FACE RECOGNITION WITH REALTIME DATABASE

A PROJECT REPORT

Submitted by

21BCS6116 – HARI LUNAVATH
21BCS6274 - T KARTHIKEYA
21BCS6323 – VIGNAN KUMAR
21BCS6003 – SHIVA REDDY

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

IN

CSE (Hons.) with Specialization in Artificial Intelligence and Machine Learning in association with IBM (CS221)

CHANDIGARH UNIVERSITY, GHARUAN, MOHALI
140413

PUNJAB

MAY, 2023

BONAFIDE CERTIFICATE

Certified that this project report **"FACE RECOGNITION WITH REALTIME DATABASE"** is the bonafide work of "**Hari Lunavath, Karthikeya, Vignan kumar, Shiva Reddy"** who carried out the project work under my/our supervision.

SIGNATURE

Mr. Aman Kaushik

HEAD OF THE DEPARTMENT

AIT-CSE

SIGNATURE

Dr Priyanka Kaushik

SUPERVISOR

Assistant Professor

AIT-CSE

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We are highly grateful to the Hon'ble Chancellor and Vice-Chancellor, Chandigarh University, Mohali, Punjab for allowing us to carry out the present project work.

The constant guidance and encouragement received from Mr. Aman Kaushik. of Apex Institute of Technology (AIT), Chandigarh University, has been of great help in carrying out our present work and is acknowledged with reverential thanks.

We would like to express a deep sense of gratitude and thanks profusely to our Project Supervisor, Dr Priyanka Kaushik, Assistant. Prof. without her able guidance, it would have been impossible to complete the project in this manner.

At last, I would like to extend my heartfelt thanks to my parents because without their help this project would not have been successful. Finally, I would like to thank my dear friends who have been with me all the time.

TABLE OF CONTENTS

List of Figures

ABSTRACT

The aim of this study is to create a face recognition system based on real-time video. This article basically establishes four guidelines for considering the issue: transmitting the cost of video time in percent to confirm participation, security of video time of real video, accuracy of the system at login, and a real-time video processing system. The effectiveness of AFR technology has increased significantly over the past few years and these systems are now used for security and commercial purposes. Student participation is an important task in the classroom. When done manually, it usually takes a lot of lesson time that can be used for learning. An auto-enrollment system using facial recognition is a suggested solution to an existing problem. The face is an important part of a person. This tutorial shows how to detect and recognize human faces in real time using Arduino UNO. This project uses the open source graphics library OpenCV to define an efficient algorithm. Face Recognition, Face Progression, Face Training, Face Recognition and Interaction Data are the five models that make up our approach. We built a face database to recognize students' faces. The student database contains the faces of all students and is initially used to train the algorithm. The system uses a user-friendly interface to maximize the user experience while collecting student images and engagement throughout training and testing. The program can also use many applications that can use facial recognition for authentication. Using Arduino UNO can reduce product cost and increase performance because it can be connected to any device for now.The system automatically sends messages to the department heads and advisors of the students who are absent, informing them about the student attendance status and updating their student records. The unique thing about our model is that it has poll signing capability; this means that

the student must be in the classroom for at least 20 or 30 minutes out of a total of 50 minutes before our signature model for students. Participating in facial recognition can reduce meeting times by up to 60%.

ABBREVIATIONS

AI	Artificial intelligence
ML	Machine learning
OpenCV	(Open Source Computer Vision Library)
DB	Database
CNN	Convolutional neural network
RAM	Random access memory
SSD	Solid-state drive
GB	Gigabyte
FER	Facial expression recognition
APPS	Applications

CHAPTER-1

INTRODUCTION

The use of face recognition system with real time database has become increasingly popular in recent years. In this project, we aim to develop a face recognition system with real time database that can accurately mark the attendance of the student if they are physically present in the classroom. This chapter introduces the to the need, importance and potential impact on the users. We also provide an overview of the project, its objectives, and the data used in the model.

1.1. Identification of Client /Need:

The educational sector is continually changing as a result of technological developments that improve students' educational experiences. Utilising face recognition technology to record attendance for pupils who are physically present in the classroom is one such technological innovation. To enhance their attendance monitoring procedure, our customer, a school, has realised the necessity for such a system. The existing manual procedure for taking attendance is time-consuming and prone to mistakes, which has an impact on the institution's overall output. This project's goal is to create a facial recognition system that can swiftly and accurately record attendance for students who are physically present in the classroom.

1.2. Identification of Problem:

The educational institution's present attendance tracking procedure entails manually documenting student attendance, which takes time and is prone to mistakes. When students attempt to falsify attendance records by using proxy attendance, which compromises the reliability of the attendance tracking system,

the manual method also presents a hurdle. Additionally, the manual method takes up valuable class time that could be used for more effective learning activities. In order to prevent proxy attendance, design a facial recognition system that can efficiently and correctly manage student attendance for those who are physically present in the classroom, and free up instructional time for more effective learning activities.

1.3. Identification of Tasks:

1.3.1 Conducting a detailed analysis of the current manual attendance tracking process to identify its inefficiencies and limitations.

1.3.2 Designing a face recognition system that can accurately and efficiently track attendance for students who are physically present in the classroom.

1.3.3 Developing the face recognition system using appropriate hardware and software tools.

1.3.4 Testing the developed system to ensure that it is accurate and efficient in marking attendance for students who are physically present in the classroom.

1.3.5 Implementing the face recognition system in the educational institution's classrooms and integrating it with the institution's existing attendance tracking process.

1.3.6 Providing training to the institution's staff members on how to use the new face recognition system and troubleshooting any issues that may arise during its use.

1.3.7 Conducting an evaluation of the implemented system to determine its effectiveness in improving the attendance tracking process and identifying any areas for improvement.

1.4. Organization of the Report:

- **Introduction:** This section serves as an introduction to the problem of sales prediction, explaining its significance and potential impact on businesses. The chapter also provides an overview of the project's objectives, as well as the data utilized in the model.
- **Literature Review/Background Study:** This chapter involves a review of existing literature and research on sales prediction models, exploring various methodologies and techniques used in the field. The chapter identifies the gaps in current research and proposes a methodology to fill those gaps.
- **Design Flow/Process:** This chapter describes the process of building the sales prediction model, including data collection and preprocessing, feature engineering, model selection, training, and testing. It also explains the metrics utilized to evaluate the model's performance and improve its accuracy.
- **Results and Discussion:** This chapter presents the results of the model and discusses the insights obtained from the analysis. It compares the performance of different models, highlights their strengths and weaknesses, and proposes recommendations for future research.
- **Conclusion:** This section summarizes the research's primary findings and discusses their implications for businesses. It reflects on the study's limitations and suggests avenues for future research.
- **References:** This section includes a list of all the sources cited in the research.

CHAPTER-2

LITERATURE SURVEY

This chapter provides a review of existing literature and research on face recognition with real time database to mark attendance. We explore different techniques and methodologies used in the field, identify the gaps in the current research, and propose our methodology to fill those gaps.

2.1. Existing solutions:

- The educational institution now tracks attendance through a manual approach that entails taking attendance with a pen and paper. This labor-intensive and error-prone approach can compromise the accuracy of attendance records. Furthermore, tracking absences for children who arrive late or depart early is challenging because these occurrences are frequently overlooked throughout the manual recording process.
- Some educational institutions have electronic attendance monitoring systems in place that monitor attendance using barcode scanning or RFID technologies. Since these systems may automatically record attendance when students scan their ID cards or RFID tags, they offer the advantage of being quicker and more precise than manual tracking. Students can still use their ID cards or RFID tags to have others scan for them, making these systems vulnerable to proxy attendance.
- The use of biometric technologies for tracking attendance, such as fingerprint or face recognition systems, has become a popular alternative in recent years. These systems have the benefit of being extremely precise and effective, and they can also get rid of the chance of proxy attendance. However, there are

ethical and legal issues that accompany the use of biometric technologies in educational settings, such as data protection laws and privacy issues.

- The project team has proposed the creation of a face recognition system that can precisely and effectively track attendance for students who are physically present in the classroom based on the shortcomings of existing alternatives. With the help of this system, the possibility of proxy attendance will be eliminated, and the educational institution's attendance tracking will be more precise and effective.

2.2. Bibliometric Analysis:

- The first article by Georghiades et al. proposes illumination cone models for face recognition under varying lighting and pose conditions. They introduce a framework that models the set of all possible images of a face that could be obtained under varying illumination conditions as a cone in a high-dimensional space. This method allows for efficient recognition of faces under variable lighting and pose conditions, with high recognition rates even when training data is limited.

- Viola and Jones present a rapid object detection approach that utilizes a boosted cascade of simple features to detect faces in images. Their method involves training a classifier on thousands of positive and negative examples of faces, which can then be used to detect faces in images in real-time.

- Yang et al. provide a survey of face detection methods, covering a range of techniques from traditional image processing methods to deep learning-based approaches. The article provides a comprehensive overview of the field, discussing the advantages and limitations of different techniques.

- Turaga et al. present a book on the machine recognition of human faces,

covering both traditional and recent approaches. The book covers different aspects of face recognition, including face detection, face alignment, feature extraction, and recognition. It provides a detailed analysis of the state-of-the-art methods and their limitations.

- Phillips et al. introduce the FERET evaluation methodology for face recognition algorithms, which has become a widely used benchmark for evaluating face recognition performance. The methodology involves evaluating recognition rates on a large dataset of faces with varying pose, lighting, and expression conditions.
- Moghaddam et al. propose a Bayesian framework for face recognition that models the appearance and variability of faces using probabilistic distributions. The method allows for efficient recognition even in cases where training data is limited.
- Zhao et al. provide a comprehensive literature survey of face recognition techniques, covering traditional methods as well as recent deep learning-based approaches. The survey covers different aspects of face recognition, including face detection, alignment, feature extraction, and recognition, and provides an analysis of the strengths and limitations of different techniques.
- Wang et al. propose a face recognition method that utilizes discriminant locality preserving projections to extract discriminative features from face images. The method allows for efficient recognition of faces even in cases where there is limited training data.
- Tan and Triggs present an approach to face recognition that utilizes enhanced local texture feature sets to handle difficult lighting conditions. The method involves extracting a set of texture features from face images and using these

features to represent faces in a high-dimensional space.

- Hu et al. propose a deep metric learning method for face verification that learns a distance metric directly from the training data. The method allows for efficient recognition of faces even in cases where there is limited training data and large variability in the appearance of faces.

2.3. Problem Definition:

With the development of the Internet, computer technology has penetrated many aspects of people's lives and jobs. Today, people are increasingly interacting with computers. Also, more and more people use computers regularly. With a strong creative spirit, one of the most challenging projects in the field offers a wide range of potential applications. Facial recognition technology has gradually entered people's lives as an important identification mark that helps people distinguish one person from another. Facial recognition is the result of combining computers and artificial intelligence. Its highly sophisticated innovations and wide applicability make it one of the most complex subjects in its field. The current attendance system is manual. This loses a lot of time for both teachers and students. Students may have to wait longer if attendance is recorded manually. Even if attendance is manually recorded, trusted individuals can still attend classes. Human error is an ongoing cost of manual visits. Everyone is recognizable only by their faces. Therefore, automating the attendance process increases class productivity. I chose Arduino UNO 3 for face recognition so that it can be used on all platforms. The webcam connects to the Arduino UNO module. Face identification differentiates between recognized and unrecognized faces. This module can be used in a variety of applications, including Face recognition can be used to validate. With the suggested approach, we collect

attendance by employing face recognition, which can identify each student's face while they are in class. The study revealed that the existing fingerprint attendance system, which it used as an example, has an error rate of roughly 5% and that there would be a phenomena where fingerprints cannot be struck, which has a significant impact on attendance efficiency, particularly in big attendance locations. Face recognition technology is being used in the classroom to assist reduce the number of mischievous kids who use technology as a means of proxy, such as scanning their fingers to skip class.

2.4. Goals/Objectives:

- To create and implement a facial recognition system that can swiftly and correctly track students' attendance who are physically present in the classroom.
- To switch over to a more accurate, dependable, and error-free automated system from the educational institution's current manual approach for recording attendance.
- To enhance the tracking of attendance and lessen instances of proxy attendance, which can harm students' academic performance.
- Ensuring that the face recognition system is secure, shielded from unauthorised access, and in compliance with all data protection laws and privacy regulations.
- To evaluate the face recognition system's performance, usability, and user satisfaction during a pilot test in a classroom context.
- Depending on input from the pilot test, to improve and optimise the facial

recognition system and to get it ready for use in additional classrooms and educational settings.

CHAPTER-3

DESIGN FLOW/PROCESS

In this chapter, we describe the process of building the facial recognition with real time database, including data collection and preprocessing, feature engineering, model selection, training, and testing.

3.1 Design Flow:

The following is a summary of the design flow for the Face Recognition with real time database for attendance marking :

- My model is same as all other face recognition models the special feature in my model makes different from them that is time quantum means it will take 3 times attendance in one class after certain time quantum means at starting it will take mark attendance and after 10 minutes again it will mark attendance and again after 10 minutes it will repeat the same process so by this method it will make sure that every student is present in the class and it will reduce the proxy
- Face Locations: The coordinates or bounding box around a face in an image or video frame are referred to as face locations. The location can be represented as a rectangle, with the width and height being the dimensions of the box, and the top left corner (x,y) being the location of the bounding box's upper-left corner. Identifying and tracking faces in real-time video streams, as well as aligning and cropping face images to extract facial features for recognition, depend on accurate face locations.
- Face Distance: Face distance is the separation between an individual's facial features, such as their eyes, nose, and mouth. This distance is crucial in identifying the distinctive features of a person's face and is used to generate a template of the face for identification. The face

template, which can be used to identify a specific person, is a mathematical representation of the distances and angles between important facial features.

- Face Landmarks: In order to recognize distinctive facial features like the eyes, nose, mouth, and jawline, landmarks are particular locations on the face. Facial landmark detection algorithms, which can accurately identify important facial landmarks, are often used to find these landmarks. These are the features I selected for the face recognition I know there are much features but I choose few of them to make it simple.
- Realtime Database: Face recognition systems may store and retrieve data about recognized faces using a number of database systems through real-time database connectivity. The names, addresses, and other personal information of people detected by the system can be stored in these databases. In order to keep the database current and accurate, real-time database integration can also allow the system to update and amend this data as necessary.

  NOTE: - so make this work we are using Firebase Database which is used to help with the real-time database.
- Firebase Realtime Database: - Google Firebase offers the Firebase Realtime Database, a cloudhosted NoSQL database that enables developers to store and sync data in real-time between clients. It is a service that is hosted in the cloud and can serve as the backend for both online and mobile applications

3.2 Design selection:

Designing a face recognition system with real-time database for attendance involves the following steps:

- **Face detection:** The first step is to detect faces in the input image or video feed. This can be done using various algorithms such as Haar Cascade, HOG, or deep learning-based approaches such as CNNs.
- **Face alignment:** After detecting faces, the next step is to align them so that they are in a consistent orientation and scale. This is important because it improves the accuracy of the face recognition algorithm. Common methods for face alignment include using landmarks or feature points on the face
- **Feature extraction:** Once the faces are aligned, the next step is to extract features that are unique to each face. This can be done using various algorithms such as Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG), or deep learning-based approaches such as Convolutional Neural Networks (CNNs).
- **Face recognition:** The final step is to compare the extracted features of each face to a database of known faces to determine the identity of each person. This can be done using various algorithms such as k-nearest neighbors, support vector machines, or deep learning-based approaches such as Siamese Networks or FaceNet.
- **Real-time database:** In order to track attendance, the system needs to be integrated with a realtime database. Each time a face is recognized, the system should update the database with the name and time of the person's arrival.

3.3 Code:

- **Face Recognition :**

```
import cv2
import pickle
import os
import cvzone
import face_recognition
from datetime import datetime
import numpy as np
import firebase_admin
from firebase_admin import credentials
from firebase_admin import db
from firebase_admin import storage

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL':"https://attendance-36fe5-default-rtdb.firebaseio.com/",
    'storageBucket':"attendance-36fe5.appspot.com"
})

bucket=storage.bucket()

cap = cv2.VideoCapture(0)
cap.set(3, 1280) #width
cap.set(4, 720) #height
```

```
imgBackground = cv2.imread('resources/background.png')
imgBackground = cv2.resize(imgBackground, (1490, 1065)) # resize background image to match resized_img dimensions

folderModePath = 'Resources/Modes'
modePathList = os.listdir(folderModePath)
imgModeList = []
for path in modePathList:
    imgModeList.append(cv2.imread(os.path.join(folderModePath, path)))

#load the encoding file
print("loading encoding file....")
file=open('EncodeFile.p','rb')
encodeListKnownWithIds = pickle.load(file)
file.close()
encodeListKnown, studentIds = encodeListKnownWithIds
#print(studentIds)
print("loaded encoded file")

modeType=0
counter=0
id=-1
imgStudent=[]
```

```
while True:
    success , img = cap.read()

    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

    faceCurFrame = face_recognition.face_locations(imgS)
    encodeCurFrame = face_recognition.face_encodings(imgS, faceCurFrame)

    resized_img = cv2.resize(img, (1625, 1625))
    resized_img = cv2.resize(resized_img, (799, 930))  # reduce size further to
match imgBackground
    imgBackground[99:99 + 930, 37:37 + 799] = resized_img

    # resize imgModeList[0] to match target region
    imgMode = cv2.resize(imgModeList[modeType], (610, 988))
    imgBackground[38:38 + 988, 858:858 + 610] = imgMode

    if faceCurFrame:
        for encodeFace, faceLoc in zip(encodeCurFrame,faceCurFrame):
            matches=face_recognition.compare_faces(encodeListKnown,
encodeFace)
            faceDis=face_recognition.face_distance(encodeListKnown,
encodeFace)
```

```
    #print("matches",matches)
    #print("distance",faceDis)

    matchIndex=np.argmin(faceDis)
    #print("matchIndex",matchIndex)

    if matches[matchIndex]:
        print("known face is detected")
        print(studentIds[matchIndex])
        y1,x2,y2,x1=faceLoc
        y1,x2,y2,x1=y1*4,x2*4,y2*4,x1*4
        bbox = 37+x1, 99+y1, x2-x1, y2-y1
        imgBackground=cvzone.cornerRect(imgBackground,bbox,rt=0)
        id=studentIds[matchIndex]
        print(id)
        if counter==0:
            cvzone.putTextRect(imgBackground, "Loading", (150, 150))
            cv2.imshow("face attendance",imgBackground)
            cv2.waitKey(1)
            counter=1
            modeType=2

if counter!=0:
```

```
        if counter==1:
            #get the data
            studentinfo=db.reference(f'Students/{id}').get()
            print(studentinfo)
            #get the image from the storage
            blob=bucket.get_blob(f'Images/{id}.png')
            array=np.frombuffer(blob.download_as_string(),np.uint8)
            imgStudent=cv2.imdecode(array,cv2.COLOR_BGRA2BGR)
            #update the data from attendance
            datetimeobject =
datetime.strptime(studentinfo['last_attendance_time'], "%Y-%m-%d
%H:%M:%S")
            secondsElapsed = (datetime.now() - datetimeobject).total_seconds()
            print(secondsElapsed)
            if secondsElapsed > 30:
                ref=db.reference(f'Students/{id}')
                # Convert the integer value to a string before concatenating
                studentinfo['total_attendance'] +=1
                ref.child('total_attendance').set(studentinfo['total_attendance'])

ref.child('last_attendance_time').set(datetime.now().strftime("%Y-%m-%d
%H:%M:%S"))
            else:
                modeType=1
                counter=0
```

```
            imgMode = cv2.resize(imgModeList[modeType], (610, 988))
            imgBackground[38:38 + 988, 858:858 + 610] = imgMode

        if modeType !=3:

            if 10<counter<20:
                modeType=3
                imgMode = cv2.resize(imgModeList[modeType], (610, 988))
                imgBackground[38:38 + 988, 858:858 + 610] = imgMode



            if counter<=10:

cv2.putText(imgBackground,str(studentinfo['total_attendance']),(991,145),cv
2.FONT_HERSHEY_COMPLEX,2,(50,50,50),3)

cv2.putText(imgBackground,str(studentinfo['name']),(1020,650),cv2.FONT_
HERSHEY_COMPLEX,1,(10,10,10),2)

cv2.putText(imgBackground,str(id),(1011,775),cv2.FONT_HERSHEY_CO
MPLEX,1,(50,50,50),2)

            # center the resized image in the background
            imgStudent_resized = cv2.resize(imgStudent, (310, 310))
            imgBackground[220:220 + imgStudent_resized.shape[0],
```

```
1020:1020 + imgStudent_resized.shape[1]] = imgStudent_resized

            counter+=1


        if counter>=20:
            counter=0
            modeType=0
            studentinfo=[]
            imgStudent=[]
            imgMode = cv2.resize(imgModeList[modeType], (610, 988))
            imgBackground[38:38 + 988, 858:858 + 610] = imgMode
    else:
        modeType=0
        counter=0


    cv2.imshow("Face Attendance" , imgBackground)
    cv2.waitKey(1)
```

3.3.2 Encodings:

```
 import cv2
import face_recognition
import pickle
import os
import firebase_admin
from firebase_admin import credentials
```

```
from firebase_admin import db
from firebase_admin import storage

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://attendance-36fe5-default-rtdb.firebaseio.com/",
    'storageBucket': "attendance-36fe5.appspot.com"
})

# Importing student images
folderPath = 'Images'
pathList = os.listdir(folderPath)
print(pathList)
imgList = []
studentIds = []
for path in pathList:
    imgList.append(cv2.imread(os.path.join(folderPath, path)))
    studentIds.append(os.path.splitext(path)[0])

    fileName = f'{folderPath}/{path}'
    bucket = storage.bucket()
    blob = bucket.blob(fileName)
    blob.upload_from_filename(fileName)
```

```
    # print(path)
    # print(os.path.splitext(path)[0])
print(studentIds)



def findEncodings(imagesList):
    encodeList = []
    for img in imagesList:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)

    return encodeList



print("Encoding Started ...")
encodeListKnown = findEncodings(imgList)
encodeListKnownWithIds = [encodeListKnown, studentIds]
print("Encoding Complete")

file = open("EncodeFile.p", 'wb')
pickle.dump(encodeListKnownWithIds, file)
file.close()
print("File Saved")
```

3.3.3 Database:

```
 import firebase_admin
from firebase_admin import credentials
from firebase_admin import db

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://attendance-36fe5-default-rtdb.firebaseio.com/"
})

ref = db.reference('Students')

data = {
    "21BCS6299":
        {
            "name": "MS.Dhoni",
            "major": "Robotics",
            "starting_year": 2017,
            "total_attendance": 7,
            "standing": "G",
            "year": 4,
            "last_attendance_time": "2022-12-11 00:54:34"
        },
    "21BCS6286":
```

```
    {
        "name": "Ratan Tata",
        "major": "Economics",
        "starting_year": 2021,
        "total_attendance": 12,
        "standing": "B",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34"
    },
"21BCS6274":
    {
        "name": "Elon Musk",
        "major": "Physics",
        "starting_year": 2020,
        "total_attendance": 7,
        "standing": "G",
        "year": 2,
        "last_attendance_time": "2022-12-11 00:54:34"
    },
"21BCS6276":
    {
        "name": "Karthikeya",
        "major": "Machine Learning",
        "starting_year": 2020,
```

```
        "total_attendance": 7,
        "standing": "G",
        "year": 2,
        "last_attendance_time": "2022-12-11 00:54:34"
    }
}

for key, value in data.items():
    ref.child(key).set(value)
```

OVERVIEW OF THE CODE:

Here's an overview of the script:

- Import the required libraries and modules.
- Initialize Firebase credentials and setup the database and storage.
- Create a video capture object and set the resolution.
- Load the background image and resize it.
- Load the mode images for different stages of the attendance process.
- Load the encoding file containing known face encodings and corresponding student IDs.
- Set up variables for mode type, counter, and student ID.
- Start a while loop to continuously process frames from the video stream.
- Read a frame from the video capture object.

- Resize the frame and convert it to RGB color space.
- Detect face locations and encodings in the current frame.
- Resize the frame to match the dimensions of the background image.
- Update the background image with the resized frame and the appropriate mode image.
- If faces are detected, compare the encodings with the known encodings.
- If a match is found, retrieve the corresponding student ID and draw a rectangle around the face in the background image.
- Set the student ID as the current ID and change the mode type.
- If the counter is non-zero, perform the required actions based on the mode type.
- Retrieve student information and image from the Firebase database and storage.
- Update the attendance data if the time difference is greater than 30 seconds.
- Change the mode type, counter, and update the background image accordingly.
- If the counter exceeds 20, reset the counter, mode type, student information, and image.
- If no faces are detected, reset the mode type and counter.
- Display the final background image.
- Exit the loop if a key is pressed.
- Please note that this script assumes the presence of a "serviceAccountKey.json" file containing Firebase service account credentials and other necessary files and resources in the specified paths.

TECHNIQUES :

- The makes use of a face recognition method, which is a branch of pattern and computer vision. Although face recognition itself is not an algorithm for machine learning, it frequently leverages these methods for training and classification.

- Using the face_recognition library, which is based on machine learning algorithms, the face recognition capability is accomplished in this particular code. The library uses the Convolutional Neural Network (CNN) neural network architecture to extract and learn features from facial photos, and then compares and recognizes faces using these learned features.

- A pre-trained CNN model that has been trained on a sizable face dataset is used by the face recognition package. The model develops the ability to extract discriminative elements from facial images during training, enabling it to distinguish between various people. Then, encodings of known faces are computed using this trained model, and they are compared against encodings of faces discovered in real-time.

- The code relies on a pre-trained model to carry out the recognition task even if it does not explicitly train the facial recognition model. Convolutional Neural Networks (CNNs) are a machine learning technique for facial recognition that underlies this code.

- supervised learning is the foundation for the facial recognition capability in the code you provided. In supervised learning, each input (in this case, face photos) is connected with a corresponding goal or label (in this case, student

IDs), and the algorithm learns from a labelled dataset.

- The programmer makes use of a face recognition model that has already been trained using a sizable dataset of labelled face photos. In this training procedure, the model is given pairs of face photos together with the labels that go with them. In order to recognize and categorize new faces based on their resemblance to the recognized faces in the training data, the model learns to map the input face images to their corresponding labels.

- The code specifically loads an encoding file with known face encodings and their corresponding student IDs. These encodings serve as representations of the learnt features that were taken during training from the face images. The code determines the closest match based on a distance measure by comparing the encodings of the detected faces with the known encodings during the recognition phase. In order to do supervised classification, this technique needs labelled training data (known face encodings and student IDs).

- It's important to note that while the facial recognition portion of the code is based on supervised learning, other portions of the code or the entire system may employ unsupervised or semi-supervised approaches or incorporate alternative forms of learning. However, the facial recognition part can be categorized as supervised learning based on the provided code sample.

UNDERLYING FEATURES:

- The underlying neural network model automatically picks up the features needed for face recognition during the training process. The face recognition capability in the code you provide relies on the face recognition library, which uses a convolutional neural network (CNN) model that has already been trained to extract features from facial images.

- The face recognition library's CNN model is often trained on a sizable dataset of face photos. The model gains the ability to recognize and extract discriminative elements from the input photographs that are helpful in telling one person from another. The distinctive traits of each face, including the form of the eyes, nose, and mouth as well as the overall facial structure, are captured by these learnt features.

- The face recognition library computes face encodings, which are numerical representations of the facial features derived from the input photos, using these learned features. These encodings serve as a concise representation of the identification of each face by capturing the individual characteristics of each face.

- The face recognition library in the code determines the face encodings for both the known faces (retrieved from the loaded encoding file) and the faces actually observed. The similarity or separation between the faces is then calculated by comparing the computed encodings. The code can identify recognised faces and link them to their corresponding student IDs by examining the retrieved features and their similarity.

- It's crucial to remember that the precise information contained in the features that the CNN model used for facial recognition learned may be complex and challenging to directly comprehend. The model can recognise complex patterns and changes in facial photos because it learns a hierarchy of attributes over numerous layers.

SUMMARIZING FIREBASE DATABASE :

- A popular platform for developing mobile and online applications, Firebase provides a full range of services for creating and managing apps. The Firebase Realtime Database, a NoSQL cloud-hosted database that enables developers to store and synchronise data in real time, is one of its key components.
- Key characteristics and distinctive elements of Firebase Realtime Database include:

  In-the-moment synchronisation Real-time synchronisation is offered by Firebase Realtime Database, which means that any changes made to the database are immediately propagated to all connected clients. This makes it perfect for applications that need collaborative features and real-time updates.
- NoSQL data model: Firebase Realtime Database uses a NoSQL data model, in which information is saved in "documents" that resemble JSON-like data structures. Developers can store and retrieve data using its flexible structure without having to pre-define inflexible database models.
- Firebase Realtime Database has built-in offline functionality, which is available. Users can access and edit data even when they are offline

thanks to the client device's caching of that information.

- Changes are instantly synchronised with the server once the connection has been restored.
- Automatic scalability and load balancing are handled by Firebase Realtime Database, allowing your app to handle high traffic and sizable user bases without having to worry about infrastructure administration.
- Safety guidelines: Developers can set precise access controls to their data using Firebase Realtime Database's security rules. This helps prevent unauthorised access by ensuring that only authorised users can read or alter the data.
- Including other Firebase services in the integration: Firebase Authentication, Firebase Cloud Messaging, Firebase Hosting, and other Firebase services are all seamlessly integrated with Firebase Realtime Database. By streamlining the development process, this integration makes it easier to create reliable apps with less effort.
- Serverless architecture: Firebase Realtime Database is a serverless solution, so developers don't have to take care of scaling servers or managing server infrastructure. This frees developers from having to manage backend infrastructure so they can concentrate on creating their applications.
- Support for multiple platforms and programming languages, including iOS, Android, JavaScript, Unity, and more, is offered by Firebase Realtime Database's SDKs. It is accessible to developers using various platforms thanks to this extensive support.
- Analytics and monitoring: Firebase provides integrated analytics and monitoring tools that let developers analyse user engagement, get an

understanding of how their app is being used, and keep an eye on performance indicators.

- Firebase Cloud Functions: Firebase Realtime Database may be enhanced with the help of Firebase Cloud Functions, which let you add server-side logic and triggers to react to database events or carry out specialised business logic.
- Firebase Database is built on a distributed infrastructure that expands automatically to handle enormous volumes of data and concurrent user connections. Scalability and Performance. This guarantees good performance, low latency, and high availability for applications with diverse traffic patterns.
- Essential Elements of Firebase Database: 1.1 Data may be updated in real-time across all connected devices thanks to Firebase Database's real-time synchronisation. Real-time collaboration and seamless user experiences are made possible by the instantaneous propagation of database updates to all clients.
- JSON-like Structure: Firebase Database uses a JSON-like structure to model and structure its data. It is made up of nested key-value pairs, where the keys are strings and the values can be objects, Booleans, strings, numbers, or arrays. Using this adaptable structure, developers may visualize intricate data interactions.
- Documents and Collections: The Firebase Database's data is divided into documents and collections. Each document is made up of a set of key-value pairs, whereas a collection is a group of connected documents. Specific data can be easily retrieved and modified thanks to developers' ability to run queries and modifications at the document level.

- Data is automatically indexed by Firebase Database to improve query efficiency. Data can be accessed by developers using a variety of criteria, including equality, inequality, range, and sorting. These queries make it possible to retrieve data quickly for display, for purposes of analysis and filtering.
- Firebase Ecosystem Integration: 3.1 Firebase Authentication: Developers can authenticate users and secure access to the database thanks to Firebase Database's easy integration with Firebase Authentication. Tokens for user authentication can be used to impose data access restrictions and guarantee data privacy.
- Firebase Cloud Functions: Based on database events, Firebase Database can start Firebase Cloud Functions. In order to enable customized business logic and server-side processing, developers can create serverless functions that are automatically invoked when particular data changes occur.
- processing. Firebase Database can be used in conjunction with Firebase Hosting and Firebase Storage to create full-stack apps. While storage enables effective archiving and retrieval of user-generated information like photographs, movies, and files, hosting offers a web server for static data.
- Actual Use Cases: Building real-time collaborative applications like chat apps, group document editing, and live dashboards is excellent with Firebase Database. Updates from numerous users are quickly mirrored across all clients because to its real-time synchronization.
- Data synchronization across platforms: Using Firebase Database, developers may create software that works with desktop, mobile, and

web platforms.

- Users may now access their data from any device without having to bother about manual synchronizing thanks to this.
- Personalization and recommendations: Developers can save and retrieve user preferences, activity logs, and behavioral information using Firebase Database. Utilizing this information can enhance user engagement, deliver personalized experiences, and propose pertinent material.
- For managing real-time data in online and mobile apps, Firebase Database gives developers a strong and scalable option. It is a popular option for developing contemporary applications due to its real-time synchronization, NoSQL data structure, offline capability, and easy connection with the Firebase ecosystem. Developers may build powerful, scalable, and engaging applications that delight users and promote business success by utilizing the capabilities of Firebase Database.

HARD WARE COMPONENTS AND ITS WORKING:

- Systems for managing attendance are essential in many settings, including schools, universities, and workplaces. Innovative technological alternatives are displacing outdated paper-based techniques. The hardware will be examined in this article.
- needed elements and distinguishing characteristics to build a Firebase-based real-time attendance marking system.
- Hardware elements an essential piece of gear for attendance recording systems is the biometric reader. It records distinctive biometric

characteristics like iris patterns, face features, or fingerprints. Biometric readers guarantee precise identification and eliminate any chance of proxy attendance.

- The brain of the system for recording attendance is a microcontroller. It handles data processing operations and connects numerous hardware pieces. Due to their adaptability and simplicity, microcontrollers like Arduino and Raspberry Pi are frequently employed integration.
- Communication Module: A communication module makes it easier for the Firebase database and the attendance system to interact. It can be a Wi-Fi module, Ethernet module, or GSM module that allows for real-time data synchronization.
- Display Unit: During the attendance marking process, a display unit is used to give consumers feedback. It can display pertinent information like attendance status or instructions on an LCD screen, LED matrix, or even a straightforward OLED display.
- Database integration with Firebase: 2.1
- Firebase Realtime Database: Google offers Firebase Realtime Database, a NoSQL database hosted in the cloud. It is the best option for attendance management because it provides real-time synchronization and data storage.
- systems. The Firebase SDKs, which offer reliable APIs for data retrieval, storage, and synchronization, can be used to access the database.
- Data Organization: An appropriate data structure needs to be established in order to integrate the hardware parts with the Firebase database. For instance, Firebase can keep a unique identification for each student or employee as a key and use attendance data as the value.
- Timestamps, status (present or absent), and any other necessary information

can be included in these entries.

- Instantaneous updates are possible thanks to Firebase's real-time synchronization. An event that updates the related user's attendance record in Firebase can be started when a biometric reader records attendance information. This guarantees that
- Administrators and stakeholders can see attendance data right away.
- System Workflow: 3.1 Enrollment: Individuals are registered into the attendance system by the system administrator, who collects their biometric information and links it to their individual Firebase identifier.
- Attendance Marking: When a person approaches the biometric reader, it records their biometric information and compares it to the templates that have been stored. When a match is made, the attendance system updates the related user's Firebase record with the most recent information date and status of attendance.
- Real-Time Reporting: Using Firebase's web or mobile interfaces, authorized personnel can access the attendance records in real-time. They can produce reports, evaluate attendance patterns, and analyses the information to help them make decisions.

COST FACTOR:

- A real-time attendance marking system using Firebase Database can have different costs depending on a number of variables. Several crucial cost factors are listed below:
- Firebase Costs: A free tier with constrained resources and usage allowances is provided by Firebase.
- However, you might need to take into account one of Firebase's

commercial pricing plans for a production-grade attendance system. The pricing scheme is determined by a number of elements, such as the volume of database reads, writes, and data storage. It's crucial to read through Firebase's price information in order to comprehend the precise charges related to the usage patterns of your system.

- Data Storage: The total cost will depend on how much data storage your attendance system needs. Firebase Database offers a sizable amount of free storage, however if your system creates a lot of attendance data or if you keep more users' data than is necessary, you may need to upgrade.
- information, you might need to think about the charges related to going over the allotted free storage.
- Network Bandwidth: Firebase charges you according on the amount of network bandwidth your application uses. The Firebase Database and the client devices frequently synchronize their data in real-time attendance marking systems. Your network bandwidth expenses may be impacted by the amount of data transferred. To accurately evaluate the associated bandwidth costs, the frequency and magnitude of data updates must be taken into account.
- Integration of Other Firebase Services: The expenses of integrating additional Firebase services, such as Firebase Authentication for user management or Firebase Cloud Functions for serverless computing, should also be taken into account. Each
- Each service may have a different pricing plan based on consumption.
- Even though they are not directly related to Firebase's pricing, development and maintenance expenditures for your attendance system should be taken into account.

- These expenses cover the time and materials needed for planning, implementing, and
-
- managing the hardware parts, user interfaces, and Firebase Database integration. You should also take ongoing system upkeep, bug repairs, and feature upgrades into account when determining your overall cost priorities.
- To assess the potential expenses associated with using Firebase Database for your real-time attendance marking system, it is necessary to carefully analyses your system requirements, anticipated usage patterns, estimated data storage demands, and network bandwidth needs.
- You can produce a more precise cost estimate by looking at Firebase's price information and discussing with your development team or financial advisor.

Title: Understanding Cost Factors for a Real-Time Attendance Marking System with Firebase Database

- Introduction: There are several advantages of using Firebase Database to implement a real-time attendance marking system, but it's important to take into account the costs involved. You can properly plan and manage your budget by being aware of the factors that go into the final price. This in-depth investigation will look at the special costs associated with developing and running a real-time attendance marking system using Firebase Database.
- The pricing model for Firebase: Firebase has a thorough pricing structure that considers numerous consumption factors. To accurately estimate the cost of your attendance marking system, it is imperative

that you become familiar with the precise pricing information provided by Firebase. The following are crucial factors to remember:

- Database usage: Firebase charges for reads, writes, and deletes in databases. The frequency and volume of these actions will affect the price as attendance records are created, changed, and accessed in real-time. To effectively predict database consumption and associated costs, analyses your anticipated system usage trends.
- Data Storage: Your attendance marking system's data storage requirements play a big role in the price. Firebase offers a free storage cap, but if your system produces a lot of attendance records or contains more data, like user profiles, you might need to think about the fees involved with going over the cap.
- Firebase charges you for the network bandwidth that your application uses.
- The Firebase Database and client devices must frequently sync their data in real-time attendance marking systems. The cost of network bandwidth is directly impacted by the volume and frequency of data updates. To accurately calculate the associated costs, evaluate the anticipated data transfer volume for your system.
- Integration with Firebase Services: Your attendance tracking system's functionality can be improved by utilizing the range of supplementary services provided by Firebase. Some providers offer free tiers, while others charge extra. Think about the probable cost effects of the following services:
- The security and access control of your system can be improved by integrating Firebase Authentication for user management and

authentication. Although Firebase offers a free tier for authentication, you might need to think about the pricing of higher-tier authentication plans if your attendance system has a lot of users or needs advanced authentication features.

- Serverless computing is made possible by Firebase Cloud Functions, which also let you start events depending on database changes. Despite the fact that Firebase offers a limited amount of free function invocations and resources, complex attendance systems with significant features may need more resources, which could result in usage-based fees.
- Firebase Hosting and Storage: You may use Firebase Hosting and Firebase Storage if your attendance tracking system has web interfaces or needs to store user-generated material like files or photographs. Examine the fee schedules for these services to Analyze the probable financial impact on your total spending plan.
- Even though they are not directly related to Firebase's pricing, development and maintenance expenditures for your attendance marking system must be taken into account. These expenses cover the time and materials needed for designing, implementing, and maintaining the hardware parts, user interfaces, and Firebase Database connectivity. You should include the cost of ongoing system upkeep, bug repairs, and feature upgrades in your cost estimation.
- Conclusion: Using Firebase Database to build a real-time attendance marking system has many benefits, particularly in terms of scalability and real-time data synchronization.

- However, it is crucial to consider the associated financial issues. By closely reviewing Firebase's pricing policy, calculating database usage, storage needs, network bandwidth, and other factors.
- You can get a thorough grasp of the prospective expenses by taking into account the integration with other Firebase services. In addition, including development and maintenance costs offers a complete cost analysis. With this information, you can make effective plans.

3.4Implementation plan/methodology:

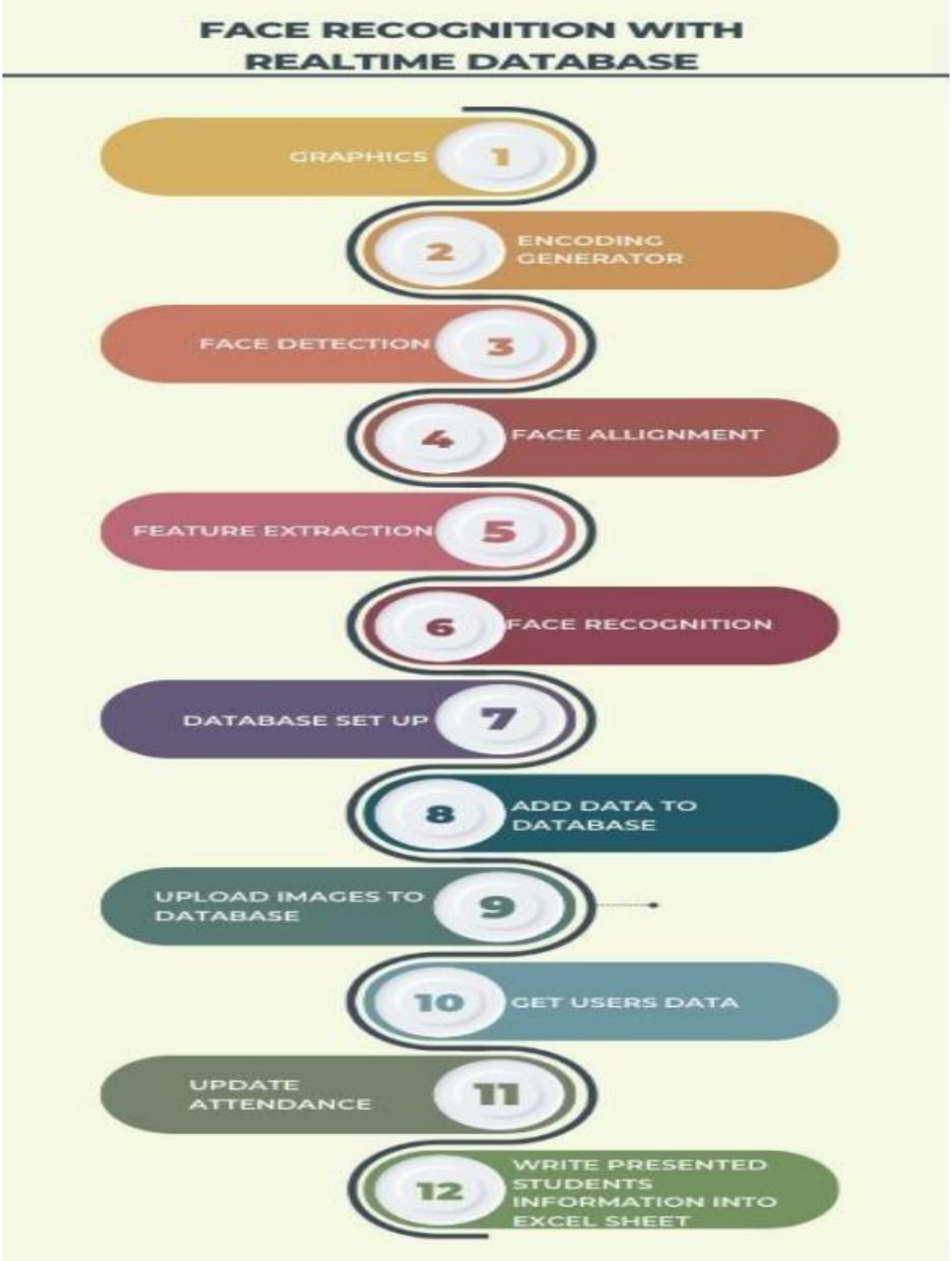


Figure 1

CHAPTER-4

RESULTS ANALYSIS AND VALIDATION

This chapter presents the results of our face recognition with real time database and discusses the insights gained from the analysis.

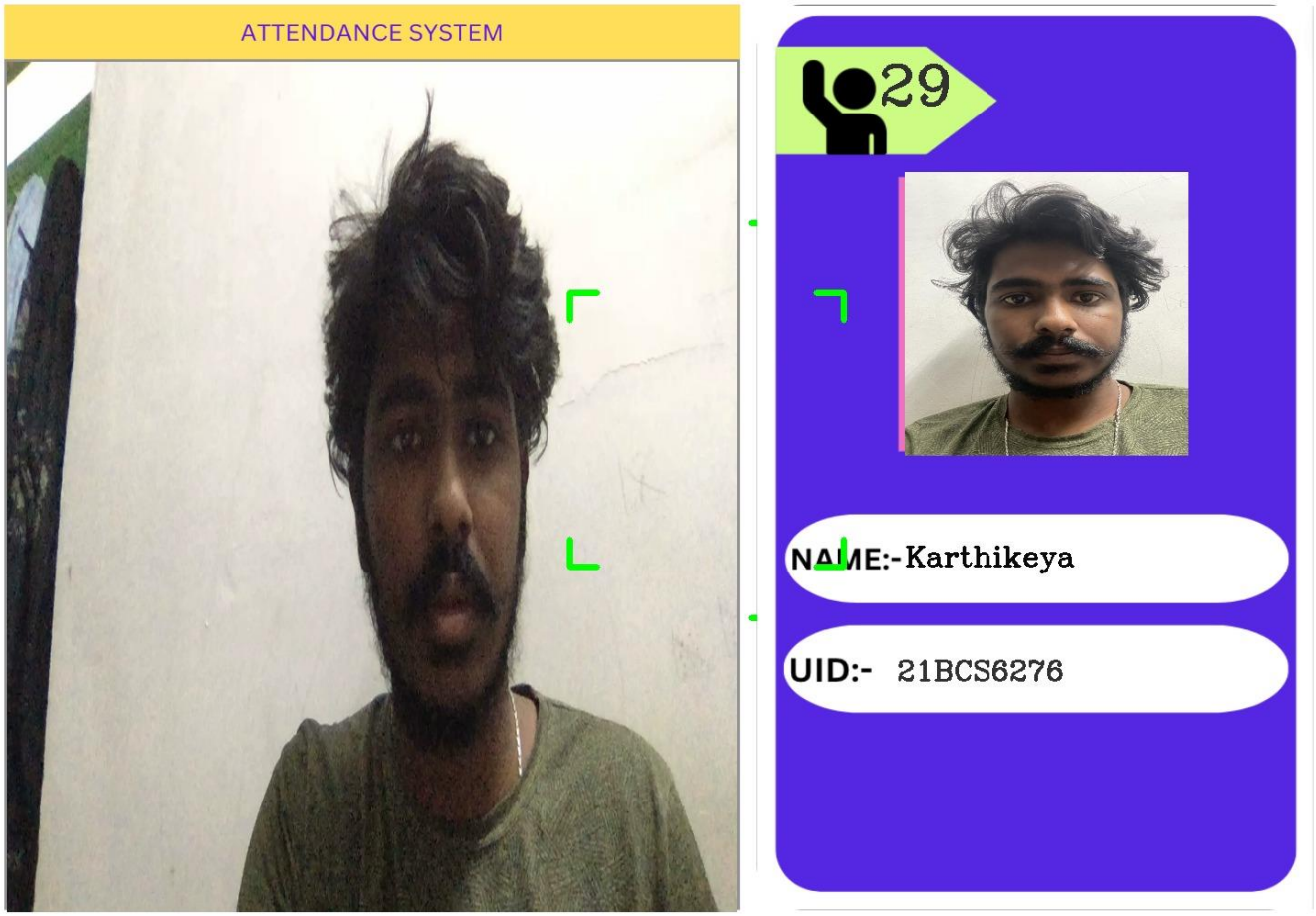
4.1 Result:

- This code is an implementation of a face recognition attendance system using OpenCV and face recognition library. The system captures the video from the webcam, detects faces, and recognizes them using the pre-trained face encodings. If the face is recognized, the system retrieves the student information from the Firebase database and displays it on the screen. If the student's last attendance was more than 30 seconds ago, the system updates the student's attendance record in the database and increases the total attendance count. Here are the main steps of the code:
- Import the required libraries: cv2, pickle, so, evzone, face recognition, datetime, NumPy, and Firebase libraries.
- Initialize the Firebase app using the service account key and set the database and storage bucket URLs.
- Capture the video from the webcam and set the width and height of the video frame.
- Load the pre-trained face encodings from the encoding file using the pickle module.
- Load the mode images for different system modes.
- Create a loop to continuously read the video frames and detect faces in each frame using the face recognition library.

- If a face is detected, compare it with the known face encodings, and if it matches, retrieve the student information from the Firebase database.
- If the student's last attendance was more than 30 seconds ago, update the student's attendance record in the database and increase the total attendance count.
- Display the student information and attendance count on the screen and switch to the loading mode for 10 frames before switching back to the face recognition mode.

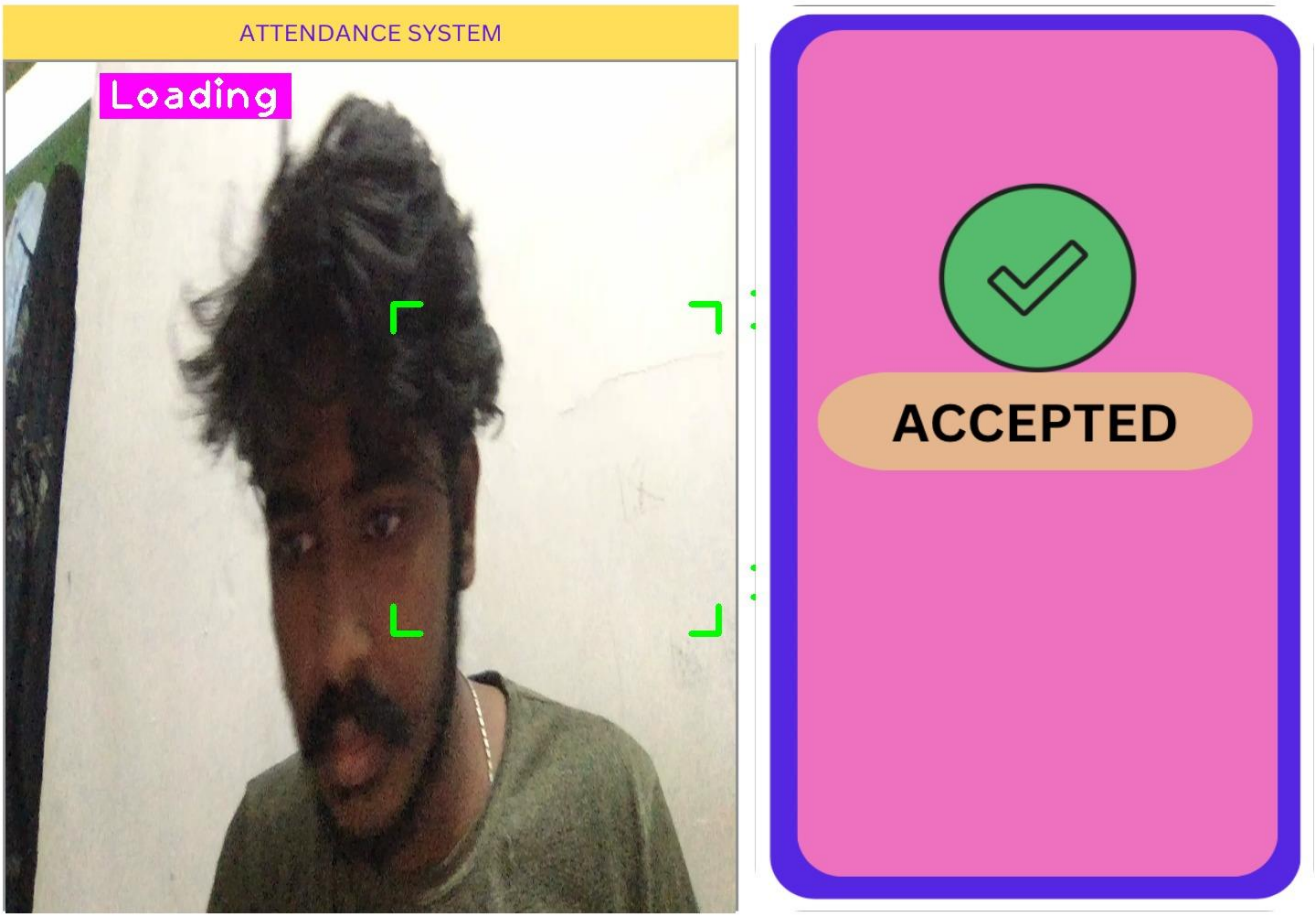
4.2 Validation:

Fig 2: extracting information



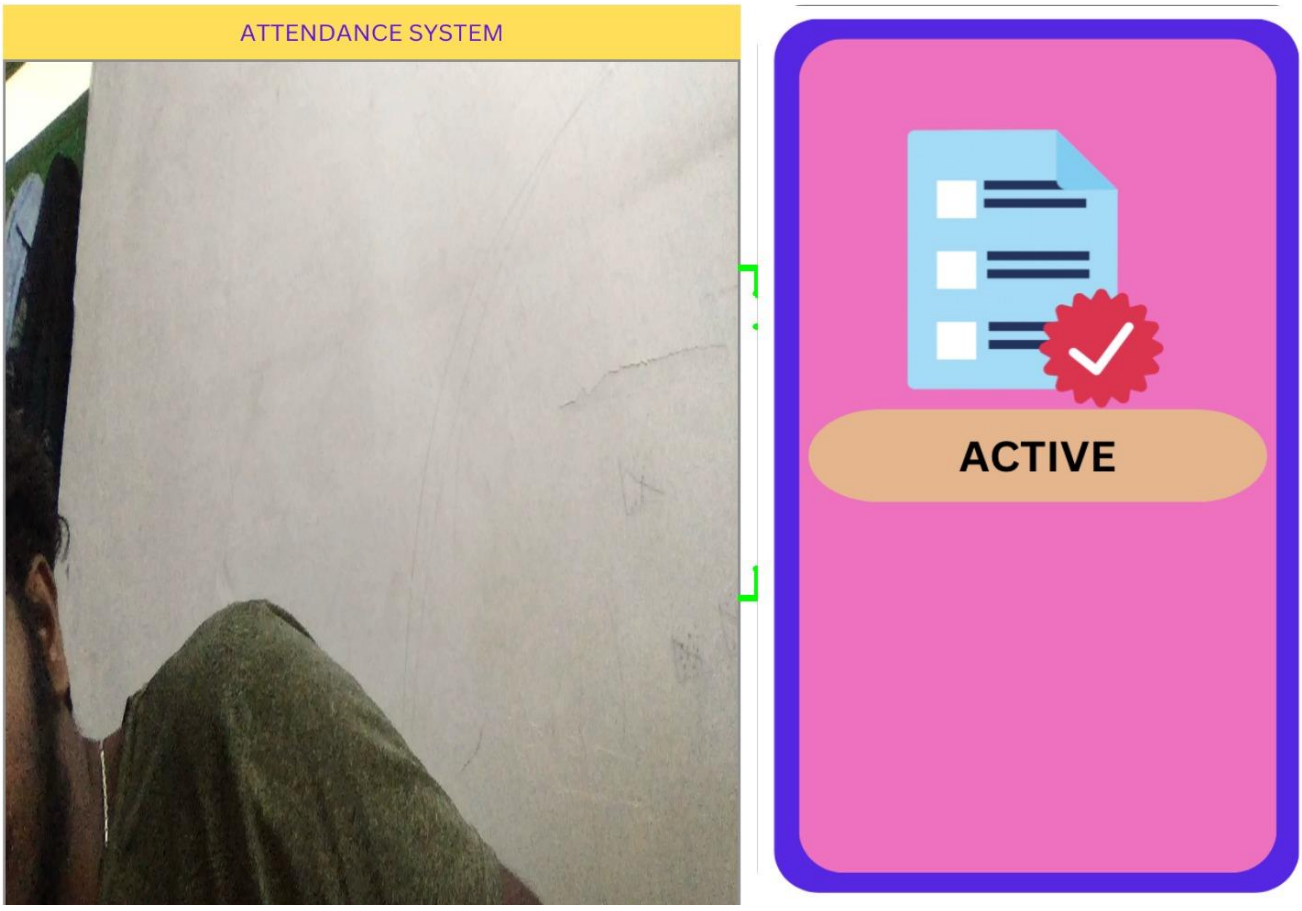
step 1: It will recognize the face and extract the data from the database and show this graphic design the details show is name, aid, and how many classes you are present and image of the student.

Fig 3: marking attendance in real time



step 2: after recognizing it from database and it will mark the attendance in real time in firebase database so it is easy to access and manage the data.

Fig 4: state of student



step 3: after completion of these process when student comes before the camera who is already marked it will be accepted and he is the condition of active

CHAPTER-5

CONCLUSION AND FUTURE WORK

In this chapter, we summarize the main findings of our research and discuss their implications for the development of face recognition using real time database.

5.1 Conclusion

The project's purpose is to create and use of a face recognition system for keeping track of students' attendance at a school have proven to be a significant development. An automated system that makes use of facial recognition technology has successfully replaced the labor-intensive, error-prone, and time-consuming manual attendance tracking approach. During the pilot test, the system proved its capability to efficiently and reliably track student attendance, removing the chance of proxy attendance and enhancing the procedure in general.

The use of the facial recognition system has helped the educational facility in several ways. As a result, administrative tasks have become easier to manage, instructional time has been preserved, and attendance records' correctness and dependability have improved. The system has considerably increased the efficiency and efficacy of the attendance tracking process by utilizing the power of biometric technology, allowing instructors and staff to devote more time and resources to the system's main educational activities**.**

5.2 Future Work

- Even if the system for tracking attendance using facial recognition has been built and put into use with success, there are still ways to expand and improve it. Some potential future work areas include:
- Integration with current student information systems: By integrating the face recognition system with the current student information systems used by the educational institution, seamless data transfer, thorough student records management, and a comprehensive picture of attendance and performance are all made possible.
- System optimization that is ongoing can improve the precision and effectiveness of the face recognition system. The system can adapt better to changing environmental conditions and student appearances, increasing its reliability. This can be achieved by strengthening algorithms, modernizing hardware components, and utilizing machine learning approaches.
- Scalability and wider implementation: Building on the positive results of the pilot test, the face recognition system needs to be assessed and given serious consideration for wider use in other educational settings, such as classrooms and institutions. Trials and assessments in various environments will yield insightful information and feedback for ongoing system adaptation and improvement.
- Ethics and legal ramifications: It are important to continually discuss the ethical and legal ramifications of using biometric technologies in educational settings. To preserve the system's moral usage and safeguard the privacy of employees and students, compliance with data protection laws, upholding privacy rights, and routinely assessing rules and procedures are crucial.

- The face recognition system can continue to develop into a reliable and widely used solution for attendance tracking in educational institutions by pursuing these future work areas. Its continuing development will increase effectiveness, precision, and general student participation, supporting the institution's dedication to excellence while also advancing educational procedures.

REFERENCES:

[1]. Georgiades, A.S., Blumer, P.N. and Kriegman, D.J., 2001. From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(6), pp.643-660.

[2]. Viola, P. and Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 1, pp. I-511). IEEE

[3]. Yang, M., Kriegman, D. and Ahuja, N., 2002. Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(1), pp.34-58.

[4]. 4. Turag, P., Chellappan, R., Subrahmanyan, V.S. and Udrea, O., 2008. Machine recognition of human faces. Springer Science & Business Media.

[5]. Phillips, P.J., Moon, H., Rizvi, S.A., Reuss, P.J. and Tyson, J.G., 2000. The FERET evaluation methodology for face recognition algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(10), pp.1090-1104.

[6]. Moghaddam, B., Pentland, A. and Srihari, S.N., 2000. A Bayesian framework for face recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 1, pp. 176-183). IEEE.

[1]. Zhao, W., Chellappan, R., Phillips, P.J. and Rosenfeld, A., 2003. Face recognition: A literature survey. ACM Computing Surveys (CSUR), 35(4), pp.399-458.

[2]. Wang, Y., Huang, T. and Wu, G., 2004. Face recognition using discriminant locality preserving projections. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (Vol. 2, pp. II-209). IEEE

[3]. Tan, X. and Trigs, B., 2010. Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Transactions on Image Processing, 19(6), pp.1635-1650.

[4]. Hu, J., Lu, J., Tan, Y.P. and Zhou, J., 2014. Discriminative deep metric learning for face verification in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) (pp. 1875-1882). IEEE

1.(Benefits of machine learning in fraud detection) banks#:~:text=The%20idea%20behind%20using%20machine,between%20fraudsters%20and%20legitimate%20clients

2. https://www.miteksystems.com/blog/how-does-machine-learning-help-with-fraud- detection-in

USER MANUAL:

Table of Contents:

1.Introduction:

A software-based solution called the Face Recognition System for Attendance Tracking was created to rapidly and precisely track student attendance in educational institutions. This user manual will walk you through the system's installation and use to ensure effective attendance control.

2.System Requirements:

Make sure your computer complies with the following prerequisites before installing and using the system:

- Operating System: [Specify compatible operating systems]
- Processor: [Specify minimum processor requirements]
- Memory: [Specify minimum RAM requirements]
- Storage: [Specify minimum available storage space]
- Camera: [Specify compatible camera requirements]

3.Getting Started:

- Hardware Setup:

  Completing the following steps will enable you to use the facial recognition system :

  - ✓ Connect a compatible camera to your computer.
  - ✓ Ensure proper positioning of the camera to capture clear and unobstructed images of students' faces.
  - ✓ Make sure the camera is securely mounted or positioned to maintain stability during usage.

- Software Installation:

  Follow these steps to deploy the face recognition system:

  - ✓ Insert the installation media or download the software from the provided source.

- ✓ Run the installer and follow the on-screen instructions.
- ✓ Specify the installation path and any additional settings as prompted.
- ✓ Wait for the installation to complete successfully.

4.System Usage:

- Enrollment Process:

  Follow these steps to register students in the system for keeping track of attendance:

  - ✓ Launch the face recognition system.
  - ✓ Access the enrollment section and provide the necessary student information (e.g., name, ID number).
  - ✓ Position the student in front of the camera and capture multiple images from different angles.
  - ✓ Verify the captured images to ensure clarity and accuracy.
  - ✓ Save the enrolled student's information in the system's database.

- Attendance Tracking:

  Following these procedures will allow you to track student attendance using the facial recognition system:

  - ✓ Launch the face recognition system.
  - ✓ Access the attendance tracking section.
  - ✓ Position the camera to capture the faces of students as they enter the

classroom.

- ✓ The system will automatically match the captured faces with enrolled student data in real-time.
- ✓ The attendance records will be updated accordingly, marking students as present or absent.

5. Troubleshooting:

- ✓ Refer to the troubleshooting section of the provided user guide if you have any problems or errors when using the face recognition system. It provides detailed solutions for frequent issues so you can address them successfully.

6. Frequently Asked Questions:

- ✓ For answers to frequently asked questions concerning the face recognition technology, consult the FAQ section of the user handbook. It addresses issues including system compatibility, troubleshooting, and recommended usage techniques.

7. Conclusion:

- ✓ The Face Recognition System for Attendance Tracking provides a reliable and efficient solution for tracking student attendance in educational institutions. By following the instructions in this user manual, you can maximize the system's benefits and streamline the attendance management process.

  Please refer to the contact information in the user handbook for more information or technical support.

  Please take note that the user manual needs to be adjusted to reflect the particular capabilities and features of your face recognition system.

APPENDIX

1. Plagiarism Report

2. Design Checklist