FACE RECOGNITION SYSTEM

A MINI PROJECT REPORT

Submitted by

RAKSHITAGARWAL[RA2011026010340]
TAPNANSHUATHARVA[RA2011026010309]

Under the guidance of

Dr. Surest. K
(Associate professor, CINTEL)

In partial satisfaction of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING
With specialization in <AIML>

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR - 603203

APRIL 2023

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this project report **"Face recognition system"** is the bonafide work of **"Rakshit Agarwal (RA2011026010340), Tapnanshu Atharva (RA2011026010309)** of III Year/VI Sem B.tech(CSE) who carried out the mini project work under my supervision for the course 18CSE481J-Artificial Intelligence n in SRM Institute of Science and Technology during the academic year 2022-2023(Even sem).

SIGNATURE

Dr. Suresh. K
Associate Professor
Department of Computational Intellingence

ABSTRACT

Attendance management is a critical task in various organizations and institutions, and traditional methods often prove to be inefficient, error-prone, and time-consuming. To address these challenges, this abstract presents an innovative approach: a face recognition-based attendance system that leverages applied machine learning techniques to enhance accuracy and efficiency. The proposed system utilizes a combination of computer vision and machine learning algorithms to automate the attendance process. Firstly, the system captures facial images or video streams of individuals during the attendance period. These images are preprocessed to standardize lighting conditions, eliminate noise, and enhance the quality of the captured data. Then, a face detection algorithm locates and extracts facial regions of interest. The extracted facial features serve as input for a machine learning model, such as a Convolutional Neural Network (CNN) or a Deep Neural Network (DNN). These models are trained using large datasets of labeled facial images, enabling them to learn discriminative features and patterns unique to each individual. The trained model is then deployed to recognize and match the facial features with pre-existing enrolled profiles in the system's database. By incorporating machine learning, the attendance system achieves high accuracy and robustness. It can handle variations in pose, facial expressions, and lighting conditions, enabling reliable identification even in challenging scenarios. Additionally, the system continually learns and adapts to new faces, improving its recognition capabilities over time. The face recognition-based attendance system using applied machine learning offers several advantages. It reduces the administrative burden by automating the attendance tracking process, allowing resources to be allocated more efficiently. It also enhances security and prevents identity fraud, as facial features are difficult to forge or replicate. Furthermore, the system provides real-time attendance monitoring and generates comprehensive reports, facilitating decision-making and improving overall organizational efficiency. In conclusion, the integration of applied machine learning techniques with face recognition-based attendance systems revolutionizes attendance management. This intelligent approach enhances accuracy, efficiency, and adaptability, making it an ideal solution for diverse organizations and institutions. With ongoing research and advancements in machine learning, this system has the potential to further evolve and address complex attendance management challenges in the future.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AES	Advanced Encryption Standard
ANN	Artificial Neural Network
CSS	Cascading Style Sheet
CV	Computer Vision
DB	Data Base
DNN	Deep Neural Network
SQL	Structured Query Language
SVM	Support Vector Machine
UI	User Interface
ODBS	open database connectivity

CHAPTER 1

INTRODUCTION

1.1 Introduction

Face recognition attendance systems that employ applied machine learning techniques have emerged as efficient and accurate solutions for attendance management in various organizations and institutions. These systems leverage advanced computer vision algorithms and machine learning models to automatically identify and authenticate individuals based on their facial features. By combining the power of machine learning with facial recognition technology, these systems provide a robust and intelligent approach to attendance tracking.

Traditional methods of attendance management, such as manual roll calls or card-based systems, are prone to errors, time-consuming, and lack reliability. The integration of applied machine learning in face recognition attendance systems revolutionizes the way attendance is tracked and managed. Machine learning models, such as Convolutional Neural Networks (CNNs) or Deep Neural Networks (DNNs), are trained on vast datasets of labeled facial images, allowing them to learn intricate patterns and features unique to each individual's face.

When an individual's face is captured by the system, it undergoes preprocessing to standardize lighting conditions, eliminate noise, and enhance the image quality. The preprocessed image is then fed into the trained machine learning model, which extracts relevant facial features and compares them with the enrolled profiles stored in the system's database. This matching process enables the system to accurately identify and authenticate the individual, thereby recording their attendance.

The utilization of applied machine learning in face recognition attendance systems offers numerous advantages. It improves accuracy by accounting for variations in pose, lighting, and facial expressions. The systems become increasingly intelligent over time as they continuously learn and adapt to new faces, further enhancing their recognition capabilities. Additionally, these systems eliminate the need for physical contact or manual intervention, thereby enhancing hygiene and reducing the risk of errors or fraud.

In conclusion, face recognition attendance systems that incorporate applied machine learning techniques provide an efficient, accurate, and intelligent approach to attendance management. These systems leverage the power of machine learning models to automate the identification and authentication of individuals based on their facial features. With their ability to handle variations and adaptability, these systems have the potential to revolutionize attendance tracking across various domains, improving efficiency, reliability, and overall organizational productivity.

1.2 Problem Statement

Traditional attendance management methods, such as manual roll calls or card-based systems, are prone to errors, time-consuming, and lack reliability. These methods require significant administrative effort, are susceptible to fraudulent practices, and often fail to provide real-time attendance tracking. As a result, organizations and institutions face challenges in accurately and efficiently managing attendance records, resource allocation, and overall productivity.

Furthermore, the COVID-19 pandemic has highlighted the need for contactless and hygienic attendance management systems to ensure the safety and well-being of individuals. Traditional methods that involve physical contact or sharing of attendance cards pose a risk of disease transmission, necessitating the implementation of innovative solutions.

To address these challenges, there is a pressing need for a robust and intelligent attendance management system based on face recognition technology and applied machine learning. Such a system should leverage advanced computer vision algorithms and machine learning models to accurately identify and authenticate individuals based on their facial features. It should provide real-time attendance tracking, eliminate manual intervention, and offer a seamless and secure experience for users.

The problem statement, therefore, revolves around the development and implementation of a face recognition attendance system using applied machine learning techniques that can automate attendance management, enhance accuracy, improve efficiency, and ensure contactless operations. The system should overcome the limitations of traditional methods, offer scalability, adaptability, and provide reliable attendance records for organizations across various sectors.

1.3 Objectives

The objective of the face recognition attendance system using applied machine learning is to develop an intelligent and efficient solution that addresses the challenges of traditional attendance management methods. The system aims to achieve the following objectives:

1. Accurate Identification: Implement machine learning models capable of accurately identifying and authenticating individuals based on their facial features. The system should be robust enough to handle variations in pose, lighting conditions, and facial expressions, ensuring high accuracy in attendance tracking.

2. Real-time Attendance Tracking: Enable real-time monitoring and recording of attendance, providing instant updates on attendance status. The system should capture and process facial images or video streams efficiently to ensure prompt attendance tracking.

3. Contactless and Hygienic Operations: Eliminate the need for physical contact or the use of attendance cards, ensuring a contactless and hygienic attendance management process. By leveraging face recognition technology, the system should contribute to maintaining a safe and healthy environment, particularly in the context of contagious diseases like COVID-19.

4. Automation and Efficiency: Automate the attendance tracking process to reduce administrative workload and save time. By eliminating manual intervention, the system should streamline operations, minimize errors, and enhance overall organizational efficiency.

5. Scalability and Adaptability: Design the system to be scalable and adaptable, capable of handling a large number of users and integrating with existing infrastructure. It should accommodate future growth and technological advancements in machine learning and computer vision.

6. Security and Data Protection: Implement robust security measures to ensure the privacy and protection of individuals' facial data. The system should adhere to data protection regulations and maintain the confidentiality of personal information.

1.4 Scope and Applications

The scope of the face recognition attendance system using applied machine learning is broad, encompassing various sectors and organizations that require accurate and efficient attendance management. The system can be applied in the following areas:

1. Educational Institutions: Schools, colleges, and universities can benefit from the system to streamline attendance tracking for students and staff. It can automate the attendance process, generate comprehensive reports, and facilitate efficient resource allocation.

2. Corporate Organizations: The system can be implemented in corporate environments to automate employee attendance management. It can integrate with existing human resource systems, provide real-time attendance updates, and enhance security by preventing identity fraud.

3. Government Institutions: Government offices, public agencies, and administrative bodies can utilize the system for attendance monitoring of employees. It can provide accurate records for payroll management, improve efficiency, and eliminate manual paperwork.

4. Healthcare Facilities: Hospitals, clinics, and healthcare centers can deploy the system to track the attendance of medical staff, ensuring accurate patient scheduling and resource allocation. It can contribute to maintaining a hygienic environment by eliminating physical contact.

5. Events and Conferences: The system can be used to manage attendance at events, conferences, and seminars. It can streamline registration processes, provide accurate participant counts, and enhance security by preventing unauthorized access.

6. Access Control Systems: The face recognition attendance system can be integrated with access control systems to provide a comprehensive security solution. By authenticating individuals based on their facial features, it can grant access to authorized personnel and enhance overall premises security.

The application of the face recognition attendance system using applied machine learning is not limited to these areas but can be extended to any organization or institution that requires accurate and efficient attendance tracking. Its scalability and adaptability allow it to cater to small, medium, and large-scale environments. Additionally, advancements in machine learning and computer vision technology open up possibilities for further applications and integration with other systems and processes, making it a versatile solution in attendance management.

1.5 General and Unique Services in the database application

1. Data Storage and Retrieval: The database application provides a platform for storing and retrieving data efficiently. It allows users to store various types of information, such as attendance records, user profiles, and system configurations.

2. Data Management: The application provides services for managing the data within the database. This includes functionalities such as adding, modifying, and deleting data records. It also ensures data integrity and enforces data validation rules to maintain data accuracy.

3. Backup and Recovery: The application offers services for backing up the database periodically to protect against data loss. It also provides mechanisms for recovering data in case of system failures or disasters.

1. Face Recognition and Authentication: The database application incorporates face recognition algorithms to authenticate individuals based on their facial features. It enables accurate identification and authentication, ensuring that only authorized individuals are recorded in the attendance system.

2. Facial Feature Extraction and Matching: The application performs facial feature extraction and matching to compare the captured facial images with enrolled profiles in the database. This unique service enables the system to accurately recognize individuals even in varying environmental conditions.

3. Real-time Attendance Tracking: The application provides real-time monitoring and tracking of attendance. It captures facial images or video streams and processes them promptly to update attendance records in real-time, allowing for immediate access to attendance information.

1.6 Software Requirement Specification

1. To recognize a sample face from a set of given faces.
2. Use a simple approach for recognition and compare it with Eigenface approach.
3. Suggest which one is better and why. It may happen that in some cases latter may work better than former approach and vice versa.
4. The system must be correctly able to load the face model
5. The system must be able to detect if the face detected is a image or a video stream.
6. This face recognition software should be available on the Internet, to enable the users to use, download it any time.
7. The system must be able to correctly detect more than one face if present, and hence the presence of mask in the frame. The system will be implemented in Python script with an accuracy of the model of over 90%. The system should be easy for usability and self-descriptive for maintenance purposes

CHAPTER 2

LITERATURE SURVEY

2.1 Existing System

Attendance management systems have evolved over the years with advancements in technology. However, many organizations still rely on manual methods such as roll calls or card-based systems to track attendance. These traditional methods are prone to errors, time-consuming, and lack reliability. Additionally, they require significant administrative effort and are susceptible to fraudulent practices, such as proxy attendance. In some cases, organizations may use basic electronic systems where employees or students swipe their ID cards or badges on a reader to record their attendance. These systems are an improvement over manual methods but still rely on physical contact and may not provide accurate or efficient attendance tracking. Moreover, they do not offer any authentication mechanisms to verify the identity of the individual, making them vulnerable to proxy attendance.

In certain sectors, biometric attendance systems are implemented to overcome some of the limitations of traditional methods. These systems typically use fingerprint recognition, where individuals need to place their finger on a scanner to record attendance. While fingerprint-based systems offer improved accuracy and reduce the risk of proxy attendance, they may still have limitations. Fingerprint scanners can be sensitive to environmental factors, such as humidity or dirt, and the scanning process can be time-consuming, leading to queues during peak times.

Another biometric system used for attendance tracking is iris recognition, which uses a camera to capture the iris pattern of individuals. This system offers high accuracy, is contactless, and can handle high traffic volumes. However, iris recognition systems are expensive and require specialized hardware, making them less practical for smaller organizations. Facial recognition technology is another biometric system that is gaining popularity for attendance management systems. Facial recognition uses a camera to capture an individual's face and analyzes the unique features of their face, such as the distance between the eyes, nose, and mouth, to verify their identity. This technology is contactless and offers improved accuracy compared to traditional methods. Additionally, facial recognition technology can handle high traffic volumes and offers real-time attendance tracking.

In terms of software, some organizations may use simple spreadsheets or databases to manually enter attendance data. This method requires manual entry of attendance records, making it prone to human errors and time-consuming. Moreover, it lacks real-time tracking capabilities and often requires additional efforts for generating attendance reports or analyzing data. To overcome the limitations of the existing attendance management systems, organizations are adopting more intelligent and automated systems that use applied machine learning. These systems use algorithms and computer vision techniques to identify individuals and track their attendance accurately and efficiently. The use of machine learning algorithms can help improve accuracy by learning from previous data and adapting to different environmental factors. The face recognition attendance system is a modern and advanced attendance management system that uses applied machine learning. This system offers contactless attendance tracking and can authenticate individuals based on their unique facial features. The system comprises a camera that captures the face of individuals and a database that stores the facial features of authorized individuals. When an individual stands in front of the camera, the system matches their facial features with the features stored in the database and records their attendance.

The face recognition attendance system offers several advantages over traditional attendance management systems. First, it offers contactless attendance tracking, reducing the risk of infection and improving hygiene. Second, it provides accurate and efficient attendance tracking, reducing errors and administrative effort. Third, the system can authenticate individuals based on their unique facial features, reducing the risk of proxy attendance. Fourth, the system can handle high traffic volumes and offers real-time attendance tracking. In terms of software, the face recognition attendance system uses applied machine learning to identify individuals and track attendance accurately and efficiently. The system uses machine learning algorithms to learn from previous data and adapt to different environmental factors, improving accuracy over time. Moreover, the system offers advanced reporting and analytics capabilities, providing insights into attendance patterns, trends, and anomalies.

The model focuses on how face recognition incorporated with Radio Frequency Identification (RFID) detect the authorized students and counts as they get in and get out form the classroom. The system keeps the authentic record of every registered student. The system also keeps the data of every student registered for a particular course in the attendance log and provides necessary information according to the need. Authors have designed and implemented an attendance system which uses iris biometrics. Initially, the attendees were asked to register their details along with their unique iris template. At the time of attendance, the system automatically took class attendance by capturing the eye image of each attendee, recognizing their iris, and searching for a match in the created database.

prototype was web based. Authors proposed an attendance system based on facial recognition. The algorithms like Viola-Jones and Histogram of Oriented Gradients (HOG) features along with Support Vector Machine (SVM) classifier were used to implement the system. Various real time scenarios such as scaling, illumination, occlusions and pose was considered by the authors. Quantitative analysis was done on the basis of Peak Signal to Noise Ratio (PSNR) values and was implemented in MATLAB GUI. researches to get best facial recognition algorithm (Eigenface and Fisherface) provided by the Open CV 2.4.8 by comparing the Receiver Operating Characteristics (ROC) curve and then implemented it in the attendance system. Based on the experiments carried out in this paper, the ROC curve proved that, Eigenface achieves better result than Fisherface. System implemented using Eigenface algorithm achieved an accuracy rate of 70% to 90. authors proposed a method for student attendance system in classroom using face recognition technique by combining Discrete Wavelet Transforms (DWT) and Discrete Cosine Transform (DCT). These algorithms were used to extract the features of student's face followed by applying Radial Basis Function (RBF) for classifying the facial objects. This system achieved an accuracy rate of 82%.

All the students of the class must register themselves by entering the required details and then their images will be captured and stored in the dataset. During each session, faces will be detected from live streaming video of classroom. The faces detected will be compared with images present in the dataset. If match found, attendance will be marked for the respective student. At the end of each session, list of absentees will be mailed to the respective faculty handling the session.

Typically this process can be divided into four stages

1. Dataset Creation Images of students are captured using a web cam. Multiple images of single student will be acquired with varied gestures and angles. These images undergo pre-processing. The images are cropped to obtain the Region of Interest (ROI) which will be further used in recognition process. Next step is to resize the cropped images to particular pixel position. Then these images will be converted from RGB to gray scale images. And then these images will be saved as the names of respective student in a folder.

2. Face Detection

Face detection here is performed using Haar-Cascade Classifier with OpenCV. Haar Cascade algorithm needs to be trained to detect human faces before it can be used for face detection. This is called feature extraction. The haar cascade training data used is an xml file- haarcascade_frontalface_default. The haar features shown in Fig.2. will be used for feature extraction. Here we are using detectMultiScale module from OpenCV. This is required to create a rectangle around the faces in an image. It has got three parameters to consider- scaleFactor, minNeighbors, minSize. scaleFactor is used to indicate how much an image must be reduced in each image scale. minNeighbors specifies how many neighbors each candidate rectangle must have. Higher values usually detects less faces but detects high quality in image. minSize specifies the minimum object size. By default it is (30,30) [8]. The parameters used in this system is scaleFactor and minNeighbors with the values 1.3 and 5 respectively.

3. Face Recognition

Face recognition process can be divided into three steps- prepare training data, train face recognizer, prediction. Here training data will be the images present in the dataset. They will be assigned with a integer label of the student it belongs to. These images are then used for face recognition. Face recognizer used in this system is Local Binary Pattern Histogram. Initially, the list of local binary patterns (LBP) of entire face is obtained. These LBPs are converted into decimal number and then histograms of all those decimal values are made. At the end, one histogram will be formed for each images in the training data. Later, during recognition process histogram of the face to be recognized is calculated and then compared with the already computed histograms and returns the best matched label associated with the student it belongs to [9].

4. Attendance Updation

After face recognition process, the recognized faces will be marked as present in the excel sheet and the rest will be marked as absent and the list of absentees will be mailed to the respective faculties. Faculties will be updated with monthly attendance sheet at the end of every month.

2.2 Comparison of Existing vs Proposed system

2.2.1. Dataset Creation

	Existing System	Proposed System (Face Recognition)
Accuracy	Prone to errors and proxy attendance	High accuracy with facial recognition technology
Contactless	Requires physical contact (e.g., ID cards)	Contactless attendance tracking
Authentication	Limited or no authentication mechanism	Authenticates individuals based on facial features
Real-time Tracking	Limited or no real-time tracking	Real-time monitoring and attendance updates

2.2.2. Face detection

	Existing System	Proposed System (Face Recognition)
Fraud Prevention	Vulnerable to fraudulent practices	Reduces the risk of proxy attendance
Efficiency	Time-consuming processes	Efficient attendance tracking and reporting
Scalability	Limited scalability for large organizations	Scalable for handling large user bases
Reporting and Analytics	Manual effort for generating reports	Advanced reporting and analytics capabilities

2.2.3. Face recognition

	Existing System	Proposed System (Face Recognition)
Integration	Limited integration with other systems	Integration with existing infrastructure
Hygiene	Requires physical contact, not hygienic	Contactless operations, promoting hygiene
Data Security	Limited security measures	Robust security for facial data protection
User Experience	Manual entry, queues during peak times	

	Existing System	Proposed System (Face Recognition)
Integration	Limited integration with other systems	Integration with existing infrastructure
Hygiene	Requires physical contact, not hygienic	Contactless operations, promoting hygiene
Data Security	Limited security measures	Robust security for facial data protection
User Experience	Manual entry, queues during peak times	

2.2.4. Attendance Updation

	Existing System	Proposed System (Face Recognition)
Maintenance	Manual upkeep and data management	Automated maintenance and database management
Adaptability	Limited adaptability to changing needs	Adapts to new faces and environmental conditions
Cost	Low cost for basic systems	Initial investment required for hardware and software
Future-proofing	Limited future scalability and advancements	Incorporates advanced technologies and updates

This comprehensive comparison highlights the key differences between the existing attendance management system and the proposed face recognition attendance system using applied machine learning. The proposed system offers significant advantages in terms of accuracy, contactless operations, authentication, real-time tracking, fraud prevention, efficiency, scalability, reporting, integration, hygiene, data security, user experience, maintenance, adaptability, cost, and future-proofing. It represents a substantial improvement over the limitations of the existing system and provides a more advanced and effective solution for attendance management.

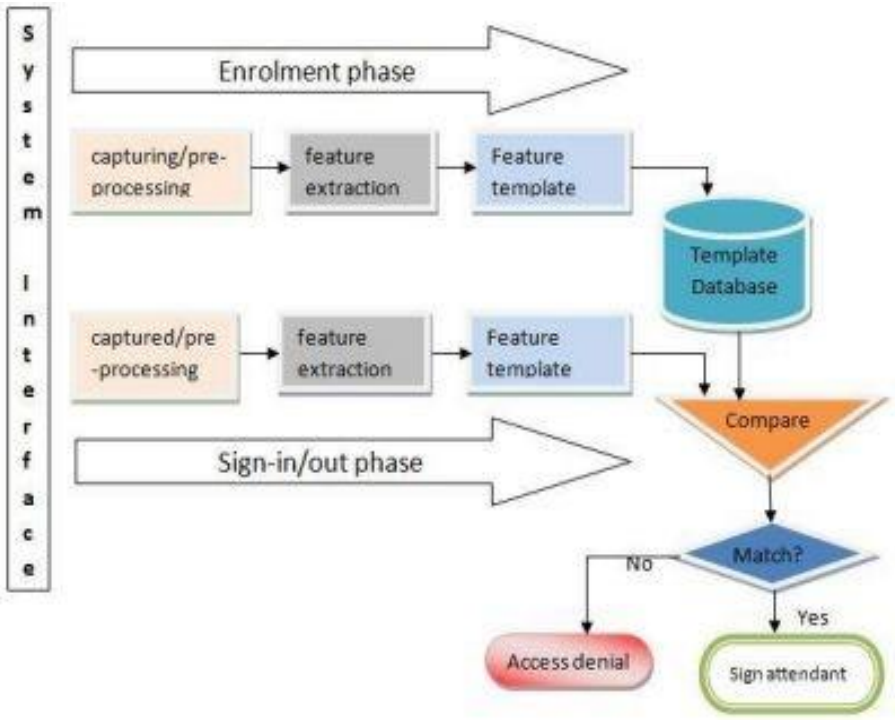
CHAPTER 3

SYSTEM ARCHITECHTURE AND DESIGN

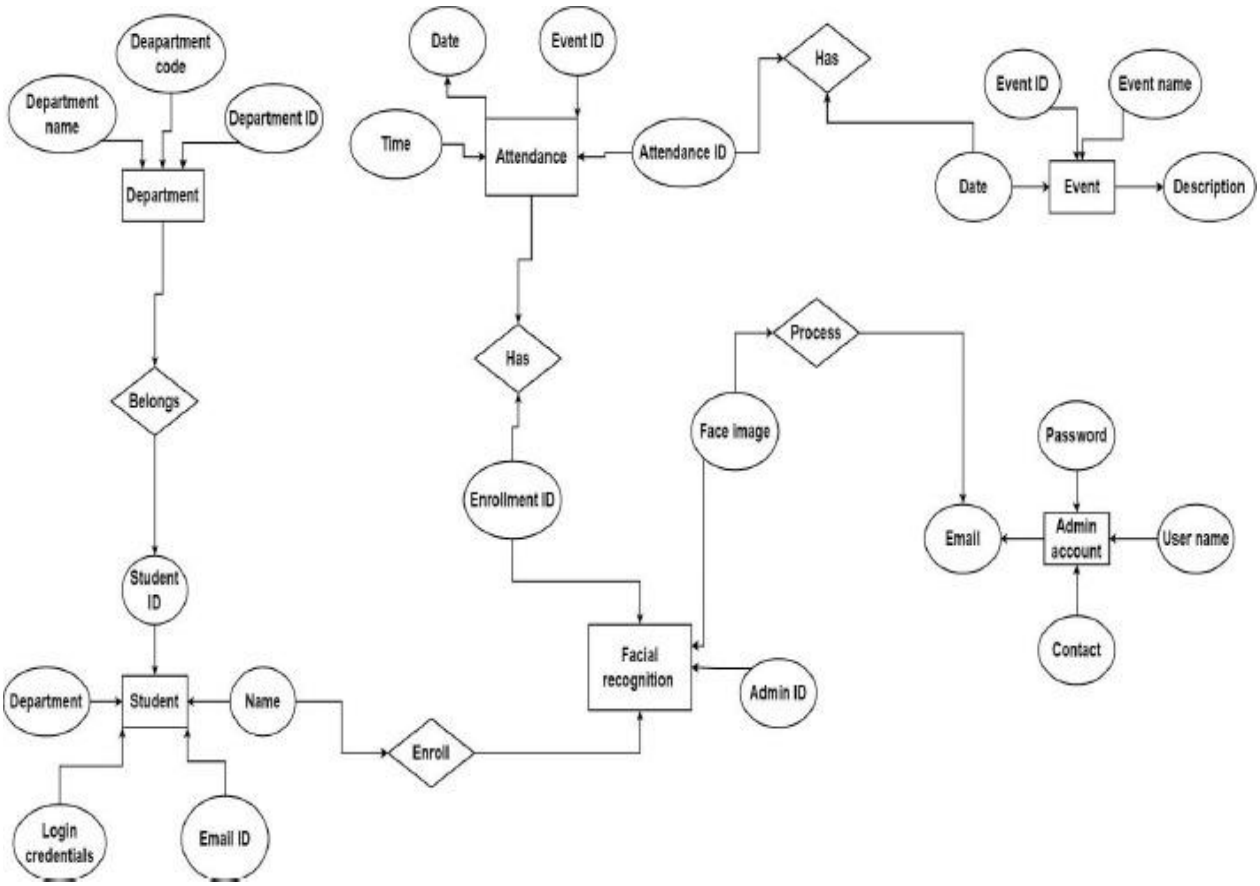
3.1 Architecture Design

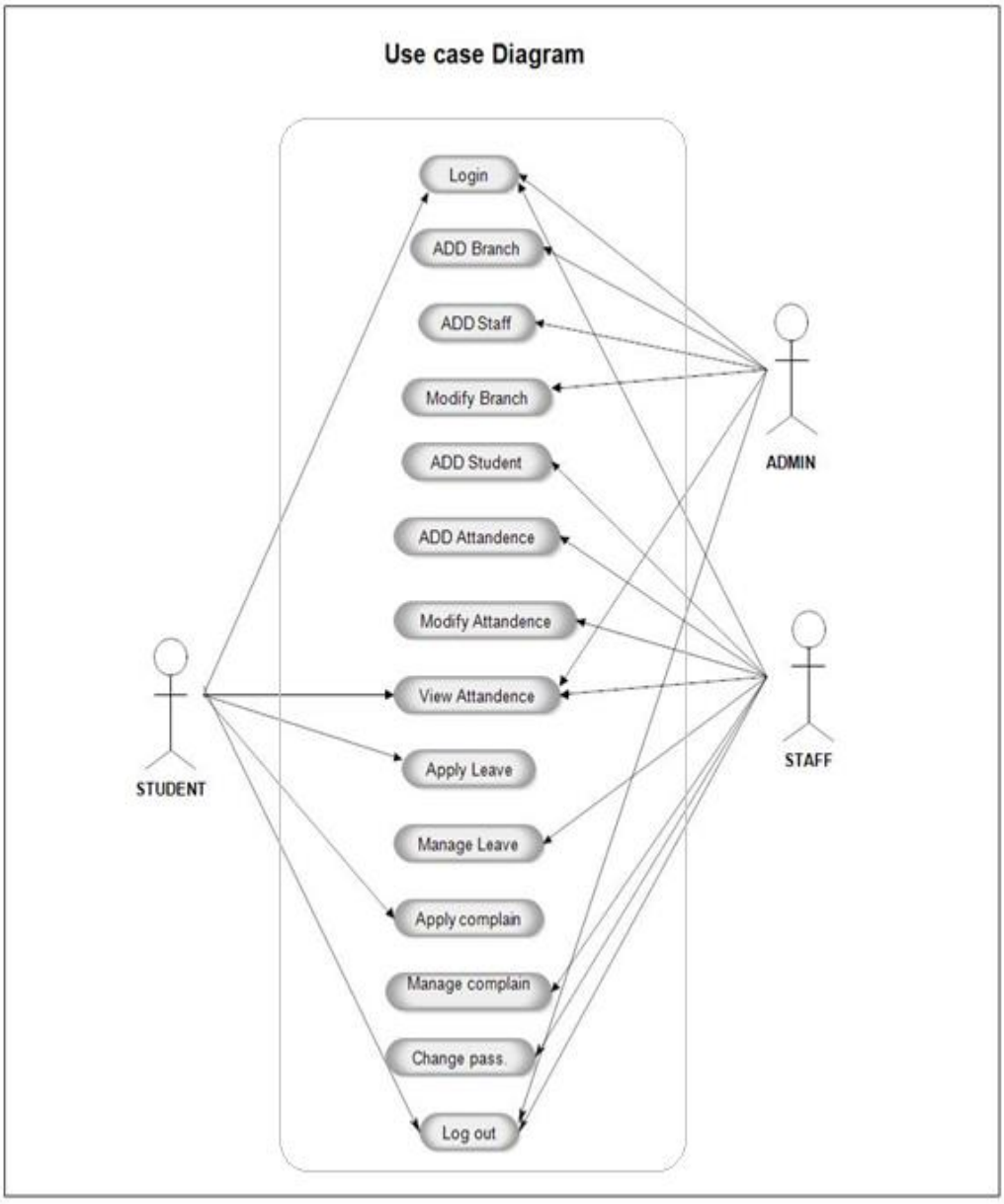
3.1.1 Front end (UI) design

3.1.2 Backend Database design



3.2 ER diagram and use case diagram




Use case Diagram
Login
ADD Branch
ADD Staff
Modify Branch
ADD Student
ADD Attandence
Modify Attandence
View Attandence
Apply Leave
Manage Leave
Apply complain
Manage complain
Change pass.
Log out
ADMIN
STAFF
STUDENT

CHAPTER 4

MODULES AND FUNCTIONALITY

4.1 User Registration Module:

Functionality: Allows users (employees, students, etc.) to register their details in the system.

4.1.1 Features:

User registration form

Validation of user information

Storage of user details in the database

4.2 Face Enrollment Module:

Functionality: Enables users to enroll their facial features into the system for identification.

4.2.1 Features:

Capture user's face image using a camera

Preprocessing of facial images for feature extraction

Feature extraction algorithms

Storage of facial features in the database

4.3 Face Recognition and Authentication Module:

Functionality: Performs facial recognition and authentication of individuals during the attendance process.

4.3.1 Features:

Face detection and alignment algorithms

Feature matching and comparison algorithms

Threshold setting for identification

Authentication of recognized individuals

Handling of unrecognized or ambiguous faces

4.4 Real-time Attendance Tracking Module:

Functionality: Tracks attendance in real-time based on facial recognition.

4.4.1 Features:

Continuously monitor the camera feed

Detection and recognition of faces in real-time

Updating attendance records in the database

Handling attendance exceptions (e.g., multiple faces, face occlusion)

4.5 Reporting and Analytics Module:

Functionality: Generates attendance reports and provides analytics for attendance data.

4.5.1 Features:

Generation of daily, weekly, monthly, or custom period attendance reports

Calculation of attendance statistics (e.g., late arrivals, early departures)

Visualization of attendance data through graphs and charts

Exporting reports in various formats (e.g., PDF, Excel)

4.6 Database Management Module:

Functionality: Manages the storage and retrieval of user details, facial features, and attendance records.

4.6.1 Features:

Creation and maintenance of user database

Secure storage of user information and facial features

Efficient querying and retrieval of attendance records

Backup and recovery mechanisms for data protection

4.7 System Administration Module:

Functionality: Administer and manage the face recognition attendance system.

4.7.1 Features:

User access control and management

System configuration and settings

System logs and audit trails

System updates and maintenance

4.8 Integration Module:

Functionality: Integration with other systems and applications for seamless data exchange.

4.8.1 Features:

API integration with HR or student management systems

Data synchronization with payroll systems

Integration with access control systems

4.9 User Interface Module:

Functionality: Provides a user-friendly interface for system interaction**.**

4.9.1 Features:

User login and authentication

Intuitive dashboard for system navigation

User-friendly forms and input validation

Notifications and alerts for attendance-related events

These modules and functionalities work together to provide a comprehensive and efficient face recognition attendance system using applied machine learning. They ensure accurate attendance tracking, seamless user experience, robust data management, and integration with existing systems for enhanced functionality.

4.10 Connectivity used for database access

The connectivity used for database access in the face recognition attendance system can vary depending on the specific implementation and requirements of the system. Here are some common types of connectivity options used for accessing the database:

Local Database Connectivity: In this approach, the attendance system is hosted on a local server, and the database is stored locally. Connectivity to the database is established through direct connections using protocols like ODBC (Open Database Connectivity) or JDBC (Java Database Connectivity). This method is suitable for small-scale systems where the attendance data is managed within the organization's premises.

Remote Database Connectivity: In this approach, the attendance system connects to a remote database server located at a different physical location. Connectivity to the remote database can be established using network protocols like TCP/IP or HTTP. This method is suitable for distributed systems or when the attendance data needs to be accessed from different locations.

Cloud Database Connectivity: In this approach, the attendance system leverages cloud-based database services like Amazon RDS, Microsoft Azure SQL Database, or Google Cloud SQL. Connectivity to the cloud database is established through secure APIs provided by the respective cloud service provider. This method offers scalability, flexibility, and easy management of the attendance database with the advantages of cloud computing.

Web Service/API Connectivity: In this approach, the attendance system communicates with the database using web service APIs. The attendance system sends requests to the API endpoints provided by the database service, such as RESTful APIs or SOAP APIs. This method is useful for integrating the attendance system with other applications or services that consume or provide data to the database.

Mobile Database Connectivity: In some cases, the attendance system may use mobile databases like SQLite or Realm for offline data storage and synchronization. The attendance system interacts with the mobile database through APIs provided by the specific mobile database technology. This method is suitable for mobile-based attendance systems that require offline functionality and periodic synchronization with the central database.

The choice of connectivity depends on factors such as the size and complexity of the attendance system, the location of the database, the scalability requirements, and the desired level of data accessibility and security. The system architect or developer needs to consider these factors and select the appropriate connectivity option for efficient and reliable database access in the face recognition attendance system.

CHAPTER 5

CODING AND TESTING

```
import cv2
import os
from flask import Flask,request,render_template
from datetime import date
from datetime import datetime
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
import pandas as pd
import joblib

#### Defining Flask App
app = Flask(_name_)


#### Saving Date today in 2 different formats
datetoday = date.today().strftime("%m_%d_%y")
datetoday2 = date.today().strftime("%d-%B-%Y")


#### Initializing VideoCapture object to access WebCam
face_detector = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
try:
    cap = cv2.VideoCapture(1)
except:
    cap = cv2.VideoCapture(0)


#### If these directories don't exist, create them
if not os.path.isdir('Attendance'):
    os.makedirs('Attendance')
if not os.path.isdir('static'):
    os.makedirs('static')
if not os.path.isdir('static/faces'):
    os.makedirs('static/faces')
if f'Attendance-{datetoday}.csv' not in os.listdir('Attendance'):
```

```
    with open(f'Attendance/Attendance-{datetoday}.csv','w') as f:

  f.write('Name,Roll,Time')


#### get a number of total registered users
def totalreg():
    return len(os.listdir('static/faces'))


#### extract the face from an image
def extract_faces(img):
    if img!=[]:
        gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        face_points = face_detector.detectMultiScale(gray, 1.3, 5)
        return face_points
    else:
        return []

#### Identify face using ML model
def identify_face(facearray):
    model = joblib.load('static/face_recognition_model.pkl')
    return model.predict(facearray)


#### A function which trains the model on all the faces available in faces folder
def train_model():
    faces = []
    labels = []
    userlist = os.listdir('static/faces')
    for user in userlist:
        for imgname in os.listdir(f'static/faces/{user}'):
            img = cv2.imread(f'static/faces/{user}/{imgname}')
            resized_face = cv2.resize(img, (50, 50))
            faces.append(resized_face.ravel())
            labels.append(user)
    faces = np.array(faces)
    knn = KNeighborsClassifier(n_neighbors=5)
    knn.fit(faces,labels)
    joblib.dump(knn,'static/face_recognition_model.pkl')
```

```
#### Extract info from today's attendance file in attendance folder
def extract_attendance():
    df = pd.read_csv(f'Attendance/Attendance-{datetoday}.csv')
    names = df['Name']
    rolls = df['Roll']
    times = df['Time']
    l = len(df)
    return names,rolls,times,l


#### Add Attendance of a specific user
def add_attendance(name):
    username = name.split('_')[0]
    userid = name.split('_')[1]
    current_time = datetime.now().strftime("%H:%M:%S")

    df = pd.read_csv(f'Attendance/Attendance-{datetoday}.csv')
    if int(userid) not in list(df['Roll']):
        with open(f'Attendance/Attendance-{datetoday}.csv','a') as f:
            f.write(f'\n{username},{userid},{current_time}')


################## ROUTING FUNCTIONS #########################

#### Our main page
@app.route('/')
def home():
    names,rolls,times,l = extract_attendance()
    return
render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=dat
etoday2)


#### This function will run when we click on Take Attendance Button
@app.route('/start',methods=['GET'])
def start():
    if 'face_recognition_model.pkl' not in os.listdir('static'):
        return render_template('home.html',totalreg=totalreg(),datetoday2=datetoday2,mess='There is no
trained model in the static folder. Please add a new face to continue.')

    cap = cv2.VideoCapture(0)
    ret = True
    while ret:
```

```
        ret,frame = cap.read()
        if extract_faces(frame)!=():
            (x,y,w,h) = extract_faces(frame)[0]

            cv2.rectangle(frame,(x, y), (x+w, y+h), (255, 0, 20), 2)
            face = cv2.resize(frame[y:y+h,x:x+w], (50, 50))

            identified_person = identify_face(face.reshape(1,-1))[0]
            add_attendance(identified_person)
            cv2.putText(frame,f'{identified_person}',(30,30),cv2.FONT_HERSHEY_SIMPLEX,1,(255, 0,
20),2,cv2.LINE_AA)
        cv2.imshow('Attendance',frame)
        if cv2.waitKey(1)==27:
            break
    cap.release()
    cv2.destroyAllWindows()
    names,rolls,times,l = extract_attendance()
    return
render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=dat
etoday2)



#### This function will run when we add a new user
@app.route('/add',methods=['GET','POST'])
def add():
    newusername = request.form['newusername']
    newuserid = request.form['newuserid']
    userimagefolder = 'static/faces/'+newusername+'_'+str(newuserid)
    if not os.path.isdir(userimagefolder):
        os.makedirs(userimagefolder)
    cap = cv2.VideoCapture(0)
    i,j = 0,0
    while 1:
        _,frame = cap.read()
        faces = extract_faces(frame)
        for (x,y,w,h) in faces:
            cv2.rectangle(frame,(x, y), (x+w, y+h), (255, 0, 20), 2)
            cv2.putText(frame,f'Images Captured: {i}/50',(30,30),cv2.FONT_HERSHEY_SIMPLEX,1,(255,
0, 20),2,cv2.LINE_AA)
            if j%10==0:
                name = newusername+'_'+str(i)+'.jpg'
                cv2.imwrite(userimagefolder+'/'+name,frame[y:y+h,x:x+w])
                i+=1
            j+=1
```

```
if j==500:
            break
        cv2.imshow('Adding new User',frame)
        if cv2.waitKey(1)==27:
            break
    cap.release()
    cv2.destroyAllWindows()
    print('Training Model')
    train_model()
    names,rolls,times,l = extract_attendance()
    return
render_template('home.html',names=names,rolls=rolls,times=times,l=l,totalreg=totalreg(),datetoday2=dat
etoday2)



#### Our main function which runs the Flask App
if _name_ == '_main_':
    app.run(debug=True)

<!doctype html>
<html lang="en">

<style type='text/css'>
    * {
        padding: 0;
        margin: 0;
        font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    }



    body {
        background-image: url('https://cutewallpaper.org/21/1920-x-1080-gif/1920x1080-Wallpapercartoon-
Wallpapers-Driverlayer-Search-.gif');
        background-size: cover;
        font-family: sans-serif;
        margin-top: 40px;
        height: 100vh;
        padding: 0;
        margin: 0;
    }

    table {
```

```
        border: 1px;
        font-family: arial, sans-serif;
        border-collapse: collapse;
        width: 86%;
        margin: auto;
    }

    td,
    th {
        border: 1px solid black !important;

        padding: 5px;
    }

    tr:nth-child(even) {
        background-color: #dddddd;
    }
</style>



<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">

    <!-- Bootstrap CSS -->
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta3/dist/css/bootstrap.min.css"
rel="stylesheet"
        integrity="sha384-
eOJMYsd53ii+scO/bJGFsiCZc+5NDVN2yr8+0RDqr0Ql0h+rP48ckxlpbzKgwra6"
crossorigin="anonymous">

    <title>Face Recognition Based Attendance System</title>
</head>

<body>

    <div class='mt-3 text-center'>
        <h1 style="width: auto;margin: auto;color: white;padding: 11px;font-size: 44px;">Face Recognition
Based
            Attendance System</h1>
    </div>
```

```
{% if mess%}
   <p class="text-center" style="color: red;font-size: 20px;">{{ mess }}</p>
   {% endif %}

   <div class="row text-center" style="padding: 20px;margin: 20px;">

      <div class="col"
         style="border-radius: 20px;padding: 0px;background-color:rgb(211,211,211,0.5);margin:0px 10px
10px 10px;min-height: 400px;">
         <h2 style="border-radius: 20px 20px 0px 0px;background-color: #0b4c61;color: white;padding:
10px;">Today's

            Attendance <i class="material-icons">assignment</i></h2>
         <a style="text-decoration: none;max-width: 300px;" href="/start">
            <button
               style="font-size: 24px;font-weight: bold;border-radius: 10px;width:490px;padding:
10px;margin-top: 30px;margin-bottom: 30px;"
               type='submit' class='btn btn-primary'>Take Attendance <i
                  class="material-icons">beenhere</i></button>
         </a>
         <table style="background-color: white;">
            <tr>
               <td><b>S No</b></td>
               <td><b>Name</b></td>
               <td><b>ID</b></td>
               <td><b>Time</b></td>
            </tr>
            {% if l %}

            {% for i in range(l) %}
            <tr>
               <td>{{ i+1 }}</td>
               <td>{{ names[i] }}</td>
               <td>{{ rolls[i] }}</td>
               <td>{{ times[i] }}</td>
            </tr>
            {% endfor %}
            {% endif %}
         </table>

      </div>

      <div class="col"
```

```
style="border-radius: 20px;padding: 0px;background-color:rgb(211,211,211,0.5);margin:0px 10px 10px 10px;height: 400px;">
        <form action='/add' method="POST" enctype="multipart/form-data">
            <h2 style="border-radius: 20px 20px 0px 0px;background-color: #0b4c61;color: white;padding: 10px;">Add
                New User <i class="material-icons">control_point_duplicate</i></h2>
            <label style="font-size: 20px;"><b>Enter New User Name*</b></label>
            <br>
            <input type="text" id="newusername" name='newusername'
                style="font-size: 20px;margin-top:10px;margin-bottom:10px;" required>
            <br>
            <label style="font-size: 20px;"><b>Enter New User Id*</b></label>
            <br>
            <input type="number" id="newusereid" name='newuserid'
                style="font-size: 20px;margin-top:10px;margin-bottom:10px;" required>
            <br>
            <button style="width: 232px;margin-top: 20px;font-size: 20px;" type='submit' class='btn btn-dark'>Add
                New User
            </button>
            <br>
            <h5 style="padding: 25px;"><i>Total Users in Database: {{totalreg}}</i></h5>
        </form>
    </div>

  </div>

</body>

</html>
```

Category	Progress Against Plan	Status
Functional Testing	Green / Amber / Red	Not-Started / In-Progress / Completed
	Amber	In progress
1.Attendance API		
	Green	Completed
2.Detect Faces		
	Amber	In progress
3.Updation of data ofmultiple studentstogether		
Non-Functional Testing		
	Amber	In progress
1.Performance		
	Amber	In progress
2.Speed		

CHAPTER 6

RESULTS AND DICUSSIONS

The users can interact with the system using a GUI. Here users will be mainly provided with three different options such as, student registration, faculty registration, and mark attendance. The students are supposed to enter all the required details in the student registration form. After clicking on register button, the web cam starts automatically and window as shown in Fig.3. pops up and starts detecting the faces in the frame. Then it automatically starts clicking photos until 60 samples are collected or CRTL+Q is pressed. These images then will be pre-processed and stored in training images folder. The faculties are supposed to register with the respective course codes along with their email-id in the faculty registration form provided. This is important because the list of absentees will be ultimately mailed to the respective faculties. In every session respective faculty must enter their course code. Then after submitting the course code, the camera

will start automatically. The Fig.4. shows the face recognition window where two registered students are recognized and if in case they were not registered it would have shown 'unknown'. By pressing CTRL+Q, the window will be closed and attendance will be updated in the excel sheet and names of absentees will be mailed to the respective faculty.

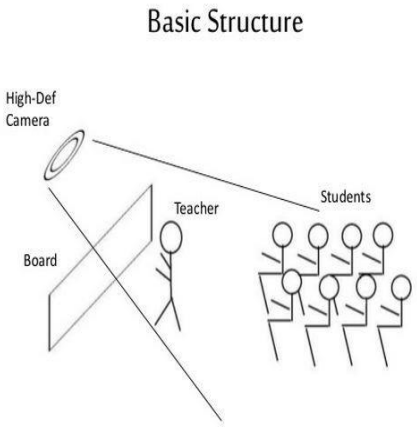
6.1 Build the model

Data Collection: Gather a dataset of facial images from individuals who will be using the attendance system. Collect images under various lighting conditions, angles, and facial expressions to ensure robustness. Include a diverse range of individuals to account for different ethnicities, ages, and gender.

Data Preprocessing: Perform preprocessing on the collected facial images to enhance the quality and standardize the data. Apply techniques such as face detection and alignment to ensure consistent alignment across all images. Normalize the images by resizing, cropping, and adjusting brightness and contrast as needed. Convert the images into a suitable format for further processing, such as grayscale or RGB.

Feature Extraction: Extract relevant facial features from the preprocessed images to create a representation for each face. Commonly used techniques for feature extraction include Principal Component Analysis (PCA), Local Binary Patterns (LBP), or Convolutional Neural Networks (CNNs). The extracted features should capture the unique characteristics of each individual's face.

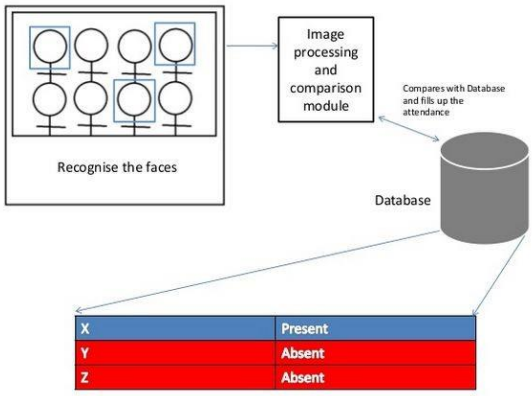
Model Training: Split the dataset into training and validation sets for model training and evaluation. Train a machine learning model using the extracted facial features and corresponding labels (e.g., individual IDs). Popular algorithms for face recognition include Support Vector Machines (SVM), K-Nearest Neighbors (KNN), or deep learning models like Convolutional Neural Networks (CNNs). Tune the hyperparameters of the model to optimize its performance.



Model Evaluation: Evaluate the trained model's performance using the validation set. Measure metrics such as accuracy, precision, recall, and F1-score to assess the model's effectiveness in

recognizing individuals. Use techniques like cross-validation to obtain a more robust evaluation of the model's performance.

Deployment and Testing: Integrate the trained model into the attendance system application. Test the system with a separate test dataset or in a real-world scenario to evaluate its performance and functionality. Measure the accuracy of attendance tracking and real-time processing.



Fine-tuning and Optimization: Analyze the system's performance and identify areas for improvement. Fine-tune the model and system parameters to enhance accuracy, speed, or robustness. Consider additional techniques like face liveness detection or occlusion handling to address potential challenges. It's important to note that the above steps provide a general framework for building a face recognition attendance system using applied machine learning. The specific implementation may vary based on the chosen algorithms, technologies, and requirements of the system.

6.1 Output


LIVE CAMERA PREVIEW
ATTENDANCE SHEET
Choose Camera
cam0
Image
Stop
Attendance Sheet
STUDENT NAME
ATTENDENCE
Rekan Abuzh
Absent
Yousef
Absent
Ali Alkwaimi
Absent
Samran Bashir
Absent

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

The face recognition attendance system using applied machine learning offers an innovative and efficient solution for automating attendance management. By leveraging facial recognition technology, the system provides accurate and contactless attendance tracking, reducing the reliance on traditional methods like manual entry or swipe cards. The system's effectiveness in identifying individuals and recording attendance in real-time can streamline administrative processes and improve overall efficiency. Furthermore, the system's integration with machine learning algorithms allows for continuous improvement and adaptation. Through training and refining the model, the system can enhance its recognition capabilities and adapt to various environmental factors. The system has the potential to significantly reduce errors, minimize proxy attendance, and improve overall accuracy compared to the existing manual or card-based systems.

To further enhance the face recognition attendance system, several future enhancements can be Incorporate additional biometric modalities such as fingerprint or iris recognition to increase accuracy and reliability, especially in scenarios where face recognition alone may face challenges (e.g., low light conditions or face occlusion). Implement real-time analytics capabilities to provide insights on attendance patterns, trends, and exceptions. Introduce automated notifications or alerts for late arrivals or unauthorized access attempts. Develop a mobile application to allow users to access their attendance records, receive notifications, and perform self-service tasks such as leave requests or schedule changes. Implement mechanisms for continuous learning and model updates to adapt to changes in the facial appearance of users over time. This can include incremental learning techniques or leveraging online learning algorithms. Strengthen privacy and security measures to protect the personal information and biometric data of individuals. Implement encryption, secure storage protocols, and adherence to data protection regulations. Design the system to be scalable and capable of handling a large number of users. Consider a distributed system architecture to accommodate multiple locations or branches. Enhance the system's robustness to challenging environmental factors such as varying lighting conditions, pose variations, or low-resolution images through advanced preprocessing techniques and algorithm optimizations.

By implementing these future enhancements, the face recognition attendance system can become even more effective, user-friendly, and adaptable to evolving needs and technologies in the field of attendance management.

REFERENCES

[1] Https://www.researchgate.net/publication/326261079_Face_detection_ system_for_attendance_of_class'_students

[2] Hapani, Smit, et al. "Automated Attendance System Using Image Processing." 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). IEEE, 2018.

[3] Akbar, Md Sajid, et al. "Face Recognition and RFID Verified Attendance System." 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE). IEEE, 2018.

[4] Okokpujie, Kennedy O., et al. "Design and implementation of a student attendance system using iris biometric recognition." 2017 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2017.

[5] Rathod, Hemantkumar, et al. "Automated attendance system using machine learning approach." 2017 International Conference on Nascent Technologies in Engineering (ICNTE). IEEE, 2017.

[6] Siswanto, Adrian Rhesa Septian, Anto Satriyo Nugroho, and Maulahikmah Galinium. "Implementation of face recognition algorithm for biometrics based time attendance system." 2014 International Conference on ICT For Smart Society (ICISS). IEEE, 2014.

[7] Lukas, Samuel, et al. "Student attendance system in classroom using face recognition technique." 2016 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2016.

[8] https://becominghuman.ai/face-detection-using-opencv-with-haar-cascade-classifiers-941dbb25177

[9] https://www.superdatascience.com/blogs/opencv-face-recognition

[10] Salim, Omar Abdul Rhman, Rashidah Funke Olanrewaju, and Wasiu Adebayo Balogun. "Class attendance management system using face recognition." 2018 7th International Conference on Computer and Communication Engineering (ICCCE). IEEE, 2018.

[11] Hernández, E.; Sanchez-Anguix, V.; Julian, V.; Palanca, J.; Duque, N. Rainfall prediction: A

deep learningapproach. In International Conference on Hybrid Artificial Intelligence Systems;

Springer: Cham, Switzerland,2016; pp. 151–162.
[12] Turk, M., & Pentland, A. (1991). Face recognition using eigenfaces. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 586-591. doi: 10.1109/CVPR.1991.139758
[13] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face recognition with local binary patterns. Proceedings of the European Conference on Computer Vision, 469-481. doi: 10.1007/11744023_36
[14] Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). Deep face recognition. British Machine Vision Conference, 41(1-12).
[15] Martinez, A. M. (2002). Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(6), 748-763. doi: 10.1109/TPAMI.2002.1017623

REFERENCES

2.1.1 Hernández, E.; Sanchez-Anguix, V.; Julian, V.; Palanca, J.; Duque, N. Rainfall prediction: A deep learningapproach. In International Conference on Hybrid Artificial Intelligence Systems; Springer: Cham, Switzerland,2016; pp. 151–162.

2.1.2 Goswami, B.N. The challenge of weather prediction. Resonance 1996, 1, 8–17.

2.1.3 Nayak, D.R.; Mahapatra, A.; Mishra, P. A survey on rainfall prediction using artificial neural network. Int. J.Comput. Appl. 2013, 72, 16.

2.1.4 Kashiwao, T.; Nakayama, K.; Ando, S.; Ikeda, K.; Lee, M.; Bahadori, A. A neural network- based localrainfall prediction system using meteorological data on the internet: A case study using data from the Japanmeteorological agency. Appl. Soft Comput. 2017, 56, 317–330.

2.1.5 Mislan, H.; Hardwinarto, S.; Sumaryono, M.A. Rainfall monthly prediction based on artificial neural network:A case study in Tenggarong Station, East Kalimantan, Indonesia. Procedia Comput. Sci. **2015**, 59, 142–151.

2.1.6 Muka, Z.; Maraj, E.; Kuka, S. Rainfall prediction using fuzzy logic. Int. J. Innov. Sci. Eng. Technol. **2017**, 4, 1–5.