UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI

BACHELOR'S DEGREE

Research and Development

BACHELOR THESIS

by

LE Phuong Thu

BI11-258

Information and Communication Technology

Title:

Facial Recognition Attendance System Using Python And Deep Learning

Supervisors: Mr. BUI Hai Dang - KSE Software Solutions

Dr. NGUYEN Minh Huong - USTH ICT Lab

Hanoi, October 2023

Attestation

I am with this, certify that my work and results are not contained plagiarism (copy/paste) from other sources.
In addition, all assessments, comments, and statistics from other authors and organizations are indicated and have been cited accordingly. In case of plagiarism in my report, I know the consequences and understand that my report will not be evaluated.
In this case, my bachelor's internship will be noted as a "failed".

Hanoi, October 2023
Signature

LE Phuong Thu

Acknowledgements

The internship opportunity that the University of Science and Technology of Hanoi gave me was an important stepping stone in my career. I appreciate this great chance to work professionally beside outstanding researchers and colleagues.

My thesis would not have been possible without the help of many people I wish to thank. I would love to express my warmest thanks to my supervisor, Mr. BUI Hai Dang, for his precious feedback and constructive instruction, which led me through the internship.

In addition, I take this opportunity to especially thank Dr. DOAN Nhat Quang and Dr. NGUYEN Minh Huong, who have been the ideal teachers, and thesis supervisors, offering advice and encouragement, always ready to give advice and the right direction, even though my project in the process has many big changes that make it difficult for me.

Not only that, I want to express my gratitude to Mr. HUYNH Vinh Nam, Mr. KIEU Quoc Viet, and Mr. NGUYEN Xuan Tung, who were so eager to assist me and point out my mistakes. Thanks should also go to all my friends and colleagues, for their encouragement and assistance.

Last but not least, I would like to express my deep and sincere gratitude to my family for their continuous and incomparable love, and unwavering support throughout my life, always supported me through my journey in USTH.

(LE Phuong Thu)
Hanoi, October 2023

Contents

List of Acronyms

AI	Artificial Intelligence.
CNN	Convolutional Neural Network.
FCN	Fully Convolutional Network.
FRAS	Facial Recognition Attendance System.
FRT	Facial Recognition Technology.
GUI	Graphical User Interface.
MTCNN	Multi-task Cascaded Convolutional Networks.
NMS	Non-Maximum Suppression.
PIN	Personal Identification Number.
RFID	Radio-Frequency Identification.

List of Figures

List of Tables

Abstract

Facial recognition technology has become increasingly popular in recent years due to its high level of accuracy and convenience in various applications. One of its most common uses is in the workplace, where it is employed in the Facial Recognition Attendance System.

This research focuses on the development of a real-time facial recognition system implemented using Python and powered by state-of-the-art deep learning techniques. The presented system incorporates two significant modules: face detection and face recognition. Face detection is facilitated through the Multi-task Cascaded Convolutional Networks (MTCNN), which locates faces within an image while providing high-precision facial landmarks. For face recognition, the model uses MobileFaceNets, a lightweight, efficient model designed explicitly for mobile and embedded vision applications. It serves to map detected faces into a latent space to facilitate comparison and recognition.

Keywords: *Face detection, Face recognition, Face verification, MobileFaceNets, MTCNN, Convolutional Neural Networks, Deep Learning*

Chapter 1

Introduction

1.1 Context and Motivation

In the contemporary milieu of technological progression, the meticulous administration of attendance persists as a pivotal endeavor across multifarious domains encompassing educational institutions, corporate enclaves, and event orchestrations. Conventional methodologies for attendance monitoring often entail manual processes or reliance upon identification tokens such as ID cards, or biometric data, including fingerprints. However, these methodologies are fraught with susceptibilities to inaccuracies, coupled with exigent concerns pertaining to scalability and user convenience. As organizations burgeon in magnitude and intricacy, the exigency for an astute and automated attendance management system becomes patently discernible.

Facial Recognition Technology (FRT) has emerged as a seminal solution to ameliorate the constraints tethered to conventional paradigms of attendance tracking. FRT capitalizes upon the idiosyncratic and innate attributes intrinsic to an individual's countenance, proffering an unobtrusive and efficient modality of identification and validation. The evolution of this technology has been conspicuously driven by the proliferation of deep learning techniques.

The crux of FRT's triumph resides in the leverage of deep learning models, particularly Convolutional Neural Networks (CNNs). These models evince an innate capacity to discern intricate patterns and attributes embedded within images, thus rendering them germane to the intricate task of facial recognition. Through erudition on diverse datasets that encapsulate a spectrum of illuminatory conditions, facial expressions, and orientations, these models manifest the propensity to generalize and discern countenances with striking acumen and resilience.

The ubiquitous assimilation of Facial Recognition Technology has impelled its assimilation into a mélange of applications within contemporary society's purview. In the precincts of security and law enforcement, FRT assumes an instrumental role in the orchestration of surveillance

systems, affording expeditious identification of potential threats and malefactors. Furthermore, the technology has permeated the consumer ambit by being inculcated into smartphones, thereby endowing users with secure and expedient mechanisms for device authentication. Social media platforms also harness FRT to proffer automated tagging suggestions for individuals in photographs, concomitantly enhancing the user experience.

Of particular importance, the prospective utility of FRT in attendance tracking has not evaded discernment. Academic institutions and corporate entities are increasingly cognizant of the merits inherent in this technology, as it promulgates the streamlining of administrative processes. A judiciously devised Facial Recognition Attendance System (FRAS) possesses the potential to alleviate the exigencies of manual attendance management, proffering contemporaneous, precision-driven, and automated monitoring solutions.

The development and implementation of facial recognition technology also push the boundaries of Artificial Intelligence, Machine Learning, and Deep Learning. The ensuing segments of this thesis undertake an exploration into the development and evaluation of a FRAS, materializing via the integration of Python and deep learning. By capitalizing upon these advancements, the system aspires to redefine the contours of attendance management paradigms, furnishing an astute and efficacious alternative to established methodologies. The ensuing sections of this exposition expound upon the architectural underpinnings, technical instantiation, and systematic assessment of the FRAS, underscored by its intrinsic potential to transfigure the landscape of attendance tracking paradigms.

Nevertheless, it always has been a challenging computer vision issue for years, even until recently. It's essential to also consider the ethical implications, such as privacy concerns and potential misuse. As we delve deeper into the realm of facial recognition technology, it becomes apparent that the delicate equilibrium between its advantages and responsible application is of paramount significance for its prospective trajectory. Hence, our proposition materializes as a response to this imperative: a solution in the form of a Facial Recognition Attendance System, employing cutting-edge models in the domain of Deep Learning, with the intent to address and resolve this intricate concern.

1.2 Research Objectives

We desire to deploy an interaction system design for the facial recognition application that is suitable for the specific needs of human recognition in companies and institutions, which can provide feedback and accept user commands. The core problem addressed in this research is the development of a robust and accurate facial recognition attendance system. This system must effectively detect faces in images or video streams, recognize individual faces, and record attendance based on these identifications. Key challenges include handling variations in lighting

conditions, facial expressions, and pose, as well as ensuring real-time performance and high accuracy. The main research objectives of this internship are as follows:

- To implement face detection using state-of-the-art techniques, specifically MTCNN (Multi-task Cascaded Convolutional Networks).
- To employ deep learning, specifically MobileFaceNets, for face recognition within the detected faces.
- To design and implement a Python-based system that integrates these components into a cohesive facial recognition attendance system.
- To evaluate the system's performance in terms of accuracy, speed, and robustness under various conditions.

1.3 Scope and Limitations

This study entails an investigation into the design, development, and assessment of a facial recognition attendance system utilizing Python and deep learning methodologies. The research primarily centers on the technical facets of system implementation and performance evaluation. Nevertheless, it is imperative to duly acknowledge specific constraints, which include:

- Ethical and legal considerations pertaining to privacy and data protection will be acknowledged in the study, although an exhaustive analysis will not be undertaken.
- The research will not extensively delve into the hardware aspects associated with facial recognition systems, such as camera selection. Rather, it will concentrate on the software and algorithmic components.
- The implementation and testing phases will be conducted within controlled environments, and while real-world deployment considerations will be addressed, a comprehensive exploration of this aspect will not be undertaken.

1.4 Organization of Thesis

In this section, we will summarize the content of each chapter thoroughly:

- **Chapter 2: Literature Review** explores existing research on facial recognition technologies, deep learning, and attendance systems.
- **Chapter 3: Methodology** elucidates the research design, tools, and data collection processes.

- **Chapter 4: Experimental Setup** details the codebase and algorithmic intricacies of the facial recognition attendance system.
- **Chapter 5: Results and Discussion** provides a comprehensive assessment of the system's performance metrics.
- **Chapter 6: Conclusion and Future Work** summarizes the research, delineates its implications, and suggests avenues for subsequent inquiries.

Chapter 2

Literature Review

This chapter aims to provide a comprehensive overview of the existing literature in the field of facial recognition, delving into the underlying principles, technological advancements, and the diverse array of applications it encompasses.

2.1 Facial Recognition Technologies

Facial recognition, situated within the domain of computer vision and artificial intelligence, represents an advanced technological discipline focused on the intricate process of ascertaining the identity or authentication of individuals based on their distinctive facial attributes. This technology has garnered substantial attention and scholarly interest in recent years, primarily attributable to its multifaceted utility across a broad spectrum of domains. Facial recognition systems are meticulously crafted to execute the intricate task of capturing, analyzing, and juxtaposing facial patterns, often quantified as mathematical descriptors, against an extensive repository of well-documented individuals. The manifold applications of facial recognition encompass critical domains such as security, surveillance, access control, and even human-computer interaction. The capacity for automatic and precise recognition and authentication of individuals predicated upon their facial characteristics holds immense promise and potential, especially in the context of attendance tracking systems, where it stands to notably enhance both efficiency and security protocols.

Facial recognition technology relies upon a sophisticated combination of image acquisition, feature extraction, and pattern-matching algorithms. In this process, a camera or sensor captures an image or video of an individual's face. Subsequently, the system extracts distinctive facial features, such as the arrangement of eyes, nose, mouth, and the geometry of facial landmarks. These features are then translated into mathematical representations, commonly known as facial descriptors or feature vectors, which serve as a basis for comparison.

Facial recognition technology boasts extensive applications with profound implications. Within the realm of security, it assumes a pivotal role in identity authentication and access management, providing a non-invasive yet exceedingly dependable method for ascertaining an individual's identity. In the context of surveillance, facial recognition empowers automated monitoring of public environments, offering the potential to promptly identify individuals of concern in real-time. Furthermore, in the sphere of human-computer interaction, this technology fosters natural and intuitive user engagements, enabling individuals to gain access to devices or systems through the simple presentation of their visage, thereby obviating the necessity for passwords or personal identification numbers (sPIN).

The potential impact of facial recognition technology on attendance tracking systems is particularly noteworthy within the academic, corporate, and public sectors. Traditional attendance tracking methods, such as manual paper-based processes or card swiping systems, are susceptible to various shortcomings, including fraud, inefficiency, and errors. Facial recognition offers a compelling alternative by automating the attendance recording process, thereby improving accuracy and mitigating these issues. Moreover, the technology can enhance security by ensuring that only authorized individuals gain access to sensitive areas or resources.

The development of facial recognition systems can be traced back to the mid-20th century when researchers first began exploring the potential of computer-based facial analysis. Notable milestones include the work of Woody Bledsoe, Helen Chan Wolf, and Charles Bisson, who developed early facial recognition algorithms in the 1960s. However, these early systems were limited by computational power and lacked the sophistication of modern deep learning-based approaches.

2.2 Related Works

Facial recognition attendance systems are relatively recent, with the first commercial systems appearing in the 2010s. There has been a considerable amount of work done in both the technology's development and its social implications. This field is rapidly evolving, with new research and developments appearing regularly. It is an area that encompasses a wide range of disciplines, from computer science to law to social science.

The field of facial recognition attendance systems encompasses several aspects, including the creation of machine learning algorithms such as DeepFace[1] and DeepID[2], the development and usage of large datasets like 'Labeled Faces in the Wild'[3] and 'MS-Celeb-1M'[4] for training, and the emergence of commercial systems that offer efficient attendance tracking. However, concerns have been raised about privacy and biases, as demonstrated by research from the MIT Media Lab. Lastly, legislation such as the EU's has been implemented to regulate the use of these technologies and protect personal data. Thus, the field is a dynamic mixture of technological progress, social impact studies, and evolving regulation.

Traditional attendance systems have relied on methods such as manual sign-in sheets, RFID cards, or fingerprint scanners. These methods are often prone to inaccuracies, time-consuming, and susceptible to fraud. Facial recognition-based attendance systems offer a promising alternative by automating the attendance tracking process. However, existing systems still face challenges related to accuracy, robustness in varying lighting conditions, and privacy concerns. Addressing these limitations is crucial for the successful implementation of facial recognition in attendance systems. Deep learning-based methods have become the primary choice to achieve the stated requirements of high accuracy and real-time performance due to their superior performance compared to traditional machine learning algorithms.

2.3 Deep Learning in Facial Recognition

Deep learning is a subset of machine learning that teaches computer systems to mimic the way humans do, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain in order to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help optimize the accuracy. Deep learning technology is advancing many industries as it can handle large volumes of data and find correlations, transforming industries and will continue to be a critical component in all forms of automation and AI. However, the technology requires a solid understanding of the underlying algorithms and theories to use effectively. Deep learning is a crucial component in facial recognition attendance systems. It helps in accurately detecting and recognizing faces, which forms the basis for identifying individuals for attendance tracking. It is important to note that deep learning models must be trained with large, diverse datasets of face images to perform well. The models should also be tested and validated for biases to ensure they work fairly across different demographics. The use of deep learning in facial recognition attendance systems allows for contactless and automated attendance tracking, but it also raises privacy and consent issues. Using these systems responsibly and complying with data protection regulations is crucial.

Deep learning has emerged as a transformative force in the field of facial recognition, revolutionizing the way we perceive, process, and authenticate human faces. Deep learning, particularly Convolutional Neural Networks (CNNs), has played a pivotal role in addressing the inherent complexities of facial recognition by offering remarkable accuracy and robustness. It has fundamentally transformed facial recognition by automating feature extraction, enabling end-to-end learning, and providing robustness to variations. It has leveraged large-scale datasets, transfer learning, and advanced techniques to achieve state-of-the-art accuracy. However, it is essential to proceed with ethical and privacy considerations to harness the full potential of deep learning in facial recognition while safeguarding individual rights and interests.

2.4 Multi-task Cascaded Convolutional Networks

MTCNN (Multi-task Cascaded Convolutional Networks) is a widely recognized face detection framework known for its ability to detect faces accurately in real-time. It was introduced by Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao in the paper titled "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks" in 2016[5]. MTCNN was specifically developed to address the challenges associated with detecting faces at various scales and orientations. MTCNN employs a cascaded architecture with three stages to detect faces at multiple scales. Each stage refines the results, leading to high precision. Several studies have explored the application of MTCNN in various domains, emphasizing its robustness and efficiency. Since its introduction, the MTCNN method has been used extensively in many research studies and practical applications. For instance, Wang et al. (2018)[6] used MTCNN for detecting faces in low-light conditions, demonstrating its robustness.

The network is called 'Multi-task' because it is designed to perform three tasks simultaneously: face detection, facial landmarks detection (i.e., the position of eyes, nose, and mouth), and bounding box regression (i.e., improving the detection window position). The 'Cascaded' part refers to its structure of three networks (P-Net, R-Net, and O-Net) arranged in a cascaded framework operating in a coarse-to-fine manner. The P-Net is the first level of this cascading structure, generating candidate windows and their corresponding bounding box regression vectors. The R-Net steps in as the second level, further refining the initial bounding boxes and discarding false positives. Lastly, the O-Net performs the final refinement and facial landmark positioning. This carefully coordinated sequence of operations ensures high precision in the face detection process.

MTCNN is structured as a cascaded architecture consisting of three stages, each responsible for different tasks:

- **Stage 1**: Proposal Network (P-Net): In the first stage, the P-Net generates a set of candidate face regions (bounding boxes) by performing convolutional and pooling operations on the input image. These candidates include faces at different scales and locations.
- **Stage 2**: Refinement Network (R-Net): The second stage, R-Net, refines the bounding box proposals from the previous stage. It evaluates the candidates produced by the P-Net and eliminates false positives by applying more complex convolutional layers and non-maximum suppression (NMS). This stage aims to reduce the number of false positives.
- **Stage 3**: Output Network (O-Net): The final stage, O-Net, further refines the remaining bounding box proposals and conducts facial landmark detection. It is responsible for precisely locating facial landmarks (e.g., eyes, nose, and mouth) within the detected faces.

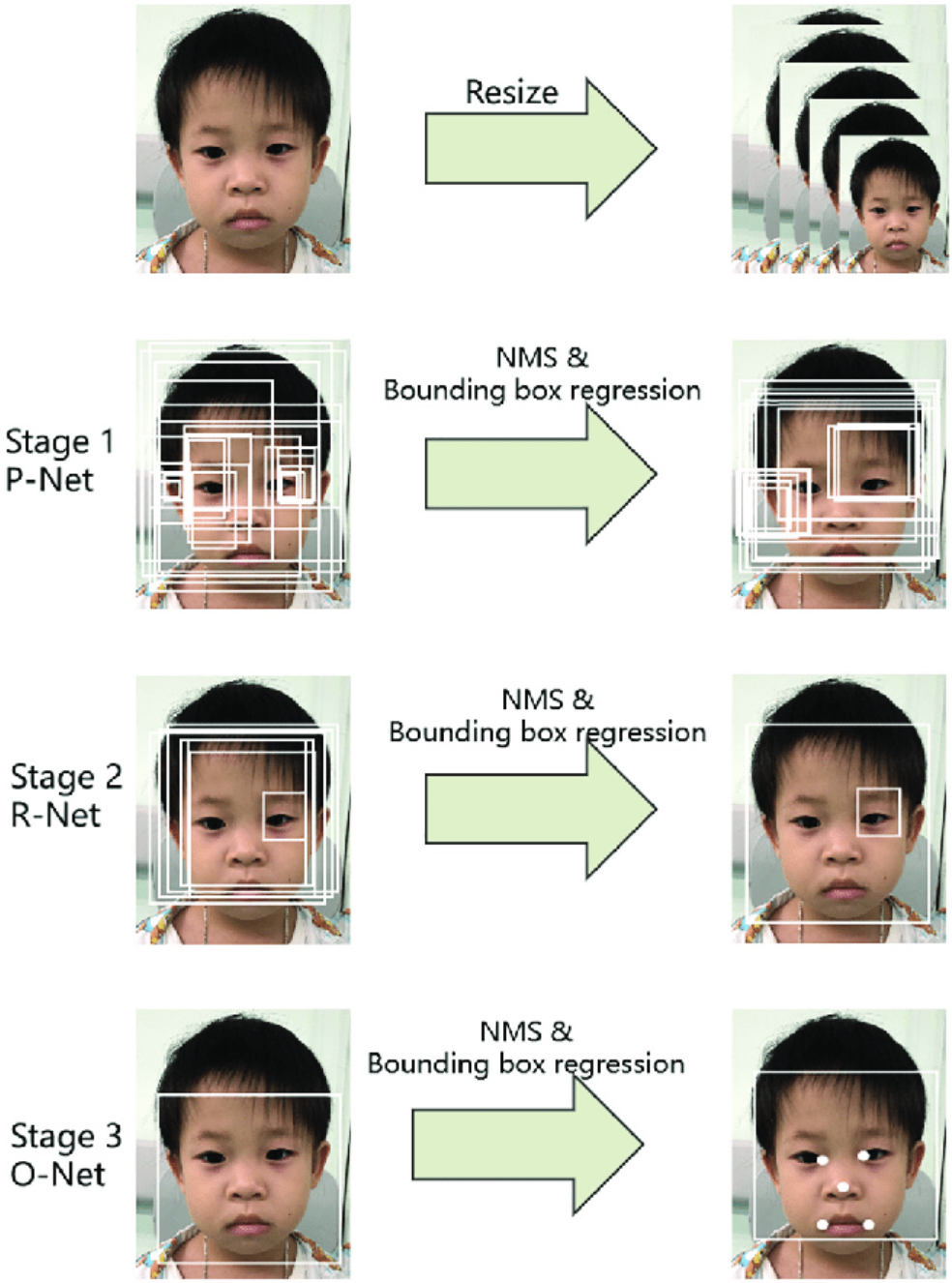

Figure 2.1 – Pipeline of the MTCNN.

2.4.1 Proposal Network (P-Net)

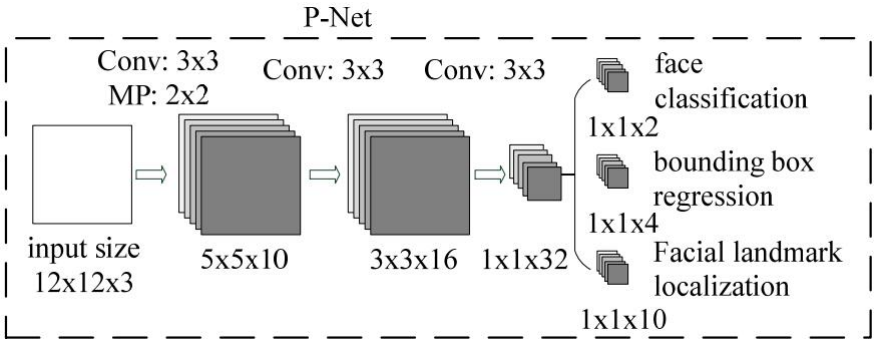

Figure 2.2 – P-Net.

In the first stage, the P-Net generates candidate windows for potential faces in the image and proposes candidate windows for face detection. This FCN systematically scans the image at different scales to identify possible areas containing faces and outputs a matrix. The output includes bounding boxes and their associated probabilities, indicating a potential face within each box.

During this stage, the input image is pyramided to different scales (creating an image pyramid), and the P-Net is run over these scaled images to generate candidate windows at different scales. These candidates are then passed through a non-maximum suppression (NMS) to reduce overlap.

2.4.2 Refine Network (R-Net)

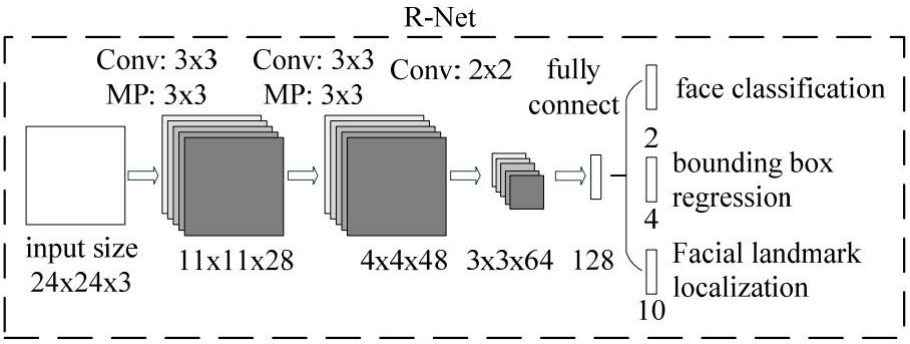

Figure 2.3 – R-Net.

The candidate windows produced by the P-Net are fed into the R-Net. It is a more complex Convolutional Neural Network (CNN) than the P-Net. In the second stage, the R-Net refines these candidate windows by rejecting a significant proportion of false candidates through more complex convolutional layers, thereby saving computational resources. It inputs candidate windows from P-Net and outputs binary classification (face / no-face) and bounding box regression vectors. At this stage, the bounding box of the remaining candidate faces is recalculated for more precision, and five facial landmarks (two eyes, nose, and two mouth corners) are predicted.

The R-Net effectively filters out a lot of false positives from the proposals of the P-Net, and it also corrects the face regions to a more accurate extent with the bounding box regression. Similar to P-Net, after the classification and regression tasks, the windows are again processed by NMS to reduce redundancy.

2.4.3 Output Network (O-Net)

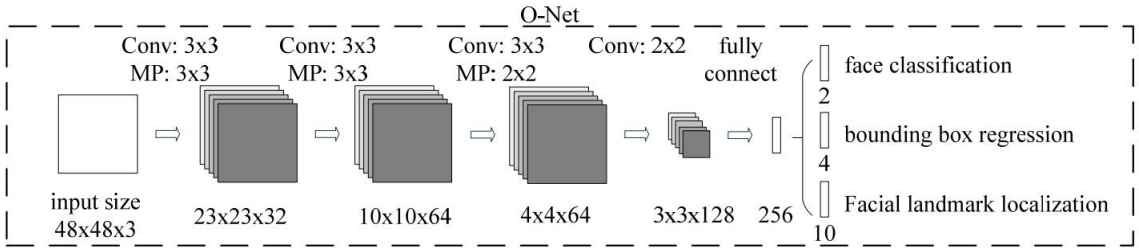

Figure 2.4 – O-Net.

In the face detection process, MTCNN's third stage plays a crucial role in predicting the positions of five facial landmarks. The final stage is the O-Net, it's similar to the R-Net but has more convolutional layers, and it further refines the output of the R-Net. In addition to performing binary classification and bounding box regression, the O-Net outputs bounding boxes and landmarks and predicts five facial landmarks' positions: the eyes, nose, and mouth's corners. These landmarks can be used for face alignment in the subsequent preprocessing steps and provide critical spatial information about each detected face's features. This further improves the accuracy of the detected face by providing more detailed information, including face position and orientation. The major downside is the computational complexity of implementing this model, which can lead to increased latency in real-time applications.

These three stages work in tandem, refining the bounding boxes at each step, eliminating nonfacial regions, and finally predicting key facial landmarks, resulting in robust face detection, even in challenging scenarios.

One of the key strengths of MTCNN is its ability to detect faces at multiple scales. It starts with a coarse detection at the first stage and gradually refines it, allowing for the detection of small and large faces in the same image. Additionally, by detecting facial landmarks, MTCNN can align the faces, making it suitable for subsequent facial recognition tasks.

MTCNN has demonstrated robustness in handling challenging conditions, such as variations in pose, lighting, and occlusions. Its cascaded structure enhances precision while maintaining computational efficiency, making it suitable for real-time applications.

MTCNN has found applications in various domains, including:

- **Facial Recognition**: It serves as a pre-processing step in facial recognition systems, accurately detecting and aligning faces for subsequent feature extraction and matching.
- **Video Surveillance**: MTCNN is employed in surveillance systems for real-time face detection in video streams.
- **Emotion Analysis**: It is used for detecting facial expressions in emotion analysis applications.
- **Access Control**: MTCNN is applied in access control systems, allowing for secure and efficient face-based access verification.

While MTCNN is highly effective, it may still face challenges in extremely crowded scenes, where faces are close together, or in situations with severe occlusions. However, while MTCNN is powerful, it is also computationally expensive compared to some newer face detection methods. Additionally, while it is typically used for face detection, it does not perform the actual face recognition (i.e., determining the identity of the detected faces). Another network, such as MobileFaceNets usually performs that task, that takes the output of MTCNN as its input.

2.5 MobileFaceNets

The concept of MobileFaceNet was introduced by Chen et al. (2018)[7], emphasizing the importance of creating lightweight yet powerful models for face recognition. Since then, it has been used in various contexts due to its small size and efficient computational speed.

There are three main contributions of MobileFaceNet:

- After the last (non-global) embedded CNN face feature layer, a global depth convolutional layer is added to replace the average pooling layer or fully connected layer.
- The facial features embedded with CNNS are designed to run efficiently on mobile phones and mobile devices.
- In LFW, AgeDB, and MegaFace datasets, face recognition has been greatly improved compared with previous networks.

Feature extraction in the MobileFaceNets model involves transforming raw face images into compact yet expressive representations known as embeddings. These embeddings encapsulate the distinctive facial characteristics that differentiate one person from another. This transformation is accomplished using several layers of the MobileFaceNets architecture, a type of Convolutional Neural Network (CNN) specifically designed to be lightweight and efficient for mobile or edge devices.

The architecture of MobileFaceNets consists of a standard convolutional layer, followed by several bottleneck structures and a final fully connected layer. As the fundamental building components

that deploy the MobileNetV2[8] residual bottlenecks. The non-linearity used by the researchers, PReLU, is more appropriate for face verification than ReLU. Additionally, the researchers utilize an efficient downsampling technique at the network's core and a linear 1×1 convolution layer as the feature output layer after a linear global depthwise convolution layer. Here is a more detailed breakdown of the MobileFaceNets architecture:

- **Initial Convolutional Layer**: The initial standard convolutional layer is the entry point for face images. It is designed to perform the first level of feature extraction, generally detecting low-level features such as edges and texture patterns.
- **Bottleneck Structures**: Following the initial convolutional layer, the model architecture includes several bottleneck structures. Each of these structures consists of:
  - A 1×1 pointwise convolutional layer is used to modify the number of input channels.
  - A depthwise convolutional layer, which applies a single filter to each input channel. The output of this layer is a set of feature maps that capture spatial and channel-wise information separately, contributing to the model's efficiency.
  - Another 1×1 pointwise convolutional layer is used to increase the number of channels again, preparing the data for the next bottleneck structure.
  - A shortcut connection (or residual connection) that skips one or more layers and connects the output of one layer to a later layer. This helps alleviate the vanishing gradient problem during training and facilitates learning identity functions, thus preventing degradation of learning performance as the network deepens.
- **Global Depthwise Pooling Layer**: There is a Global Depthwise Pooling layer before the final output layer. This layer reduces the dimensionality of the feature maps while preserving significant information about the face. The output of this layer is a 1D vector that summarizes the spatial information for each feature map, providing a compact representation of the face image.
- **Fully Connected Layer (Embedding Layer)**: The final layer in MobileFaceNets is a fully connected layer, which takes the output of the Global Depthwise Pooling layer and transforms it into a compact feature vector, also known as an embedding. This embedding represents the unique features of the face and can be used for further tasks such as face verification or identification.

Input	Operator	t	c	n	s
$112^2 \times 3$	conv3x3	-	64	1	2
$56^2 \times 64$	depthwise conv3x3	-	64	1	1
$56^2 \times 64$	bottleneck	2	64	5	2
$28^2 \times 64$	bottleneck	4	128	1	2
$14^2 \times 128$	bottleneck	2	128	6	1
$14^2 \times 128$	bottleneck	4	128	1	2
$7^2 \times 128$	bottleneck	2	128	2	1
$7^2 \times 128$	conv1x1	-	512	1	1
$7^2 \times 512$	linear GDConv7x7	-	512	1	1
$1^2 \times 512$	linear conv1x1	-	128	1	1

Figure 2.5 – The detailed architecture of MobileFaceNet for feature embedding.[7]

MobileFaceNet[7] is actually an improved version of MobileNetV2[8], although they have a lot of similarities in terms of their notations. Each line describes a sequence of operators, repeated n times. The author uses global depth convolution (GDConv) instead of global average pooling. The kernel size of the GDConv layer is equal to the size of the input dimension:

$$pad = 0,\ \text{stride}\ = 1 \tag{2.1}$$

The calculation of GDConv is represented as follows:

$$G_m = \sum_{i,j} K_{i,j,m} \cdot F_{i,j,m} \tag{2.2}$$

- F is the input feature map size

$$W \times H \times M \tag{2.3}$$

- K is the size of the depthwise convolution kernel

$$W \times H \times M \tag{2.4}$$

- G is the output feature map with size

$$1 \times 1 \times M \tag{2.5}$$

Computational cost and parameter quantities are:

$$W \cdot H \cdot M \tag{2.6}$$

2.6 Attendance Systems: From Traditional to Automated

Attendance systems have traditionally been manual, relying on paper-based methods or card-swiping mechanisms. The introduction of biometric systems, such as fingerprint and iris scanners, marked a significant advancement. However, these methods often require physical contact or close proximity, which may not always be feasible or hygienic. Facial recognition offers a contactless alternative that is both efficient and secure. Various institutions, from educational settings to corporate offices, have started to adopt facial recognition-based attendance systems. These systems not only streamline the attendance process but also mitigate issues related to proxy attendance and identity fraud.

Chapter 3

Methodology

This chapter provides a comprehensive exposition of the research design, methodologies, data collection instruments, and procedures employed in this study.

3.1 System Design and Architecture

3.1.1 Research Design

In this study, we adopted an applied research design aimed at solving the practical problem of manual attendance systems by developing an automated facial recognition attendance system. The research design encompasses the exploration and utilization of deep learning techniques, primarily focusing on MTCNN for face detection and MobileFaceNets for face recognition, integrated into a Python environment. The stages of the research design include the literature review, tool selection, data collection and preprocessing, system development, and performance evaluation.

3.1.2 System Architecture

Detection Phase

The detection phase of the facial recognition system is a critical process that involves the preparation and optimization of the dataset to ensure precise and reliable face recognition. This phase is methodically structured into four distinct stages as elucidated below:

- **Face Detection using MTCNN**: In this initial stage, each labeled image sourced from the database undergoes a face detection process leveraging the capabilities of the Multi-Task Cascaded Convolutional Networks (MTCNN) as detailed in the seminal work of Ku Hongchang (2020)[9]. This technique, facilitated through the TensorFlow library, is instrumental in accurately identifying and isolating facial regions within the images.

- **Normalization and Preprocessing**: Subsequent to face detection, the isolated facial images are subjected to a normalization and preprocessing regimen. This step is crucial in standardizing the dataset, ensuring uniformity in scale and orientation, thereby facilitating a streamlined feature extraction process in the subsequent stages.

- **Feature Extraction using MobileFaceNets**: In this pivotal stage, distinctive facial features are extracted utilizing the FaceNet model, a pre-trained deep neural network renowned for its efficacy in facial recognition. This process is central to the identification of unique facial signatures which are imperative for the recognition process.

- **Database Compilation**: The final stage of the training phase involves the systematic storage of the extracted features into a facial feature database. This repository serves as the reference point for the facial recognition system, harboring the essential data required to authenticate individuals in the identification phase.

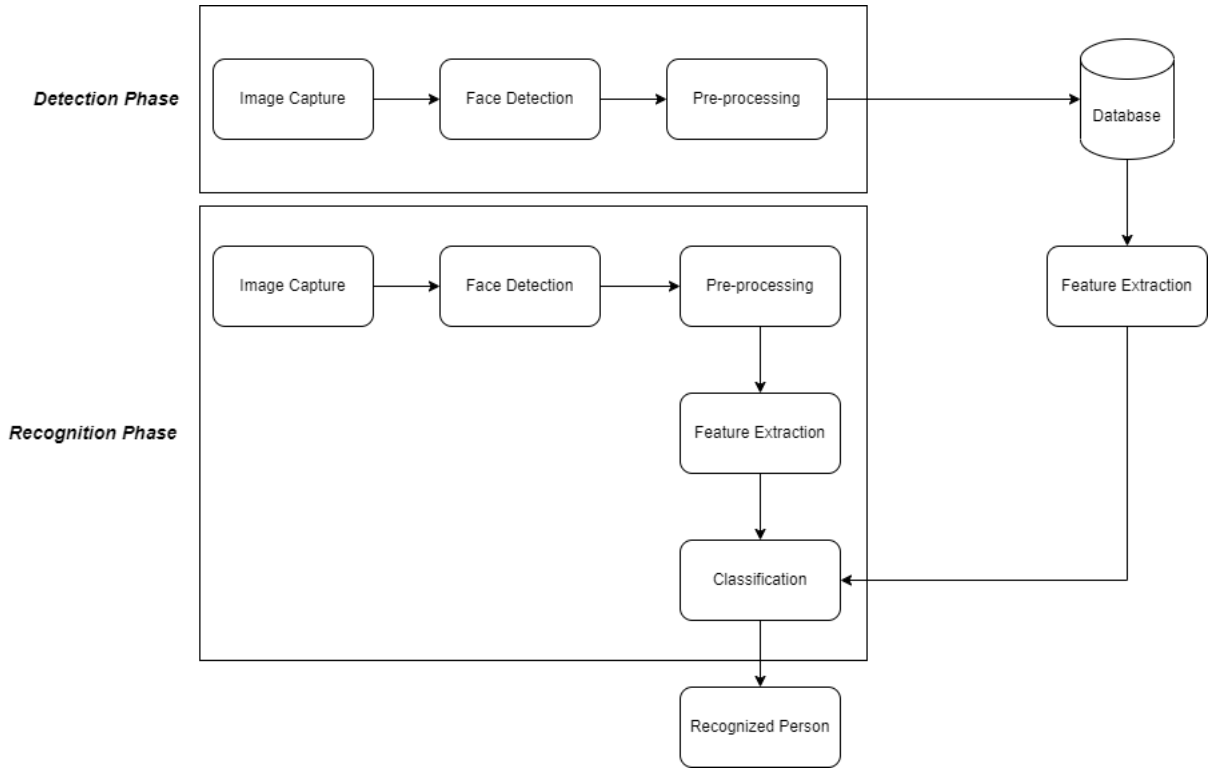

Figure 3.1 – System Architecture of Facial Recognition.

Recognition Phase

The recognition phase is executed through a meticulous procedure designed to authenticate individuals accurately based on the pre-trained data. This process unfolds over a series of well-defined steps as described below:

- **Face Detection**: The initial step involves executing a face detection algorithm on an input image, captured through a smartphone camera, to identify potential face regions within the image.

- **Normalization and Preprocessing**: Similar to the detection phase, the detected facial images are then normalized and preprocessed to standardize the facial data, ensuring it is primed for the subsequent feature extraction process.

- **Feature Extraction using MobileFaceNets**: In this stage, the preprocessed facial images are analyzed using the MobileFaceNets model to extract distinctive facial features. This deep learning model excels in identifying the nuanced details that are unique to each individual's face.

- **Feature Comparison and Classification**: Following feature extraction, a comprehensive comparison is undertaken where the extracted features are matched against the facial feature database compiled during the training phase. This classification process is integral in identifying the individual by finding the closest match in the database.

- **Result Generation**: The final step culminates in the generation of facial recognition results, where the system verifies the individual's identity with a considerable degree of precision, thereby authenticating the person's identity.

3.1.3 Pre-trained Model

In the internship of this research, a pre-trained model was employed to facilitate face recognition with MobileFaceNets. This model was provided by my external supervisor, Mr. Bui Hai Dang, and is proprietary to *KSE Software Solutions*. Due to the proprietary nature of the model, it is not publicly available for distribution. Interested parties wishing to access the model for academic or research purposes are encouraged to contact Mr. Bui Hai Dang at *dang.bui@kse-solutions.com* or the respective department at *KSE Software Solutions*.

By using this pre-trained model, the research benefitted in the following ways:

- **Time and Resource efficiency**: Pre-trained model MobileFaceNets is trained on large data sets, which often require significant computational resources. Research time during the internship is allowed, it is possible to take advantage of this model without having to train them from scratch, saving time and computing power.

- **Faster Development and Generalizability**: During research and internships, be able to quickly prototype and develop applications using pre-trained models as a starting point, possibly fine-tune these models for their specific tasks, adapting them to their needs without completely retraining from scratch. This speeds up the development cycle and allows for faster testing.

- **Reduced Data requirements**: Fine-tuning pre-trained models typically requires less labeled data than training the model from scratch. This is especially valuable in situations where collecting large amounts of labeled data is expensive or challenging.

- **Scalability**: These models are often designed to be scalable, meaning they can be tweaked across a variety of hardware configurations, making them adaptable to research environments. different. Therefore, they can be suitable for computer configuration for experimentation during this internship.

For further details on the model's architecture, performance, and applicability, please refer to *Chapter 2: Literature Review*.

3.1.4 Face Detection using MTCNN

Before recognizing an individual, it is essential to detect the presence of a face in the image. For this purpose, we employ MTCNN, a popular and efficient method for face detection.

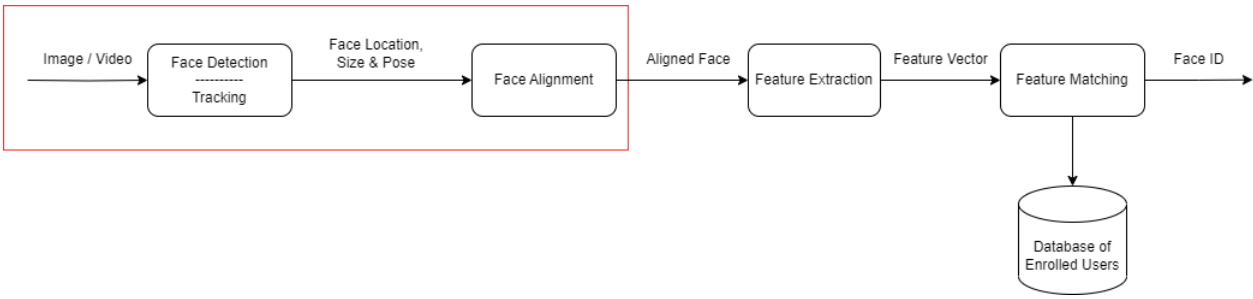

Figure 3.2 – Pipeline of Facial Recognition - Face Detection

In the context of a facial recognition attendance system, the face detection module is the initial stage where faces are identified from the input source. The face detection stage does not identify who the faces belong to; it only identifies that a face is present and where it is located in the image. Identifying the individuals (face recognition) is a separate process that comes after face detection.

- **Input**: The input to a face detection algorithm is typically an image or a video frame. In a real-world setting, this could come from various sources like a webcam, CCTV camera, smartphone camera, or even pre-stored images or video clips. The image or video frame may contain one or more faces, possibly along with other objects, and the faces may be in various poses, expressions, lighting conditions, or occlusions.

- **Output**: The output of a face detection algorithm is generally the locations of the detected faces within the input image or video frame. This is typically represented as a list of bounding boxes, where each bounding box is specified by the coordinates of its top left corner and its

width and height. Each bounding box corresponds to a region of the image the algorithm identified as a face.

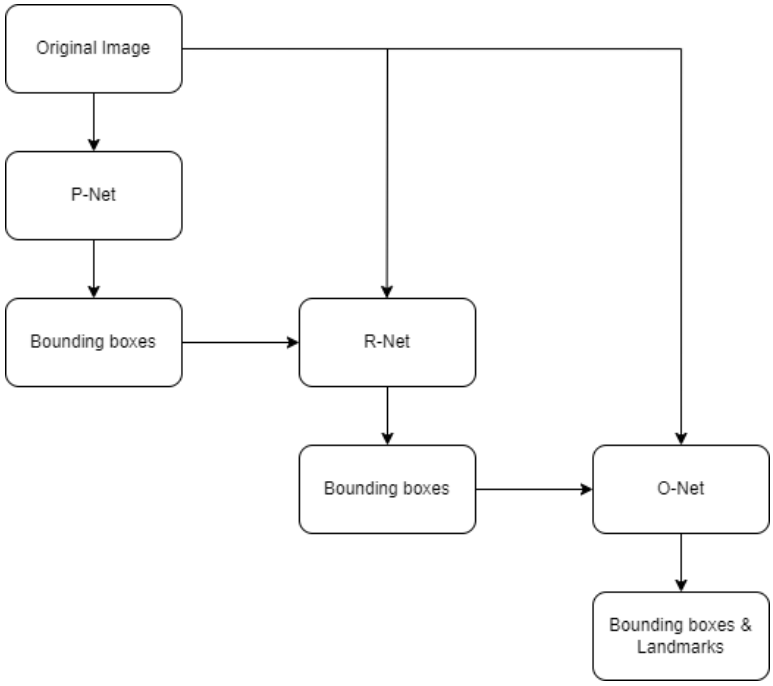

Figure 3.3 – Flow chart of the training data preparation.[10]

The MTCNN model requires a mix of positive (with face), negative (without face), and partial (part of a face) images for training. Alongside, the facial landmark positions are also required for the images. The networks are trained sequentially. First, the P-Net is trained, then the R-Net, and finally the O-Net. During the training of the R-Net and O-Net, the preceding networks are frozen, ensuring the cascaded structure's efficacy. Given an input image, it is first resized to different scales to build an image pyramid. This is to detect faces of varying sizes. The image pyramid is then fed into the P-Net. Face windows are generated. The face windows are refined by the R-Net. Finally, the O-Net takes the refined windows and outputs the final face detections along with their landmarks. After each stage, NMS is applied to reduce the number of overlapping boxes. This ensures that each face is detected only once.

3.1.5 Face Recognition using MobileFaceNets

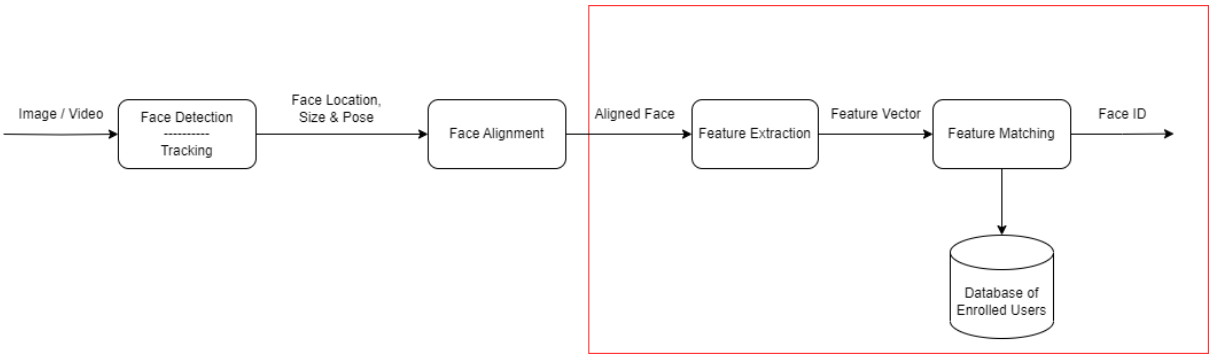

Figure 3.4 – Pipeline of Facial Recognition - Face Recognition

Face Detection and Face Recognition are two essential aspects of computer vision and have been extensively used in many applications, including security systems, social media, and automated attendance systems. Once a face is detected, the next step is to recognize the individual. We utilize MobileFaceNets, a lightweight and efficient neural network, for this task.

MobileFaceNets is a deep-learning architecture optimized for mobile devices. It achieves high accuracy in face recognition tasks while being computationally efficient. The architecture is based on depthwise separable convolutions, which reduces the number of parameters and computational cost compared to standard convolutions. Given a detected face, MobileFaceNets extracts a feature vector that captures the unique characteristics of the face. This feature vector is then compared to a database of known face feature vectors to identify the individual.

- **Input**: The input to a face recognition algorithm is typically an image or video frame that has one or more faces. In a real-world application, the input could come from various sources. Ideally, the input should already be processed by a face detection module, meaning that each face is properly cropped and normalized (for instance, aligned, resized to a standard dimension, or adjusted for lighting conditions).

- **Output**: The output of a face recognition algorithm is typically the identification of the person(s) present in the input image or video frame. This identification usually relies on a database or a list of known individuals and their corresponding facial features, which the system uses to match the input face(s). If the face recognition system cannot match the input face to any record in the database, the system might return an "unknown" status.

In simple terms, while face detection concerns the question "Where are the faces?" face recognition deals with the question "Whose face is this?". A facial recognition system usually involves a face detection step first to find faces in an image or video, followed by face recognition to identify whose faces they are.

Faces should be labeled with their corresponding identities. A typical dataset might include multiple images per person to capture different poses, expressions, and lighting conditions.

3.2 Dataset

The datasets used experimentally for the face recognition system to verify individual identities include the *Yale-Face-Database dataset* and the 'Real-world' dataset.

The Yale-Face-Database dataset contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised, and wink.

The "Real-world" dataset is a specially curated collection of images amassed during the tenure of the internship and is structured to proficiently support the facial recognition project. The dataset was collected from a diverse group of individuals to ensure the system's versatility and inclusivity. It comprised high-resolution images captured in different lighting conditions to facilitate a robust training process. It is comprised of a collection of photographs of a specific number of individuals, the data set includes nearly 400 photos of 20 people with different facial angles and expressions, and each individual has between 10 to 20 color images labeled with a name label. Each individual's image is saved in the same photo folder and assigned an identifier according to the structure: Full name.

The acquisition of the dataset adhered to stringent guidelines to maintain a high-quality standard:

- Photos taken with a smartphone camera have a resolution of 828 x 1792 pixels or higher for screenshots, photos taken with the front camera, and the wide camera option is available to minimize blur.
- The camera was held vertically to ensure a uniform orientation of all the images.
- Individuals were centered in the frame with a focus on the mid-face region to capture detailed and balanced facial features under well-lit conditions.
- A camera-to-subject distance of 1 to 2 meters was maintained to obtain clear and sharp images of the faces.

3.3 Facial Recognition Attendance System

Developing graphical user interfaces (GUIs) in Python using Qt Designer is a popular choice for creating cross-platform desktop applications. Qt Designer is a graphical tool that allows us to design the user interface of the application visually, and then we can integrate it with the Python code.

In the context of this internship program, we engage in the deployment of machine learning models subsequent to a rigorous experimentation phase, with the objective of constructing a rudimentary facial recognition-based attendance system. This system is architecturally designed utilizing Qt Designer, a graphical user interface tool renowned for its cross-platform capabilities and ease of integration with Python programming language. The system is bifurcated into three distinct windows to facilitate user interaction and data presentation:

- **The Main Window**: This serves as the primary interface for the end-users, offering functionalities such as initiating the facial recognition process, capturing real-time images, and marking attendance. It is the gateway through which the underlying deep learning algorithms are triggered to perform facial identification tasks.

- **The Information Window**: This secondary window functions as a dashboard that displays pertinent information, including but not limited to, the list of attendees, time stamps of attendance, and any anomalies or errors encountered during the facial recognition process. This window serves an administrative role by providing a comprehensive overview of the attendance data.

- **The Attendance History Window**: This window will present the attendance history, featuring five distinct columns of information: name, time, date, status, and duration. Particular emphasis should be directed toward the final two columns, as they provide vital data regarding an individual's check-in or check-out activities. The duration column provides a comprehensive record of the total time an individual spent within the premises, coupled with their check-in status.

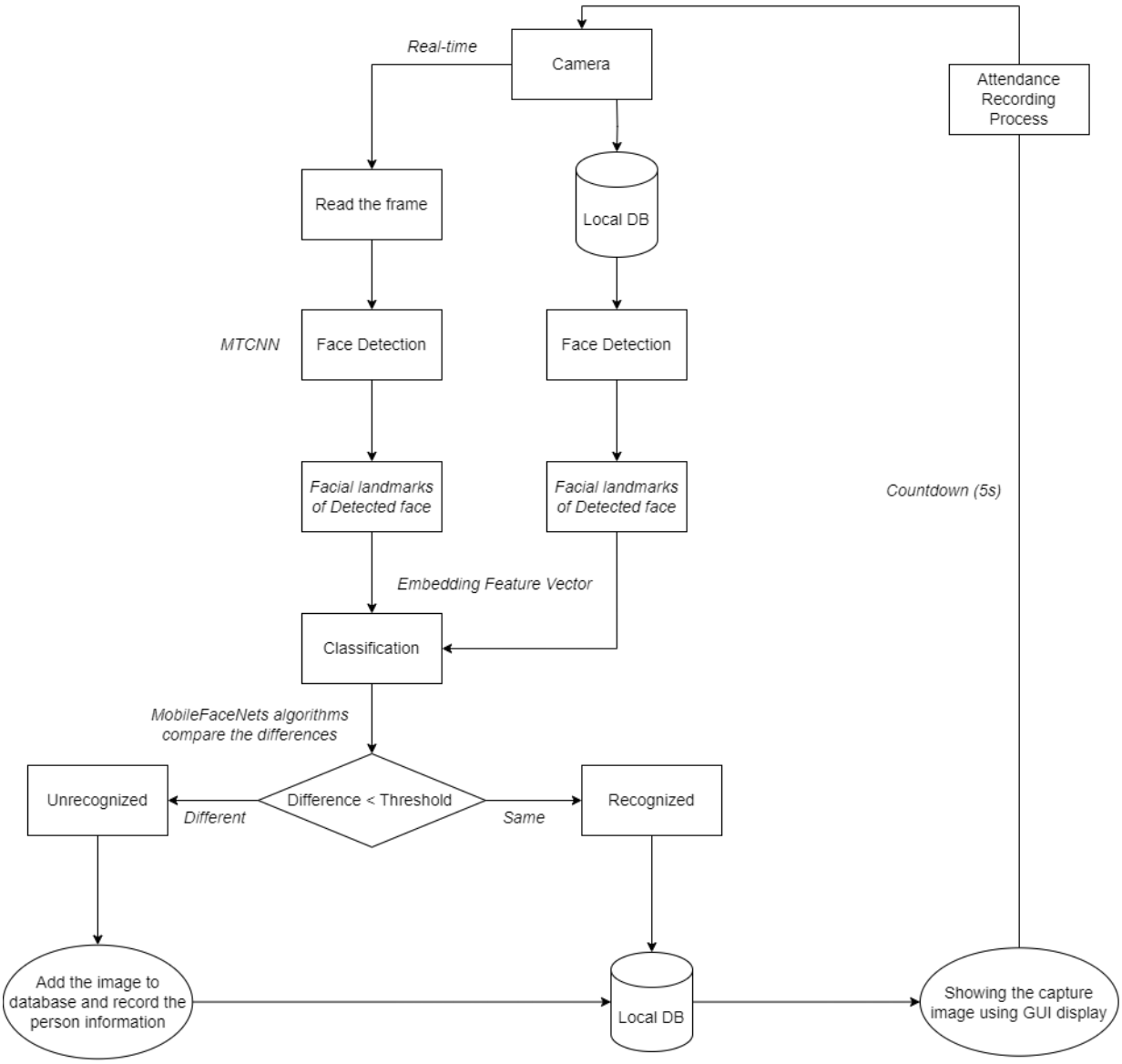

Figure 3.5 – Overview of Facial Recognition Attendance System.

For actual application, when the system necessitates the addition of new individual data into the database, a straightforward approach involves the creation of a new directory, denominated by the individual's name or identification number, within which clear facial photographs of that person shall be encompassed.

LOCAL DATABASE

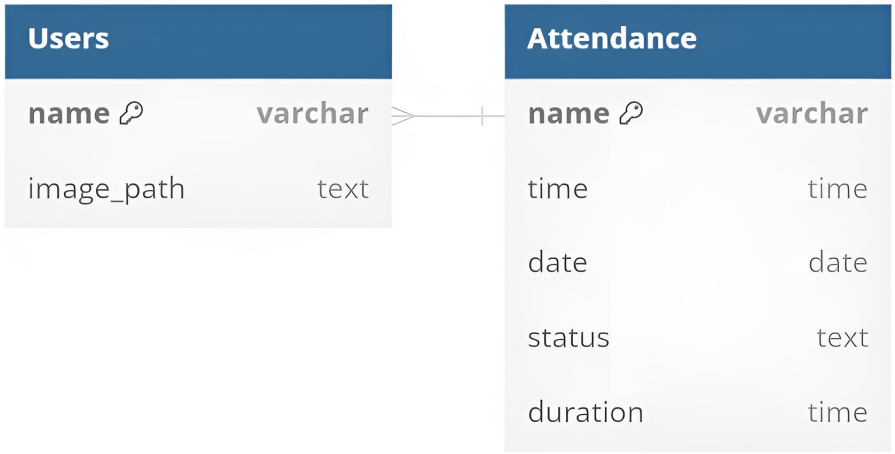

Figure 3.6 – Overview of Local Database.

The Users table is designed to store information about users, specifically their unique identifier, name, and the path to their image.

- **name (varchar)**: Represents the name of the user. Marked as the primary key ([pk]), meaning each name in the table must be unique. The system will not allow the insertion of duplicate names. As the primary key, the name column cannot have NULL values. Every record must have a name. Being the primary key also implies that this column serves as the main identifier for each record in the table. Any references from other tables to this table will likely use the name column.

- **image_path (text)**: Contains the path (likely a file path or a URL) to an image associated with the user. This could be a profile picture or any other relevant image. The path might be used in an application to display the user's image. There's no constraint mentioned, so it's possible for multiple users to have the same image_path (though that might not be common in real-world scenarios).

The Attendance table is designed to store information about users attendance history, specifically their unique identifier, name, time, date, status (Check-in or Check-out), and duration (this is the total time that person is present, which can be understood as the time between check-in and check-out).

- **name (varchar)**: Represents the name of the user. Marked as the primary key ([pk]), meaning each name in the table must be unique. The system will not allow the insertion of duplicate names. As the primary key, the name column cannot have NULL values. Every record must have a name. Being the primary key also implies that this column serves as the

main identifier for each record in the table. Any references from other tables to this table will likely use the name column.

- **time (time)**: The "time" column is designed to capture a precise point in time during the day, typically recorded in the format of hours, minutes, and seconds. It is intended for the purpose of logging and referencing events or activities that transpire at a specific moment in time. This column serves as a means to record time-related data, making it especially useful for applications where time precision is essential.

- **date (date)**: The "date" column serves as a repository for calendar dates, typically represented with the year, month, and day format. It is well-suited for documenting events or information that are associated with particular dates. This column is commonly employed for storing dates of significant occurrences, such as appointments, historical milestones, or transaction dates.

- **status (text)**: The "status" column accommodates textual data and is utilized to articulate the present state or condition of an entity. Its purpose extends to a wide range of applications, where it serves as a descriptive indicator of the status of an object, process, or event. For example, it can be used to convey the attendance status of individuals, including descriptors like "checked-in" or "checked-out."

- **duration (time)**: The "duration" column is designated for recording time intervals, typically denoted in hours, minutes, and seconds. Its primary role is to document the temporal extent of activities, actions, or processes. In the context of attendance tracking, this column can be employed to record the length of time taken for attendance-related actions, providing insights into the duration of these activities.

Chapter 4

Experimental Setup

This chapter provides a comprehensive evaluation of the system's performance metrics, in addition to evaluating the performance of the facial recognition attendance system when using these models.

Experiments are performed on CPUs, Python programming language, OpenCV, and TensorFlow library.

4.1 Data Augmentation and Preprocessing

To enhance the model's performance, data augmentation techniques such as rotation, flipping, and scaling were applied to the dataset. Preprocessing involved cleaning the data by removing low-quality images and performing normalization to reduce computational complexity. Face alignment was also conducted to ensure uniformity in the dataset, paving the way for accurate face recognition.

Data after being collected will include nearly 400 photos of 20 people with different facial angles and expressions. These are raw, unprocessed data. To make the data usable effectively, the data cleaning and normalization process is described as follows:

- Data Classification and Filtering:
  - Image Classification: Sort images by individual persons and allocate them into separate folders.
  - Data Cleaning: Remove noisy and blurred images to retain only the high-quality ones.
- Labeling: Assign the full name of the individual as the label for each folder.
- Image Data Reformatting: Check and convert all images to *.jpg format.

Figure 4.1 – Illustration of the 'Real-world' dataset.

Large-scale datasets constitute a prerequisite for the successful training of neural networks. Image augmentation technology leverages a series of random modifications applied to training images to generate analogous yet distinct training samples, thereby expanding the dataset's size. To optimize the model's precision in identification tasks, the employment of the MTCNN library is advocated to enhance image quality. The specific procedures encompass the following:

- Create images with 20-degree left- and right-tilt angles;
- Flip horizontally or vertically
- Rotate randomly
- Gaussian blur
- Sharpening

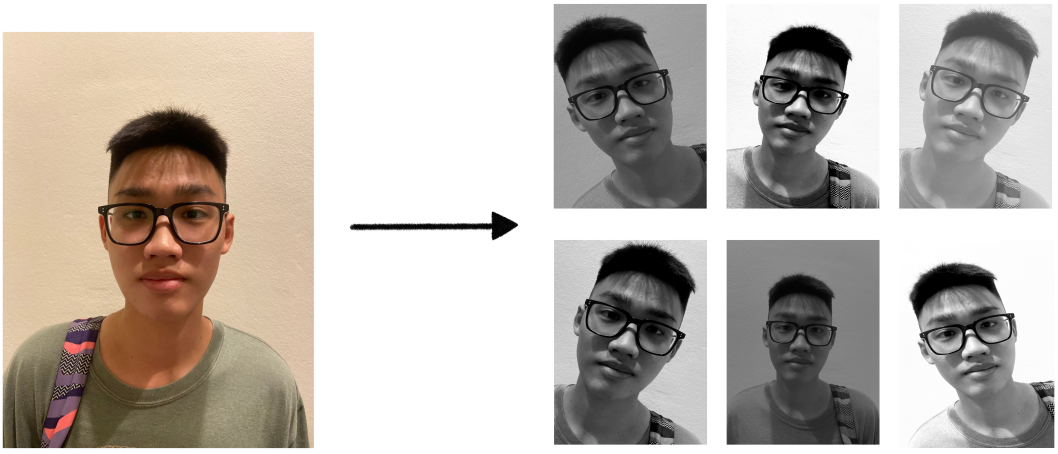

Figure 4.2 – Illustration of image data augmentation.

4.2 Implementation

To handle the complete pipeline of a face recognition task using the Yale dataset and the 'Real-world' dataset:

- **Data Preparation**: Organize the data sets, divide them into training, validation, and test sets, and create image label pairs for each set.
- **Face Detection**: Use the MTCNN detector to detect faces in photos.
- **Feature Extraction**: Load the MobileFaceNet model to extract embeddings (or feature vectors) from detected faces.
- **Face Recognition**: Comparing the extracted features to a database to identify or verify the individual. Using embeddings from the training set, train a neural network classifier to recognize different objects in the Yale dataset and the 'Real-world' dataset.

With an input image in the test set, the facial recognition system produces a prediction to identify the image, this is the basic information of the label of the folder containing similar images of the input image. Each recognition prediction can be correct or incorrect, thereby calculating the Accuracy of the test set.

Similarly, with 'Real World' data, producing a resulting image that resembles the name of the person being identified.

Chapter 5

Results and Discussion

5.1 Result Analysis

5.1.1 Yale Dataset

Experimental results of face recognition on ***the Yale dataset*** *(The Yale Face Dataset contains 165 images of 28 human subjects under 9 poses and 64 illumination conditions)* are shown below, with average accuracy for each image folder (subject), with an average recognition time of 0.0061 seconds.

Subject	Number of Images	Accuracy
subject01	26	0.84
subject02	26	0.90
subject03	26	0.84
subject04	26	0.85
subject05	26	0.71
subject06	26	0.95
subject07	26	0.88
subject08	26	0.73
subject09	26	0.76
subject10	26	0.88
subject11	26	0.83
subject12	26	0.76
subject13	26	1.00
subject14	26	0.83
subject15	26	0.80
Total	390	
Average		0.84

Table 5.1 – Recognition results on the Yale dataset

This column indicates the ID or label of the individual in the dataset. For example, subject01 represents the first individual, subject02 the second, and so on. This column indicates how many test images are present in the dataset for each subject. In this case, each subject has 26 images. The recognition accuracy of the model for each subject. The accuracy is represented as a fraction,

where 1.00 would mean 100% accuracy, and 0.80 would mean 80% accuracy. For instance, for subject01, the recognition accuracy is approximately 84.85%.

Each subject in the dataset is represented by an equal number of images, which is 26. This uniformity ensures that one subject doesn't dominate the experiment process due to a higher number of images. The accuracy varies across subjects. For instance, subject13 has a perfect accuracy of 100% (1.00), whereas subject05 has the lowest accuracy of about 71.43%. This variability can be due to various factors like the quality of images, lighting conditions, facial expressions, and occlusions present in the images of different subjects. By glancing at the accuracy values, it's evident that the model performs reasonably well for most subjects, with many subjects having an accuracy of over 80%.

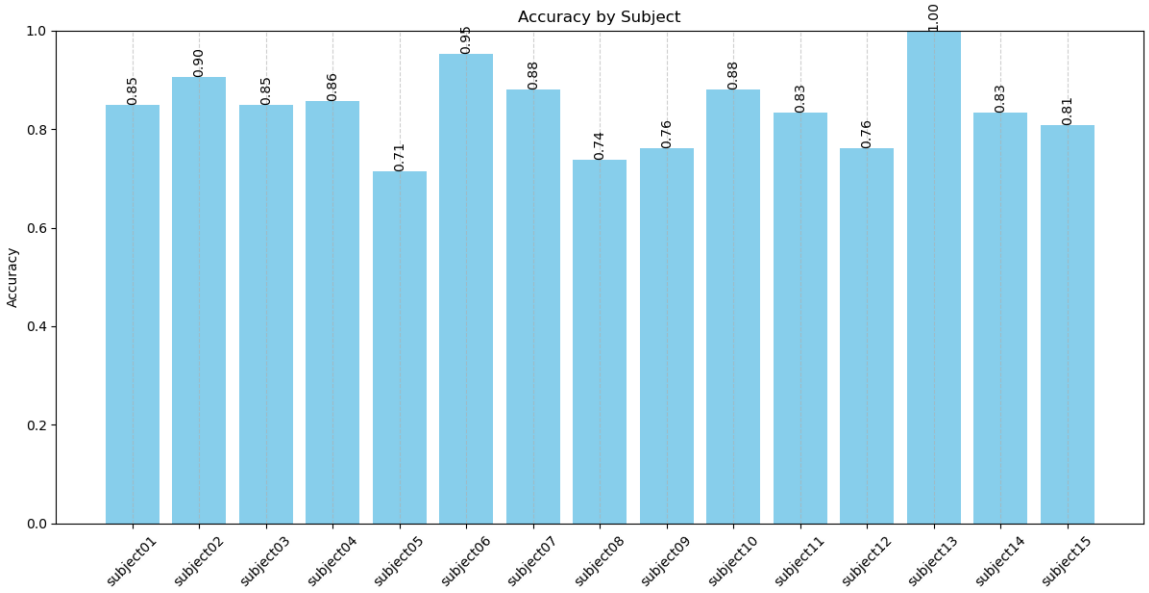

Figure 5.1 – Recognition results on the Yale dataset

The chart above provides a visual way to quickly grasp the information that's presented in the tabular. Each bar corresponds to a subject from the Yale dataset, labeled from "subject01" to "subject15". The height of each bar represents the recognition accuracy for each subject. The taller the bar, the higher the recognition accuracy for that subject. Many bars are of medium height, suggesting decent recognition performance but with some room for improvement.

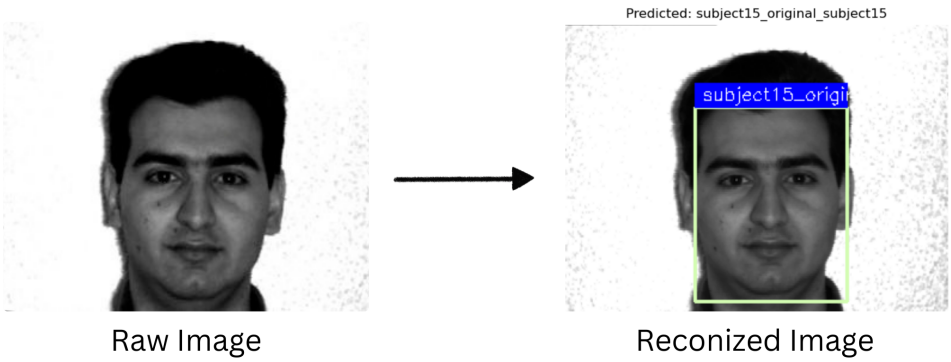

Figure 5.2 – Example result when experimenting on the Yale dataset

5.1.2 'Real-world' Dataset

Experimental results of face recognition on ***the 'Real-world' dataset*** *(The "Real-world" dataset comprises a total of 389 images capturing 20 distinct human subjects in various poses and lighting conditions, bearing certain similarities to the Yale Dataset)* are shown below, with accuracy for several image directories (name of person), with an average recognition time of 0.0024 seconds.

Subject	Number of Images	Accuracy
Ha Tuan Anh	32	0.93
Ha Tuan Khoi	32	0.90
Le Ngoc Linh	41	0.92
Le Phuong Thu	35	1.00
Le Thu Ha	28	0.71
Nguyen Thi Binh	28	0.95
Nguyen Tuan Ngoc	30	0.88
Nong Thanh Ngoc Quang	35	0.83
...	...	...
Total	812	
Average		0.89

Table 5.2 – Recognition results on the 'Real-world' dataset

The accuracy varies for different individuals. For example, "Le Phuong Thu" has a perfect accuracy score of 1.00, indicating that the system correctly identified all instances of this person's face. On the other hand, "Le Thu Ha" has a much lower accuracy of 0.71, suggesting that the system struggled to correctly recognize this individual.

These results can be useful for evaluating the performance of the face recognition system. We can use this data to identify areas where the system performs well and areas where it may need improvement. It's also essential to consider the number of images available for each person, as having more training data can lead to better recognition performance.

5.2 System Analysis

When the face is matched with the list of faces available in the database, it recognizes the person successfully and displays 'Name' of the person. Provided that the threshold must be greater than or equal to the system's established threshold (>= 0.75)

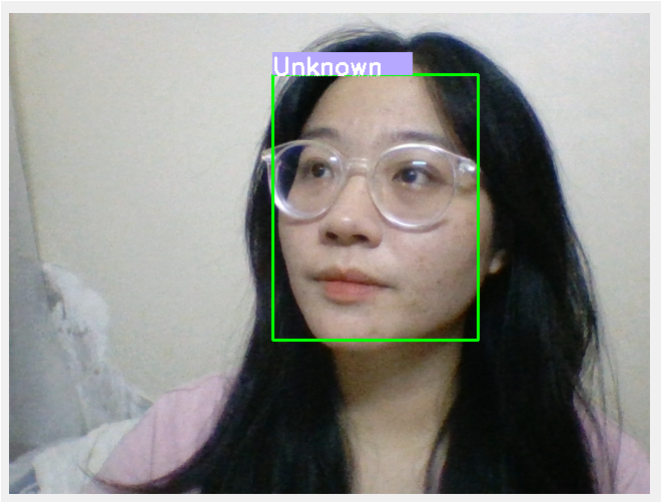

Figure 5.3 – Detecting single person.

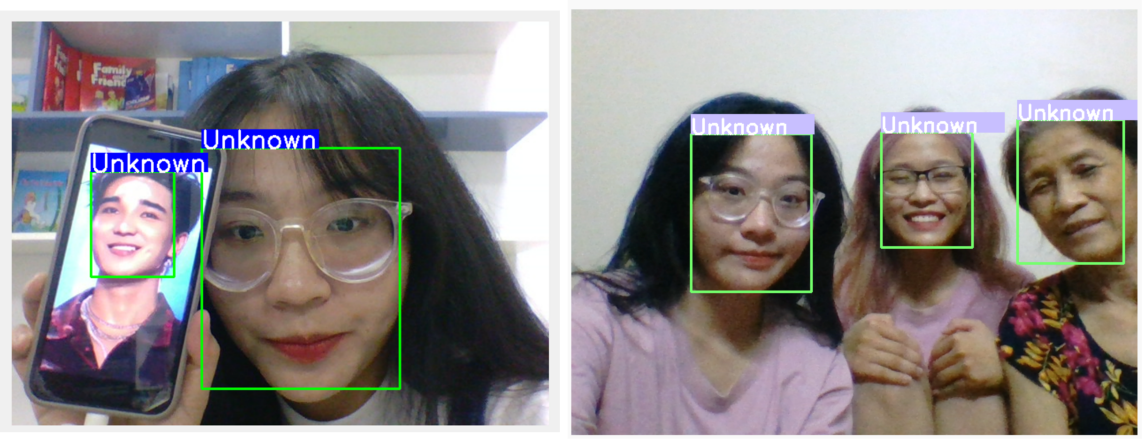

Figure 5.4 – Detecting multiple person.

Pleased to hear that the face detection component of the system is performing exceptionally well. It's particularly commendable that the system is capable of accurately identifying faces from various angles and is not limited to frontal facial recognition. Furthermore, the ability to simultaneously detect multiple faces in real-time significantly enhances its utility and applicability in diverse scenarios. Overall, these capabilities demonstrate the robustness and versatility of the facial recognition attendance system.

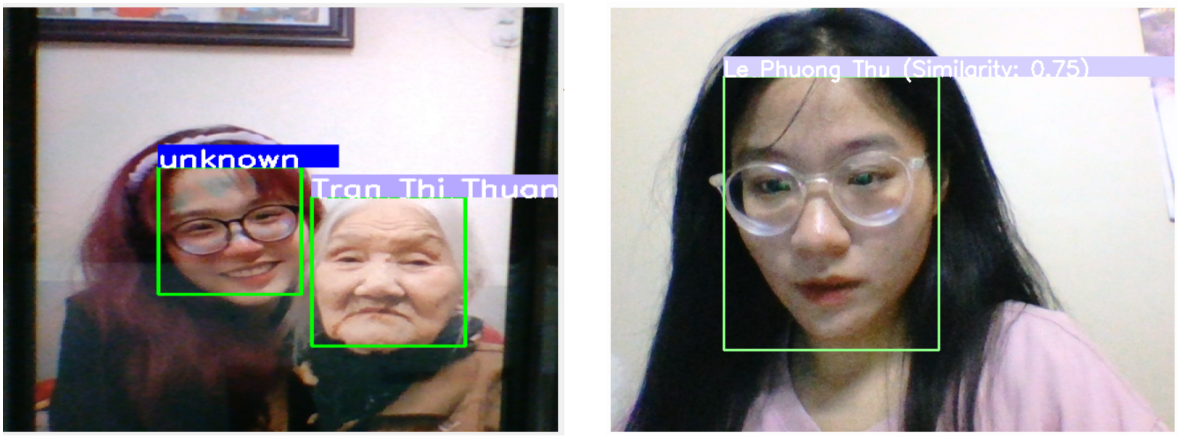

Figure 5.5 – System results when recognizing faces.

The current facial recognition attendance system is constrained by its capability to identify only a single individual at a given time. This limitation inhibits the system's scalability and practicality, particularly in scenarios requiring simultaneous identification of multiple individuals. Therefore, the existing model falls short of providing a comprehensive solution for real-time, multi-person attendance systems.

Upon successful facial recognition, the system programmatically updates a Microsoft Excel spreadsheet to record the individual's attendance. This update includes appending the recognized individual's name along with the corresponding date and time of recognition, thereby ensuring a structured and timestamped record for attendance tracking.

5.3 Discussion

The recorded mean processing times of 0.0061 seconds and 0.0024 seconds indeed reflect efficiency in the system's performance. It is, however, imperative to acknowledge that such metrics may not comprehensively encapsulate scenarios involving outlier cases or operational variances induced by heavy loads.

Upon a holistic assessment, it becomes evident that the Yale Face Database, as the chosen preprocessed dataset, confers notable advantages to the experimental process. Its utilization streamlines proceedings and enhances convenience in comparison to unprocessed data sources.

Nonetheless, a pivotal aspect to underscore is the commendable speed and precision exhibited when applying the system to a 'Real-world' dataset. Notably, an impressive average accuracy rate of 89% emerges, surpassing the performance of the Yale dataset, which records an accuracy rate of 84.1%. This discernible difference in accuracy underscores the system's robustness and commendable performance when employing deep learning models, specifically the MTCNN and

MobileFaceNets. It substantiates the proposition that the facial recognition attendance system, underpinned by these models, is well-founded and merits profound appreciation for its efficacy.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have deployed a recognition model based on the project's requirements. In addition to evaluating the above-mentioned datasets with quite satisfactory accuracy, we have implemented a simple recognition system to review the model's recognition and intuitiveness in real-time. The breadth of usable algorithms extends beyond the confines of this thesis; select an appropriate model depending on the visualization's objectives.

Even if the model is effective, the outcomes have several difficulties:

- In real-time recognition, the model can detect multiple people at the same time, but can only identify one person, not in parallel (two people at the same time). If two faces are entered at the same time for recognition, the model will only recognize one face, the other face will be detected as 'Unknown'.
- The distance the model can detect lying is about 1.5 meters. If the distance is more than 1.5 meters, the model will not be recognized because of the large distance.
- Sometimes, the model has a little problem recognizing some people with similar faces (as mentioned above, it could be a sibling of the same sex or twins, or it could be several individuals with similar facial features), the model can be confused but only in rare cases.

6.2 Future Work

Many adaptations, tests, and experiments have been left for the future due to lack of time. These are the ideas that we may experiment with in the future.

- Try to perform more in-depth research into facial recognition techniques and have a better grasp of how they work.
- Develop a better facial recognition system that can recognize multiple faces at the same time.

References

[1] Mei Wang and Weihong Deng. "Deep face recognition: A survey." In: *Neurocomputing* 429 (2021), pp. 215–244.

[2] Wanli Ouyang et al. "Deepid-net: multi-stage and deformable deep convolutional neural networks for object detection." In: *arXiv preprint arXiv:1409.3505* (2014).

[3] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments." In: *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*. 2008.

[4] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition." In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*. Springer. 2016, pp. 87–102.

[5] Jia Xiang and Gengming Zhu. "Joint face detection and facial expression recognition with MTCNN." In: *2017 4th international conference on information science and control engineering (ICISCE)*. IEEE. 2017, pp. 424–427.

[6] Mei Ma and Jianji Wang. "Multi-view face detection and landmark localization based on MTCNN." In: *2018 Chinese Automation Congress (CAC)*. IEEE. 2018, pp. 4200–4205.

[7] Sheng Chen, Yang Liu, Xiang Gao, and Zhen Han. "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices." In: *Biometric Recognition: 13th Chinese Conference, CCBR 2018, Urumqi, China, August 11-12, 2018, Proceedings 13*. Springer. 2018, pp. 428–438.

[8] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 4510–4520.

[9] Hongchang Ku and Wei Dong. "Face recognition based on mtcnn and convolutional neural network." In: *Frontiers in Signal Processing* 4.1 (2020), pp. 37–42.

[10] Mei Ma and Jianji Wang. "Multi-View Face Detection and Landmark Localization Based on MTCNN." In: *2018 Chinese Automation Congress (CAC)* (2018), pp. 4200–4205. URL: `https://api.semanticscholar.org/CorpusID:59235632`.

Appendix A

Appendix B