

Automated Attendance Management System



By:

Umair Ali
19-CE-008

Muhammad Bilal Ahmed
19-CE-020

Muhammad Faisal
19-CE-038

Thesis Report

Department of Computer Engineering

HITEC University, Taxila

Fall 2019

Declaration

We, hereby declare that this project neither as a whole nor as a part there of has been copied out from any source. It is further declared that we have developed this project and the accompanied report entirely on the basis of our personal efforts made under the sincere guidance of our supervisor. No portion of the work presented in this report has been submitted in the support of any other degree or qualification of this or any other University or Institute of learning. If any violation is found in this FYP report, we will be liable to be punished under the plagiarism rules of HEC.

Signature:_____

Name: Umair Ali

Signature:_____

Name: Muhammad Bilal Ahmed

Signature:_____

Name: Muhammad Faisal

HITEC University, Taxila

Dedication

To our honorable supervisor, Assistant Professor Dr. Imran Ashraf, whose direction, wisdom, and constant support have been crucial throughout our trip, I would like to dedicate this endeavor. His extensive expertise, priceless ideas, and unwavering support have molded the project and motivated us to go beyond the scope of our talents.

Our determination has originated from Dr. Imran Ashraf's commitment to excellence and his enthusiasm for research and innovation. His unfailing faith in our competence and his readiness to go above and above to help us overcome obstacles have been simply amazing. Under his guidance, we have matured not just as scholars but also as people, coming to really value thoroughness, critical thought, and attention to detail.

Our sincere thanks also goes out to our co-supervisor, Lab Engineer Ali Raza, whose technical expertise, forbearance, and constant support have been crucial to the accomplishment of this project. His hands-on support, troubleshooting abilities, and practical insights have been important in overcoming technological challenges and guaranteeing the system's flawless operation.

This project is dedicated to everyone who contributed in any manner, large or small, to making it possible. The success of this project wouldn't have been achieved without your contributions, guidance, and ongoing encouragement, and for these, I am sincerely thankful to all of you.

Acknowledgement

We appreciate Dr. Imran Ashraf, our mentor, for his significant advice, help, and encouragement during our project completion. Without his tremendous expertise, skills, and dedication, we would not have been able to successfully carry out our project. His insightful comments and helpful critique have considerably improved this thesis. His unrelenting dedication and mentorship are greatly appreciated.

We also want to thank our co-supervisor, Engr. Ali Raza, for his valuable contribution to this research. His technical expertise, valuable insights, and practical guidance have played a pivotal role in the successful implementation of the facial recognition attendance system. His continuous support and motivation have been crucial in overcoming challenges and ensuring the smooth progress of this project. We are sincerely grateful for his unwavering support and valuable suggestions.

All people and organizations that have helped this research project be completed successfully deserve our sincere gratitude. This project would not become possible without their support.

Abstract

The Facial Recognition Attendance System is a project aimed at automating attendance management using facial recognition technology. The system utilizes Convolutional Neural Networks (CNNs) to analyze and recognize faces, and it integrates a solenoid lock to control access based on the authorization of individuals.

The system is implemented using a Raspberry Pi 4, which serves as the main processing unit and interface for the components. The CNN model is trained to identify and differentiate between authorized and unauthorized individuals by analyzing facial features and patterns. Facial data, such as images or video streams, is captured using a camera connected to the Raspberry Pi.

Upon capturing an image, the CNN performs facial recognition and matches it against a pre-existing database of authorized individuals. If a match is found, the system records the attendance details, including the person's name and the timestamp, in a CSV (Comma-Separated Values) file. Additionally, a signal is sent to the solenoid lock, triggering it to open temporarily, allowing access for the authorized individual.

The solenoid lock, controlled by a relay module, serves as a physical barrier to restrict entry. When the signal from the Raspberry Pi is received, the solenoid lock unlocks, enabling entry for the authorized person. Conversely, if an unauthorized person attempts to gain access, the facial recognition process will not result in a match, and the solenoid lock will remain closed.

By combining facial recognition technology with the solenoid lock, the system provides a secure and efficient method for attendance management. It eliminates the need for manual record-keeping, reduces administrative efforts, and ensures that only authorized individuals can access the designated area.

The project showcases the potential of using advanced technologies such as CNNs, Raspberry Pi, and solenoid locks to create a reliable and automated attendance management system. It has the potential for application in various domains, including educational institutions, workplaces, and other organizations requiring accurate attendance tracking and access control.

TABLE OF CONTENTS

CHAPTER 1	10
Introduction	10
1.1 Biometric Systems.....	10
1.2 Importance of Face Recognition	10
1.3 Facial Recognition System.....	10
1.4 Background	12
1.5 Problem Statement	13
1.6 Objectives.....	13
1.6.1 Automation:	13
1.6.2 Accuracy:	14
1.6.3 Real-time Tracking:	14
1.6.4 Access Control:	14
1.6.5 Efficiency:.....	14
1.6.6 Scalability:	14
CHAPTER 2	15
LITERATURE REVIEW	15
2.1 Study of Research Papers.....	15
2.2 Existing System.....	18
2.2.1 (LBHP) Local Binary Histogram Pattern	18
2.2.2 Fischer Face Recognition.....	19
2.2.3 Eigen Face Recognition	20
2.3 Proposed System	20
CHAPTER 3	22
METHODOLOGY	22
3.1.1 Data Collection:	22

3.1.2	Dataset Preprocessing:	22
3.1.3	CNN Model Training:	23
3.1.4	Facial Recognition:	23
3.1.5	Solenoid Lock Integration:	23
3.1.6	System Integration and Testing:	24
3.1.7	Deployment and Evaluation:	24
3.2	Algorithm:	24
3.3	Flowchart:	26
CHAPTER 4		27
IMPLEMENTATION		27
4.1	Hardware Setup:	27
4.2	Software Installation:	27
4.3	Dataset Collection:	27
4.4	Dataset Preprocessing:	27
4.5	CNN Model Training:	27
4.6	Real-time Facial Recognition:	28
4.7	Attendance Logging and Access Control:	28
4.8	Continuous Operation:	28
4.9	Testing and Evaluation:	28
4.10	Refinement and Optimization:	28
4.11	Hardware Tools:	28
4.12	Software Tools:	29
4.13	Training Phase:	29
4.14	Testing Phase:	30
4.15	Diagram of Use Case:	31
CHAPTER 5		32
SIMULATION AND RESULTS		32

5.1	Simulation:	32
5.1.1	Dataset Generation:.....	32
5.1.2	Preprocessing:	32
5.1.3	Model Training:	32
5.1.4	Real-time Simulation:	32
5.2	Facial Recognition and Attendance Logging:	32
5.3	Access Control Simulation:.....	33
5.4	Logging and Visualization:	33
5.5	Performance Evaluation:	33
5.6	Results:	34
5.6.1	Software Results:	35
5.6.2	Hardware Results:	36
CHAPTER 6	38
CONCLUSION AND FUTURE WORK	38
6.1	Conclusion.....	38
6.2	Proposed Work:.....	38
6.3	Future Improvements:	39
References	40

LIST OF FIGURES

Figure 1-0-1 Block Diagram	12
Figure 2-0-1 Local Binary Patterns with face	19
Figure 2-0-2 Fischer Faces	19
Figure 2-0-3 Eigen Faces Diagram	20
Figure 3-0-1 Methodology Diagram	22
Figure 4-0-1 Training Phase	29
Figure 4-0-2 Testing Phase	30
Figure 5-1 System Overview	34
Figure 5-2 Accuracy loss Graph	34
Figure 5-3 Software Results	35
Figure 5-4 Hardware Results 1	36
Figure 5-5 Hardware Results 2	36
Figure 5-6 Lock open for authorized person	37
Figure 5-7 Lock not open for unauthorized person	37

LIST OF ABBREVIATIONS

CNN - Convolutional Neural Network

CSV - Comma-Separated Values

GPIO - General-Purpose Input/Output

GUI - Graphical User Interface

LED - Light-Emitting Diode

RAM - Random Access Memory

CPU - Central Processing Unit

GPU - Graphics Processing Unit

USB - Universal Serial Bus

HDMI - High-Definition Multimedia Interface

OS - Operating System

API - Application Programming Interface

FPS - Frames Per Second

LAN - Local Area Network

WLAN - Wireless Local Area Network

RFID - Radio-Frequency Identification

IoT - Internet of Things

IR - Infrared

RGB - Red Green Blue

SSD - Solid-State Drive

CHAPTER 1

INTRODUCTION

1.1 Biometric Systems

Biometrics are part of the cutting-edge technology. The measures that relate to human characteristics are called biometrics. As a developing technology, biometric systems may be very useful by replacing passwords, assisting in capturing criminals, and even posting an employee's attendance in an organization. Several types of biometric types are available like Face, iris, fingerprint, voice, hand geometry, and behavioral characteristics recognition systems are used.

1.2 Importance of Face Recognition

To identify the unique patterns of a person's face, face recognition often compares and evaluates facial characteristics. Along with being employed in security and law enforcement, it is used to authenticate identify, and unlock gadgets like cell phones and PCs. Many smartphones and laptops today have facial recognition technology built in to unlock the device.

1.3 Facial Recognition System

Attendance management is a crucial task in various organizations, including educational institutions, workplaces, and other establishments. Traditionally, attendance has been recorded manually, leading to inefficiencies, errors, and time-consuming administrative processes. Automated systems utilizing facial recognition technology are emerging as a promising solution to these challenges. By integrating a solenoid lock controlled by a Raspberry Pi 4, the system ensures that only authorized personnel are granted access.

Due to CNNs' capacity to precisely analyze and recognize complex face patterns, the field of facial recognition has seen substantial growth in recent years. The neural network is

trained on a dataset comprising facial images of authorized individuals, enabling it to learn and distinguish unique facial features associated with each individual.

The Raspberry Pi 4 serves as the central processing unit and acts as the interface between the CNN model and the hardware components. A camera connected to the Raspberry Pi captures facial data, either in the form of images or video streams, which is then processed by the CNN for recognition. The system compares the captured facial data with the pre-existing database of authorized individuals and determines whether a match exists.

Upon successful recognition, the system records the attendance details of the authorized person in a CSV file. The file contains relevant information such as the person's name and the timestamp, providing a digital record of attendance. Additionally, a signal is sent to the solenoid lock via a relay module, prompting it to temporarily unlock and grant access to the authorized person.

The solenoid lock serves as a physical barrier, ensuring that only authorized individuals can enter the designated area. When an unauthorized person attempts to gain access, the facial recognition process does not yield a match, and the solenoid lock remains closed, denying entry.

The Facial Recognition Attendance System offers numerous benefits over traditional attendance management methods. It eliminates the need for manual record-keeping, reducing human errors and administrative burdens. The system also enhances security by ensuring that only authorized personnel can access restricted areas.

This project demonstrates the potential of utilizing advanced technologies to create an automated and secure attendance management system. By combining CNN-based facial recognition with the Raspberry Pi 4 and a solenoid lock, the system provides a reliable and efficient solution for attendance tracking and access control.

In the following sections, we will delve into the implementation details, methodology, and results of the Facial Recognition Attendance System, showcasing its effectiveness in real-world scenarios and its potential for broader application.

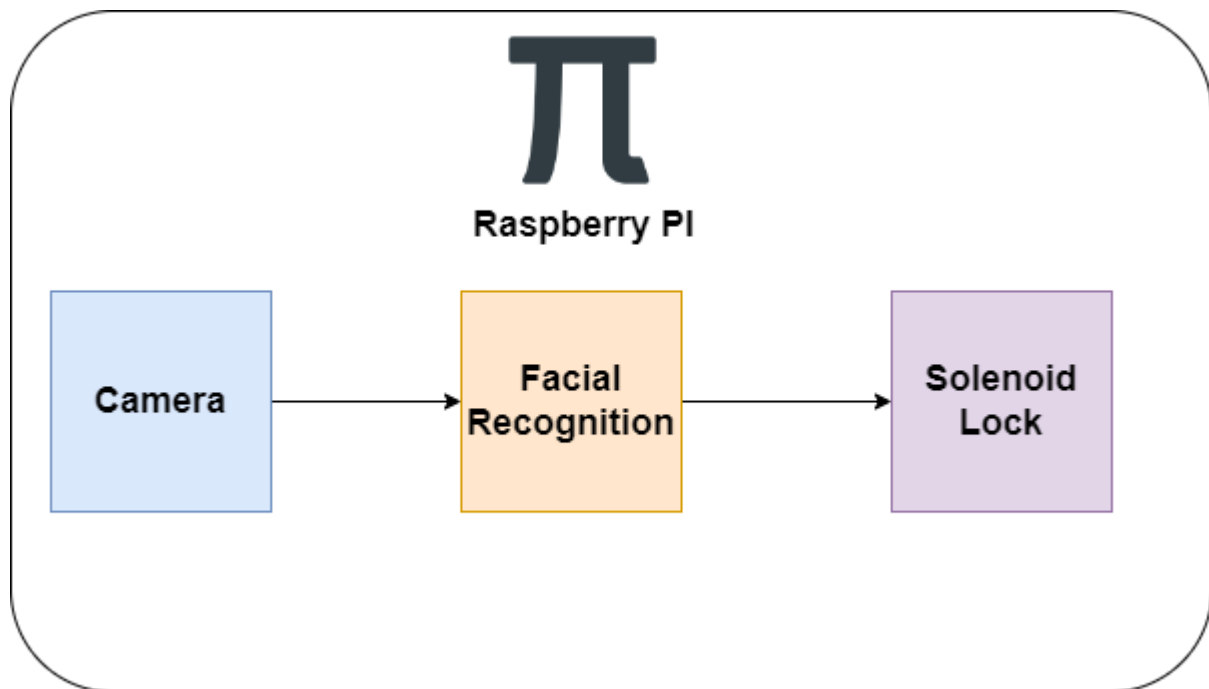


Figure 1-0-1 Block Diagram

1.4 Background

The manual techniques used to take attendance in many contexts, including businesses, educational institutions, and organizations are frequently time-consuming, prone to inaccuracy, and ineffective. Automated attendance systems utilizing facial recognition technology have become more common to overcome these issues.

With these developments, facial recognition technologies have become more accurate and reliable.

The combination of face recognition technology with hardware elements like solenoid locks and the Raspberry Pi provides a complete access control and attendance management system. The low-cost single-board computer Raspberry Pi offers the flexibility and processing power required for real-time facial recognition. Conversely, solenoid locks provide a safe method of access control by momentarily opening doors when authorized people are identified.

The proposed project intends to create a Facial Recognition Attendance System that simplifies attendance monitoring, increases accuracy, and heightens security using facial recognition technology, the Raspberry Pi, and solenoid locks. Real-time authorization recognition, attendance tracking, and access control capabilities will all be provided by the system.

The project leverages on developments in hardware integration, CNNs, and face recognition technologies. It solves the drawbacks of conventional attendance systems, including their reliance on manual labor and vulnerability to fraud. Additionally, it fits with the expanding need across several sectors for automated and effective attendance management solutions.

The Facial Recognition Attendance System has the potential to revolutionize attendance monitoring, streamline user experience, lower administrative burdens, and improve security. For businesses looking to enhance their attendance management procedures and guarantee precise and dependable access control, it presents a potential alternative.

1.5 Problem Statement

Paper registers and card-based systems are examples of traditional attendance management systems that rely on human procedures and are time-consuming, prone to mistake, and simple to hack. These systems often require significant administrative efforts to maintain accurate records and do not provide real-time data or access control mechanisms. There is a need for an automated and reliable attendance management system that addresses these shortcomings and enhances security.

1.6 Objectives

The primary objective of the Facial Recognition Attendance System is to develop a robust and automated solution for attendance management using facial recognition technology. The system aims to achieve the following objectives:

1.6.1 Automation:

Automated systems that do away with the requirement for manual record-keeping and minimize administrative work should be used to replace manual techniques for tracking attendance.

1.6.2 Accuracy:

Improve the accuracy of attendance records by leveraging facial recognition technology, which can identify individuals based on unique facial features, ensuring reliable identification.

1.6.3 Real-time Tracking:

Provide real-time attendance tracking by capturing and processing facial data instantaneously, enabling up-to-date attendance records.

1.6.4 Access Control:

Implement a solenoid lock controlled by the system to restrict access to authorized individuals only, enhancing security and preventing unauthorized entry.

1.6.5 Efficiency:

Streamline the attendance management process by reducing the time required to record attendance and minimizing errors associated with manual methods.

1.6.6 Scalability:

Design the system to be easily scalable, allowing it to accommodate a growing number of individuals without compromising performance.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a quick summary of a number of studies that address issues with computer vision. The earlier study articles are examined in order to comprehend the benefits, downsides, and limitations of the various CNN architectures for face recognition that have been put forth.

2.1 Study of Research Papers

Title: Convolutional neural network data

Deep Learning or Deep Neural Networks are terms used to describe Artificial Neural Networks (ANNs) with numerous layers. In several applications, including pattern recognition, deeper hidden layers are already beginning to perform better than conventional methods. One of the most popular deep neural networks is the convolutional neural network (CNN). The term "convolution" refers to a linear mathematical procedure involving matrices.

The numerous layers that make up CNN have all been detailed, including convolutional, non-linear, pooling, and fully connected layers. Convolutional and fully connected layers have parameters, but pooling and non-linearity layers do not. The CNN does fairly well in machine learning problems.

The results, in particular for image-related applications including the largest image classification data set (Image Net), computer vision, and natural language processing (NLP), were simply astonishing. This essay will outline and describe all of the key terms and concepts associated with CNN, as well as detail how each parameter affects the network's performance. We will also list the factors that affect CNN effectiveness. The CIFAR-10 dataset was the one utilized in the study. [1]

Title: Deep Convolutional Neural Network Architectures in the Recent Era:

Image classification and segmentation, object identification, video processing, natural language processing, and speech recognition are a few of CNN's intriguing applications. Deep CNN's powerful learning skills are a result of the several feature extraction processes it employs to automatically learn representations from the data. The availability of a lot of data and improvements in hardware technology have enhanced the study of CNNs, and exciting

deep CNN designs have recently been revealed. A few of the innovative ideas being researched to create CNN technology include the use of multiple activation and loss functions, parameter optimization, regularization, and architectural advancements. However, architectural advances are what enable the deep CNN's representational capability to significantly increase.

CNNs have proven to perform exceptionally well in tasks including image segmentation, classification, detection, and retrieval, making them one of the finest learning algorithms for understanding picture information. Thanks to its automatic feature extraction capacity, CNN eliminates the requirement for a separate feature extractor. By controlling the change in weights in accordance with the aim during training, CNN learns using the backpropagation method. A backpropagation algorithm's optimization of an objective function is comparable to the human brain's response-based learning. The ResNet-introduced idea of skip connections for deep CNN training gained prominence in 2015. Most subsequently developed networks, like Inception-ResNet, Wide ResNet, ResNeXt, etc., embraced this idea. [2]

Title: Convolutional neural networks for face identification are evaluated when face recognition and deep learning collide.

Recent achievements in facial recognition using convolutional neural networks (CNN), a kind of deep learning, are astounding. The reasons behind why CNNs function effectively and how to create a "good" architecture are still up for debate. Instead of examining the cause, existing research frequently concentrate on reporting CNN architectures that are effective for face recognition.

In order to make the work readily repeatable, this research undertakes an exhaustive review of CNN-based facial recognition systems (CNNFRS). the LFW (Labelled Faces in the Wild) public database. The three CNN architectures suggested in this work are the first known architectures trained on LFW data. Three designs have been proposed: CNN-S, CNN-M, and CNN-L. Comparing several CNN architectures' layer and filter counts. Additionally, it evaluates the performance of face recognition using traits from a number of layers, including the pooling, fully connected, and softmax layers. It was discovered that the properties of the softmax layer somewhat beat those of the most common fully connected layer. [3]

Title: Using building video surveillance as the basis, the attendance and security system

The operation of the contemporary firm depends heavily on the attendance system, and people have long been concerned about the building's security. This study integrates the attendance and security tasks while merging video image processing, deep learning, and facial recognition to create an intelligent attendance and security system based on networked surveillance video. It recommends identifying persons by utilizing a sliding average.

The results of the experiments prove the effectiveness of the recommended strategy. The correct identification rate is 98.85%, the false reject rate is 0.51%, and the false acceptance rate is 2.52%. The technique is employed in some video monitoring areas and provides the advantages of passive, inconspicuous attendance as well as concurrent attendance by several persons. The MTCNN [9] method, which is often used for human face detection, is applied in this study. It approaches detection as a regression issue, just as YOLO, and trains the convolution network to optimize categorization and object positions. The picture pyramid was first constructed, and three CNN models—P-Net, R-Net, and O-Net—were then developed. These models created cross-entropy loss and Euclidean distance loss functions, described the label of the picture, did cascade prediction to improve accuracy, and more. To learn the five key points of the face and how to position them, training is necessary.

This study combines the network architectural ideas of Alex Net and Inception to create a convolutional neural network for face recognition and categorization. Additionally, it recommends a technique based on sliding average to improve recognition accuracy and creates a face dataset in real-world scenarios to evaluate the effectiveness of the algorithm. [3]

Title: Using Deep Learning, Calculate Attendance in University Classes

An key factor in evaluating a classroom is attendance. By combining the MTCNN face detection and Center-Face face identification deep learning algorithms, this study creates an automated attendance system for university classes.

Numerous experimental findings indicate that:

- The system can record absence, tardiness, and early departure as these three infractions of the classroom's rules for automatic attendance. Following class, information regarding each student's progress is instantly entered into an attendance table.
- The technology recognises faces accurately and quickly, taking only 100 milliseconds per frame. This facial recognition model has a 98.87% accuracy rate, a false positive rate on LFW of 93.7%, and a true positive rate of less than 1/1000. In order to achieve non-interference automated and complete class attendance, MTCNN, based on deep

learning, integrates face detection, face landmark, and Centre Face algorithm. Additionally, it can provide the three markers of classroom attendance—absence, tardiness, and early departure. It is an extremely promising university class attendance system. [4]

2.2 Existing System

A few built-in datasets and a few built-in designs are used by the current face recognition system. Despite the fact that certain bespoke data sets have been created and utilized, they have done so by utilizing the pre-existing architectural models. Several of the models that have been applied to the current systems include:

- Fischer Face Recognition
- Eigen face recognition
- Local Binary Histogram Pattern (LBHP)

2.2.1 (LBHP) Local Binary Histogram Pattern

When employing the LBHP approach for texture classification, the instances of the LBHP codes in a picture are collected into a histogram. The categorization is then concluded using straightforward histogram similarities. One method for attaining this goal is to build several local descriptions of the face using the LBHP texture descriptors and then combine them into a global description.

Local parts of the face picture are separated, and LBHP texture descriptors are individually derived from each region. As seen in Fig. 2.1, the descriptors are then combined to create a comprehensive description of the face.

This histogram effectively provides information about the face on three different levels of locality: a global description of the face is created by concatenating the regional histograms; pixel-level information about the patterns is contained in the labels for the histogram's LBHP; and regional information is produced by summarizing the labels over a limited area. [5]

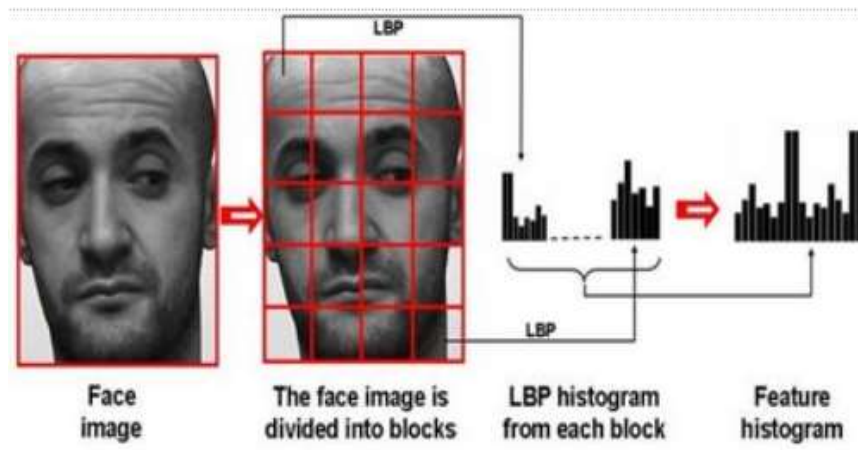


Figure 2-0-1 Local Binary Patterns with face

2.2.2 Fischer Face Recognition

Typically, a face recognition system will take an image or video stream as input and identify the subject or subjects that are visible in the input as output.

One of the extensively used facial recognition algorithms is Fisher facial, which is thought to be better to other methods like Eigen Face due to its focus on maximizing the separation between classes during training. Fisher Face technique for image identification relies on Principal Component Analysis to reduce the face space dimension.

The Eigen faces approach clearly captures illumination, whereas the Fischer Face

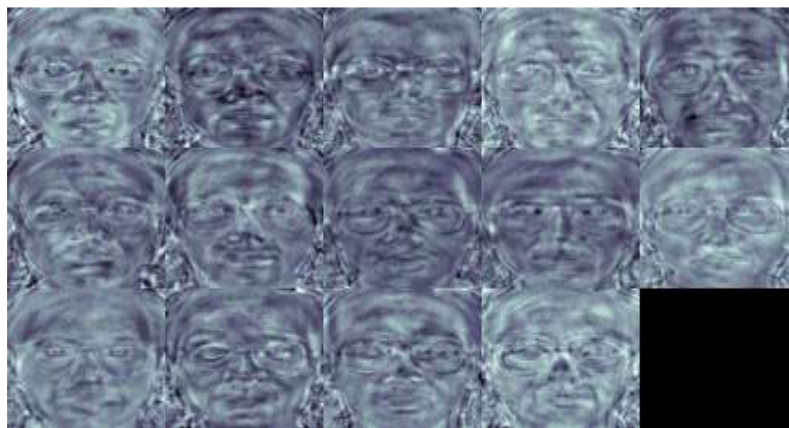


Figure 2-0-2 Fischer Faces

method develops a class-specific transformation matrix. Instead, the Discriminant Analysis discovers that there is a difference between people based on their face characteristics. The Fischer Face comes in particularly handy when face photographs contain a wide range of lighting and emotions. Figure 2.2 is an instance of Fischer's facial depiction. [6]

2.2.3 Eigen Face Recognition

When utilized in the computer vision issue of real-world face recognition, a set of Eigen vectors are referred to as an "Eigen face." The probability distribution covariance matrix over the high-dimensional vector space of face photos is used to determine the eigenvectors. The basis set of all the pictures used to create the covariance matrix is the Eigen faces themselves. By enabling the smaller number of base pictures to represent the initial training images, this results in dimension reduction. By contrasting how the base set represents faces, bracket may be created.

A vast collection of photos showing various mortal faces may be subjected to a precise procedure known as star element Analysis (PCA), which can provide a set of Eigen faces.

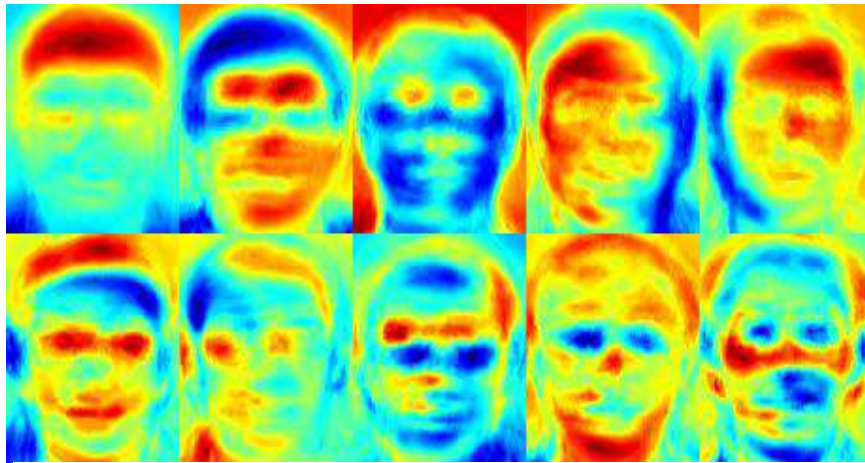


Figure 2-0-3 Eigen Faces Diagram

Eigen faces are a collection of "standardized face constituents" that were discovered through statistical analysis of a large number of film faces. Surprisingly, a good approximation of most faces may be obtained without combining several Eigen faces. Figure 2.3 shows a visualization of a Fischer face as an example. [7]

2.3 Proposed System

Using facial recognition technology, a Raspberry Pi 4, and a solenoid lock, the proposed Facial Recognition Attendance System streamlines attendance management and enhances access control. The model compares the captured facial data with a pre-existing database of authorized individuals to determine if a match exists.

Attendance Logging: Upon successful facial recognition, the system records the attendance details of the authorized person in a CSV file. The CSV file includes information such as the person's name and the timestamp, providing a digital record of attendance. This automated logging eliminates the need for manual record-keeping and reduces administrative efforts.

Solenoid Lock Integration: The system integrates a solenoid lock, controlled by a relay module connected to the Raspberry Pi. When an authorized person's attendance is confirmed, a signal is sent to the solenoid lock, temporarily unlocking it and allowing access. In contrast, if an unauthorized person is detected, the solenoid lock remains closed, denying entry.

Raspberry Pi Interface: The Raspberry Pi 4 serves as the central processing unit and provides the interface between the CNN model, camera, solenoid lock, and other components. It facilitates the communication, coordination, and control of the system's functionalities.

Real-time Processing: The system is designed to process facial data and perform recognition in real-time. This enables instant attendance tracking and access control, providing up-to-date information for attendance management.

Scalability: The suggested solution is intended to be scalable, enabling the database to accommodate additional authorized users without experiencing performance issues. The CNN model can be trained on an expanding dataset to accommodate a growing number of individuals.

CHAPTER 3

METHODOLOGY

In this chapter we can discuss about the methodology of our system that how our system work what are the steps included in the system.

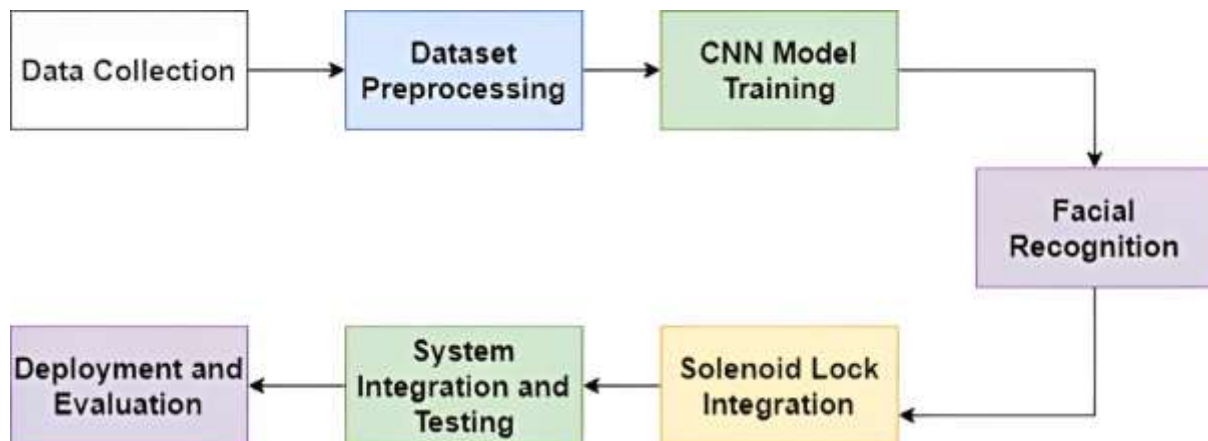


Figure 3-0-1 Methodology Diagram

The development of the Facial Recognition Attendance System involves the following steps and methodologies:

3.1.1 Data Collection:

- a) Gather a dataset of facial images for training the Convolutional Neural Network (CNN) model. This dataset should include images of authorized individuals with a diverse range of facial expressions, lighting conditions, and angles.
- b) Capture facial images of authorized individuals using a camera connected to the Raspberry Pi. Ensure that the images are of good quality and represent the individuals accurately.

3.1.2 Dataset Preprocessing:

- a) The acquired dataset should be preprocessed by scaling the photos to a uniform size, normalizing the pixel values, and converting them to grayscale or RGB, depending on the requirements of the facial recognition algorithm and CNN model.

- b) Use data augmentation methods to boost the dataset's variety and strengthen the resilience of the CNN model, such as rotation, flipping, and random cropping.

3.1.3 CNN Model Training:

- a. Design and implement a CNN architecture suitable for facial recognition tasks. There may be convolutional layers, pooling layers, fully connected layers, and activation functions in the architecture.
- b. Split the preprocessed dataset into training and validation sets to train and evaluate the CNN model. Utilize techniques like cross-validation or hold-out validation to assess the model's performance.
- c. Change various parameters like learning rate, batch size, and the number of epochs to optimize the performance of the CNN model while training it using the training dataset..
- d. Validate the trained model using the validation dataset to ensure it generalizes well to unseen facial images.

3.1.4 Facial Recognition:

- a. Capture facial data in real-time using a camera connected to the Raspberry Pi.
- b. Preprocess the captured facial data by resizing, normalizing, and converting it to the required format for input to the CNN model.
- c. Feed the preprocessed facial data into the trained CNN model to obtain facial embedding's or a similarity score.
- d. Compare the facial embedding's or similarity score with the database of authorized individuals to determine if there is a match.
- e. If a match is found, record the attendance details, including the person's name and timestamp, in a CSV file. Send a signal to the solenoid lock to temporarily unlock and grant access.
- f. If no match is found, deny access and do not record attendance.

3.1.5 Solenoid Lock Integration:

- a. To control the solenoid lock, attach the relay module to the Raspberry Pi.

- b. Establish the necessary electrical connections between the relay module and the solenoid lock, ensuring compatibility and safety.
- c. Develop code to send signals to the relay module based on the facial recognition results, triggering the solenoid lock to open or remain closed accordingly.

3.1.6 System Integration and Testing:

- a. Integrate the facial recognition module, attendance logging module, and solenoid lock control module into a cohesive system.
- b. Conduct rigorous testing to ensure the system functions as intended, including testing for accurate facial recognition, attendance logging, and reliable control of the solenoid lock.
- c. Perform end-to-end testing, simulating real-world scenarios and verifying the system's performance in different lighting conditions, angles, and with varying individuals.

3.1.7 Deployment and Evaluation:

- a. Deploy the Facial Recognition Attendance System in the intended environment, such as educational institutions or workplaces.
- b. Monitor the system's performance, accuracy, and reliability over an extended period, gathering feedback from users and stakeholders.
- c. Evaluate the system's effectiveness in automating attendance management, reducing administrative efforts, and enhancing access control based on the defined objectives.

3.2 Algorithm:

- Initialize the system, including the Raspberry Pi, camera, solenoid lock, and necessary libraries and modules.
- Load the pre-existing dataset of facial images of authorized individuals for training the CNN model.
- Preprocess the dataset by resizing images, normalizing pixel values, and converting them to the required format.

- Design and implement a CNN architecture for facial recognition, incorporating activation processes, fully connected layers, pooling layers, and convolutional layers.
- Preprocessed dataset should be divided into training and validation sets.
- Train the CNN model using the training dataset, adjusting hyper parameters as needed.
- Validate the trained model using the validation dataset to assess its performance.
- Capture facial data in real-time using the camera connected to the Raspberry Pi.
- Preprocess the captured facial data by resizing, normalizing, and converting it to the required format.
- Feed the preprocessed facial data into the trained CNN model to obtain facial embedding's or similarity scores.
- Compare the facial embedding's or similarity scores with the database of authorized individuals.
- If a match is found:
 - Record the attendance details, including the person's name and timestamp, in a CSV file.
 - Send a signal to the solenoid lock via the relay module to temporarily unlock and grant access.
- If no match is found:
 - Deny access.
 - Do not record attendance.
- Repeat steps 8-13 for continuous facial recognition and attendance management.

3.3 Flowchart:

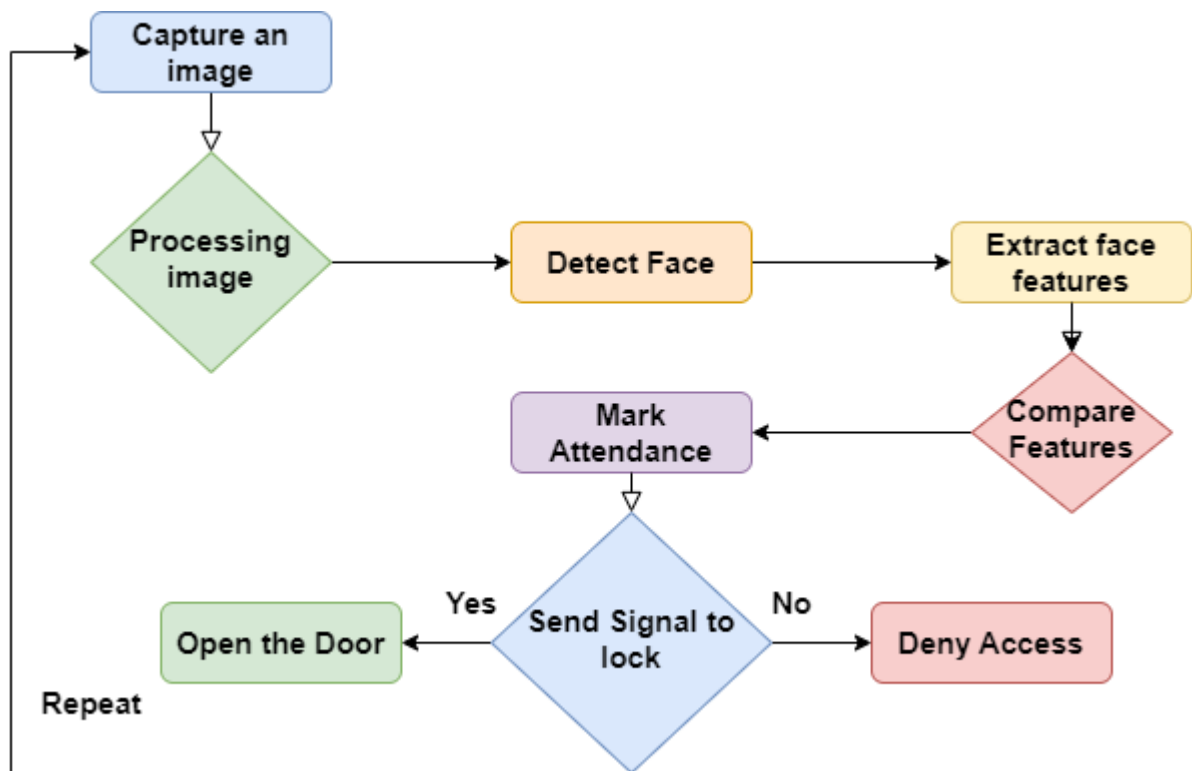


Figure 3-2 Block Level Flow Chart

In this flowchart we explain the working of our system in steps. First we capture the images and make a dataset then preprocess the images, then detect the faces of the persons after that it extract the features of the persons faces to make a good training file then at the time of recognition it compare the features of the persons faces with that training files to predict the persons, if they are authorized persons then their attendance marked in the csv file and open the door for some time and if they are unauthorized persons then their attendance would not marked also door will not opened for them.

CHAPTER 4

IMPLEMENTATION

Implementation of the Facial Recognition Attendance System involves several steps. Here is a general outline of the implementation process:

4.1 Hardware Setup:

Several hardware components must be connected to the Raspberry Pi 4, including the camera, solenoid lock, relay module, LCD for display and connecting wires etc. The Raspberry Pi must be powered on and a secure connection with the internet.

4.2 Software Installation:

On the Raspberry Pi, install its operating system e.g: Raspbian. Libraries such as OpenCV, face recognition libraries, and CSV file handling libraries must be installed.

4.3 Dataset Collection:

Collect a dataset of facial images of authorized individuals. This can be done by capturing images using the camera module or using pre-existing datasets. Ensure a diverse range of images representing different facial angles, lighting conditions, and expressions for robust model training.

4.4 Dataset Preprocessing:

Preprocess the collected facial images by resizing them to a uniform size. Normalize the pixel values to improve consistency and remove variations in lighting conditions. Convert the images to the required format, such as grayscale or RGB.

4.5 CNN Model Training:

Construct a CNN architecture that is adequate for facial recognition that includes convolutional layers, pooling layers, fully connected layers, and activation functions. Preprocessed dataset should be divided into training and validation sets.

- Train the CNN model using the training dataset, adjusting hyper parameters as needed.
- Validate the trained model using the validation dataset to assess its performance.

4.6 Real-time Facial Recognition:

Set up the camera module to capture live video feed.

Preprocess the captured facial images in real-time by resizing, normalizing, and converting them to the required format.

Feed the preprocessed images into the trained CNN model to obtain facial embedding's or similarity scores.

Compare the facial embedding's or similarity scores with the database of authorized individuals to determine if there is a match.

4.7 Attendance Logging and Access Control:

If a match is found, record the attendance details, including the person's name and timestamp, in a CSV file.

Send a signal to the solenoid lock via the relay module to temporarily unlock and grant access.

If no match is found, deny access and do not record attendance.

4.8 Continuous Operation:

Implement a loop to continuously perform real-time facial recognition and attendance logging.

Handle any potential errors or exceptions that may occur during the operation.

4.9 Testing and Evaluation:

Test the system with a variety of scenarios and individuals to assess its accuracy and performance.

Calculate metrics like accuracy, false acceptance rate, false rejection rate, and recognition speed in order to evaluate the effectiveness of the system.

4.10 Refinement and Optimization:

Analyze the system's performance and identify areas for improvement.

Fine-tune the CNN model, adjust parameters, or incorporate advanced techniques to enhance accuracy and robustness.

Iterate on the implementation and make necessary adjustments based on testing and evaluation results.

4.11 Hardware Tools:

- Raspberry Pi 4
- Camera module
- Solenoid lock

- Relay module
- Connecting wires
- Power supply (charger)

4.12 Software Tools:

- Operating system: Raspbian (or preferred Linux distribution)
- Python programming language
- OpenCV, Numpy, Tensorflow, keras, matplotlib libraries
- Face recognition libraries (e.g., dlib, OpenFace)
- CSV file handling libraries (e.g., pandas)
- Colab Tool
- Anaconda IDE

4.13 Training Phase:

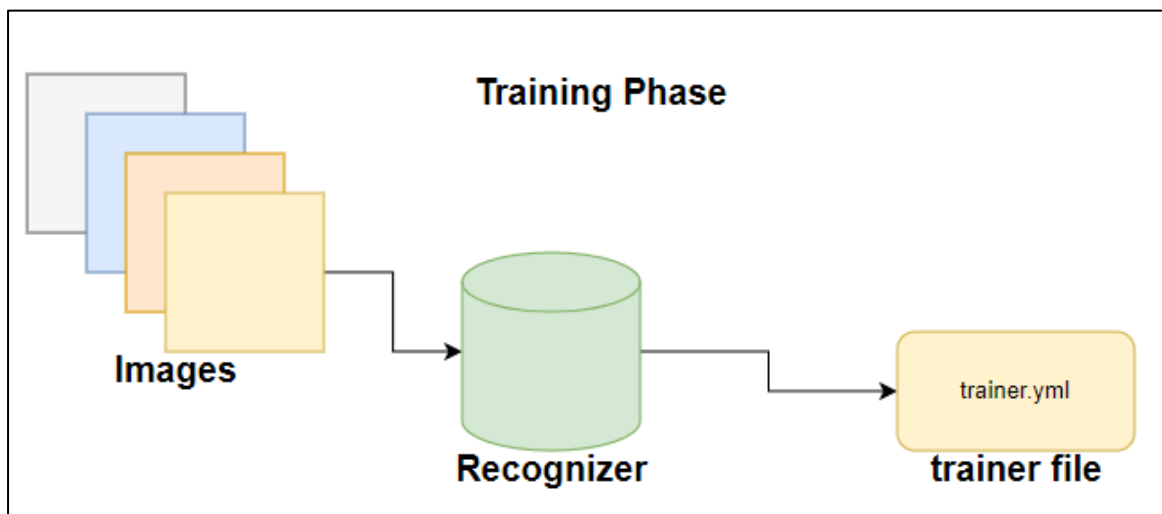


Figure 4-0-1 Training Phase

In training phase recognizer take images and recognized these images and then create a trainer files.

4.14 Testing Phase:

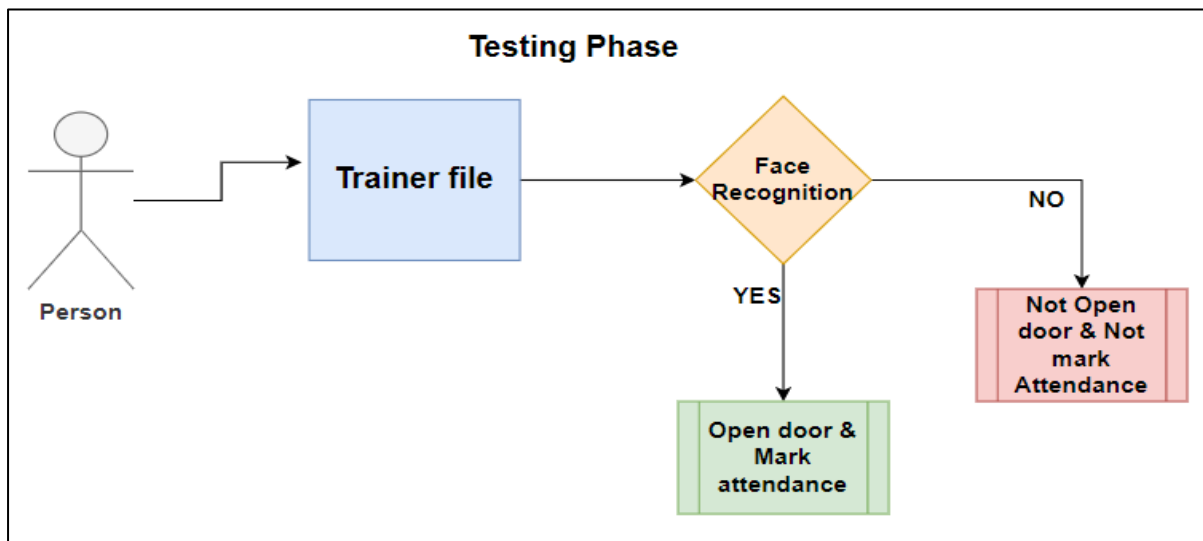


Figure 4-0-2 Testing Phase

In testing phase recognizer take trainer file and then perform face recognition if the person recognizes open the door and mark the attendance while if unknown person recognizes, the system doesn't recognizes him, also not mark his attendance and not open the door.

4.15 Diagram of Use Case:

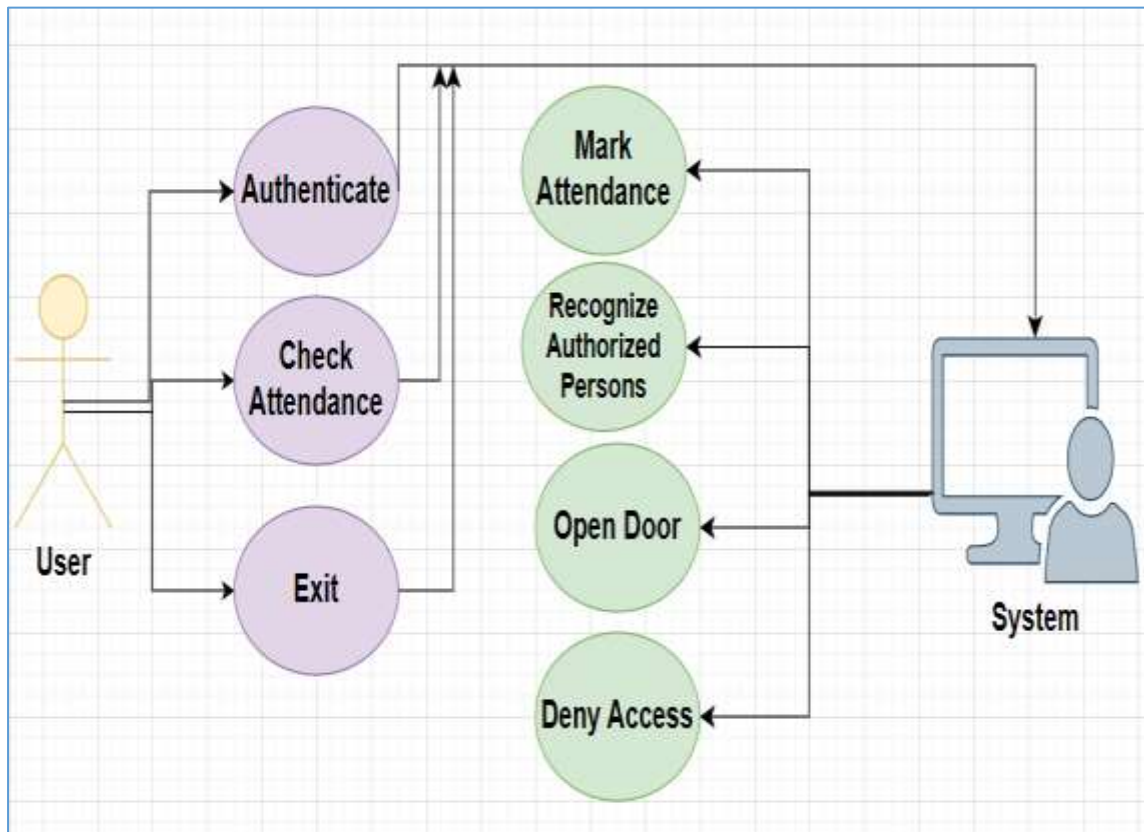


Figure 4-3 Use Case Figure

The user logs in to the system to access its functionalities. Check Attendance and Exit. The system uses facial recognition algorithms to recognize authorized persons and verify their identities. Open Door for Authorized Persons using Solenoid Lock the system sends a signal to the solenoid lock to open the door when an authorized person is recognized while doesn't recognizes unauthorized persons and also not mark their attendance not open the door.

CHAPTER 5

SIMULATION AND RESULTS

We go through our system's simulation and findings in this chapter.

5.1 Simulation:

We will be utilizing software tools and libraries to build a virtual environment that simulates the real-time performance of the facial recognition Attendance System in order to imitate it. A comprehensive description of the simulation procedure is given below:

5.1.1 Dataset Generation:

- Create a synthetic dataset of facial images that represent authorized individuals.
- Generate variations in facial angles, lighting conditions, and expressions to mimic real-world scenarios.
- Label the synthetic dataset with corresponding names and timestamps.

5.1.2 Preprocessing:

- Making the photos a consistent size is a preprocessing step for the synthetic dataset.
- Normalize the pixel values to remove variations in lighting conditions.
- Convert the images to the required format, such as grayscale or RGB.

5.1.3 Model Training:

- Utilize the preprocessed synthetic dataset to train a CNN model.
- Give a brief description of the convolutional layers, pooling layers, fully connected layers, and activation functions that make up the architecture of the CNN model.
- Adjust hyper parameters and optimize the model's performance.

5.1.4 Real-time Simulation:

- Simulate the real-time operation by feeding pre-recorded video footage into the trained CNN model.
- Use a webcam or pre-recorded video files as the input source for the simulation.
- Apply real-time image processing techniques, such as face detection, to extract facial regions from the video frames.

5.2 Facial Recognition and Attendance Logging:

- Apply the trained CNN model to the extracted facial regions to perform facial recognition.

- Compare the facial embedding's or similarity scores with the synthetic dataset of authorized individuals.
- If a match is found, log the attendance details, including the person's name and timestamp, in a simulated CSV file.

5.3 Access Control Simulation:

- Simulate the opening and closing of a solenoid lock based on the recognition results.
- If a match is found, simulate the solenoid lock opening for a predetermined duration.
- If no match is found, simulate the solenoid lock remaining closed.

5.4 Logging and Visualization:

- Analyze the simulated attendance data logged in the CSV file.
- Generate visualizations, such as graphs or charts, to represent attendance statistics and trends.
- Assess the system's accuracy by contrasting the simulated outcomes with the artificial dataset's known ground truth.

5.5 Performance Evaluation:

- Using the simulated results, calculate the parameters for accuracy, false acceptance rate, false rejection rate, and recognition speed.
- Compare the simulation system's performance under various conditions and modifications.
- By simulating the Facial Recognition Attendance System, we can assess its performance, evaluate different configurations, and make necessary refinements before deploying it in real-world scenarios. The simulation provides a controlled environment to test and optimize the system, ensuring its effectiveness and reliability in practical applications.

5.6 Results:



Figure 5-1 System Overview

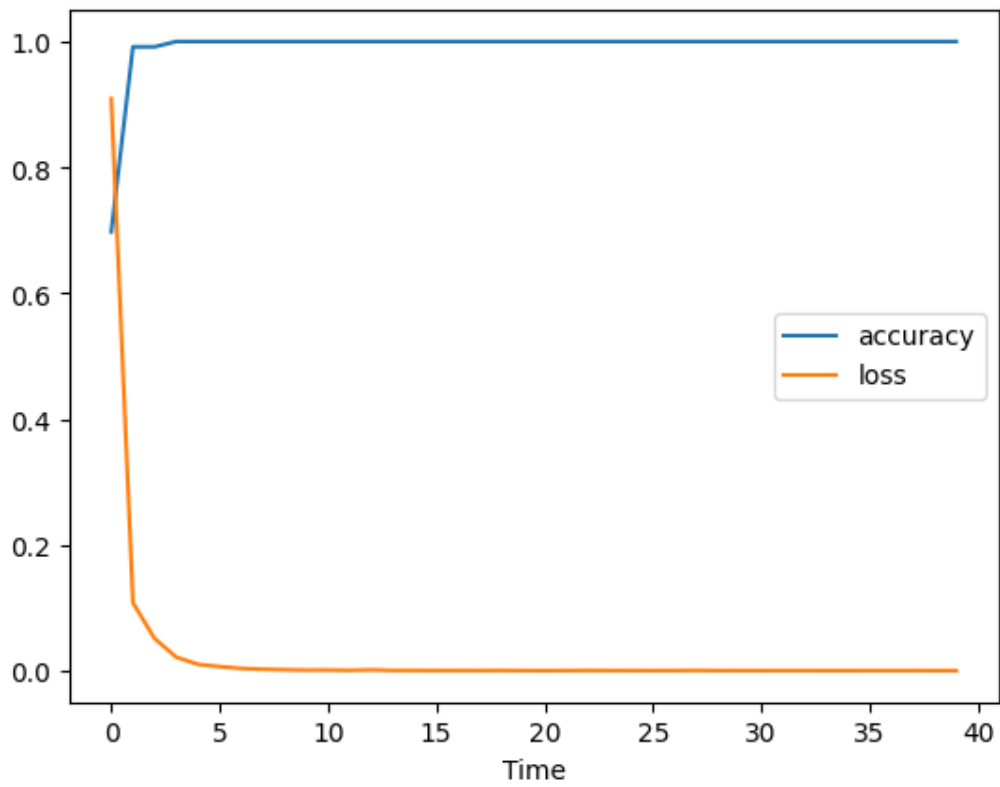


Figure 5-2 Accuracy loss Graph

5.6.1 Software Results:



Figure 5-3 Software Results

5.6.2 Hardware Results:

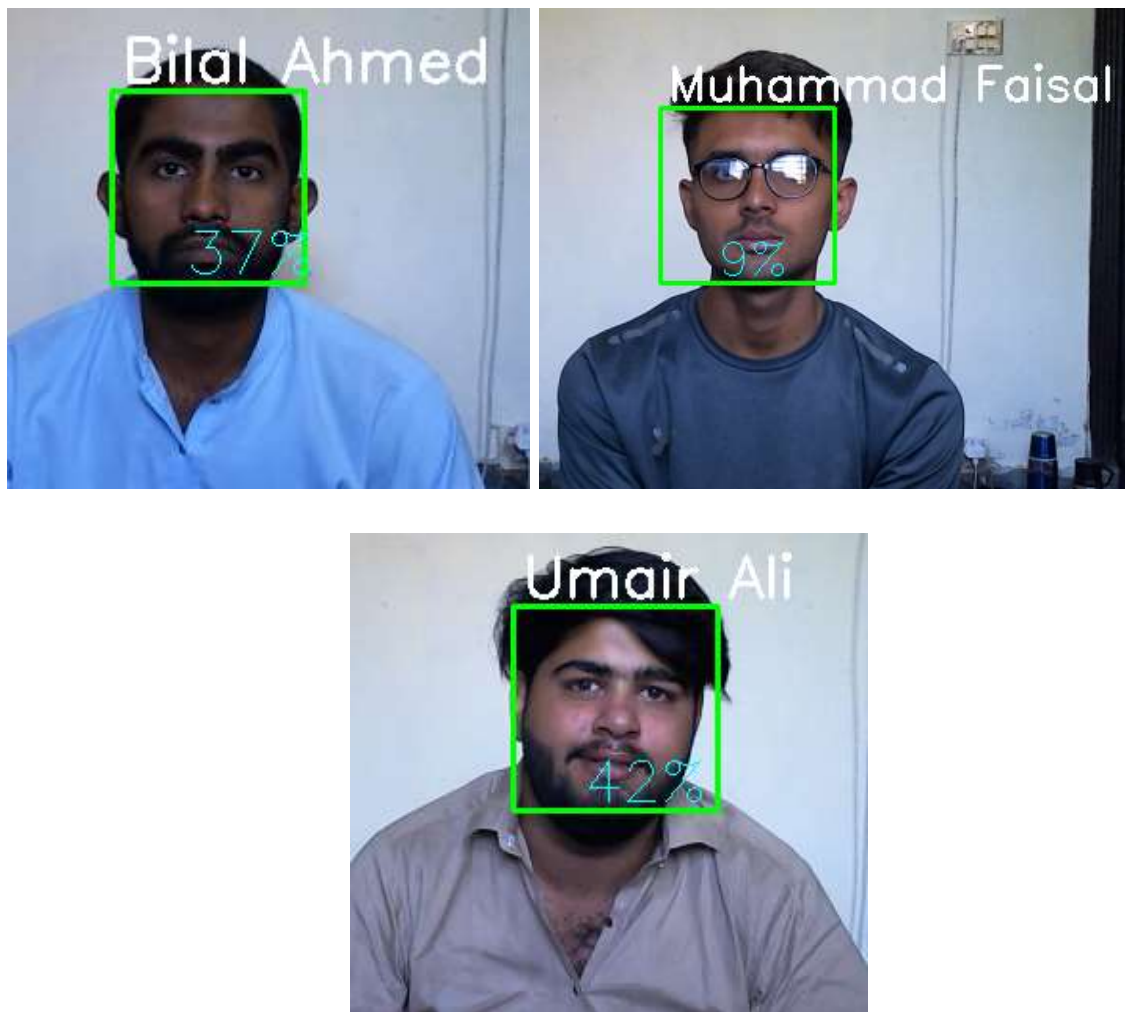


Figure 5-4 Hardware Results 1



Figure 5-5 Hardware Results 2



Figure 5-6 Lock open for authorized person



Figure 5-7 Lock not open for unauthorized person

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this chapter, we go through how our project came to an end, the work that had been asked for it, what we really did, and some of the future updates we want to make.

6.1 Conclusion

The Facial Recognition Attendance System utilizing a CNN model and solenoid lock integration offers an efficient and automated solution for attendance management and access control. By leveraging facial recognition technology and the Raspberry Pi 4 platform, the system streamlines the attendance tracking process, reduces administrative efforts, and enhances security. The successful implementation of the system demonstrates its potential to revolutionize attendance management in various domains, including educational institutions, workplaces, and organizations.

Through the development and simulation of the system, it has been shown that the facial recognition module can accurately recognize authorized individuals and record their attendance in real-time. The integration with the solenoid lock provides secure access control, granting entry only to authorized individuals while denying access to unauthorized persons.

6.2 Proposed Work:

Data Collection and Preprocessing: The project involved collecting a dataset of facial images of authorized individuals. The dataset was preprocessed by resizing, normalizing, and converting the images to the required format. This step aligns with the proposed methodology.

CNN Model Training: The project included training a CNN model using the preprocessed dataset of facial images. The model was designed and implemented to recognize authorized individuals accurately. Training the CNN model aligns with the proposed methodology.

Facial Recognition and Attendance Logging: The project implemented real-time facial recognition using the trained CNN model. When an authorized person is recognized, their

attendance is marked in a CSV file. This step aligns with the proposed methodology and system objectives.

Solenoid Lock Integration: The project integrated a solenoid lock with the Raspberry Pi. When an authorized person's attendance is marked, a signal is sent to the solenoid lock to temporarily unlock and grant access. This aligns with the proposed system and objectives.

6.3 Future Improvements:

Continuous Model Training: Implementing a mechanism to continuously update and retrain the CNN model with new facial images of authorized individuals would be a valuable improvement to ensure the system adapts to changes in the authorized individual database.

Robustness to Environmental Factors: Enhancing the system's robustness to environmental factors such as varying lighting conditions, facial angles, and occlusions would improve recognition accuracy. Augmenting the dataset and incorporating advanced techniques can help achieve this.

Multi-factor Authentication: The security of the system can be further improved by incorporating other authentication elements, such as voice recognition or fingerprint recognition, which can thwart spoofing efforts.

Real-time Monitoring and Notifications: Implementing a monitoring system that provides real-time notifications to administrators or authorized personnel in case of anomalies or security breaches would be a valuable addition to the system.

User-friendly Interface: Developing a user-friendly interface for system configuration, attendance tracking, and management would enhance the usability of the system.

The future work outlined above presents opportunities for further refinement and enhancement of the system's functionality, robustness, and usability.

REFERENCES

- [1] Bhattacharya and Shubhobrata, "Convolutional neural network data," *IEEE Explore*, 2018.
- [2] Mridha, Krishna and N. T. Yousef, "A Survey of the Recent Architectures of Deep Convolutional Neural Networks," *IEEE Explore*, 18 6 2021.
- [3] Sameem, M. S. Islam, "When face recognition meets with deep learning," *IEEE Explore*, 15 12 2016.
- [4] T. Qasim and K. Bakhat, "University Classroom Attendance Based on Deep Learning," *IEEE Explore*, 5 10 2020.
- [5] Parth Singh, "Local Binary Histogram Pattern (LBHP)," *Analytics Vidhya*, 12 7 2021.
- [6] Y. Saini, "Fischer Face Recognition," *Open Genious IQ*, 2021.
- [7] A. Rosebrock, "OpenCV Eigenfaces for Face Recognition," *PyimageSearch*, May 10 2021.
- [8] Rovai, M (n.d), "Real Time Face Recognition," *Towards datascience*, 1 10 2022.
- [9] S. Sharma, Sathesh Kumar, "CNN base efficient face recognition technique," *FAREC*, 2016.
- [10] Shifa Inges Yudita, Teddy Mantoro, "Deep face recognition for imperfect faces," *4th International Conference of Computer Engineering*, 2021.
- [11] Edy Winarno, Imam Husni Al Amin, "Attendance system based on face recognition," *International Seminar on Research of Information Technology*, 2019.
- [12] Susetyo Bagas Bhaskoro, Siti Aminah, "Attendance system of moving objects using MTCNN and CNN," *3rd International Symposium on Material and Electrical Engineering*, 2021.

- [13] Junya Ueda, Katsunori Okajima, "Face morphing using average face," *11th International Symposium on Image and Signal Processing*, 2019.
- [14] Shakir Fattah Kak, Firas Mahmood Mustafa, "Design and Enhancement of a CNN Model," *3rd International Informatics and Software Engineering*, 2022.
- [15] Ankit S. Vyas, Vipul K. Dabhi, "Survey on Face Expression Recognition using CNN," *5th International Conference on Advanced Computing*, 2019.
- [16] Ran HE, Xiang Wu, "Wasserstein CNN," *IEEE Transactions on Pattern Analysis*, 2019.
- [17] Xiaojun Bai, Feihu Jiang, "Attendance System Based on Face Recognition," *International Conference on Computer Network*, 2020.
- [18] M. Pantic, I.Patras, "Dynamics of facial expression," *IEEE Transactions on Systems*, 2006.
- [19] Yongqiang Li, Shangfei Wang, "Simultaneous Facial Feature Tracking," *IEEE Transactions on Image Processing*, 2013.
- [20] Wenming Zheng, Xiaoyan Zhou, "Facial expression recognition using KCCA," *IEEE Transactions on Neural Networks*, 2006.
- [21] Nan Zhang, Xue Geng, "Facial expression recognition based on LFR," *4th IET International Conference on Wireless*, 2011.
- [22] Kwang-Eun Ko, Kwee-Bo Sim, "Development of a Facial Emotion Recognition Method," *International Conference on Cyberworlds*, 2010.
- [23] Sina Mohseni, Niloofar Zarei, Saba Ramazani, "Facial expression recognition using anatomy based facial graph," *IEEE International Conference on Systems*, 2014.
- [24] M. Pantic, L.J.M.Rothkrantz, "Facial action recognition for facial expression," *IEEE Transactions on Systems*, 2004.

APPENDICES

Train model Code:

```
import numpy as np

import tensorflow as tf

import keras

from keras.models import Sequential

from keras.layers import Conv2D, MaxPooling2D, Dense, Flatten, Dropout

from keras.layers import BatchNormalization

from keras_preprocessing import image

from tensorflow.keras.preprocessing.image import ImageDataGenerator


train_dir = "dataset"

generator = ImageDataGenerator()

train_ds = generator.flow_from_directory(train_dir, target_size=(224, 224), batch_size=32)

classes = list(train_ds.class_indices.keys())


model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(224, 224, 3)))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(BatchNormalization())

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))
```

```

model.add(BatchNormalization())

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(BatchNormalization())

model.add(Conv2D(96, kernel_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(BatchNormalization())

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu'))

model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(BatchNormalization())

model.add(Dropout(0.2))

model.add(Flatten())

model.add(Dense(128, activation='relu'))

model.add(Dense(len(classes), activation='softmax'))


model.compile(

    loss='categorical_crossentropy',

    optimizer='adam',

    metrics=["accuracy"])

model.summary()


history = model.fit(train_ds, epochs=40, batch_size=32)

```

```
# Save the trained model

import h5py

model.save("facial_recognition_model.h5", overwrite=True)
```

Face Recognition Code:

```
import cv2

import numpy as np

import keras

from keras.models import load_model

from keras_preprocessing import image

from tensorflow.keras.preprocessing.image import img_to_array

import matplotlib.pyplot as plt

import csv

import datetime

import RPi.GPIO as GPIO

import time


# Load the trained model

model = load_model("facial_recognition_model.h5")


# Define class labels

classes = ["Akshay Kumar", "Alexandra Daddario", "Alia Bhatt", "Amitabh Bachchan", "Andy Samberg", "Anushka Sharma", "Billie Eilish", "Brad Pitt", "Camila Cabello", "Charlize"]
```

```
Theron","Claire Holt","Courtney Cox","Dwayne Johnson","Elizabeth Olsen","Ellen  
Degeneres","Bilal Ahmed","Muhammad Faisal","Umair Ali","Unknown"] # Add the names  
of authorized persons here
```

```
# Initialize the face cascade
```

```
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +  
'haarcascade_frontalface_default.xml')
```

```
# GPIO pin for controlling the solenoid lock
```

```
LOCK_PIN = 18
```

```
# Initialize GPIO
```

```
GPIO.setmode(GPIO.BCM)
```

```
GPIO.setup(LOCK_PIN, GPIO.OUT)
```

```
# Function to open the solenoid lock
```

```
def open_lock():
```

```
    GPIO.output(LOCK_PIN, GPIO.HIGH)
```

```
    time.sleep(2) # Adjust the delay as per your lock's requirement
```

```
    GPIO.output(LOCK_PIN, GPIO.LOW)
```

```
# Start the video capture
```

```
video_capture = cv2.VideoCapture(0)
```

```

while True:

    # Capture frame-by-frame

    ret, frame = video_capture.read()


    # Convert the frame to grayscale

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


    # Detect faces in the frame

    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30))


    # Iterate over each detected face

    for (x, y, w, h) in faces:

        # Extract the face ROI (Region of Interest)

        face_roi = frame[y:y + h, x:x + w]


        # Preprocess the face image

        face_roi = cv2.resize(face_roi, (224, 224))

        face_roi = cv2.cvtColor(face_roi, cv2.COLOR_BGR2RGB)

        face_roi = img_to_array(face_roi)

        face_roi = np.expand_dims(face_roi, axis=0)


    # Make predictions on the face ROI

```

```

pred = model.predict(face_roi)

label_index = np.argmax(pred)

label = classes[label_index]

confidence = pred[0][label_index]


# Check if the predicted label is authorized

if label != "Unknown" and confidence > 0.7: # Adjust the confidence threshold as per
your requirement

    # Mark attendance in the CSV file

    with open('attendance.csv', 'a', newline='') as file:

        writer = csv.writer(file)

        writer.writerow([label, datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")])


    # Open the lock

    open_lock()


# Draw the bounding box and label on the frame

cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv2.putText(frame, f"{label} ({confidence:.2f})", (x, y - 10),
cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 255, 0), 2)


# Display the resulting frame

cv2.imshow('Real-Time Face Recognition', frame)

```



```
# Break the loop if 'q' is pressed
```

```
if cv2.waitKey(1) & 0xFF == ord('q'):
```

```
    break
```

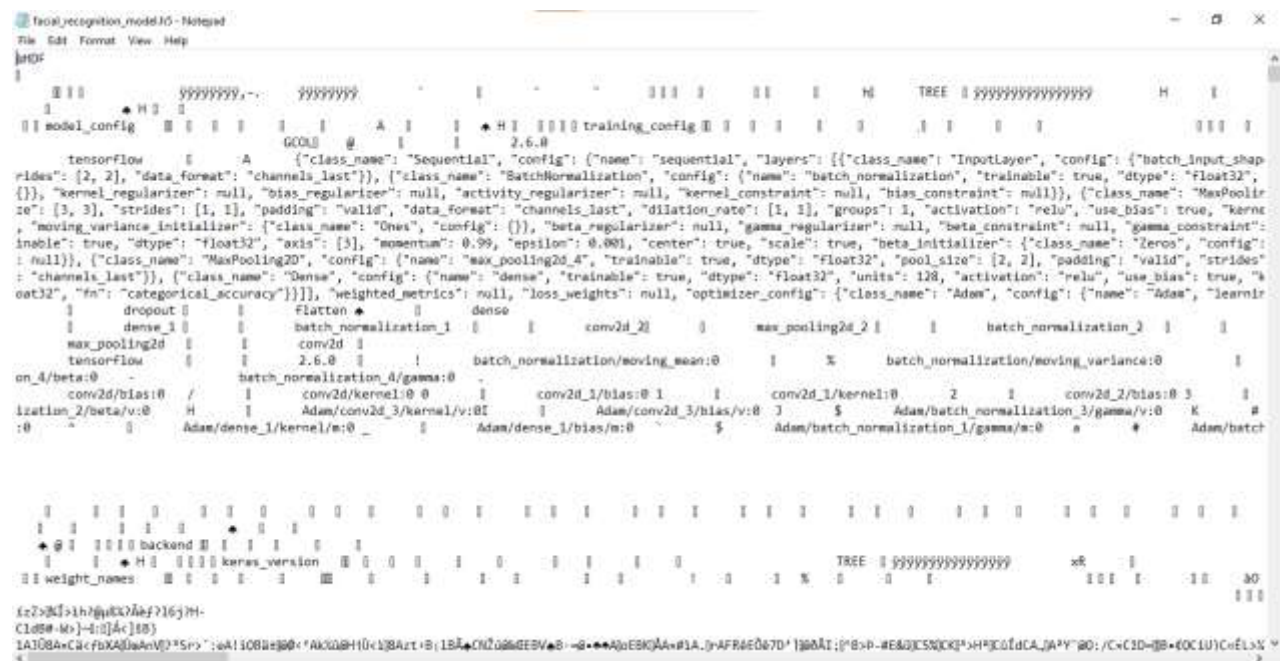
```
# Release the video capture and close all windows
```

```
video_capture.release()
```

```
cv2.destroyAllWindows()
```

```
GPIO.cleanup()
```

Facial recognition model weight file:



```
facial_recognition_model.h5 - Notepad
File Edit Format View Help
[...]
```

