A SMART-BASED STUDENTS' ATTENDANCE SYSTEM USING FACIAL RECOGNITION TECHNIQUES

BY

USUNOBUN CHRISTY OSEILHENOME

PSC1908957

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF PHYSICAL SCIENCES,

UNIVERSITY OF BENIN, BENIN CITY,

EDO STATE, NIGERIA.

MAY 2024.

A SMART-BASED STUDENTS' ATTENDANCE SYSTEM USING FACIAL RECOGNITION TECHNIQUES

BY

USUNOBUN CHRISTY OSEILHENOME

PSC1908957

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF PHYSICAL SCIENCES, UNIVERSITY OF BENIN, BENIN CITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE.

MAY 2024.

CERTIFICATION

This is to certify that this project report was carried out by **Usunobun Christy Oseilhenome** with matriculation number **PSC1908957** of the Department of Computer Science, University of Benin, under my supervision, and is adequate in scope and content for the award of Bachelor of Science (B.Sc.) degree in Computer Science of the University of Benin.

……. …….

Mr. D.N. Idehen **Date**

(Supervisor)

APPROVAL

This project work is hereby approved in partial fulfilment of the requirements for the award of Bachelor of Science (B.Sc.) degree in Computer Science from the University of Benin.

……. …….

PROF. G.O. EKUOBASE (PHD) DATE

Head of Department

DEDICATION

This project is dedicated to God Almighty for blessing me with the strength and wisdom to see it through to completion. I also dedicate this to my parents, Mr. and Mrs. Timothy Usunobun, for their unwavering support, love and guidance throughout my academic journey.

ACKNOWLEDGEMENT

First and foremost, I want to thank God for providing me strength, wisdom, perseverance and direction needed to undertake and complete not just this project, but my academic journey. I am immensely grateful to my project supervisor, Mr. D.N. Idehen, who devoted his time to guiding and supporting me along the course of this project. I am indeed very grateful for his time, support and advice within the time of this project work.

I also acknowledge all the lecturers in the department of Computer Science for their teachings and support. From the H.O.D, Prof. Godspower O. Ekuobase (PHD), Prof. (Mrs.) V.V.N. Akwukwuma, Prof. (Mrs.) F. Egbokhare, Prof. A.A. Imianvan, Prof. (Mrs.) A.O. Egwali, Prof. F.I. Amadin, Prof. F.A.U. Imouokhome, Prof. (Mrs.). S. Konyeha, Prof. (Mrs.) V.I, Osubor, Dr. (Mrs.) G.O. Aziken, Mr. E.E. Obasohan, Mr. S.O.P. Oliomogbe, Mr. P.E.B. Imiefoh, Dr. E. Nwelih, Dr. (Mrs.) A.R. Usiobaifo, Dr. F.O. Oliha, Dr. (Mrs.) R. O. Osaseri, Dr. E.C. Igodan, Mr. K. O. Otokiti, Mr. E. Obayagbona, Mrs. R.I. Izevbizua, Mr. J.O. Okhuoya, Mr. O. Evbuomwan and special thanks to other staff in the Department of Computer Science.

To my parents, Mr. and Mrs. Timothy Usunobun, and my siblings, I extend my heartfelt gratitude for their endless love, belief, prayers, support and encouragement in every step of this journey.

Finally, I am thankful to my friends – Mega, Esther, Jojo, Mira and others I can't mention – whose assistance and suggestions have largely contributed to making this work a success.

May God continue to bless and protect you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

API: Application Programming Interface. This is a way for two or more computer programs to communicate with each other.

CNTK: Microsoft Cognitive Toolkit, this is a deprecated deep learning framework developed by Microsoft research.

GSM: The Global System for Mobile Communications was developed by the European Telecommunications Standards Institute (ETSI) to describe the protocols for second-generation (2g) digital cellular networks used by mobile devices such as mobile phones and tablets.

SMTP: The Simple Mail Transfer Protocol is an internet communication protocol for electronic mail transmission.

ABSTRACT

Attendance tracking in classes is a very important activity in any institution, and taking attendance of students using facial recognition is a more efficient and accurate method than the traditional methods which includes paper- based and roll call methods. The facial recognition system is an application of computer vision that can perform two major tasks of identifying and verifying a person from a given database. Facial recognition proves to be more effective in taking attendance than the traditional method, which is inaccurate, time-consuming and vulnerable in most cases of large class environments. This system is designed with a login page for authentication, it also provides a mailing platform where the attendance will be sent to the school authority for record keeping. This study develops a deep-learning-based facial recognition system used to detect the face of students in a class environment for the sole purpose of taking their attendance records.

CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND OF THE STUDY

Attendance is the concept of people, individually or as a group, appearing at a location for a previously scheduled event. Measuring attendance is a significant concern for many organizations, which can use such information to gauge the effectiveness of their efforts and to plan for future efforts (Wikipedia). In the education system, attendance is used as a record to assess students' level of consistency in participance of a class. Educational institutions use paper-based systems to track attendance in numerous countries (Abdalkarim, 2022). These methods have been in use for years and have some challenges as it is tedious.

To maintain track of each student’s attendance, it takes time to call out their names at the beginning of the course. False signatures, names missing from spreadsheets, manually inputting data into systems and the possibility of proxy attendance are further problems. To track attendance more effectively and efficiently, it is crucial to swap these outdated practices for modern ones. These modern practices involve automatic recognition of a particular individual based on distinguishing characteristics such as QR code, ID and password, face recognition and fingerprint recognition. (Abdalkarim, 2022). This research will focus on developing a student's attendance system using face recognition techniques. Biometrics is the science of identifying or verifying the identity of a person based on physiological or behavioral characteristics. (Ruud *et al.* 2003). It is an extremely effective way to identify individuals since the users do not have to remember to possess anything. Biometric authentication is used in computer science as a form of identification and access control. Biometrics are classified into two: behavioral and physiological. Behavioral biometrics include signature, voice, lip motion, etc. while physiological biometrics include face, fingerprint, hand geometry, etc. Due to the high variance found in readings of

behavioral biometrics, physiological biometrics are more suitable for identification mechanism. In recent times, due to the significant improvement in camera and processing quality, face recognition has been proven to be a very convenient and effective biometric method of identification and authentication.

A face recognition system is an application of computer vision that can perform two major tasks, which are identifying and verifying a person from a given database (Choudary *et al,* 2018). The system captures facial patterns such as the distance between eyes, nose shape and the contour of the jaw line, and converts them into a mathematical representation known as a facial template. This template is then compared with a database of pre-registered templates to determine the identity of the person. (Zhang & Wallace, 2019). The facial recognition system is an evolution capable of matching stored data and real-time images, frames or videos to identify a person whose data is already stored in the database by using a deep learning algorithm on the system.

1.2 STATEMENT OF THE PROBLEM

The traditional system of taking attendance is still used in lecture rooms in most institutions today. Lecturers will give out a sheet of paper containing list of student's names to sign on or in some cases, the students write out their names, matriculation number and their signature to indicate their presence for a particular class (Mijanur *et al*, 2016). The problem with this method is that a student can easily sign attendance on behalf of another student. Also, this method is not efficient and requires more time to arrange, record and calculate the average attendance of each student (Unnati and Priya, 2014). This research will help address these limitations and the Students Attendance system using face recognition techniques which will be designed will be a more faster and reliable platform for students attendance to be taken.

1.3 AIM AND OBJECTIVES

The aim of this study is to enhance attendance management at the University of Benin by proposing and validating the feasibility of implementing a facial recognition system.

The Objectives are :

1. Identify existing problems of current attendance systems.
2. Analyze the identified problems, considering factors such as manual data entry errors, time consumption, and potential security issues associated with traditional attendance systems.
3. Develop a comprehensive model for a Smart Based Attendance System utilizing facial recognition techniques.
4. Implement the designed model using appropriate technologies.
5. Conduct rigorous testing to ensure the accuracy and reliability of the facial recognition system under various conditions.

1.4 SCOPE OF STUDY

The scope of this study is to develop a student's attendance system using face recognition techniques to improve the attendance management in the University of Benin. The system will be able to capture students' image frames in order to detect their faces and it will be developed by using the python programming language.

1.5 SIGNIFICANCE OF STUDY

This study will replace the traditional attendance system with a faster, more efficient and reliable alternative, which is facial recognition technology. It provides a secure means of sending the attendance log to authorized personnel (staff), by using a Simple Mail Transfer Protocol (email) and an authentication method to secure the backend of the

program. In summary, this research aims to improve the overall attendance management process thereby delivering a dependable and user-friendly solution.

1.6 LIMITATIONS OF THE STUDY

- Lack of data frames for training: recognizing a student's face will be very difficult because each student will have to be captured with a large number of picture frames so the deep learning can predict with high accuracy. To solve this limitation, this project will be using three subjects for its simulation because capturing many students for this project will result in downtime to complete the study.
- Steady Power Supply: there must be a steady supply of power because without power, the devices to be used cannot function.
- Internet Access: internet connectivity must be available so that when sending the mail, the system will not produce an error.
- Quality Capturing Devices: the camera to be used for capturing must be of good quality, because a bad quality camera would cause the machine to provide an inaccurate result.

CHAPTER TWO

LITERATURE REVIEW

Attendance taking is a very important activity in most institutions as it is used for checking the performance of employees and students.Each and every institute has its own method in this regard. Some of these Institutions take attendance manually using the old paper or file-based approach while some have adopted methods of automatic attendance using some biometric techniques, but in these methods, employees/students have to wait for a long time in making a queue at the time they enter the office/class (Senthamil *et al,*2024).

2.1 ASSESSMENT OF DIFFERENT ATTENDANCE MANAGEMENT SYSTEM

2.1.1 Bluetooth-based Attendance

In 2013, Vishal Bhalla *et al* proposed the attendance system which can take attendance using Bluetooth. Attendance is taken using an instructor's mobile phone, an application software is installed in the instructor's mobile phone, and it enables it to query students' mobile telephone via Bluetooth connection, and through transfer of student's mobile telephone Media Access Control [MAC] addresses to the instructor's mobile phone, the presence of the student can be confirmed. The problem of this system is that student's phone is required for attendance, therefore the presence of a student is not guaranteed, only his/her should be in coverage area.

2.1.2 Iris-based Attendance System

In 2010, Seifedine Kadry and Mohamad Smaili proposed a system. In their paper, a wireless iris recognition attendance management system is designed and implemented using Daugman's algorithm (Daugman,2003). This system-based biometrics and wireless technique solves the problem of spurious(fake) attendance and the trouble of laying the

corresponding network. It can make the user's attendance more easily and effectively. In this paper, RF wireless technique is beimg used for employee identification. The primary issue with this method is its cost; each student must wait in a lengthy line to have their presence marked by an iris scanner, and it is extremely short in distance.

2.1.3 Fingerprint-based Attendance System

Mohamed *et al* (2012) designed a fingerprint device that is used in fingerprint attendance system. The students mark their presence by placing their finger on the devices sensor. The components of this system are:

- Handheld device which was constructed and controlled by microcontroller with components (fingerprint module, Real Time Clock, buttons, Graphic liquid crystal display, memory,etc).
- Host computer with GUI application for managing the attendance, the application is used to transfer the student's data to the device. The attendance details can be accessed through USB interface and finally stored in the database. The problem in this system is the fingerprint device as it gets damaged very frequently. Waiting in line for a long time for pupils to use the fingerprint device is another problem.

(Rao *et al,* 2013) proposed an employee attendance management system using fingerprint recognition. Every check-in and check-out time, employees needed to scan their fingerprint to record attendance. The suggested attendance system uses alignment-based greedy matching along with minutiae-based matching to identify scanned fingerprints. The suggested system is not suitable for a course attendance system, even though the authors claimed that it is inexpensive and simple to use. This is because the system needs a lot of fingerprint recording devices when there are many classes going on at once. Moreover, if there is a large number of students in a course, the system will cause a long and time-

consuming queue.

Later on, (Rao *et al*, 2013; Zainal *et al*, 2016; Zainal *et al*, 2014) proposed a portable device for student attendance system based on fingerprint recognition. This proposed system asked a student to scan his/her fingerprint to the device for attendance process. The attendance data was only stored on the devices. The device was not directly connected to the server, as consequent, the lecturer needs to back up the data to the server manually after class hour. In addition, many devices are needed if there are many classes at the same time.

Soewito *et al* (2015) proposed an employee attendance system on android smartphone using fingerprint and GPS integrated with payment system. From the user smartphone, the system recorded fingerprints and attendance time, and the coordinate of position through GPS that is available on the smartphone to avoid a long queue and fake attendance. The problem with this system was that not all android smartphones are equipped with a fingerprint scanner. Moreover, recording the user position through GPS on an android smartphone is inaccurate. According to Baver (2013), GPS on android smartphone had a deviation about 10-93m from the actual position. Therefore, employees who are outside the office but still close enough to the office can be recorded as present.

Almost all attendance systems based on fingerprint recognition didn't report recognition accuracy except the system proposed by Zainal *et al* (2016), which is 85% with total recognition time around 7-9 mins for 27 students. Moreover, there is a drawback in the system, as reported by Zainal *et al* (2016), the system cannot recognize a fingerprint if it is wet, dirty or broken.

2.2 OVERVIEW OF FACIAL RECOGNITION FOR ATTENDANCE

Sharanya *et al* (2020) proposed a face recognition attendance system. This system proposes a single image-based face likeness detection method for discriminating 2-D paper masks from the live faces. In this method, the camera is fixed within the classroom and it will capture the image, the faces are detected and then it is recognized within the database and finally the attendance is marked. Still images taken from live faces and 2-D paper masks were found with the differences in terms of shape and detailedness. Frequencies and texture information were exploited using various algorithms in order to effectively employ such differences.

Piyushi (2017) proposed face liveness and disguise detection system which eliminates the chances of a person to fake his/her identity. The face recognition systems available in the market fail to detect the fake faces shaped using high-end silicone masks and prosthetics. Also, these systems misinterpret faces from physical photographs as real faces. These are the vulnerabilities that are present in almost all face recognition systems. This system works on the principle that the surface temperatures of masks are close to ambient temperatures. This system captures image from the webcam connected to Raspberry Pi and then it is processed by OpenCV to detect the face in the image. The temperature of the face captured by the camera is obtained by IR temperature sensor. If the face is detected in the image and it's temperature is more than the threshold value (skin temperature), then the face is real, otherwise it is fake.

Mr. Kaisseluanathan *et al* (2018) presented facial recognition as a technology of biometrics which has been used in many areas like security systems, human machine interaction and image processing techniques. The main objective of this paper is to calculate the attendance of students in an easier way. A system called automated attendance management system was proposed. This system calculated attendance

automatically by recognizing the facial dimensions. The algorithm used in this system is Eigenfaces. This system does not only detect the faces but also the distance of the facial characters under varying conditions.

Nandhini *et al* (2019) presented automatic face recognition technologies have made many improvements in the changing world. Numerous algorithms and techniques have been developed for improving the performance of face recognition but the concept to be implemented here is Deep Learning. It helps in conversion of the frames of the video into images so that the face of the student can be easily recognized for their attendance so that this attendance database can be easily reflected automatically.

2.3 DEEP LEARNING

Before discussing deep learning, we need to understand where deep learning stands in the context of computer technology.

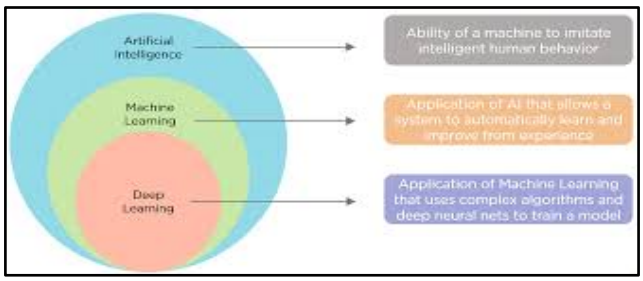

Figure 2.1: A Diagram showing the relationship between Artificial Intelligence, Machine

Learning and Deep Learning (Google; Simplilearn.com; 2024).

2.3.1 ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is a science and engineering subject that deals with intelligent behavior. It is a subfield of computer science that has improved human existence in different ways. Artificial Intelligence is a combination of reasoning, learning, problem-solving perception and language understanding (Agrawal et al, 2022). AI is made of two words: 'Artificial' [meaning something made by humans or non-natural things] and 'Intelligence' [meaning the ability to understand or think accordingly]. Therefore, AI can also be simply defined as training machines/computers to mimic the human brain and its thinking capabilities. AI focuses on three major skills: learning, reasoning and self-correction, to obtain the maximum efficiency possible (Zhu,2022).

2.3.2 MACHINE LEARNING

Machine Learning is the study/process of allowing a system (computer) to learn automatically from its own experiences and improve accordingly without being explicitly programmed. Machine Learning (ML), a subset of AI, allows computers to detect patterns from large complex datasets automatically and uses these patterns to make predictions (Wang et al. 2019). ML focuses on the development of programs so that it can access data to use it for itself. The entire process makes observations on data to identify the possible patterns being formed and make better future decisions as per the examples provided to them.

2.3.3 DEEP LEARNING

Deep Learning is a type of machine learning that uses artificial neural networks with

multiple layers of processing units. These networks are trained on large datasets of labeled examples, and they can learn to perform complex tasks such as image recognition, natural language processing, and speech recognition (Zhang et al. 2019). It is a sub-part of the broader family of machine learning. Deep Learning algorithms focus on information processing patterns to identify patterns in data, just like the human brain does, and classify the information accordingly. Deep learning works on larger sets of data when compared to machine learning and the prediction mechanism is self-administered by machines of technology. Deep learning is successful in several domains, ranging from acoustics, images, and to natural language processing. However, applying deep learning to ubiquitous graph data is non-trivial because of the unique characteristics of graphs. Recently, substantial research efforts have been devoted to applying deep learning methods to graphs, resulting in beneficial advances in graph analysis techniques (Zhang et al, 2022). Deep learning imitates the human brain's neural pathways in processing data, using it for decision-making, detecting objects, recognizing speech and translating languages. Deep learning can be implemented using several programming languages but one of the most effective languages that are used in the world is Python. Python provides a wide variety of packages that can be used to develop programs with large or medium impact to the world of deep learning.

Python is a high-level, general-purpose programming language, it is dynamically-typed and garbage-collected. Its design philosophy emphasizes code readability with the use of significant indentation. Python supports multiple programming paradigms, including structured, object-oriented and functional programming. In this project, the python programming language will help in developing the facial recognition system, using a framework called Keras. Keras is embedded in a Python package called TensorFlow, additional packages that will be used are OpenCV, pandas, NumPy and SMTP to design the facial recognition system.

2.4 BIOMETRIC SYSTEM FOR ATTENDANCE

A Biometric system is made up of diverse methods to identify individuals. This session will be addressing the different method of biometrics that was adopted for the attendance base system. Many authors have written about attendance systems but the more efficient method is biometrics which makes the signing of attendance automatic. The application of biometric recognition in personal authentication enables the growth of this technology to be employed in various domains. The implementation of biometric recognition systems can be based on physical or behavioral characteristics, such as the iris, voice, fingerprint, and face. Currently, the attendance tracking system based on biometric recognition for education sectors is still underutilized, thus providing a good opportunity to carry out interesting research in this area. As evidenced in a typical classroom, educators tend to take the attendance of their students by using traditional methods such as calling out names or signing off an attendance sheet. Yet, these types of methods have proved to be time-consuming and tedious, and sometimes, fraud occurs. As a result, significant progress had been made to mark attendance automatically by making use of biometric recognition. This progress enables a new and more advanced biometric-based attendance system being developed over the past ten years. The setting up of biometric-based attendance systems requires both software and hardware components. Since the software and hardware sections are too broad to be discussed in one paper, this literature survey only provides an overview of the types of hardware used. Emphasis is then placed on the microcontroller platform, biometric sensor, communication channel, database storage, and other components to assist future researchers in designing the hardware part of biometric-based attendance systems (Hoo *et al,* 2019).

In a comprehensive review by Chatzis (2022), the state-of-the-art in biometric attendance

systems is thoroughly explored, encompassing various biometric data types such as fingerprints, facial recognition, and iris scans for attendance tracking. Both commercial and open-source biometric attendance systems are discussed, highlighting their benefits in terms of accuracy, security, and convenience, along with the challenges they pose, including implementation costs, privacy concerns, and user acceptance issues. This review serves as a valuable resource for understanding the current landscape of biometric attendance systems and offers valuable insights into potential improvements in this rapidly evolving field.

2.4.1 OPENCV

OpenCV (Open Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel. The library is cross-platform. It focuses mainly on real-time image processing. The library was originally portable to some specific platforms such as digital signal processors (Kumbhar *et al*, 2017).

The attendance management system is a required tool for attaining attendance in any habitat where attendance is essential. Yet, many of the available techniques consume time, are invasive and it demands manual work from the users. This research is directed at building a less invasive, cost-effective, and more efficient automated student attendance management system using face recognition that leverages OpenCV functions for facial recognition. The system provides a GUI for marking attendance. It provides an interface for updating attendance using facial recognition libraries of OpenCV. The system stores attendance in a database which is maintained by the administrator. The administrator can view, update, and change the attendance of the students. The students can view and update their attendance. The system is developed on an Open-Source image processing library and the interface is developed using the Python Tkinter module. The Tkinter module

is an open-source module by which programmers can develop GUI screens, hence, it is not software-dependent. The OpenCV module used for image processing is interfaced using Python (Duggempudi, 2021).

This research aimed at studying the current methods of attendance used at higher institutions of learning in Uganda and the feasibility of using facial biometrics as a new method of capturing attendance. Facial biometrics is distinct from other biometrics because it can be carried out without the consent of the person involved. As a result, the researcher developed a face recognition attendance system using OpenCV and Microsoft Azure CS. Questionnaires, interviews, and observations were used to capture data for the research. The data were analyzed using SPSS to get the requirements and systems functionalities. Object-Oriented Design tools were used to model the architecture of the system. A data Flow Diagram, Use-Case Diagram, Activity Diagram, and Flow Chart were used for processing whereas s Entity Relation Diagram was used for data modelling. The system was designed to facilitate the attendance management of a large number of attendees with ease. Efficiency and reliability were essential features of the system. Data visualization was provided to help management make informed and timely decisions on management matters that are related to attendance. The system was developed using Python Tkinter, OpenCV, and Azure CS as mentioned above. The data (images) used by the system were stored in the cloud for accessibility by multiple users. The system was tested thoroughly using various testing types to uncover and fix errors and minimize the severity of failures (Alqqmeee, 2022).

2.5 SECURITY IN FACIAL RECOGNITION SYSTEM FOR ATTENDANCE RECORD

With the use of facial recognition in the attendance system, concerns about security and

privacy have been raised. As facial recognition systems gather biometric data, there are concerns regarding data breaches, unauthorized access, and potential misuse of this data. Therefore, it is crucial to implement security measures to safeguard the facial recognition attendance system. This involves ensuring secure storage and transmission of biometric data, adopting encryption techniques, implementing access controls, and regularly updating the system to patch security vulnerabilities. Additionally, educating users about the importance of data privacy and obtaining their consent for using their biometric data is vital in maintaining the trust and ethicality of the facial recognition attendance system (Jindal & Singh, 2020).

2.5.1 SECURED LOGIN INTERFACE

Chandavarkar (2021) conducted groundbreaking research on enhancing account security through a login interface, addressing the vulnerabilities of traditional username and password authentication systems. The exponential technological advancements have rendered the conventional login methods susceptible to various cyber-attacks, jeopardizing user privacy and data. To counter these risks, Chandavarkar's study introduces a cutting-edge two-factor authentication system, incorporating the dynamic One Time Password (OTP) mechanism. OTP involves generating a randomly fixed-digit code sent to the user's physical device upon entering the correct password. This additional layer ensures that only authorized users can access the system, significantly bolstering security. The dynamic nature of OTP makes it applicable in various scenarios, such as secure bank transactions, social media account deletions, and cloud platform access. Chandavarkar's research delves into the lifecycle of OTP, analyzing challenges and issues in the current technological landscape, further contributing to the field of cybersecurity.

2.5.2 SECURED TRANSFER SYSTEM

Helm (2021) “supports the contention that a cyber secure Internet voting system that significantly reduces the opportunity for mail-in voter fraud, helps to ensure privacy for the voter, including nonrepudiation, non-attribution, receipt freeness, and vote acknowledgment can be created using existing technology”.

CHAPTER THREE

INVESTIGATION, SYSTEM ANALYSIS AND DESIGN

3.1 INTRODUCTION

Regular attendance at educational institutions is crucial, as it signifies students' dedication and is frequently used as a criterion in determining final grades. The methodology employed in this study's data acquisition and analysis is outlined in this chapter.

3.1.1 DATA COLLECTION

Data collection was done with the use of e-questionnaires, which were meticulously designed using Google Forms to enhance accessibility, convenience, and ease of use. These e-questionnaires were sent out to students via the WhatsApp messaging platform, aiming to gather their perspectives on the existing attendance system and identify potential areas for improvement.

3.1.2 POPULATION OF THE STUDY

The targeted population happens to be undergraduate students in tertiary Institutions in Nigeria. E-questionnaires were used to collect data from this category of respondents. With a sample size of 100 undergraduate students, 68 responses were obtained.

3.2 SYSTEM ANALYSIS

A system can be said to be ‘an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal’. (Hitesh, 2016).

System analysis is the act, process or profession of studying an activity in order to define its goals or purposes and to discover operations and procedures for accomplishing them

most effectively. (Hitesh, 2016). It is a method of problem-solving that breaks down a system into its constituent parts in order to examine how well those parts function and interact to achieve the goal.

3.3 ANALYSIS OF THE EXISTING SYSTEM

The current system employs a manual method of capturing attendance through paper. During classes, lecturers distribute paper forms to students, who record their presence by writing their names and signatures. The complete attendance records for each session are maintained in a logbook. At the conclusion of the session, these records are utilized to generate reports for assessing student participation and performance.

3.3.1 PROBLEMS OF THE EXISTING SYSTEM

From the distributed e-questionnaires, diverse data were gathered from undergraduate students in federal, state, and private tertiary institutions across the nation. Of the respondents, 97% utilize the current system (paper-based method) for attendance recording in their respective institutions. Some of the questions asked are as follows:

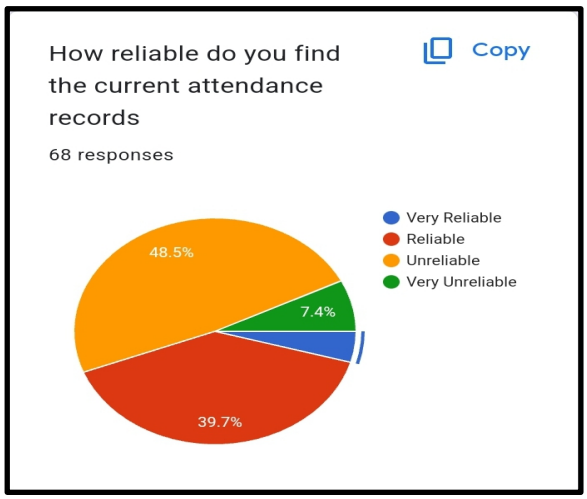

Figure 3.1: Question 1

Based on Figure 3.1 above, 48.5% of the sample population perceive the current attendance system as unreliable. Furthermore, 7.4% classify it as extremely unreliable, while only 39.7% consider it reliable and 4.4% consider it highly reliable. In summary, the data suggests that a substantial majority of the sample population holds reservations about the trustworthiness of the current attendance system.

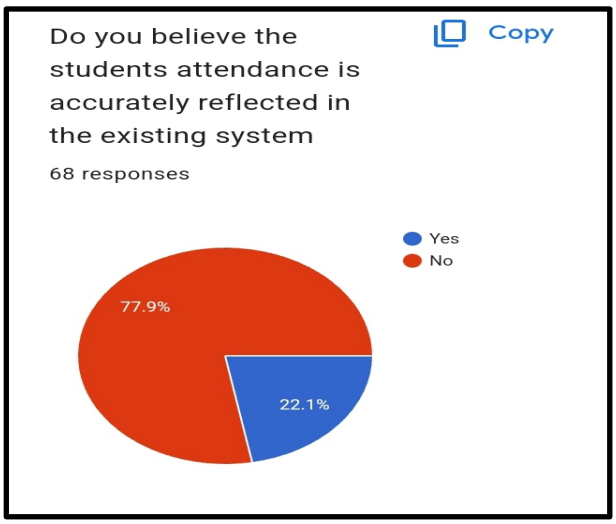

Figure 3.2: Question 2

According to the data presented in figure 3.2 above, a mere 22.1% of respondents express confidence in the accuracy of the current student attendance system while the substantial

majority of 77.9% disagree with this. These inaccuracies may be as a result of manual data entry errors.

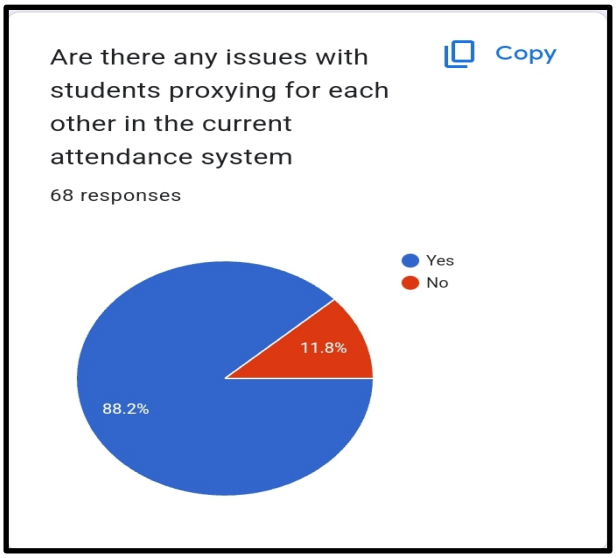

Figure 3.3: Question 3

Figure 3.3 highlights an additional concern with the current system - the prevalence of students acting as attendance proxies for others. 88.2% respondents agreed that students actually proxy for each other while 11.8% disagreed.

Proxy attendance involves one student signing the attendance register on behalf of another student who is not present in class.

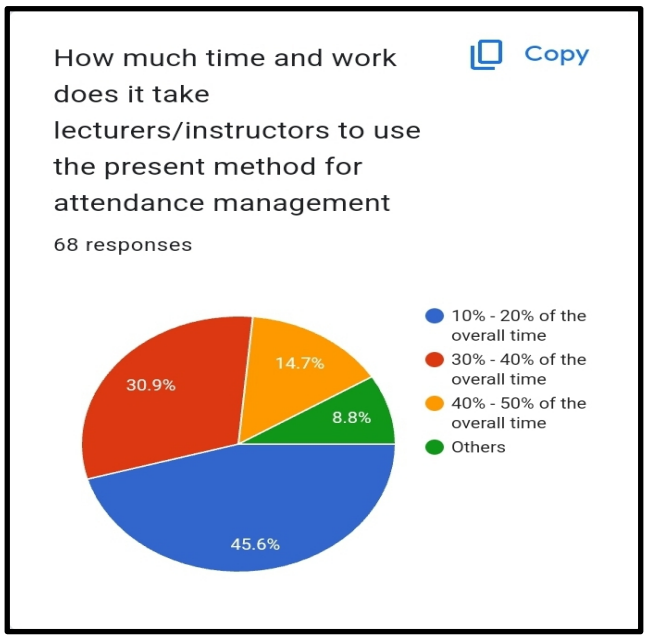

Figure 3.4: Question 4

Figure 3.4 illustrates the proportion of time spent by lecturers/instructors on attendance-taking in the existing system. As evident from the chart, attendance management using paper-based methods consumes a substantial amount of time.

In summary, using the data gathered from the e-questionnaires sent out, it has been concluded that the existing system i.e. paper-based system is unreliable as students' attendance are sometimes recorded inaccurately and data is not maintained efficiently. All

calculations to generate reports are done manually so it is error-prone. The system also requires a lot of paperwork and it is time-consuming. Proxy attendance and falsification of signatures by students affect the correctness of generated reports.

The problems listed above led to the need for the proposed system which takes students' attendance using facial verification.

3.4 ANALYSIS OF THE PROPOSED SYSTEM

The proposed system is a smart based students attendance system using facial recognition techniques. In line with its title, this system is intended to record student attendance using their facial features. For identifying a face, a secondary dataset was used for the system to know and identify the shape of a face. In order to enable the machine to learn and verify a student's face with an accuracy of around 98.9% to 99.99%, the student's image frame will also be acquired for facial verification.

3.4.1 FACE CAPTURED BY THE SYSTEM

Figure 3.5 shows how the system will capture the image of the student; it will take a 1x1 inch size for the face. With the help of OpenCV and “haarcascade_frontalface_default” in XML format, it was able to locate the position of the face in the camera, so that the system will crop the path where the face is, hence, saving the image in a folder named with the student id, for example, “AFIT-CYS-18-0027”. When taking the image of the students the system will capture over 501 images which will be used to train the system.

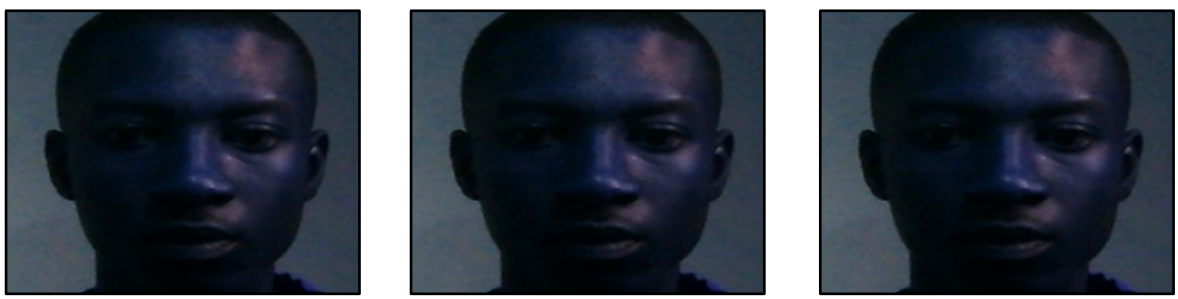

Figure 3.5: Captured images of a student

3.5 MODEL PROCESS

This is the steps that will be used for the project; it will also demonstrate how the system works from the beginning, with resource acquisition occurring before system deployment and system management after.

3.5.1 DEVELOPMENT LIFE CYCLE

The build-up/ development of the system will be achieved in a series of steps, the first step is acquiring resources which will be used for the development. The research obtained a software that will be used for developing the program called “PyCharm”, and it got some graphical images for the GUI (Graphical user interface). The next phase is the coding and implementation. This phase covers the programming language used to develop the project which was Python, and also the package (modules) in Python that enabled us complete the project work (to complete the objective which was listed in chapter 1). This next phase is the capturing of student images which are shown in figure 3.1 and below in figure 3.6, it explains how the capturing is carried out. The testing phase follows next, here, the whole system will be debugged and will make sure the program achieves its objective. Then the program will be deployed to the system. The last phase is management, in this phase,, the registration of students will be conducted and the system will be maintained in an active state.

3.5.2 ACTIVITY DIAGRAM SHOWING HOW THE SYSTEM WILL FUNCTION

Figure 3.6 shows how the model and how the program activity will be processed from one stage to another, it gives a view of the available navigations in the system and what you

will find on each home page of the system. From the diagram, the system “connects to DB(Database)”, before the system lets you do anything on it. The system will have to connect to the database so that when the admin is trying to log in, it will check from the database if the admin information matches the one in the database, thereafter the admin will be permitted access to the system, on the other hand, if the system does not connect to the database, the program will end and display an error.

Subsequently, the administrator proceeds to log in or register. In the event of registering, the admin is required to submit their chosen username, email address, and password for secure storage in the system's database. Upon successful registration, the admin is prompted to log in using their credentials to gain authorization within the system.

The next step in this model is to register students by asking them for their department and student ID (matric number). Thereafter, the student's image is captured and saved in a folder with his/her student ID(matric number) and the image will be saved with a file name ranging from 1 to 501.png. The system's interaction with the camera to recognize faces and take images is illustrated in Figure 3.6. Figure 3.7 below illustrates how the verification process works. It takes attendance, verifies the student's identification, and logs the student's name into an Excel sheet.

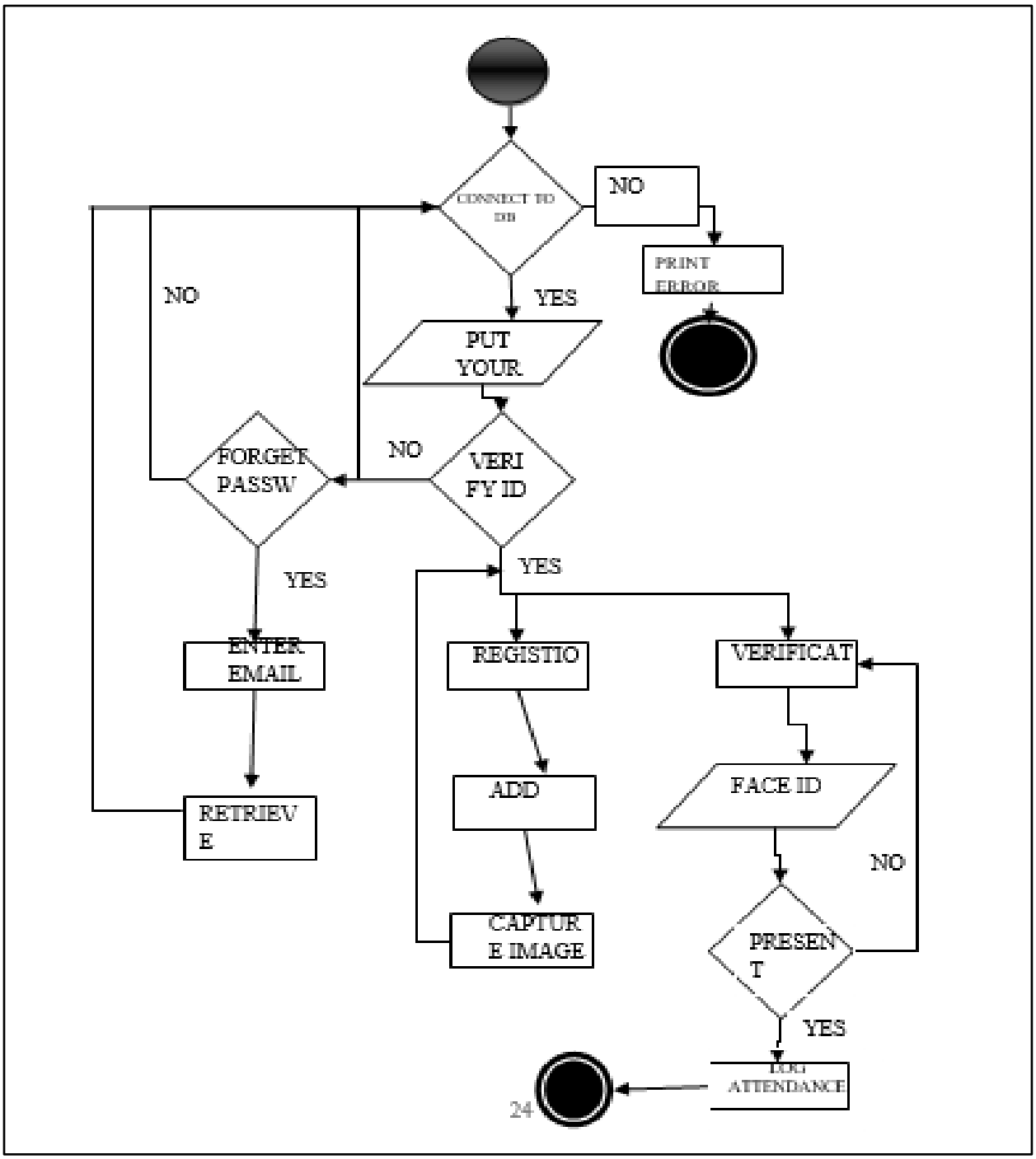

Fig 3.6: Activity Diagram showing how the system will function

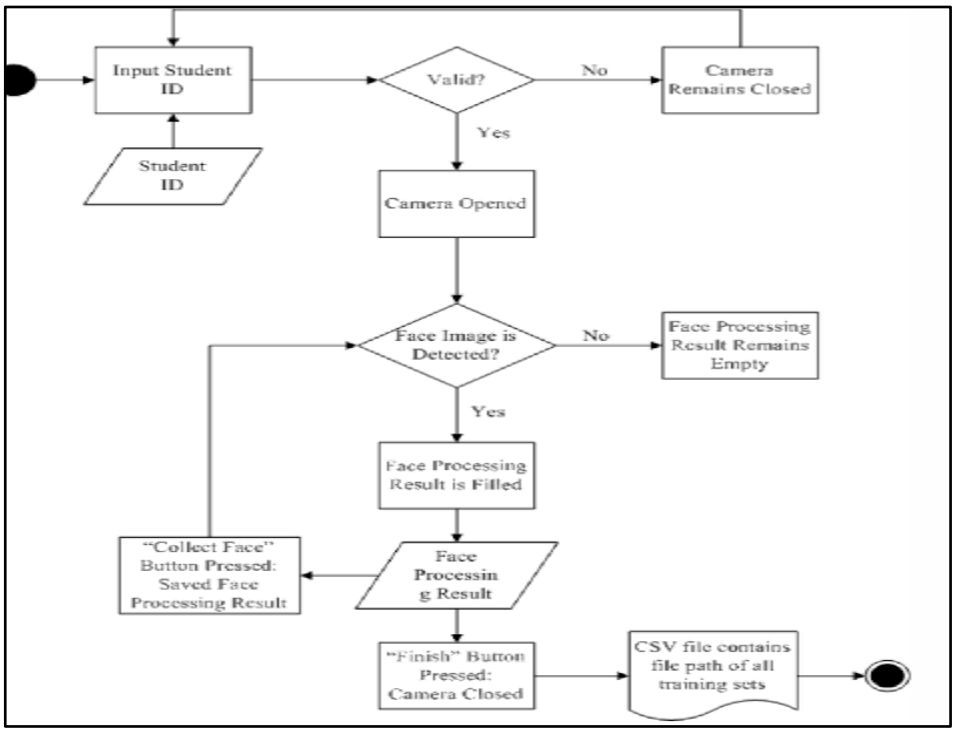

Figure 3.7 Model for the facial recognition system when capturing the data

Figure 3.7 above shows how the system will capture the image of the student. The model provides the system with the ability to identify if a face is on the camera, it proceeds to crop the student's image and repeats this action 501 times. Then, the model ceases its

operation and save the cropped images in a designated folder.

3.5.3 THE MODEL FOR TESTING THE DATA CAPTURED

Figure 3.8 illustrates how the machine will be able to check and identify the right person on the system and input the student ID in the excel sheet. In figure 3.8, the feature database is where the trained model will be saved, it is called “trained _data.h5”, it will use OpenCV

and TensorFlow to be able to identify the face of the student present.

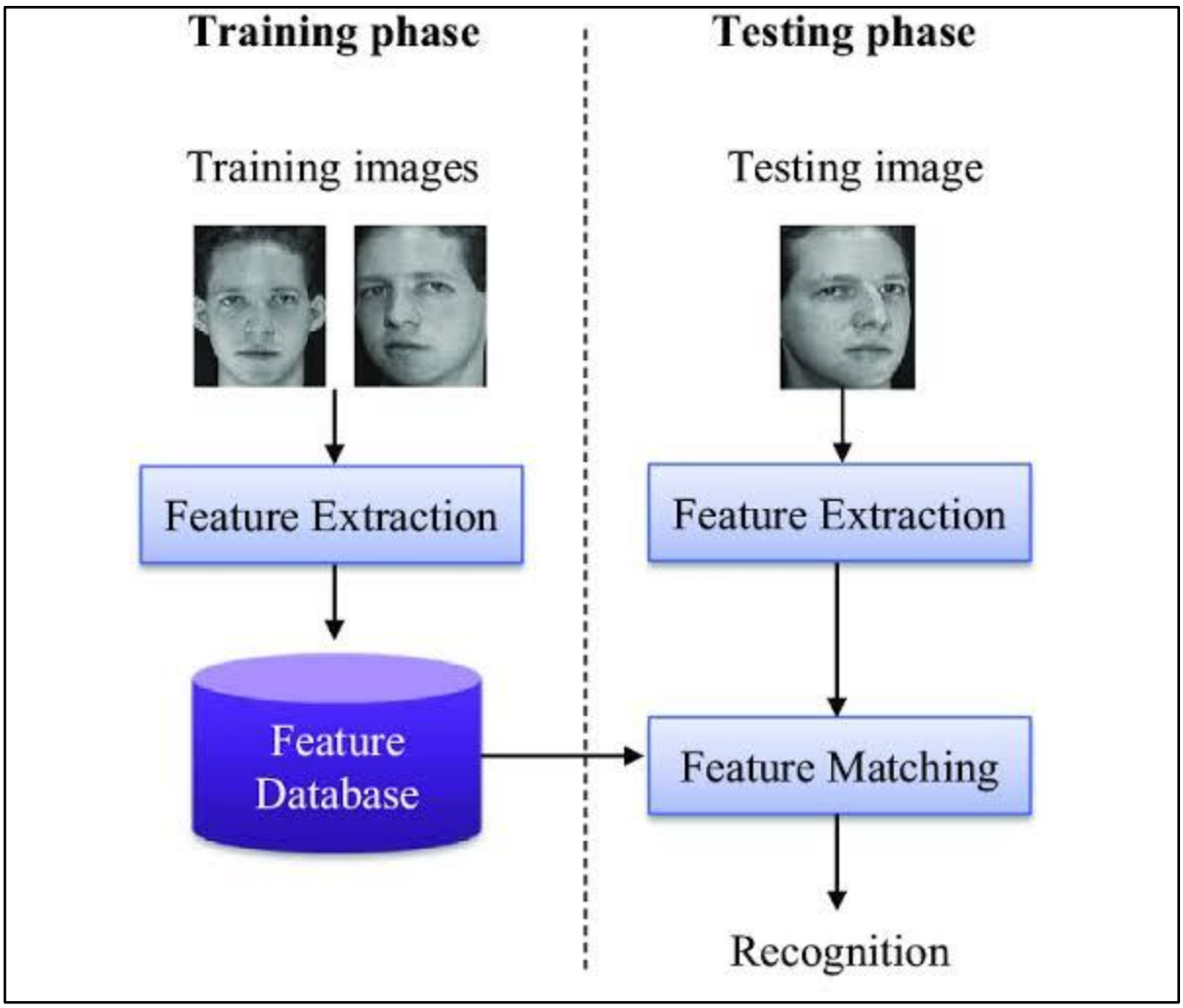

Figure 3.8: The model for testing the data captured

[https://images.app.goo.gl/kBzAcZRG87dD8sTVA]

3.5.4 THE MODEL USED TO SEND THE ATTENDANCE VIA MAIL (SMTP)

For the system to send the model, it used the SMTP (Simple Mail Transfer Protocol) package in Python. We can see that the model in Figure 3.9 showed automated email encryption, for this purpose the system used the google mail server to help encrypt the message and the document. This package requests the Gmail and the password to be passed before it can send the mail to the recipient, also this project created an email account called “projectfestus@gmail.com” which will be the default email used to send the mail to the sender. Google mail also scans the file sent to find out if it has a virus in it or not.

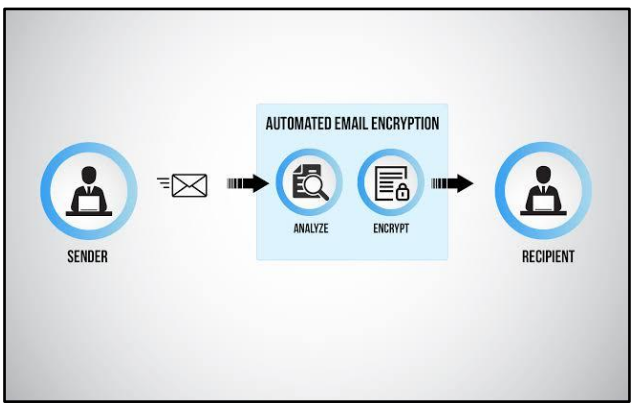

Figure 3.9: The model used to send the attendance using mail (SMTP)

[https://images.app.goo.gl/t8KpSqzczpCk13mY6]

3.5.5 THE MODEL USED FOR THE SECURITY OF THE FACIAL RECOGNITION SYSTEM

The security login page is configured as shown in Figure 3.10. Due to the reason that conditional statements are used to control the number of times an admin tries to log in to the system, the application will terminate if the admin enters an incorrect password five times.

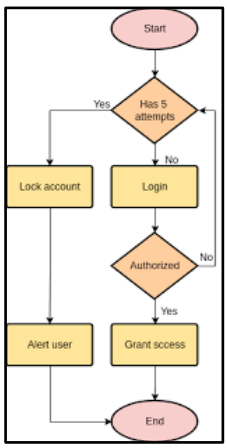

Figure 3.10 The Model Used For The Security Of The Facial Recognition System

[https://images.app.goo.gl/AHxTcAgubdeWizW36]

3.6 CODING AND IMPLEMENTATION

For this research, the following packages will be used for implementing the research, both the graphical user interface and the command line interface.

3.6.1 PYTHON FOR IMPLEMENTATION

TensorFlow will be used to provide the processing power to run a deep learning algorithm, where layers of training frame will be defined, validation and training data variable will be assigned, compilation of trained model will be stored in a file for easy testing so as to reduce the time for training and testing, and processing of image files which will be used to train the model (using a dataset).

3.6.2 TRAINING ILLUSTRATION

Table 3.1 describes how the system is trained. The system takes 1 minute to train the machine for one registered student and 6 minutes for 2 registered students.

Registered Users	Epochs	Time	Accuracy
1 person	1	1 minute	100%
2 people	1	6 minutes	96%
2 people	2	13 minutes	100%

Table 3.1: Training Illustration

3.6.3 TRAINING THE CNN (CONVOLUTIONAL NEURAL NETWORK)

The image in figure 3.11 is showing the hidden layers in the model, we can see that over 13 layers are being used, passing the image in through the input. This process is used to train the machine, also the input determines the output. This will help the model easily identify the student in different classes.

The system was able to reduce the time for testing and verifying the face by saving the trained model in a file named “trained_data.h5” (.h5 is one of the file formats which is used to save the trained model in TensorFlow). The file format is better than the others because it takes less file space and has full information about the trained model.

Figure 3.12 illustrates how the image is inputted into the convolutional layer; this layer reshapes the inputted data from 36x36 to 11x11. The data moves from one convolutional layer to another, using max pooling to compress the data. Furthermore, the CNN is

connected together to give an output.

```
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d (Conv2D)              (None, 254, 254, 256)     7168

 max_pooling2d (MaxPooling2D  (None, 127, 127, 256)    0
 )

 conv2d_1 (Conv2D)            (None, 125, 125, 128)     295040

 max_pooling2d_1 (MaxPooling  (None, 62, 62, 128)       0
 2D)

 conv2d_2 (Conv2D)            (None, 60, 60, 64)        73792

 max_pooling2d_2 (MaxPooling  (None, 30, 30, 64)        0
 2D)

 conv2d_3 (Conv2D)            (None, 28, 28, 32)        18464

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)        0
 2D)

 conv2d_4 (Conv2D)            (None, 12, 12, 16)        4624

 max_pooling2d_4 (MaxPooling  (None, 6, 6, 16)          0
 2D)

 flatten (Flatten)            (None, 576)               0

 dense (Dense)                (None, 256)               147712

 dense_1 (Dense)              (None, 2)                 514

=================================================================
Total params: 547,314
Trainable params: 547,314
Non-trainable params: 0
_________________________________________________________________
```

Figure 3:11: The layers used in the model for training the machine

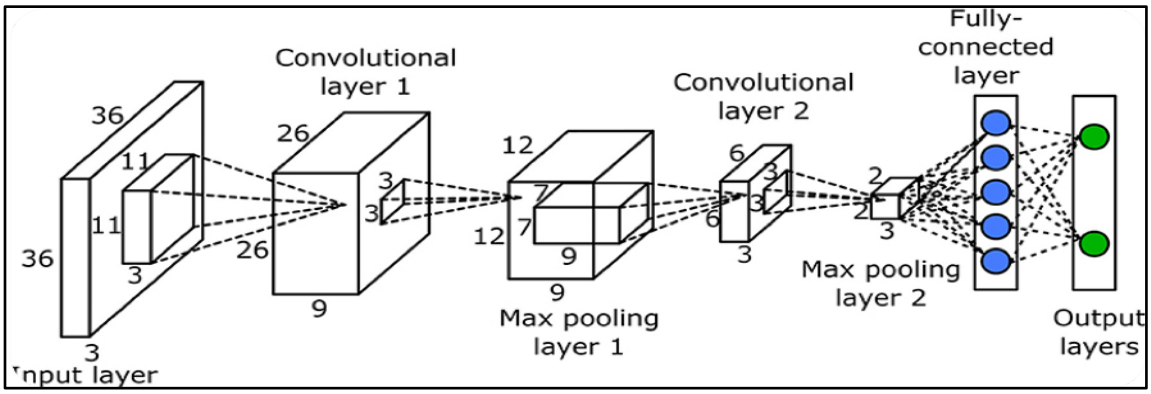

Figure 3:12: Training the CNN

3.6.3.1 TKINTER

Tkinter is a Python binding to the Tk (GUI toolkit). It is the standard Python interface to the Tk GUI toolkit and is Python's de facto standard GUI. Tkinter is included with standard Linux, Microsoft Windows and macOS installs of Python. The name Tkinter comes from the Tk interface. This will be used to design the graphical user interface for the research, it will have a login page where the admin will be verified.

3.6.3.2 OPEN CV

OpenCV will be used to capture the face of the student. By using this package, real-time detection of the face will be possible.

3.6.3.3 HOW THE IMAGE WILL BE CAPTURED AND RESHAPED

The Python package OS (operating system), is used to navigate through an OS and select files in the operating system being used. For this program, OS will be used to get the names of nested folders which are labeled (named with the student ID) for training the model.

Figure 3:13: How the image will be captured and reshaped

To secure the login page, the program used MySQL (Structured Query Language) to store the user ID and password, the program also hashes the password of the admin with MD5,

so that the password won't be in plain text.

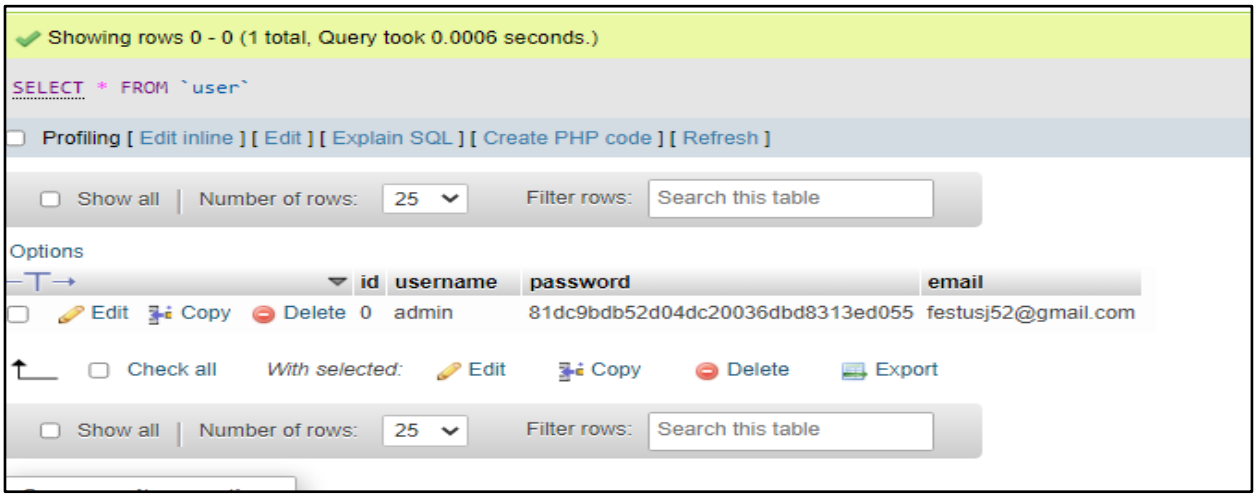

Figure 3:14: MySQL user table

3.7 SECURITY

For protecting the backend (CLI: Command Line Interface), an authentication page will be created to control access to the admin page. This will ensure confidentiality and integrity in the attendance record, also, unauthorized users cannot gain access to the information

and cause a change in data.

3.8 CAPTURING THE DATA

For the system to be able to capture the face, which will be used to train the model, a dataset was used to help the system to identify the location of the face in a camera. The dataset is known as “haarcascade_frontalface_default.xml”. It is an open-source dataset popularly used in face recognition or face identification.

However, the advent of artificial intelligence (AI) and machine learning (ML) opened new possibilities for big data capture, enabling self-learning tools that can automate the gathering, processing, and analyzing of enormous datasets for business use cases.

3.9 LIMITATIONS

An outline showing the limitations the study encountered:

- When building the model, the Sequential, compilation, and fit developed a bug in which the model was not able to identify more than two students. To fix the problem, the system used the “categorical_crossentropy” for the loss and “softmax” for the activation (Figure 24).

- When adding the layers in the model, the “Dense” layer developed an error. To fix this error the “Dense” layer used the number of the class index, to enable the model to be able to classify each class based on the class name. (For example, if four student’s data are registered, the system recognizes it as four classes, hence, the ID of each student is

mapped to an index value).

CHAPTER FOUR

IMPLEMENTATION AND DOCUMENTATION

4.1 SYSTEM REQUIREMENTS

System testing and implementation are critical steps in the building of a reliable system. For proper operation and execution, all computer software requires a certain set of component resources. The system requirements for implementing the facial recognition-based Smart-based Attendance System are listed below.

4.1.1 HARDWARE REQUIREMENTS

- CPU speed: 2.30GHz
- Processor: 64 bits
- External drives: USB
- RAM: 8gb
- Camera

4.1.2 SOFTWARE REQUIREMENTS

- Operating System: Windows 10 pro
- Excel
- PyCharm
- Python

4.2 CHOICE OF PROGRAMMING LANGUAGE USED

Python is a general-purpose language that traces back to the late 1980s. It was created by Guido Van Rossum and has grown to be one of the widest used programming languages in the world. Python is used in fields such as web development, data science, artificial intelligence, etc. Its design philosophy emphasizes code readability and its syntax allows

programmers to express concepts in fewer lines of code compared to other languages. This study will use the python programming language to code and develop the attendance system. TensorFlow will be used to provide the processing power to run a deep learning algorithm, where layers of training frame will be defined, validation and training data variable will be assigned, compilation of trained model will be stored in a file for easy testing so as to reduce the time for training and testing, and processing of image files which will be used to train the model (using a dataset).

4.3 SYSTEM IMPLEMENTATION

This section will provide a screenshot of the result, the code, and how each page line of code maps to its result.

4.3.1 LOGIN PAGE

This is the first page that is used for the authentication of the admin, it takes the input of the username and password or adds a new admin user. Figure 4.3 shows the graphical user interface of the security page whose model was illustrated in Figure 3.6. This GUI was used just to make the interface of the system friendly to the admin.

Fig 4.1: GUI Login Page

4.3.2 SIGN-UP PAGE FOR THE ADMIN

The sign-up page is used to register an admin, and in the backend, it saves the data in the database which was shown in Figure 3.11. It requires an input of “username”, “password”, and “email”. This information will be used to authenticate the new admin.

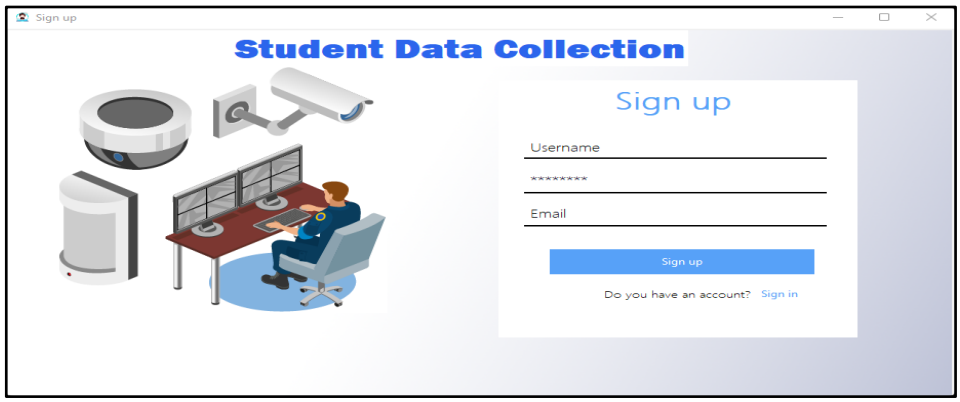

Fig 4.2: GUI Sign-Up Page

4.3.3 HOME PAGE FOR REGISTRATION OF STUDENTS, MODEL TRAINING AND VERIFICATION OF STUDENTS IDENTITY

Figure 4.3 below shows the graphical user interface of the security page whose model was illustrated in Figure 3.6. This GUI was used just to make the interface of the system friendly to the admin. The page in Figure 4.3 also shows the navigation. If the admin wants to register a new student or train the model, a class is used to code the register, train model, and verify users.

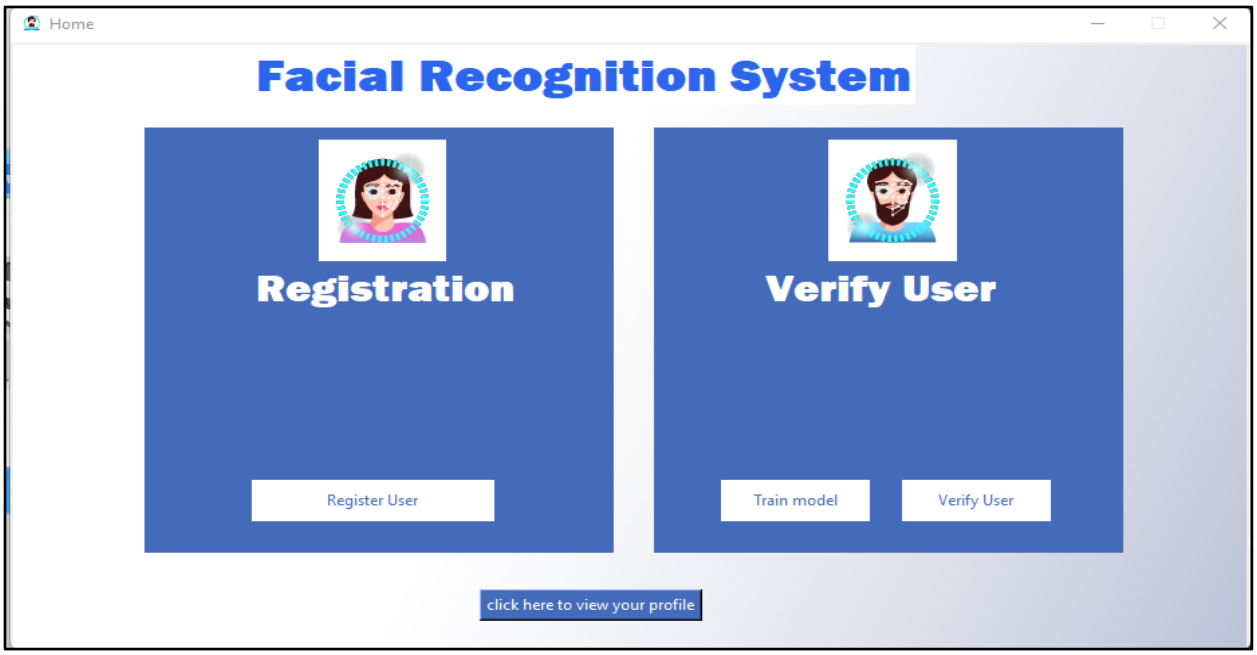

Fig 4.3 : GUI Home Page Where Registering of Students, Train, Or Verify The Student Identity

4.3.4 REGISTRATION PAGE OF STUDENTS

After clicking on 'register user', this page shows a button to add a new user, this button makes sure the admin wants to add a new user, as illustrated in Fig 4.4.

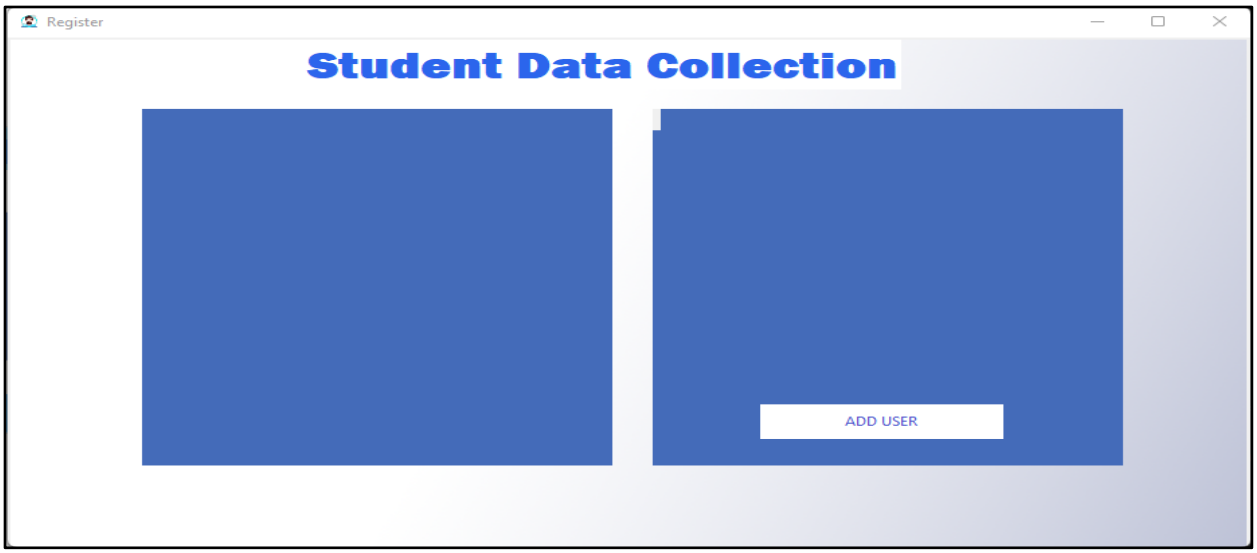

Fig 4.4: GUI Registration page of Students

4.3.5 REGISTRATION PAGE OF THE STUDENT (FILL IN STUDENT INFORMATION)

Fig 4.5 shows how the admin user will fill in the student information so that it can be stored in the right directory, the code will create a folder named {the department you inputted}, and {the student ID you inputted}, and the image will be saved in {the student ID you inputted} folder. The machine will take the folder where the image is saved and use

the “student id” as the class name.

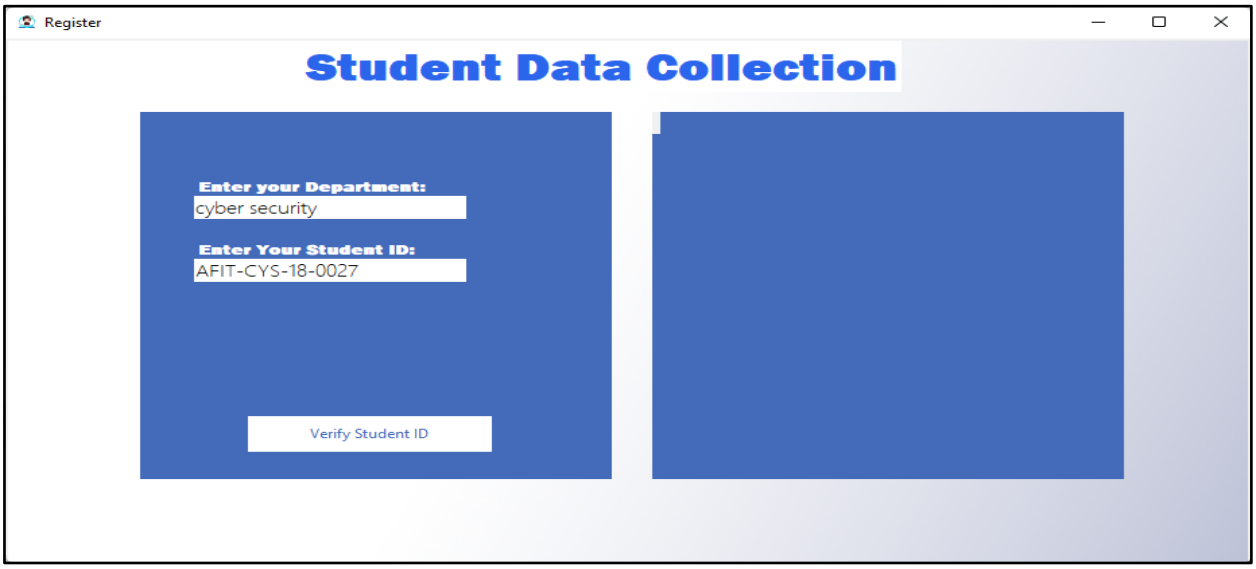

Fig 4.5: GUI Registration Page of the Student

4.3.6 ADMIN PROFILE PAGE FOR SENDING MAILS

For the email page which was designed, as shown in Fig 4.6, the admin could send the attendance to the right person. Without internet access, this program can’t send mail,

because the system has to connect to the mail server before sending the email.

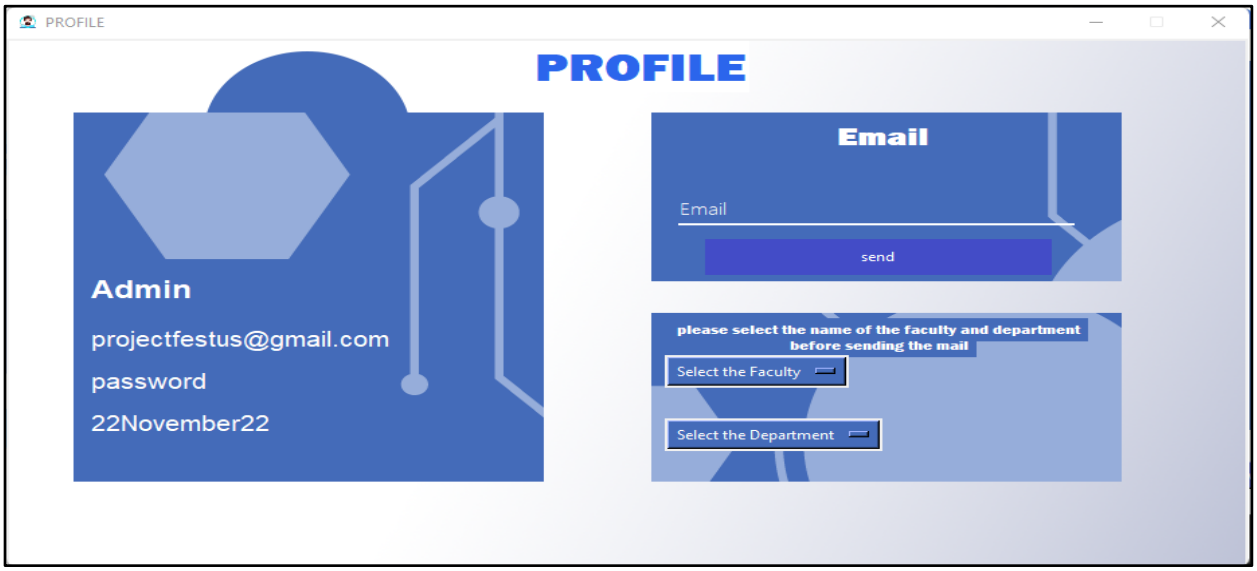

Fig 4.6: The GUI Admin Profile's Page for Sending Mails

4.4 RESULTS

4.4.1 SENDING OF MAIL

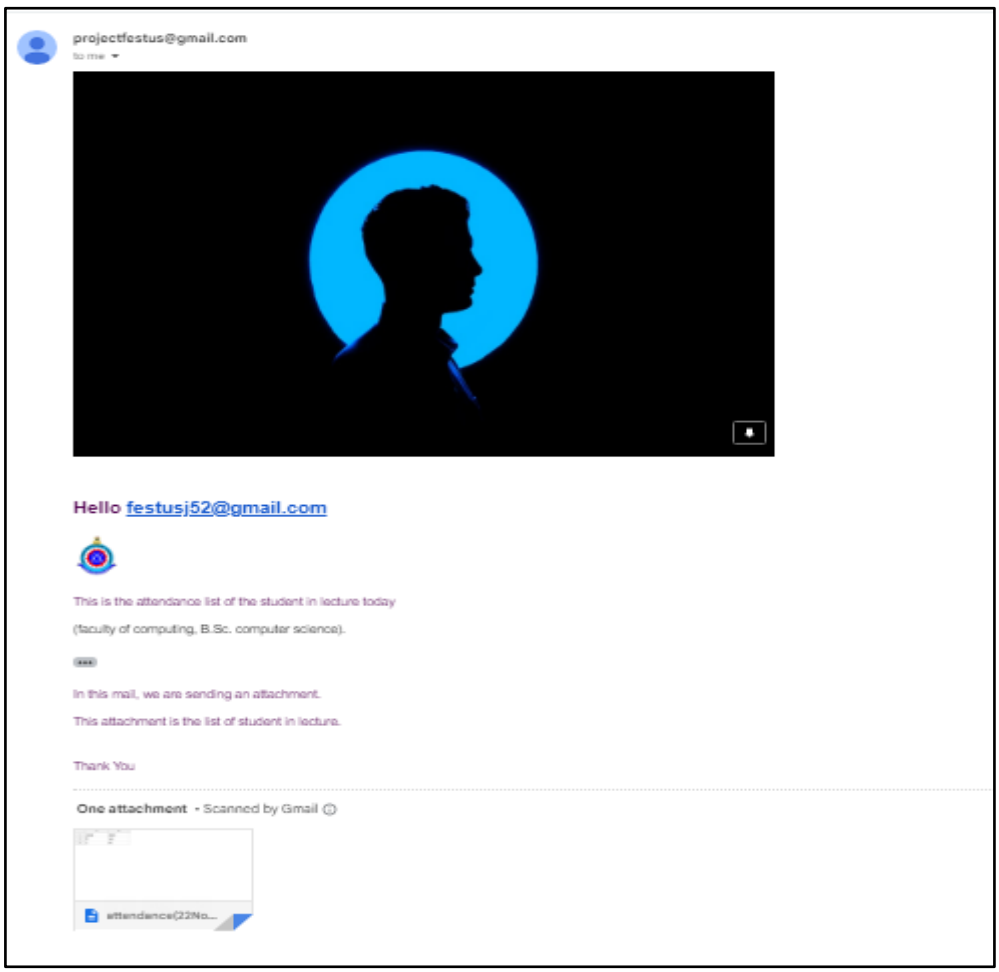

Fig 4.7: Sent Mail

This mail format in figure 4.7 above uses HTML and CSS. This mail has a header, a paragraph with a link carrying the email of the receiver, and a file is sent with the email. This file is the CSV file that carries the attendance list of the students.

4.4.2 THE RECORD OF THE ATTENDANCE LIST

Figure 4.8 illustrates how the attendance record will be saved with the student's ID and the state of the student if they are present or not. This was done by using the Python package called pandas, it converts a Python dictionary into a data frame of pandas and saves it in a CSV file.

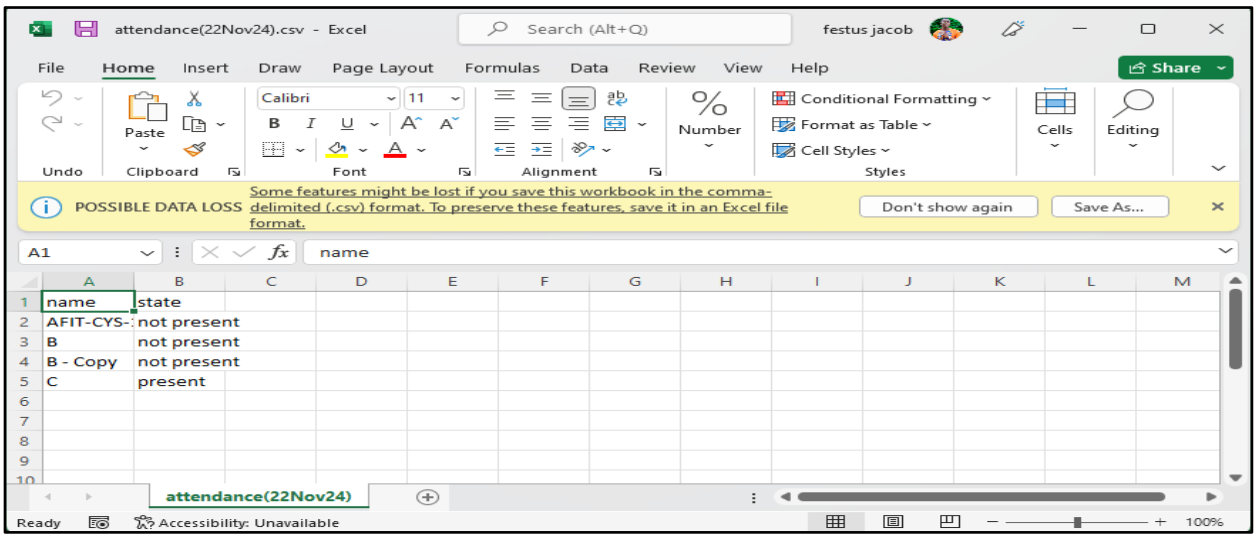

Fig 4.8: The Record of the Attendance List

4.5 DEPLOYMENT

4.5.1 THE SYNTAX FOR REGISTRATION

The code in Figure 4.9 is the syntax for the registration page, and it also shows the code for the graphical user interface (Figures 4.2 and 4.3) of the code. The program used some packages in the code, which is “Tkinter, PIL, OS, and CV2(also known as OpenCV)”. This code makes the program able to use the camera on the system to save files and to give a message when its operation is complete.

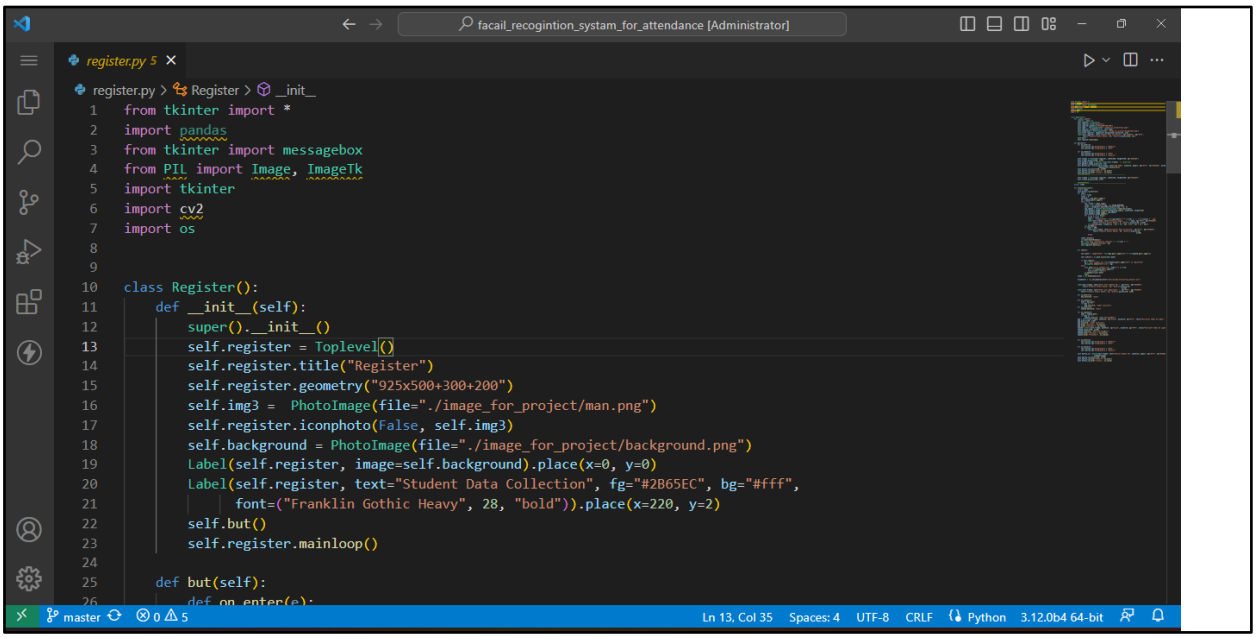

Fig 4.9: The Syntax for Registration

4.5.2 THE CODE FOR THE HOME AND LOGIN PAGE

Figure 4.10 below is the code and syntax for the home page, which graphical user interface was shown in figure 4.3.

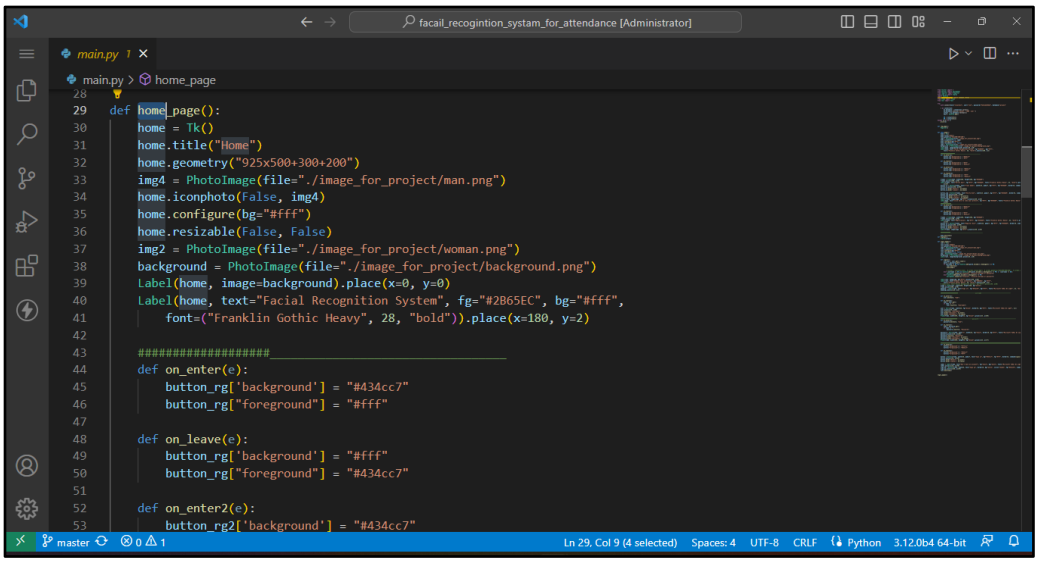

Fig 4.10: The Code for the Home and Login Page

4.5.3 THE CODE TO CONVERT THE LOG TO CSV

The code in Figure 4.11 below is used to convert a data frame into a CSV file and to prevent file name duplication. The program uses date and time to save each CSV attendance log. A record of each attendance file name will be stored in a log_dats.txt as shown in Figure 4.11.

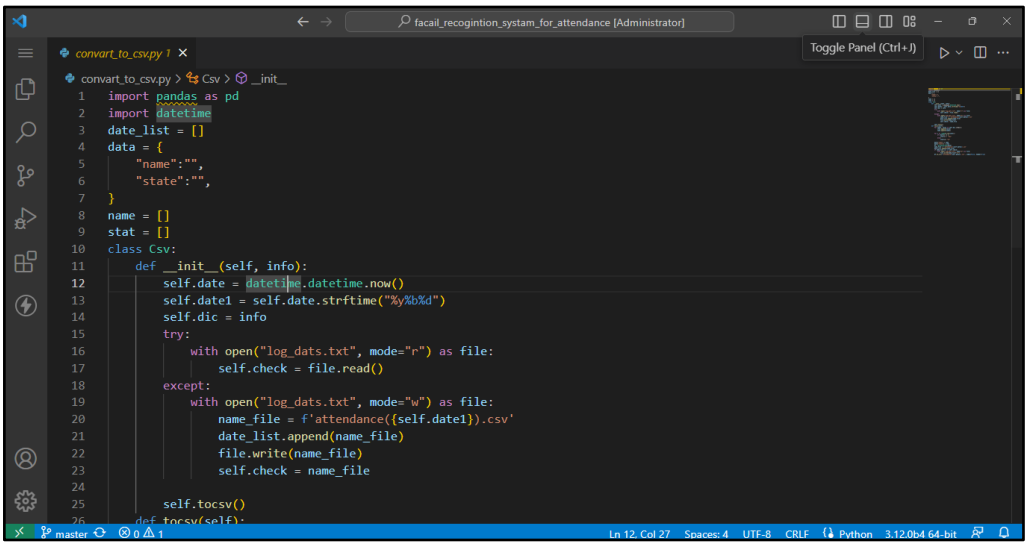

Fig 4:11: The Code to convert the Log to CSV

4.5.4 THE CODE THAT CREATES LAYERS USED TO TRAIN THE MODEL

The code in Figure 4.12 is the deep learning model that shows the layers that are used to train the model. The “relu” in the code means rectified linear unit, also the “softmax” is a

mathematical function that converts a vector of numbers into a vector of probabilities.

```
    tf.config.experimental.set_memory_growth(gpu, True)
model = tf.keras.models.Sequential([ tf.keras.layers.Conv2D(256,(3,3), 1, activation = "relu", input_shape=(256, 256, 3)),
                        tf.keras.layers.MaxPool2D(),

                        tf.keras.layers.Conv2D(128,(3,3), 1, activation = "relu"),
                        tf.keras.layers.MaxPool2D(),

                        tf.keras.layers.Conv2D(64, (3, 3), 1, activation="relu"),
                        tf.keras.layers.MaxPool2D(),

                         tf.keras.layers.Conv2D(32, (3, 3), 1, activation="relu"),
                         tf.keras.layers.MaxPool2D(),
                        tf.keras.layers.Conv2D(16, (3, 3), 1, activation="relu"),
                        tf.keras.layers.MaxPool2D(),

                         tf.keras.layers.Flatten(),
                        tf.keras.layers.Dense(256, activation="relu"),
                        tf.keras.layers.Dense(len(train_dataset.class_indices), activation="softmax")
```

Fig 4.12: Code to create layers for training the model

4.5.5 THE CODE TO SEND THE ATTENDANCE RECORD

The code in Figure 4.13 is used to send the attendance list using SMTP, with a default email address, also the code requests for the receiver's email address, the faculty, and the

department as shown in Figure 4.6 above.

```
class Mail:
    def __init__(self, faculty, department,receiver_mail):
        with open("mail_style.html", mode='r') as file:
            mail_content = file.read()
            mail_content =mail_content.replace("{receiver_mail}", receiver_mail)
            mail_content =mail_content.replace("{faculty}", faculty)
            mail_content =mail_content.replace("{department}", department)
        #The mail addresses and password
        sender_address = '[redacted]'
        sender_pass = '[redacted]'
        receiver_address = receiver_mail
        #Setup the MIME
        message = MIMEMultipart()
        message['From'] = sender_address
        message['To'] = receiver_address
        message['Subject'] = 'Attendance list'
```

Fig 4.13: The Code to send the Attendance Record

4.6 MANAGEMENT

4.6.1 A CLOUD STORAGE ACCOUNT TO SAVE THE CODE

Below are the files which are stored in a cloud environment to prevent loss of data and violation of integrity. In the management phase, the program must keep functioning

without error.

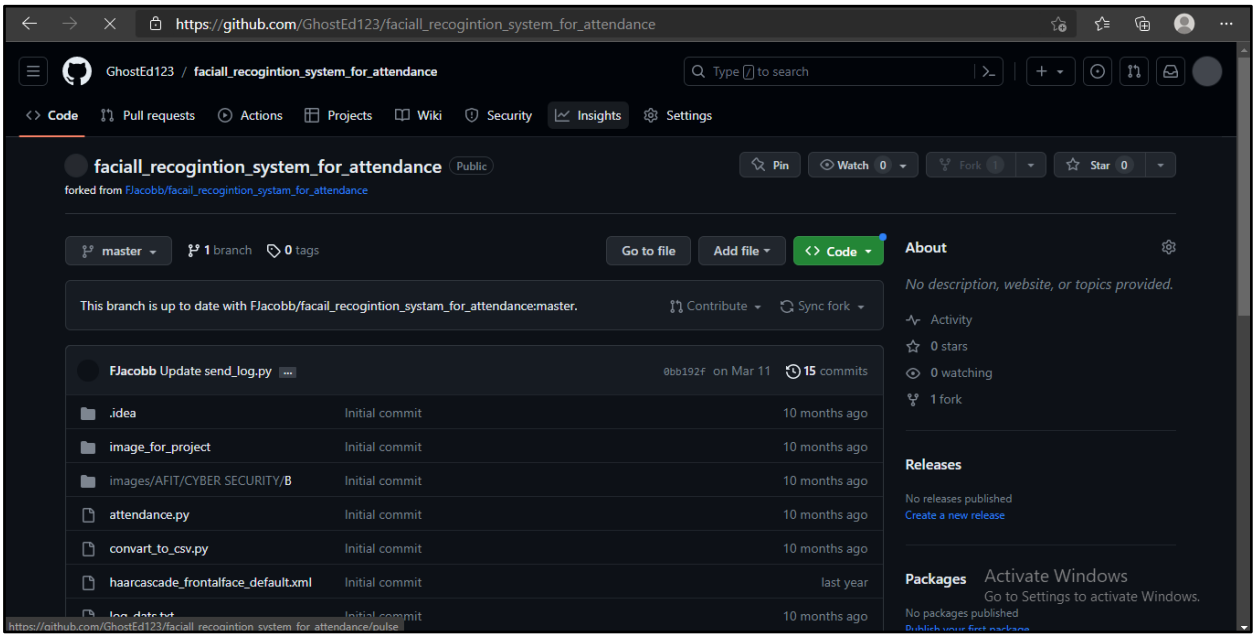

Fig 4.14:- A Cloud Storage Account To Save The Code

(https://github.com/GhostEd123/faciall_recogintion_system_for_attendance.git)

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATION

5.1 SUMMARY

This project focused on the development and implementation of a smart-based student attendance system using facial recognition techniques. Using modern technologies, this system intended to revolutionize existing attendance tracking systems in educational institutions. Attendance is usually taken using paper sheets and the old file system. This method which has been in use for a long time is tedious, inaccurate most of the time and inefficient.

In a nutshell, this project work expressed the aim and objectives of the study, its limitations and its significance. Different reviews of literature on attendance systems - as iris-based attendance system, fingerprint-based attendance systems - were discussed too. The proposed system was analysed in order to describe it and how it should work. Lastly, the system requirements, implementation and documentation were covered.

In summary, the Smart-based student attendance system using facial recognition techniques offers numerous benefits including streamlined attendance management and enhanced security measures. By modernizing attendance tracking processes, this system contributes to creating a more efficient and secure learning environment for both students and educators

5.2 CONCLUSION

In conclusion, the development of a smart-based student attendance system that employs

facial recognition algorithms is a significant achievement in educational technology. This system, which incorporates facial recognition technology, provides a seamless and efficient means for monitoring student attendance while decreasing educators' administrative burdens. By utilizing facial recognition, the system eliminates the need for manual attendance monitoring methods such as paper-based systems or barcode scanning, which are error-prone and time intensive. Instead, it offers a simple and accurate method of verifying student identity, guaranteeing that attendance records are valid and up to date. Furthermore, the use of this technology provides a safer learning environment by limiting access to illegal users. With facial recognition verification, only enrolled students can obtain access to classrooms or campus facilities, lowering the likelihood of security breaches, and improving safety standards.

5.3 RECOMMENDATION

When developing a deep learning model, the study recommends that you use a system with a higher GPU. The facial recognition algorithm should be updated regularly based on feedback and emerging technologies. Also ensure the authorized personnel knows how to use the system.

REFERENCES

- Abdalkarim B.A. and Devrim A. (2022). A literature review on smart attendance systems. 3rd International Conference on Applied Engineering and Natural Sciences.
- Abu S.S., Molla R.H., Rumana T., Ahmed F., Ahsan S.K., Anowar H.M., Mijanur R. (2023). Computer Vision Based Automated Attendance System using Face Recognition. Proceedings of the 6th Industrial Engineering and Operations Management Bangladesh Conference, Dhaka.
- Agrawal A., Ambad R., Lahoti R., Muley P., Pande P. (2022). Role of Artificial Intelligence in PCOS detection. Journal of Data Meghe Institute of Medical Sciences University,17(3),49-494.

- Alqqmeee M.S. (2022). OpenCV Face Detection and Recognition Attendance System. In 2022 3rd International Conference on Inventive Communication and Computational Technologies (ICICCCT) (pp 1-6). IEEE.
- Basheer M.K.P and Raghu C.V. (2012). Fingerprint Attendance System for Classroom Needs. In Proc. India Conference (INDICON) Annual IEEE, pp 433-438.
- Baver C. (2013). The (in-) accuracy of GPS measures of smartphones: A study of running tracking applications. Proceedings of International Conference on Advances in Mobile Computing & Multimedia. pp 335.
- Chandavarkar A. (2021). Securing the Login Interface: A Review of Current Practices and Recommendations. Computers & Security.
- Chatzis S. (2021). Biometric Attendance Systems: A Review of the State-of-the-art. Computers & Security, 104, 101916.
- Choudary S., Kakaji A., Pranay K. and Prabhu P. (2018). Efficient Attendance Management System based on Facial Recognition. *International Journal of Engineering and Technology*, 7,565.
- Duggempudi S.V.S. (2021). Implementation of a Face Recognition Attendance System using OpenCV. In 2021 5th International Conference on Inventive Communication and Computational Technologies (ICICCCT) pp1-5.
- Hoo S. and Ibrahim H. (2019). Biometric-based Attendance Tracking System for Education Sectors: A Literature Survey on Hardware Requirements. Journal of Sensors, 1-25.
- Jindal V., Singh V. (2020). Security Issues in Facial Recognition Attendance System. International Journal of Computer Applications (IJCA).

- Khushbu G. and Aakansha S.C. (2020). A Review on Face Detection based Attendance System with Temperature Monitoring. International Research Journal of Engineering and Technology. Vol 07 Issue 12.
- Kumbhar P.Y., Mohammad A., Shubham D., Shivkumar H. (2017). Real Time Face Detection and Tracking using OpenCV.
- Mijanur R., Sifat N.R., Mahbubur R. and Firoz H. (2016. Biometric Student Registration and Verification System. *Discovery* 52: 2399-2407.
- Rao S. and Satoa K. (2013). An Attendance Monitoring System using Biometrics Authentication. International Journal of Advanced Research in Computer Science and Software Engineering. p3.
- Ruud M.B., Jonathan H.C., Sharath P., Nalini K.R., Andrew W.S. (2003). Guide to Biometrics.
- Seifedine K. and Mohammad S. (2010). Wireless Attendance Management System based on Iris Recognition. Scientific Research and Essays. Vol 5(12) pp1428-1435.
- Senthamil S.K., Chitrakala P., Jenitha A. (2024). Face Recognition Based Attendance Marking System. International Journal of Computer Science and Mobile Computing. Vol 3 Issue 2, pp337-342.
- Sharanya T., Katsuri U., Sucharith P., Trisheeka M., Dhivya V. (2020). Online Attendance using Facial Recognition. International Journal of Engineering Research and Technology. Vol 9 Issue 06.
- Soewito B., Gaol F.L., Simanjuntak B., Gunawan F.E. (2015). Attendance System on Android Smartphone. International Conference on Control Electronics, Renewable Energy and Communications.pp208-211.
- Unnati A.P. and Dr. Swaminarayan P.R. (2014). Development of a Student Attendance Management System using RFID and Face Recognition: a review.

International Journal of Advance Research in Computer Science and Management Studies.

- Vishal B., Tapodhan S., Ankit G., Vijay G. (2013). Bluetooth-based Attendance Management System. International Journal of Innovations in Engineering and Technology. Vol 3 Issue 1.
- Wikipedia (2024); Definition of attendance.
- Zainal N.I., Sidek K.A., Gundawan T.S. (2016). Portable Anti-forgery Recognition for Attendance System using Fingerprint based Biometrics. Journal of Engineering and Applied Sciences.pp396-403.
- Zainal N.I., Sidek K.A., Gundawan T.S., Manser H., Kartiwi M. (2014). Design and Development of Portable Classroom Attendance System based on Arduino and Fingerprint Biometric. Information and Communication Technology for the Muslim World (ICT419) The 5th International Conference on IEEE.pp1-4.
- Zhang Z. and Wallace B. (2019); A survey in deep learning for natural language processing.
- Zhu C.W. (2022). Fundamentals of Artificial Intelligence and Machine Learning. Springer Nature Singapore Pte Ltd.

APPENDIX

The model source code

main.py code

```
from tkinter import *
from tkinter import messagebox
from register import Register
from sign_up import Signup
import hashlib
from mysql.connector import connect, Error
from train import Train
from test import Test
from profile import Profile
try:
    with connect(host="localhost", user="root", password="Fe$tu$245618", database="project"
    ) as connection:
        my_database = connection.cursor()
        my_database.execute("SELECT * FROM `user`")
        output = my_database.fetchall()
        hash = hashlib.md5()
        UR = output[0][1]
        PW = output[0][2]
except Error as e:
    print(e)
def reg_page():
    Register()
def profile_page():
    Profile()
def home_page():
    home = Tk()
    home.title("Home")
    home.geometry("925x500+300+200")
    img4 = PhotoImage(file="./image_for_project/man.png")
    home.iconphoto(False, img4)
    home.configure(bg="#fff")
    home.resizable(False, False)
    img2 = PhotoImage(file="./image_for_project/woman.png")
    background = PhotoImage(file="./image_for_project/background.png")
    Label(home, image=background).place(x=0, y=0)
    Label(home, text="Facial Recognition System", fg="#2B65EC", bg="#fff",
        font=("Franklin Gothic Heavy", 28, "bold")).place(x=180, y=2)
    ###################_____________________________
    def on_enter(e):
        button_rg['background'] = "#434cc7"
        button_rg["foreground"] = "#fff"
    def on_leave(e):
        button_rg['background'] = "#fff"
        button_rg["foreground"] = "#434cc7"
    def on_enter2(e):
        button_rg2['background'] = "#434cc7"
        button_rg2["foreground"] = "#fff"
    def on_leave2(e):
        button_rg2['background'] = "#fff"
        button_rg2["foreground"] = "#434cc7"
    def on_enter3(e):
        profile_link['background'] = "#434cc7"
        profile_link["foreground"] = "#fff"
    def on_leave3(e):
```

```
        profile_link['background'] = "#fff"
        profile_link["foreground"] = "#434cc7"
    frame1 = Frame(home, width=350, height=350, bg="#446bb9")
    frame1.place(x=480, y=70)
    Label(frame1, text="Verify User", fg="#fff", bg="#446bb9", font=("Franklin Gothic Heavy", 23,
"bold")).place(x=80,
        y=110)
    button_rg = Button(frame1, text="Train model", width=15, pady=7, bg="#fff", fg="#446bb9",
border=0, command=Train)
    button_rg.place(x=50, y=290)
    button_rg.bind("<Enter>", on_enter)
    button_rg.bind("<Leave>", on_leave)
    button_rg2 = Button(frame1, text="Verify User", width=15, pady=7, bg="#fff", fg="#446bb9",
border=0, command=Test)
    button_rg2.place(x=185, y=290)
    button_rg2.bind("<Enter>", on_enter2)
    button_rg2.bind("<Leave>", on_leave2)
    Label(frame1, image=img4, bg="#fff").place(x=130, y=10)
    profile_link = Button(home, text="click here to view your profile", fg="#fff", bg="#446bb9",
command=profile_page )
    profile_link.place(x=350,y=450)
    profile_link.bind("<Enter>", on_leave3)
    profile_link.bind("<Leave>", on_enter3)
    ##############_____________________________________
    def on_enter(e):
        button_lg['background'] = "#434cc7"
        button_lg["foreground"] = "#fff"
    def on_leave(e):
        button_lg['background'] = "#fff"
        button_lg["foreground"] = "#434cc7"
    frame2 = Frame(home, width=350, height=350, bg="#446bb9")
    frame2.place(x=100, y=70)
    Label(frame2, text="Registration", fg="#fff", bg="#446bb9", font=("Franklin Gothic Heavy", 23,
"bold")).place(x=80,
        y=110)
    button_lg = Button(frame2, text="Register User", width=25, pady=7, bg="#fff", fg="#446bb9",
border=0, command=reg_page)
    button_lg.place(x=80, y=290)
    button_lg.bind("<Enter>", on_enter)
    button_lg.bind("<Leave>", on_leave)
    Label(frame2, image=img2, bg="#fff").place(x=130, y=10)
    ##############_____________________________________
    home.mainloop()
def Signups():
    Signup(output)
def login_pages():
    root = Tk()
    root.title("LOGIN")
    root.geometry("925x500+300+200")
    img4 = PhotoImage(file="./image_for_project/man.png")
    root.configure(bg="#fff")
    root.iconphoto(False, img4)
    root.resizable(False, False)
    img = PhotoImage(file="./image_for_project/Asset 111.png")
    background = PhotoImage(file="./image_for_project/background.png")
    Label(root, image=background).place(x=0, y=0)
    def signin():
```

```
        username = user.get().lower()
        pword = password.get()
        if username == UR and hashlib.md5(pword.encode()).hexdigest() == PW:
            root.destroy()
            home_page()
            # screen = Toplevel(prog) # screen.title("App") # screen.geometry("925x500+300+200") #
screen.config(bg="white") # # Label(screen, text="Hello Everyone", bg="#fff", font=("calibri(Body)",
50, "bold")).pack(expand=True) # screen.mainloop()
        elif (username != UR and hashlib.md5(pword.encode()).hexdigest() != PW) or (username !=
UR):
            messagebox.showerror("Invalid", "invalid username or password")
        elif hashlib.md5(pword.encode()).hexdigest() != PW:
            messagebox.showerror("Invalid", "invalid username or password")
    Label(root, image=img, bg="#fff").place(x=50, y=50)
    Label(root, text="Student Data Collection", fg="#2B65EC", bg="#fff",
        font=("Franklin Gothic Heavy", 28, "bold")).place(x=220, y=2)
    # Label(prog, text="Facial Recognition System", bg="#fff").place(x=50, y=60)
    frame = Frame(root, width=350, height=350, bg="white")
    frame.place(x=480, y=70)
    heading = Label(frame, text="Sign in", fg="#57a1f8", bg="#fff", font=("Microsoft YaHei UI Light",
23, "bold"))
    heading.place(x=110, y=5)
    ########################----------------------username-----------------------------------
    def on_enter(e):
        user.delete(0, "end")
    def on_leave(e):
        name = user.get()
        if name == "":
            user.insert(0, "Username")
    user = Entry(frame, width=25, fg="black", border=0, bg="#fff", font=("Microsoft YaHei UI Light",
11))
    user.place(x=30, y=80)
    user.insert(0, "Username")
    user.bind("<FocusIn>", on_enter)
    user.bind("<FocusOut>", on_leave)
    Frame(frame, width=295, height=2, bg="black").place(x=25, y=105)
    ########################------------------password---------------------------------------------
    def on_enter(e):
        password.delete(0, "end")
    def on_leave(e):
        name = password.get()
        if name == "":
            password.insert(0, "Password")
    password = Entry(frame, show="*", width=25, fg="black", border=0, bg="#fff", font=("Microsoft
YaHei UI Light", 11))
    password.place(x=30, y=150)
    password.insert(0, "Password")
    password.bind("<FocusIn>", on_enter)
    password.bind("<FocusOut>", on_leave)
    Frame(frame, width=295, height=2, bg="black").place(x=25, y=177)

##########################################################################
#####################################
    def on_enter(e):
        button['background'] = "#fffccc"
        button["foreground"] = "#434cc7"
    def on_leave(e):
```

```
        button['background'] = "#434cc7"
        button["foreground"] = "#fff"
    button = Button(frame, width=36, pady=7, text="Sign in", bg="#434cc7", fg="#fff", border=0,
command=signin)
    button.place(x=50, y=210)
    button.bind("<Enter>", on_enter)
    button.bind("<Leave>", on_leave)
    label = Label(frame, text="Don't have an account?", fg="black", bg="white", font=("Microsoft
YaHei UI Light", 9))
    label.place(x=110, y=270)
    sign_up = Button(frame, width=6, text="Sign up", border=0, bg="white", cursor="hand2",
fg="#57a1f8", command=Signups)
    sign_up.place(x=250, y=270)
    root.mainloop()
login_pages()
register.py
from tkinter import *
from tkinter import messagebox
from PIL import Image, ImageTk
import tkinter
import cv2
import os
class Register():
    def __init__(self):
        super().__init__()
        self.register = Toplevel()
        self.register.title("Register")
        self.register.geometry("925x500+300+200")
        self.img3 = PhotoImage(file="./image_for_project/man.png")
        self.register.iconphoto(False, self.img3)
        self.background = PhotoImage(file="./image_for_project/background.png")
        Label(self.register, image=self.background).place(x=0, y=0)
        Label(self.register, text="Student Data Collection", fg="#2B65EC", bg="#fff",
              font=("Franklin Gothic Heavy", 28, "bold")).place(x=220, y=2)
        self.but()
        self.register.mainloop()
    def but(self):
        def on_enter(e):
            self.button_rg['background'] = "#434cc7"
            self.button_rg["foreground"] = "#fff"
        def on_leave(e):
            self.button_rg['background'] = "#fff"
            self.button_rg["foreground"] = "#434cc7"
        self.frame1 = Frame(self.register, width=350, height=350, bg="#446bb9")
        self.frame1.place(x=480, y=70)
        self.paneeli_image = tkinter.Label(self.frame1) # ,image=img)
        self.paneeli_image.place(x=0, y=0)
        self.button_rg = Button(self.frame1, text="ADD USER", width=25, pady=7, bg="#fff",
fg="#446bb9", border=0,
                                command=self.facecollect)
        self.button_rg.place(x=80, y=290)
        self.button_rg.bind("<Enter>", on_enter)
        self.button_rg.bind("<Leave>", on_leave)
        ##############_____________________________________
        self.frame2 = Frame(self.register, width=350, height=350, bg="#446bb9")
        self.frame2.place(x=100, y=70)
        ##############_____________________________________
```

```
global video
def facecollect(self):
    global video
    self.button_rg.destroy()
    def cam():
        global frame
        count = 0
        partment = dep.get().upper()
        id = nameID.get().upper()
        while True:
            ret, frame = video.read()
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            faces = facedetect.detectMultiScale(frame, 2.0, 5)
            img_update = ImageTk.PhotoImage(Image.fromarray(frame))
            self.paneeli_image.configure(image=img_update, width=350, height=350)
            self.paneeli_image.image = img_update
            self.paneeli_image.update()
            for x, y, w, h in faces:
                count = count + 1
                name = './images/AFIT/' +str(partment)+"/"+ str(id) + '/' + str(count) + '.jpg'
                Label(self.frame2, text="Creating Images....."+ name, fg="#fff", bg="#446bb9",
                      font=("Franklin Gothic Heavy", 4, "bold")).place(x=20, y=160)
                cv2.imwrite(name, frame[y:(y - 32) + (h + 20), x:(x - 32) + (w + 20)])
            cv2.waitKey(1)
            if count >500:
                Label(self.frame2, text="Successful Data Collection", fg="#fff", bg="#446bb9",
                      font=("Franklin Gothic Heavy", 10, "bold")).place(x=20,
                                                                        y=180)
                break
        video.release()
        cv2.destroyAllWindows()
        mg = "You have successfully register ("+ str(id) + ") ."
        messagebox.showinfo("Message", mg)
        self.register.destroy()
    def check():
        self.path = 'images/AFIT/' +str(dep.get().upper())+"/"+ str(nameID.get().upper())
        self.isExist = os.path.exists(self.path)
        if self.isExist:
            mg = "This student ID ("+str(nameID.get().upper())+") is registered"
            messagebox.showerror("error", mg)
        else:
            with open("id of student.txt", mode='a') as file:
                fm = str(nameID.get().upper())
                file.write(f"{fm}\n")
            os.makedirs(self.path)
            cam()
    video = cv2.VideoCapture(2)
    facedetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
    Label(self.frame2, text="Enter Your Student ID: ", fg="#fff", bg="#446bb9",
          font=("Franklin Gothic Heavy", 10, "bold")).place(x=40,
                                                            y=120)
    Label(self.frame2, text="Enter your Department: ", fg="#fff", bg="#446bb9",
       font=("Franklin Gothic Heavy", 10, "bold")).place(x=40, y=60)
    def on_enter1(e):
        dep.delete(0, "end")
    def on_leave1(e):
        name = dep.get()
```

```
            if name == "":
                dep.insert(0, "cyber security")
        def on_enter2(e):
            nameID.delete(0, "end")
        def on_leave2(e):
            name = nameID.get()
            if name == "":
                nameID.insert(0, "AFIT-CYS-18-0027")
        dep = Entry(self.frame2, width=25, fg="black", border=0, bg="#fff", font=("Microsoft YaHei UI
Light", 11))
        dep.place(x=40, y=80)
        dep.insert(0, "cyber security")
        dep.bind("<FocusIn>", on_enter1)
        dep.bind("<FocusOut>", on_leave1)
        nameID = Entry(self.frame2, width=25, fg="black", border=0, bg="#fff", font=("Microsoft YaHei
UI Light", 11))
        nameID.place(x=40, y=140)
        nameID.insert(0, "AFIT-CYS-18-0027")
        nameID.bind("<FocusIn>", on_enter2)
        nameID.bind("<FocusOut>", on_leave2)
        def on_enter(e):
            self.button_rg['background'] = "#434cc7"
            self.button_rg["foreground"] = "#fff"
        def on_leave(e):
            self.button_rg['background'] = "#fff"
            self.button_rg["foreground"] = "#434cc7"
        self.button_rg = Button(self.frame2, text="Verify Student ID", width=25, pady=7, bg="#fff",
fg="#446bb9", border=0,
                                command=check)
        self.button_rg.place(x=80, y=290)
        self.button_rg.bind("<Enter>", on_enter)
        self.button_rg.bind("<Leave>", on_leave)
sign up.py
from tkinter import *
from tkinter import messagebox
from mysql.connector import connect, Error, cursor
class Signup():
    def __init__(self, list_of_output):
        super().__init__()
        self.signup = Toplevel()
        self.signup.title("Sign up")
        self.signup.geometry("925x500+300+200")
        self.img3 = PhotoImage(file="./image_for_project/man.png")
        self.signup.iconphoto(False, self.img3)
        self.background = PhotoImage(file="./image_for_project/background.png")
        self.img = PhotoImage(file="./image_for_project/Asset 111.png")
        Label(self.signup, image=self.background).place(x=0, y=0)
        Label(self.signup, image=self.img, bg="#fff").place(x=50, y=50)
        Label(self.signup, text="Student Data Collection", fg="#2B65EC", bg="#fff",
              font=("Franklin Gothic Heavy", 28, "bold")).place(x=220, y=2)
        self.check_list = list_of_output
        with open("id.txt", mode="r") as file:
            self.id = int(file.read())
        self.but()
        self.signup.mainloop()
    def signin(self):
        self.signup.destroy()
```

```
    def sign_up(self):
        self.nr = self.user.get()
        self.pw = self.password.get()
        self.em = self.email.get()
        userlist = []
        emaillist = []
        for i in self.check_list:
            userlist.append(i[1])
            emaillist.append(i[-1])
        if self.nr in userlist or self.em in emaillist:
            messagebox.showerror("Invalid", "Someone already have this username or Email.")
        elif self.nr != ""and self.pw != ""and self.em != '':
            self.id = self.id + 1
            try:
                with connect(host="localhost", user="prog", password="Fe$tu$245618",
database="project") as connection:
                    my_database = connection.cursor()
                    text = f"INSERT INTO `user`(`id`, `username`, `password`, `email`) VALUES ('{self.id}',
'{self.nr}', MD5('{self.pw}'), '{self.em}')"
                    my_database.execute(text)
                    connection.commit()
                    messagebox.showinfo("Message", "You have successfully created an account.")
                    with open("id.txt", mode="w") as file:
                        file.write(f"{self.id}")
                    self.signin()
            except Error as e:
                print(e)
    def but(self):
        self.frame = Frame(self.signup, width=350, height=350, bg="white")
        self.frame.place(x=480, y=70)
        self.heading = Label(self.frame, text="Sign up", fg="#57a1f8", bg="#fff",
            font=("Microsoft YaHei UI Light", 23, "bold"))
        self.heading.place(x=110, y=5)
        # screen = Toplevel(prog) # screen.title("App") # screen.geometry("925x500+300+200") #
screen.config(bg="white") # # Label(screen, text="Hello Everyone", bg="#fff", font=("calibri(Body)",
50, "bold")).pack(expand=True) # screen.mainloop()
        ########################----------------------username-----------------------------------
        def on_enter(e):
            self.user.delete(0, "end")
        def on_leave(e):
            name = self.user.get()
            if name == "":
                self.user.insert(0, "Username")
        self.user = Entry(self.frame, width=25, fg="black", border=0, bg="#fff", font=("Microsoft YaHei
UI Light", 11))
        self.user.place(x=30, y=80)
        self.user.insert(0, "Username")
        self.user.bind("<FocusIn>", on_enter)
        self.user.bind("<FocusOut>", on_leave)
        Frame(self.frame, width=295, height=2, bg="black").place(x=25, y=105)
        ########################-------------------password-------------------------------------------
        def on_enter(e):
            self.password.delete(0, "end")
        def on_leave(e):
            name = self.password.get()
            if name == "":
                self.password.insert(0, "Password")
```

```
        self.password = Entry(self.frame, show="*", width=25, fg="black", border=0, bg="#fff",
font=("Microsoft YaHei UI Light", 11))
        self.password.place(x=30, y=125)
        self.password.insert(0, "Password")
        self.password.bind("<FocusIn>", on_enter)
        self.password.bind("<FocusOut>", on_leave)
        Frame(self.frame, width=295, height=2, bg="black").place(x=25, y=152)
        ######################----------password----------------------------------------
        def on_enter(e):
            self.email.delete(0, "end")
        def on_leave(e):
            name = self.email.get()
            if name == "":
                self.email.insert(0, "Password")
        self.email = Entry(self.frame, width=25, fg="black", border=0, bg="#fff",
                           font=("Microsoft YaHei UI Light", 11))
        self.email.place(x=30, y=170)
        self.email.insert(0, "Email")
        self.email.bind("<FocusIn>", on_enter)
        self.email.bind("<FocusOut>", on_leave)
        Frame(self.frame, width=295, height=2, bg="black").place(x=25, y=197)

#########################################################################
#####################################
        Button(self.frame, width=36, pady=7, text="Sign up", bg="#57a1f8", fg="#fff", border=0,
command=self.sign_up).place(x=50,
                                                                    y=230)
        label = Label(self.frame, text="Do you have an account?", fg="black", bg="white",
                      font=("Microsoft YaHei UI Light", 9))
        label.place(x=100, y=280)
        sign_up = Button(self.frame, width=6, text="Sign in", border=0, bg="white", cursor="hand2",
fg="#57a1f8",
                         command=self.signin)
        sign_up.place(x=250, y=280)
train.py
from keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
from keras.utils.vis_utils import *
import graphviz, pydot
from tkinter import messagebox
#
class Train():
    def __init__(self):
            student_id =[]
            listd = []
            train = ImageDataGenerator(rescale=1/255)
            validation = ImageDataGenerator(rescale=1/255)
            print(train)
            path = "images/AFIT/CYBER SECURITY/"
            train_dataset = train.flow_from_directory(path, batch_size=3, target_size=(256, 256))
            gpus = tf.config.experimental.list_physical_devices("GPU")
            for gpu in gpus:
                tf.config.experimental.set_memory_growth(gpu, True)
            model = tf.keras.models.Sequential([ tf.keras.layers.Conv2D(256,(3,3), 1, activation = "relu",
input_shape=(256, 256, 3)),
                                tf.keras.layers.MaxPool2D(),
                                tf.keras.layers.Conv2D(128,(3,3), 1, activation = "relu"),
```

```
                    tf.keras.layers.MaxPool2D(),
                    tf.keras.layers.Conv2D(64, (3, 3), 1, activation="relu"),
                    tf.keras.layers.MaxPool2D(),
                     tf.keras.layers.Conv2D(32, (3, 3), 1, activation="relu"),
                     tf.keras.layers.MaxPool2D(),
                    tf.keras.layers.Conv2D(16, (3, 3), 1, activation="relu"),
                    tf.keras.layers.MaxPool2D(),
                     tf.keras.layers.Flatten(),
                    tf.keras.layers.Dense(256, activation="relu"),
                    tf.keras.layers.Dense(len(train_dataset.class_indices), activation="softmax")
        ])
        print(model.summary())
        validation_dataset = validation.flow_from_directory(path, batch_size=3, target_size=(256,
256))
        model.compile(loss="categorical_crossentropy", optimizer= "adam", metrics=["accuracy"])
        model.fit(train_dataset, epochs=2, validation_data=validation_dataset)
        model.save("trained_data.h5")
        messagebox.showinfo("Training result", "The model has completed it training")
convert_to_csv.py
import pandas as pd
import datetime
date_list = []
data = {
   "name":"",
   "state":"",
}
name = []
stat = []
class Csv:
   def __init__(self, info):
      self.date = datetime.datetime.now()
      self.date1 = self.date.strftime("%y%b%d")
      self.dic = info
      try:
         with open("log_dats.txt", mode="r") as file:
            self.check = file.read()
      except:
         with open("log_dats.txt", mode="w") as file:
            name_file = f'attendance({self.date1}).csv'
            date_list.append(name_file)
            file.write(name_file)
            self.check = name_file
      self.tocsv()
   def tocsv(self):
      for name1, value in self.dic.items():
         stat.append(value)
         name.append(name1)
      for i in range(0,len(stat)):
         if stat[i] == 1:
            stat[i] = "Present"
         else:
            stat[i]= "Absent"
      data["name"] = name
      data["state"] = stat
      df = pd.DataFrame(data)
      name_file = f'attendance({self.date1}).csv'
      date_list.append(name_file)
```

```
        if name_file not in self.check:
            with open("log_dats.txt", mode="a") as file:
                file.write(name_file+"\n")
        df.to_csv(f'attendance({self.date1}).csv', index=False, header=True)
test.py
from keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf
from tensorflow import keras
import numpy as np
import cv2
from keras.models import load_model
import numpy as np
from convart_to_csv import Csv
class Test:
def __init__(self):
# from alexa import alexa
train = ImageDataGenerator(rescale=1/255)
train_dataset = train.flow_from_directory(f"C:/Users/festu/OneDrive/Documents/100days of
code/festus_project/images/AFIT/CYBER SECURITY/", batch_size=3, class_mode="binary",
target_size=(256, 256))
mark = {}
for name, value in train_dataset.class_indices.items():
mark[name] = 0
class_of_id = train_dataset.class_indices
facedetect = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
cap=cv2.VideoCapture(2)
font=cv2.FONT_HERSHEY_COMPLEX
model = load_model('C:/Users/festu/OneDrive/Documents/100days of
code/festus_project/trained_data.h5')
def get_className(classNo):
print(classNo)
for name, value in class_of_id.items():
if classNo == value:
if mark[name] == 0:
mark[name] = 1
return name
else:
return name
else:
return "unknown"
play = True
while play:
sucess, imgOrignal=cap.read()
faces = facedetect.detectMultiScale(imgOrignal,1.3,5)
for x,y,w,h in faces:
crop_img=imgOrignal[y:y+h,x:x+h]
img=cv2.resize(crop_img, (256,256))
img=img.reshape(1, 256, 256, 3)
prediction=model.predict(img)
classIndex = np.expand_dims(prediction,axis=1)
probabilityValue=np.amax(prediction)
print(classIndex)
cv2.rectangle(imgOrignal,(x,y),(x+w,y+h),(0,255,0),2)
cv2.rectangle(imgOrignal, (x,y-40),(x+w, y), (0,255,0),-2)
print(classIndex)
cv2.putText(imgOrignal, str(get_className(classIndex)),(x,y-10), font, 0.75, (255,255,255),1,
cv2.LINE_AA)
```

```
cv2.imshow("Secured Facial Recognition Model",imgOrignal)
if cv2.waitKey(1)==ord('q'):
cap.release()
cv2.destroyAllWindows()
play = False
break
Profile.py
import datetime
from tkinter import *
from tkinter import messagebox
from send_log import Mail
class Profile():
    def __init__(self):
        self.prog = Toplevel()
        self.prog.title("PROFILE")
        self.prog.geometry("925x500+300+200")
        self.img4 = PhotoImage(file="./image_for_project/Ellipse 1.png")
        self.img3 = PhotoImage(file="./image_for_project/man.png")
        self.img6 = PhotoImage(file="./image_for_project/Asset 3.png")
        self.background = PhotoImage(file="./image_for_project/background.png")
        self.prog.iconphoto(False, self.img3)
        self.prog.configure(bg="#fff")
        self.prog.resizable(False, False)
        self.login_pages()
        self.prog.mainloop()
    def login_pages(self):
        Label(self.prog, image=self.background).place(x=0, y=0)
        Label(self.prog, text="PROFILE", fg="#2B65EC", bg="#fff",
              font=("Franklin Gothic Heavy", 28, "bold")).place(x=390, y=2)
        # Label(prog, text="Facial Recognition System", bg="#fff").place(x=50, y=60)
        Label(self.prog, image=self.img4, bg="#fff").place(x=145, y=10)
        frame = Frame(self.prog, width=350, height=350, bg="#446bb9")
        frame.place(x=50, y=70)
        heading = Label(frame, text="Sign in", fg="#57a1f8", bg="#fff", font=("Microsoft YaHei UI Light",
23, "bold"))
        frame2 = Frame(self.prog, width=350, height=160, bg="#446bb9")
        frame2.place(x=480, y=70)
        Label(frame, image=self.img6, bg="#446bb9").place(x=-330, y=-490)
        Label(frame2, image=self.img6, bg="#446bb9").place(x=-220, y=-50)
        frame3 = Frame(self.prog, width=350, height=160, bg="#446bb9")
        frame3.place(x=480, y=260)
        Label(frame3, image=self.img6, bg="#446bb9").place(x=-390, y=-150)
        Label(frame3, text="before sending the mail ", fg="#fff", bg="#446bb9", font=("Franklin Gothic
Heavy", 9)).place(
            x=100, y=20)
        Label(frame3, text="please select the name of the faculty and department ", fg="#fff",
bg="#446bb9", font=("Franklin Gothic Heavy", 9)).place(x=16, y=5)

#####################################################################
###################################
        def click_in():
            if faculty.get() != '' and department.get() != ""and user.get() != "":
                selected_faculty = faculty.get()
                selected_department = department.get()
                Mail(selected_faculty, selected_department, user.get())
                messagebox.showinfo("Status","Mail sent")
            else:
```

```
            messagebox.showerror("Invalid", "you most select an option for each box and put in your
email")
        Label(frame2, text="Email", fg="#fff", bg="#446bb9",font=("Franklin Gothic Heavy", 18,
"bold")).place(x=135, y=5)
        def on_enter(e):
            user.delete(0, "end")
        def on_leave(e):
            name = user.get()
            if name == "":
                user.insert(0, "Email")
        def on_enter1(e):
            button_email['background'] = "#fffccc"
            button_email["foreground"] = "#434cc7"
        def on_leave1(e):
            button_email['background'] = "#434cc7"
            button_email["foreground"] = "#fff"
        user = Entry(frame2, width=25, fg="#fff", border=0, bg="#446bb9", font=("Microsoft YaHei UI
Light", 11))
        user.place(x=20, y=80)
        user.insert(0, "Email")
        user.bind("<FocusIn>", on_enter)
        user.bind("<FocusOut>", on_leave)
        Frame(frame2, width=295, height=2, bg="#fff").place(x=20, y=105)
        button_email = Button(frame2, text="send", width=36, pady=7, bg="#446bb9", fg="#fff",
border=0, command=click_in)
        button_email.place(x=40,y=120)
        button_email.bind("<Enter>", on_enter1)
        button_email.bind("<Leave>", on_leave1)

########################################################################
###################################
        options_faculty = ["faculty of computing", "faculty of air engineering", "faculty of ground and
communication"
                                                    "engineering", "faculty of sciences", "faculty of social
management sciences"]
        faculty = StringVar()
        faculty.set("Select the Faculty")
        drop = OptionMenu(frame3, faculty, *options_faculty,)
        drop.config(fg="#fff", bg="#446bb9")
        drop.place(x=10, y=40)
        #-------------------------------------------------------------------------------------------------------
        options_department = ['B.Eng. Aerospace Engineering','B.Eng. Mechanical Engineering','B.Eng.
Electrical and Electronics Engineering',
    'B.Eng. Information &Communication Technology',
    'B.Eng. Automotive Engineering',
    'B.Eng. Civil Engineering',
    'B.Sc. Accounting',
    'B.Sc. Business Administration',
    'B.Sc. Economics',
    'B.Sc. Marketing',
    'B.Sc. Cyber Security',
    'B.Eng mechatronics engineering',
    'B.Eng metallurgical and material engineering',
    'B.Eng telecommunication engineering',
    'B.Sc. computer science',
    'B.Sc. chemistry',
    'B.Sc. mathematics',
```

```
    'B.Sc. physics',
    'B.Sc. physics with electronics',
    'B.Sc. marketing',
    'B.Sc. statistics',
    'B.Sc. international relation',
    'B.Sc. banking and finance']
        department = StringVar()
        department.set("Select the Department")
        drop = OptionMenu(frame3, department, *options_department)
        drop.config(fg="#fff", bg="#446bb9")
        drop.place(x=10, y=100)

#########################################################################
###################################
        label1 = Label(frame, text="Admin", fg="#fff", bg="#446bb9", font=("Arial Rounded MT", 18,
"bold"))
        label1.place(x=10,y=150)
        label2 = Label(frame, text="projectfestus@gmail.com", fg="#fff", bg="#446bb9", font=("Arial
Rounded MT", 15))
        label2.place(x=10, y=200)
        label3 = Label(frame, text="password", fg="#fff", bg="#446bb9", font=("Arial Rounded MT", 15))
        label3.place(x=10, y=240)
        label4 = Label(frame, text=f"{datetime.datetime.now().strftime('%y%B%d')}", fg="#fff",
bg="#446bb9", font=("Arial Rounded MT", 15))
        label4.place(x=10, y=280)

#########################################################################
###################################
```