m	6m
1	Preliminary studies: Local market analysis, Choice of business model.	√	√				
2	Platform development			√	√	√	
3	Testing and Launch	√					
4	Marketing and Promotion						√

Innovative Aspects

1. First in the Region

- **Pioneering Technology**: First system in the region to implement real-time facial recognition for both security and attendance management. This advanced technology sets a new standard for institutional security and operational efficiency.

2. Advanced Recognition Algorithms

- **High Precision and Speed**: Utilizes cutting-edge facial recognition algorithms that ensure high accuracy and quick identification, even in real-time and in crowded environments. This ensures seamless and reliable performance in various institutional settings.

3. Comprehensive Security and Attendance Solution

- **Dual Functionality**: Combines security monitoring with automated attendance tracking, providing a comprehensive solution that addresses multiple institutional needs with a single system. This integration simplifies operations and improves efficiency.

4. User-Centric Design

- **Intuitive Interface**: The system is designed with a user-friendly interface that makes it easy for administrators to manage and monitor. Users (employees, students) experience a seamless and unobtrusive process, enhancing overall satisfaction.

5. Data Privacy and Security

- **Secure Data Handling**: Implements robust data privacy and security measures to protect sensitive information. This includes compliance with data protection regulations (such as GDPR), ensuring that user data is handled responsibly and securely.

6. Scalability and Flexibility

- **Adaptable System**: The system is designed to be scalable and adaptable to various institutional sizes and requirements. This flexibility allows for easy customization and expansion as the institution grows or as needs change.

7. Environmental Commitment

- **Energy-Efficient Technology**: Incorporates energy-efficient hardware and algorithms to minimize the environmental impact. This commitment to sustainability aligns with broader institutional goals of reducing carbon footprints and promoting eco-friendly practices.

8. New Market Segment Targeting

- **Broad Institutional Appeal**: Targets a new segment of institutions that prioritize advanced security measures and operational efficiency. By addressing specific needs in educational facilities, corporate offices, and secure government buildings, the system enhances convenience and satisfaction for a wide range of users.

<u>Strategic Market Analysis</u>

1. Market Segment

- **Potential Market**: Institutions needing enhanced security and attendance management.
- **Target Market**:
  - Educational institutions (schools, universities).
  - Corporate offices and enterprises.
  - Government buildings and secure facilities.
  - Healthcare facilities.
- **Key Stakeholders**:
  - Security personnel.
  - HR departments.
  - IT departments.

2. Measuring the Intensity of Competition

- **Competitors**:
  - Established security system providers.
  - Companies offering biometric authentication solutions.
- **Competitor Strengths**:
  - Established customer base.
  - Advanced technology and experience.
- **Competitor Weaknesses**:
  - Lack of integrated attendance management features.
  - Limited adaptability to various institutional needs.

3. Marketing Strategy

- **Cost Reduction**: Offer competitive pricing with high-value features to attract institutions.
- **Promotional Offers**: Free trials, discounts for early adopters, and referral programs.
- **Market Penetration**: Target initial markets with high security needs and expand based on success.

Production and Organization Plan

1. **Production Process**
    - **System Design and Development**:
        - Facial recognition algorithm development.
        - Integration with existing security and attendance systems.
        - User interface design for administrators and users.
    - **Partnerships**:
        - Collaboration with camera manufacturers.
        - Partnerships with security firms for system deployment.
    - **Testing and Deployment**:
        - Pilot testing in a selected institution.
        - Gathering feedback and refining the system.
    - **Marketing Launch**:
        - Announce the system launch through industry events and digital marketing.
2. **Procurement**
    - **Hardware**: Surveillance cameras, servers, and networking equipment.
    - **Software**: Facial recognition algorithms, database management systems, and user interface design tools.
3. **Workforce**
    - **Development Team**: Software developers, data scientists, UI/UX designers.
    - **Implementation Team**: Security experts, IT specialists, installation technicians.

- **Support Team**: Customer support representatives, maintenance technicians.

4. Main Partners

- **Technology Partners**: Companies providing advanced facial recognition technology and camera hardware.
- **Deployment Partners**: Security firms and IT service providers for installation and maintenance.
- **Institutional Partners**: Early adopter institutions for pilot testing and feedback.

Financial Plan

1. Costs and Charges

To efficiently manage and deliver a robust facial recognition system, it is crucial to accurately determine all the costs and investments required. This includes initial, operational, and recurring costs. Key considerations include:

1.1 Initial Costs

1.1.1 Infrastructure:

- **Office Space**: Purchase or lease of office premises.
- **Setup**: Office layout, furniture, and essential infrastructure.

1.1.2 Equipment:

- **Hardware**:
    - Surveillance cameras with high resolution and facial recognition capabilities.
    - Servers for data processing and storage.
    - Workstations for development and monitoring.
- **Material Handling Equipment**:
    - Mounting equipment for cameras.
    - Network setup tools and cables.

1.1.3 Technology:

- **Software Development**:
    - Facial recognition algorithm development tools.

    - Database management systems.
    - Security software for data protection.
- **Order Management Systems**:
    - ERP software for internal management.
    - Delivery tracking and integration systems.

1.2 Operational Costs

1.2.1 Personnel:

- **Salaries**:
    - Wages for software developers, data scientists, and UI/UX designers.
    - Compensation for IT support, installation technicians, and customer service representatives.
- **Training**:
    - Continuous training for staff on the latest facial recognition technologies and security protocols.

1.2.2 Logistics:

- **Installation and Maintenance**:
    - Travel expenses for installation teams.
    - Costs associated with vehicle maintenance and fuel.
- **Packaging**:
    - Secure packaging for transporting sensitive hardware components.

1.2.3 Marketing and Customer Service:

- **Marketing Campaigns**:
    - Digital marketing efforts to promote the system.

    - Industry events and trade shows to showcase the technology.
- **Customer Service Management**:
    - Support services for handling inquiries, complaints, and system troubleshooting.

1.2.4 Other Costs:

- **Insurance**:
    - Insurance for equipment and liability coverage.
- **Licenses and Permits**:
    - Necessary legal permits and software licenses.

1.3 Recurring Costs

- **Maintenance**:
    - Regular maintenance of cameras, servers, and other hardware.
    - Software updates and system improvements.
- **Licenses and Permits**:
    - Annual renewal of licenses and permits.
- **IT System Maintenance**:
    - Continuous updates and cybersecurity measures for computer systems.

2. Methods and Sources of Financing

To finance this project, several funding methods and sources can be explored:

2.1 Internal Financing

- **Company's Own Funds**:

    - Utilizing available company funds.
- **Reinvestment of Profits**:
    - Reinvesting earnings from initial deployments and services.

2.2 External Financing

2.2.1 Bank Loans:

- **Long-term Loans**:
    - Loans to cover significant upfront costs such as equipment and infrastructure.
- **Lines of Credit**:
    - Short-term credit lines for operational expenses.

2.2.2 Investors:

- **Private Investors and Venture Capital**:
    - Attracting investors interested in innovative security technologies.

2.2.3 Grants and Aids:

- **Government Grants**:
    - Applying for government support for innovative security solutions.
- **Incubation Programs**:
    - Joining start-up incubation and acceleration programs for additional support and funding.

2.2.4 Crowdfunding:

- **Crowdfunding Platforms**:
    - Leveraging platforms like Kickstarter or Indiegogo to raise funds from small investors.

3. Obtaining the Refund

A detailed payment schedule can help plan the repayment of borrowed or invested funds. This table should include:

3.1 Repayment Planning

3.1.1 Payment Schedule:

- **Detailed Repayment Plans**:
    - Specific amounts to be repaid and timelines for each funding source.
    - Including grace periods and applicable interest rates.

3.1.2 Cash Flow:

- **Forecasting**:
    - Projecting cash flow to ensure the company can meet repayment deadlines.
- **Expenditure Adjustment**:
    - Balancing expenditures and revenues to avoid financial deficits.

Date	Amount Payable	Funding Type	Maturity	Balance Remaining	Notes
01/07/2024	1000000 DZD	Bank loan	Monthly	900000 DZD	Interest rate 5%
01/08/2024	1000000 DZD	Bank loan	Monthly	800000 DZD	
01/09/2024	1000000 DZD	Bank loan	Monthly	700000 DZD	
01/10/2024	500000 DZD	Private investor	Quarterly	200000 DZD	Profit sharing

Business Model Canvas

Designed For:

Designed by: BOURDIMA YOUSSEF

Date: 24-06-2024

Version:

Key Partners

- Hardware Manufacturers: Companies that provide high-quality facial recognition cameras, servers, and access control hardware.
- Software Development Partners: Collaboration with AI and machine learning experts to enhance facial recognition algorithms.
- Installation and Integration Services: Companies with expertise in installing and integrating the system with existing security infrastructure.
- Data Security Specialists: Partners with a strong track record in data security to ensure compliance and build trust with customers.

Key Activities

- Proposed Value
  - Develop and improve facial recognition algorithms for accuracy, speed, and security.
  - Integrate access control features with facial recognition for real-time identification.
  - Design an attendance management system for automatic student and staff tracking.
  - Ensure data security and privacy compliance for facial recognition information.

Key Resources

- Material Resources: High-quality surveillance cameras, facial recognition software, computing infrastructure, and networking equipment.
- Human Resources: Trained personnel for system installation, technical support, and customer service.

Value Proportions

- Product: Facial recognition security and attendance system with real-time identification and access control.
- Enhanced security: Real-time monitoring and alerts for unauthorized entry.
- Automated attendance tracking: Simplified and accurate recording of employee and student attendance.
- User-friendly interface: Easy-to-use system for administrators.
- Data storage and retrieval: Secure and efficient management of attendance and entry data.

Customer Relationships

- Attraction: Free demos, case studies showcasing successful implementations, highlighting return on investment (ROI) in security and efficiency.
- Retention: Software updates with new features, flexible pricing plans based on university size and needs, loyalty programs for referrals.

Channels

- Direct Sales: Sales representatives contacting universities directly to understand their needs and propose customized solutions.

Customer Segments

- Educational institutions: Universities, colleges, and schools.
- Corporate offices: Companies requiring secure access control.
- Event organizers: Large-scale events needing enhanced security.

Cost Structure

- Highest-Cost Key Resources: Personnel costs for software development and technical support teams.
- Most Costly Key Activities: Research and development for continuous improvement of the facial recognition system, customer acquisition and support, and marketing expenses.
- Major Cost Categories:
- Personnel: Salaries, benefits, and training for employees.
- Technology: Software licensing fees, hardware purchases, and maintenance costs.

Revenue Streams

Value Propositions: Customers are willing to pay for enhanced security, streamlined attendance management.

- Payment Methods: One-time sales for hardware and software licenses, and potentially transaction-based fees for additional features or services.

Contribution to Total Revenue:

- Subscription Model: Expected to contribute a significant portion of recurring revenue, providing stability and predictability.
- Hardware Sales: Initial hardware sales may generate substantial revenue upfront, followed by additional revenue from maintenance contracts and upgrades.
- Service Fees: Additional revenue from training, support services, and customization requests.