AUTOMATIC ATTENDANCE SYSTEM USING FACE RECOGNITION

A MINI PROJECT REPORT

18CSC305J - ARTIFICIAL INTELLIGENCE

Submitted by

ANIRUDDH Y S [RA2011003010141]
AKSHAY RAM SANJAYAN M S [RA2011003010173]
KAVIE NIVESH V [RA2011003010187]

Under the guidance of

Mrs.Rajalakshmi M

Assistant Professor, Department of Computer Science and Engineering

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Chengalpattu District

MAY 2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that Mini project report titled **"Automatic Attendance System Using Face ecognition"** is the bona fide work of **ANIRUDDH Y S [RA2011003010141], AKSHAY RAM ANJAYAN M S [RA2011003010173], KAVIE NIVESH V [RA2011003010187]** who carried out the inor project under my supervision. Certified further, that to the best of my knowledge, the ork reported herein does not form any other project report or dissertation on the basis of hich a degree or award was conferred on an earlier occasion on this or any other candidate.

Department of Computing Technology Kattankulathur 603 203 SRM IST

M. Pushpalatha

M. R 12/6/23

SIGNATURE

Mrs.Rajalakshmi M
Assistant Professor
Department of Computing Technologies

SIGNATURE

Dr. M. Pushpalatha
Professor & Head
Department of Computing Technologi

ABSTRACT

In the recent time automated face recognition has become a trend and has been developed very much , this is mainly due to two reasons; first it is due to availability of modern technologies and second is due to the ability to save time using face recognition in the process of taking attendance of students. Its usage will grow vast in the future as it saves a lot of time. It consumes a lot of time to take attendance manually and few might also fake the attendance, in order to prevent time consumption and avoid faking the attendance face recognition is used to identify the person present in the class and mark his attendance , this is done with the help of image or video frame. We proposed an automatic attendance management system using machine learning techniques such as CNN algorithm. The face detection and recognition will automatically detect the students in the classroom and mark the attendance by recognizing the person. The faculty has access to add the student details such as name, USN, phone number, email-id. Then the image is captured through a high definition camera during the class hours. When the lecturing is going on faces of students are detected, segmented and stored for verification with database using the Convolutional Neural Networks (CNN) algorithm of machine learning technique. Deep learning uses multiple layers to discover the meaning of data at different levels of extraction. Improves appearance for facial research. By introducing deep learning in face recognition, state-of-the art application has been developed and success has been achieved in practical applications. Convolutional neural network is a deep neural network model that has proven success in face recognition. In real time, models must be built before CNNs can be used.

TABLE OF CONTENTS

Chapter No.	Title	Page No.
	ABSTRACT	**iii**
	TABLE OF CONTENTS	**iv**
	LIST OF FIGURES	**vi**
	ABBREVIATIONS	**vii**
1	**INTRODUCTION**	**1**
1.1	Introduction	1
1.2	Problem statement	1
1.3	Software Requirements Specification	2
2	**LITERATURE SURVEY**	**3**
2.1	Existing system With Limitation	3
2.2	Proposed system	3
3	**SYSTEM ARCHITECTURE AND DESIGN**	**5**
3.1	Architecture Diagram	5
3.1.1	Job Master	5
3.1.2	Job Workers	6
3.2	Convolution Neural network	7
3.2.1	Convolution layer	7
3.2.2	Pooling layer	7
3.2.3	Fully Connected Layer	7
3.2.4	Dropout	8
3.2.5	Activity Functions	8
4	**METHODOLOGY**	**9**
4.1	Methodology	9
4.1.1	Data Entry	9
4.1.2	Dataset Training	9
4.1.3	Facial Recognition	10
4.1.4	Attendance Entry	10
4.1.5	Notify	10

5 CODING AND TESTING 11

5.1 main.py **11**

5.2 AddDataToDatabase.py **15**

5.3 EncodeGenerator.py **17**

6 RESULTS AND DISCUSSIONS 19

6.1 Results 19

6.2 Encodings 21

6.3 Database Recordings 22

7 CONCLUSION AND FUTURE ENHANCEMENT 23

7.1 Conclusion **23**

7.2 Future Enhancement **23**

REFERENCES 24

LIST OF FIGURES

Figure No.	Figure Name	Page No.
3.1	Architecture diagram	5
3.2	Job Master and Job Workers	6
3.3	Convolution Neural Network Architecture	6
6.1	Attendance Marking	18
6.2	Active Attendance	18
6.3	Attendance Already Marked	19
6.1	Realtime Database	21

ABBREVIATIONS

AAS	Automatic attendance system
CNN	Convolution Neural Network
FR	Face Recognition
CV	Computer Vision
DB	Data Base
`ML	Machine Learning
``DL	Deep Learning

CHAPTER1

INTRODUCTION

1.1 Introduction

Many scientific advances and technologies have been created in this recent epoch of automation to save time, improve accuracy, and reduce costs improve the quality of our life System of automated help is the advancement that has been made in the field of traditional jobs will be replaced by automation Call action on one presence. Assistive technology Bio-metrics are commonly used in these systems. data, web-based and based on smart cards. These In a variety of organizations, systems are commonly employed. The traditional way of calculating attendance time is as follows: When the strength isn't there, it's time-consuming and complicated. The automated attendance system has an advantage over the conventional method because It saves time and can also be utilized for security purposes. This also aids in the prevention of erroneous assistance.Machine learning techniques are used. We can take automatic attendance on paper. In recent years, face recognition from stationary and moving images has been an active and demanding research area in the field of image processing, pattern recognition and so on [1-5].Automatic attendance systems have been around for quite some time, but with the advent of new artificial intelligence technology like Convolution Neural Network (CNN), these systems have become more accurate and efficient.Facial recognition is one such technology that has been integrated into automatic attendance systems. With the help of CNN, facial recognition can be made even more effective, making it an ideal solution for organizations looking to improve their attendance tracking.As a form of computer application, face recognition is being widely used in recent times on mobile platforms [6, 7]. It has also seen wider uses in other technological forms, such as robotics [8, 9]. Convolutional Neural Networks are used to create this model. Face recognition algorithm in the classroom .The presence and absence of the presence and absence of the presence and absence of the presence and absence of the If the student receives a message, he or she will be notified.

1.2 Problem Statement

In schools, colleges and workplaces, attendance tracking is an important task for recording and analyzing student or employee performance. The process of participating in the written record is always time-consuming and error-prone. Therefore, it should be an automatic system that can be held correctly and efficiently. The solution is to use CNN face recognition to create an automated attendance system that can

recognize the faces of students or staff and enter attendance. The system must be able to process large amounts of data, work in different lighting conditions and angles, and perform accurate facial recognition.The system should be user-friendly and easy to integrate with other systems. The aim of this project is to create a more efficient and effective attendance system using CNN for facial recognition that can improve the attendance process in schools, school colleges and offices.

1.3 Software Requirement Specification

Python: Python is a popular programming language used for developing machine learning applications. The face recognition system can be implemented using Python.

TensorFlow: TensorFlow is a popular open-source machine learning library developed by Google. It provides a wide range of APIs for building and training deep learning models. It can be used to train a face recognition model based on CNN.

OpenCV: OpenCV is an open-source computer vision library that provides various algorithms for image and video processing. It can be used to implement face detection and image preprocessing in the face recognition system.

MySQL: MySQL is an open-source relational database management system that can be used to store the attendance data.

Flask: Flask is a lightweight web framework that can be used to create web applications for the face recognition system. It can be used to create a user interface for the system and integrate it with the database.

PyCharm: PyCharm is an integrated development environment (IDE) that can be used for developing Python applications. It provides various features such as code completion, debugging, and version control.

Anaconda: Anaconda is a distribution of Python and its packages for scientific computing. It includes many packages required for building machine learning applications, including TensorFlow and OpenCV.

GIT: Git is a version control system that can be used to track changes in the source code and collaborate with other developers.

CHAPTER 2

Literature Survey

2.1 Existing System With Limitations

In recent times, different techniques, methods and algorithms have been used to perform facial recognition and increase the accuracy of facial recognition. In [10], an automatic student attendance system was proposed that can be utilized in small and crowded classrooms. In this model, after the training stage, the user, e.g. the teacher, can get the attendance by taking one or multiple images of the classroom using his/her smartphone. The implemented system detects the faces in the images and recognizes which students are present in order to mark the attendance. But real-time video attendance was not possible with this system.

The Bilateral Filter, Haar-like features [11] techniques were applied to identify the model of the human face in [12]. The Simplified Fuzzy Adaptive Resonance Theory Map Neural Network (SFAM-NN) method has been compared to the Cascade Classifier Adaboost method to evaluate the efficiency of the proposed method. In [13], the authors implemented a system where the Viola and Jones algorithm [14] has been utilized for detecting face bounding box, constrained local model-based face tracking and face landmark identification algorithms. It is also called the AdaBoost algorithm for face recognition. To perform facial recognition in this model, Principle Component Analysis (PCA) has been used. In [15], an automatic attendance management system was proposed using face recognition algorithms. A camera at the doorway captures students' image while entering into the class. But, that system faced limitations as it could not define two persons at the same time.

2.2 Proposed System

The goal of this project is to overcome the limitations of previous facial recognition techniques, a CNN-based automatic attendance system has been proposed. This system uses deep learning algorithms to learn facial features and can recognize faces even under challenging conditions.New features such as in-time and out-time recording have been added to improve the system's functionality and usability. In addition, measures have been taken to prevent spoofing attacks by using liveness detection techniques.

2.2.1 Automatic Time counting and notifications:

When the students enter the class the face recognition system recognizes the face and starts a timer for that particular student and if the student is present in the class for more than 40 minutes he/she would get the attendance for that particular class. Also when the student does not attend any class a notification will automatically be sent to their parents.

2.2.2 Predictive Analytics

Another exciting feature of this automatic attendance system is its use of predictive analytics. By analyzing attendance data over time, the system can identify patterns and trends that may indicate potential issues such as chronic absenteeism or tardiness. This allows teachers and administrators to intervene early and provide support to students who may be struggling.Furthermore, the system can generate reports that provide insights into attendance patterns across different classes, grade levels, and even entire schools. This information can be used to identify areas for improvement and inform decision-making around resource allocation and scheduling.

2.2.3 Mobile App Integration

To make the attendance-taking process even more convenient, this automatic attendance system can be integrated with a mobile app. Students can use the app to check-in when they arrive in class, eliminating the need for physical sign-in sheets or cards. Additionally, the app can send push notifications to remind students of upcoming classes and deadlines.Teachers and administrators can also use the app to access attendance data in real-time, view reports, and communicate with students and parents. This mobile integration makes the attendance process more streamlined and accessible for everyone involved.

2.2.4 Security and Privacy

As with any technology that involves personal data, security and privacy are of utmost importance. This automatic attendance system uses state-of-the-art encryption and security protocols to ensure that student data is protected from unauthorized access or theft.Additionally, the system is designed to be fully compliant with relevant data protection regulations such as GDPR and CCPA. Students and parents can rest assured that their personal information is being handled responsibly and transparently.

CHAPTER 3

SYSTEM ARCHITECTURE AND DESIGN

3.1 Architecture design

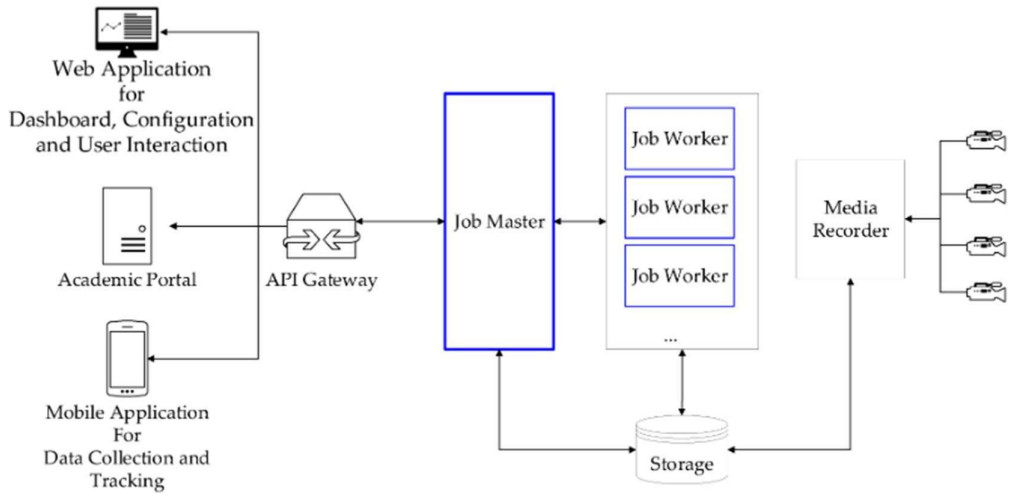

Figure 3.1

3.1.1 Job Master

As a navigator of the system, the job master is responsible as the navigator of the data streams and process—the system runs according to the schedule data. The corresponding FR model is also loaded based on the list of students instead of constructing a significant model for identifying the whole students. The API Gateway allows APIs to communicate with existing systems. The system APIs are available with the building management system configured via the administrator's web application, for example, synchronized list attendees, floors list, rooms list, and so on. The process of data synchronization allows the system to be compatible with existing systems, thus enhancing the adaptability.

Figure 2 demonstrates the process of dividing the archived frames into tasks. This process allows speeding up the calculation by adding more job workers. The job master acts as a workload balancer, which controls the works among the workers to avoid bottleneck and race conditions.

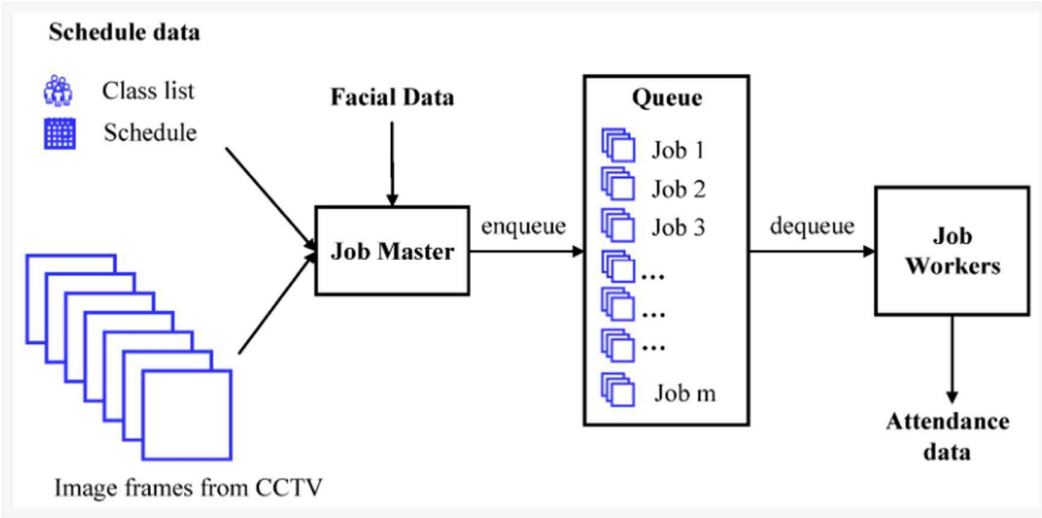

Figure 3.2

3.1.2 Job Workers

Another module that plays an essential part in the architecture is job worker. It performs the simple task of an FR problem. All descriptions of FR building blocks are available in this module .The video frames are processed directly in this module. All processes are in parallel through the arrangement of the job master. They are constructed based on master-slave architecture . This architecture allows us to work efficiently with extensive data when deploying systems on a larger scale (hundreds of cameras).

3.2 Architecture of convolution neural network

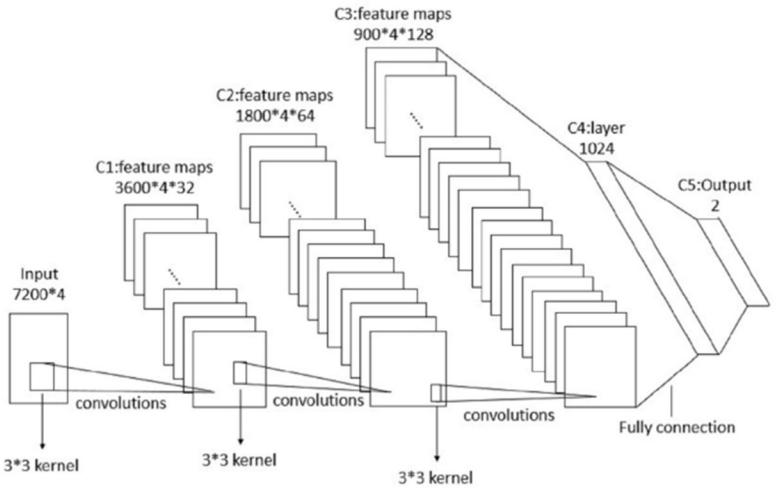

Figure 3.3

3.2.1 Convolutional Layer

This layer is the first layer that is used to extract the various features from the input images. In this layer, the mathematical operation of convolution is performed between the input image and a filter of a particular size MxM. By sliding the filter over the input image, the dot product is taken between the filter and the parts of the input image with respect to the size of the filter (MxM).

The output is termed as the Feature map which gives us information about the image such as the corners and edges. Later, this feature map is fed to other layers to learn several other features of the input image.

The convolution layer in CNN passes the result to the next layer once applying the convolution operation in the input. Convolutional layers in CNN benefit a lot as they ensure the spatial relationship between the pixels is intact.

3.2.2 Pooling Layer

In most cases, a Convolutional Layer is followed by a Pooling Layer. The primary aim of this layer is to decrease the size of the convolved feature map to reduce the computational costs. This is performed by decreasing the connections between layers and independently operates on each feature map. Depending upon method used, there are several types of Pooling operations. It basically summarises the features generated by a convolution layer.

In Max Pooling, the largest element is taken from feature map. Average Pooling calculates the average of the elements in a predefined sized Image section. The total sum of the elements in the predefined section is computed in Sum Pooling. The Pooling Layer usually serves as a bridge between the Convolutional Layer and the FC Layer.

This CNN model generalises the features extracted by the convolution layer, and helps the networks to recognise the features independently. With the help of this, the computations are also reduced in a network.

3.2.3Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place. The reason two layers are connected is that two fully connected layers will perform better than a single connected layer. These layers in CNN reduce the human supervision

3.2.4 Dropout

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model's performance when used on a new data.To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

Dropout results in improving the performance of a machine learning model as it prevents overfitting by making the network simpler. It drops neurons from the neural networks during training.

3.2.5 Activation Functions

Finally, one of the most important parameters of the CNN model is the activation function. They are used to learn and approximate any kind of continuous and complex relationship between variables of the network. In simple words, it decides which information of the model should fire in the forward direction and which ones should not at the end of the network.

It adds non-linearity to the network. There are several commonly used activation functions such as the ReLU, Softmax, tanH and the Sigmoid functions. Each of these functions have a specific usage. For a binary classification CNN model, sigmoid and softmax functions are preferred an for a multi-class classification, generally softmax us used. In simple terms, activation functions in a CNN model determine whether a neuron should be activated or not. It decides whether the input to the work is important or not to predict using mathematical operations.

CHAPTER 4

METHODOLOGY

4.1 Methodology:

The different parts of the system can be grouped into four main stages.

These are:

- Data Entry
- Dataset Training
- Face Recognition
- Attendance Entry

4.1.1 Data Entry

The first step is to include the faces of the students in the system for creating a dataset. For this, continuous photos of each of the enrolled students are taken by the system from a live video stream one person at a time, along with their names and IDs. The default setting is set to take 20 pictures at a 2 second interval from a live video stream. It is preferred that the students have different head positioning during this time to create a better dataset. The setting can be changed to increase the number of pictures taken to make a more accurate dataset. A folder for each student is created with the corresponding student's name and ID as the label. Each of the pictures of faces is then saved in that student's designated folder. Besides this process, previously taken pictures of the enrolled students can be added to the dataset for making it more diverse. In this case, the new photos will be saved in that student's previously created folder. After every data entry, the system is automatically trained using the currently available dataset.

4.1.2 Dataset Training

The training is done by a triplet training step. This method consists of three different face images, two of which belong to the same person. The CNN extracts 128 facial measurements, called embeddings from each face. These are stored as 128-d vectors. For the two images belonging to the same person, the CNN tweaks their weights to make the vectors closer while also making them slightly further away than the third picture. This whole process is done using the face_encodings function of the face_recognition library. The extracted data is then stored as a pickle file, which is used later for comparing and recognizing faces in the next stage. This step also automatically creates a spreadsheet that contains the names and IDs of all the students of the class, whose data has been entered in the previous stage.

4.1.3 Face Recognition:

A video camera on a good position, preferably on the doorway of the classroom or inside the classroom itself where it has a clear view of the students. The system can then detect the faces of the students from the ongoing video stream of the camera. The detected faces are then compared to the trained dataset. A confidence value is assigned to each of the matches. The match with the highest confidence is selected and the label, which is the name and ID of the student, is extracted. If there is not a match of a high enough accuracy, then the student is labeled as 'Unknown'.

4.1.4 Attendance Entry

In each session of the video stream, which would be each period of classes, the names and IDs of the recognized students are automatically logged on a daily attendance spreadsheet along with the date, time and period name. There is also the option to calculate the total attendance during a specific time span, which can be a semester or month or year depending on the time range. The system can automatically calculate the total number of classes and also show the total attendance of the enrolled students for those classes.

4.1.5 Notify

If any person is absent for the class a notification will be sent to their mobile as well as their parents mobile.

CHAPTER 5

CODING AND TESTING

5.1 main.py

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

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


bucket = storage.bucket()

cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)

imgBackground = cv2.imread('Resources/background.png')

# Importing the mode images into a list
folderModePath = 'Resources/Modes'
modePathList = os.listdir(folderModePath)
imgModeList = []
for path in modePathList:
    imgModeList.append(cv2.imread(os.path.join(folderModePath, path)))
# print(len(imgModeList))
```

```
# Load the encoding file
print("Loading Encode File ...")
file = open('EncodeFile.p', 'rb')
encodeListKnownWithIds = pickle.load(file)
```



```
file.close()
encodeListKnown, studentIds = encodeListKnownWithIds
# print(studentIds)
print("Encode File Loaded")

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)

    imgBackground[162:162 + 480, 55:55 + 640] = img
    imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

    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("faceDis", faceDis)

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

            if matches[matchIndex]:
                # print("Known Face Detected")
                # print(studentIds[matchIndex])
```

```
    y1, x2, y2, x1 = faceLoc
        y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
        bbox = 55 + x1, 162 + y1, x2 - x1, y2 - y1
        imgBackground = cvzone.cornerRect(imgBackground, bbox, rt=0)
        id = studentIds[matchIndex]
        if counter == 0:
            cvzone.putTextRect(imgBackground, "Loading", (275, 400))
            cv2.imshow("Face Attendance", imgBackground)



                cv2.waitKey(1)
            counter = 1
            modeType = 1

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 data of attendance
        datetimeObject = datetime.strptime(studentInfo['last_attendance_time'],
                                           "%Y-%m-%d %H:%M:%S")
        secondsElapsed = (datetime.now() - datetimeObject).total_seconds()
        print(secondsElapsed)
        if secondsElapsed > 10:
            ref = db.reference(f'Students/{id}')
            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 = 3
            counter = 0
            imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

    if modeType != 3:
```

```
if 10 < counter < 20:
            modeType = 2

        imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]

        if counter <= 30:
            cv2.putText(imgBackground, str(studentInfo['total_attendance']), (861, 125),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 1)
            cv2.putText(imgBackground, str(studentInfo['major']), (1006, 550),
                        cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)
            cv2.putText(imgBackground, str(id), (1006, 493),
```



```
                        cv2.FONT_HERSHEY_COMPLEX, 0.5, (255, 255, 255), 1)
            cv2.putText(imgBackground, str(studentInfo['standing']), (910, 625),
                        cv2.FONT_HERSHEY_COMPLEX, 0.6, (100, 100, 100), 1)
            cv2.putText(imgBackground, str(studentInfo['year']), (1025, 625),
                        cv2.FONT_HERSHEY_COMPLEX, 0.6, (100, 100, 100), 1)
            cv2.putText(imgBackground, str(studentInfo['starting_year']), (1125, 625),
                        cv2.FONT_HERSHEY_COMPLEX, 0.6, (100, 100, 100), 1)

            (w, h), _ = cv2.getTextSize(studentInfo['name'], cv2.FONT_HERSHEY_COMPLEX, 1, 1)
            offset = (414 - w) // 2
            cv2.putText(imgBackground, str(studentInfo['name']), (808 + offset, 445),
                        cv2.FONT_HERSHEY_COMPLEX, 1, (50, 50, 50), 1)
            resizeV=cv2.resize(imgStudent,(640,480),fx=0,fy=0,interpolation=cv2.INTER_CUBIC)

            imgBackground[162:162 + 480, 55:55 + 640] = resizeV

        counter += 1

        if counter >= 20:
            counter = 0
            modeType = 0
            studentInfo = []
            imgStudent = []
            imgBackground[44:44 + 633, 808:808 + 414] = imgModeList[modeType]
```

```
else:
    modeType = 0
    counter = 0
# cv2.imshow("Webcam", img)
cv2.imshow("Face Attendance", imgBackground)
cv2.waitKey(1)
```

5.2AddDataToDatabase.py

```
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://automaticattendncesystem-default-rtdb.firebaseio.com/"
})
ref = db.reference('Students')

data = {
"621104":
    {
        "name": "Aakash",
        "major": "Engineering",
        "starting_year": 2022,
        "total_attendance": 5,
        "standing": "A",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34"
    },
"524211":
    {
        "name": "Sandeep",
        "major": "Engineering",
        "starting_year": 2022,
        "total_attendance": 11,
        "standing": "A",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34"
    },
```

```
"321123":
      {
        "name": "Shiva",
        "major": "Engineering",
        "starting_year": 2020,
        "total_attendance": 16,
        "standing": "B",
        "year": 3,
        "last_attendance_time": "2022-12-11 00:54:34"
      },
   "852741":
      {
        "name": "Emly Blunt",
        "major": "Economics",
        "starting_year": 2021,
        "total_attendance": 12,
        "standing": "B",
        "year": 1,
        "last_attendance_time": "2022-12-11 00:54:34"
      },

"963852":
      {
        "name": "Elon Musk",
        "major": "Physics",
        "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)
```

5.3 EncodeGenerator.py

```
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
import cv2

cred = credentials.Certificate("serviceAccountKey.json")
firebase_admin.initialize_app(cred, {
    'databaseURL': "https://automaticattendncesystem-default-rtdb.firebaseio.com/",
    'storageBucket': "automaticattendncesystem.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")
```

CHAPTER 6

RESULTS

The primary goal of the system is to flawlessly mark and record attendance. For doing that, the main focus is to elevate the accuracy of the facial recognition system which is the cornerstone of this work. The built-in webcam of a laptop is used as a default video recorder for testing the system. The accuracy of the system in relation to the number of input images per person . The dataset consists of various number of images of 17 distinct people. The accuracy was measured by training a certain number of images per person and running the system to recognize them from five different frames from video source. It is also observed that increasing input images for training the dataset also increases the accuracy of the system. This accuracy increases swiftly when the number of image per person in the dataset is between 5 – 20.We have also found from our experiments, that if there is a discrepancy among the number of inputs for each student, the system sometimes becomes flawed and the student with a higher number of inputs is selected to be the one recognized by the system. To stop this error, the same number of inputs should be chosen during data entry stage. It is also perceived that the time taken to train the dataset increases drastically with the increase of the dataset. The time can be reduced if the system is running on a relatively high-performance computer. We also found that the system can't detect the faces of people who are too far away from the camera. So, it is recommended to place the camera where it can have the best view.

6.1 Results

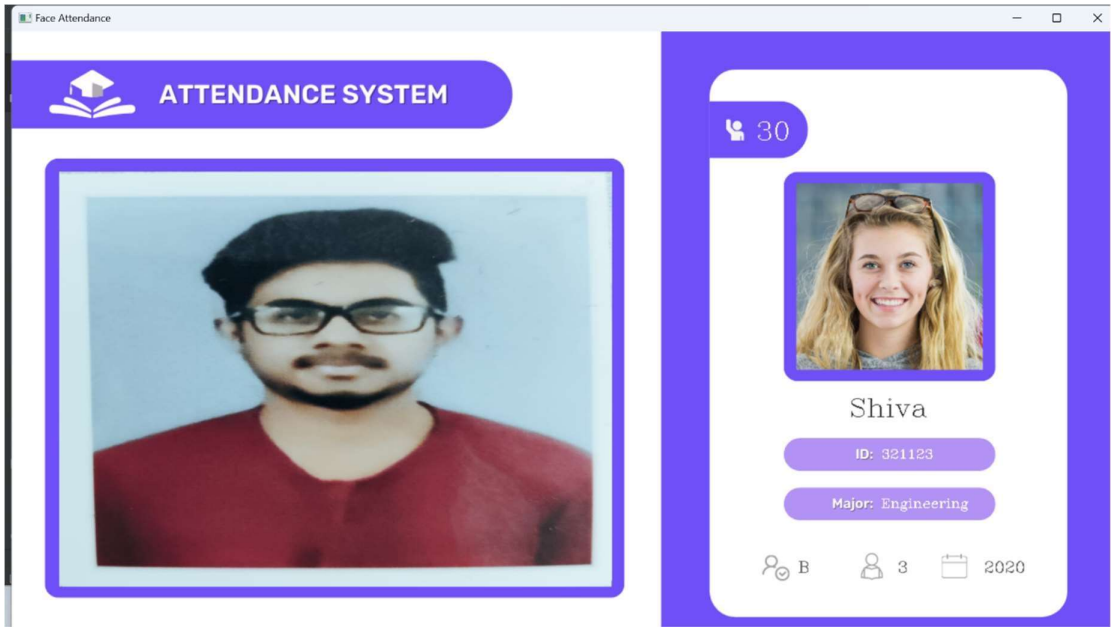

Figure 6.1

The attendance is marked when the person enter the classroom.

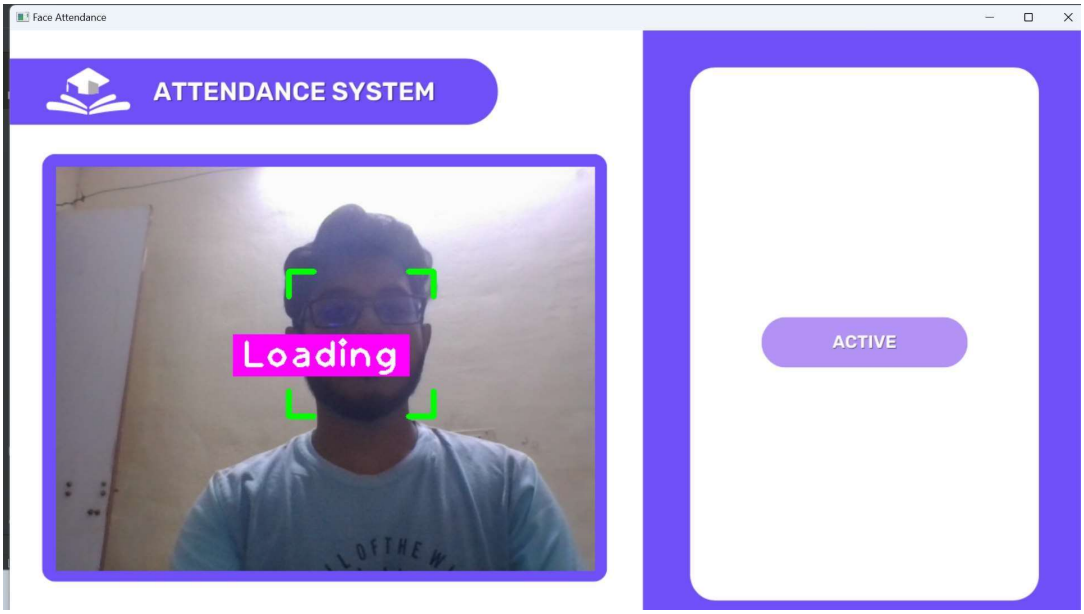

Figure 6.2

The above image shows that the person is active in the class.

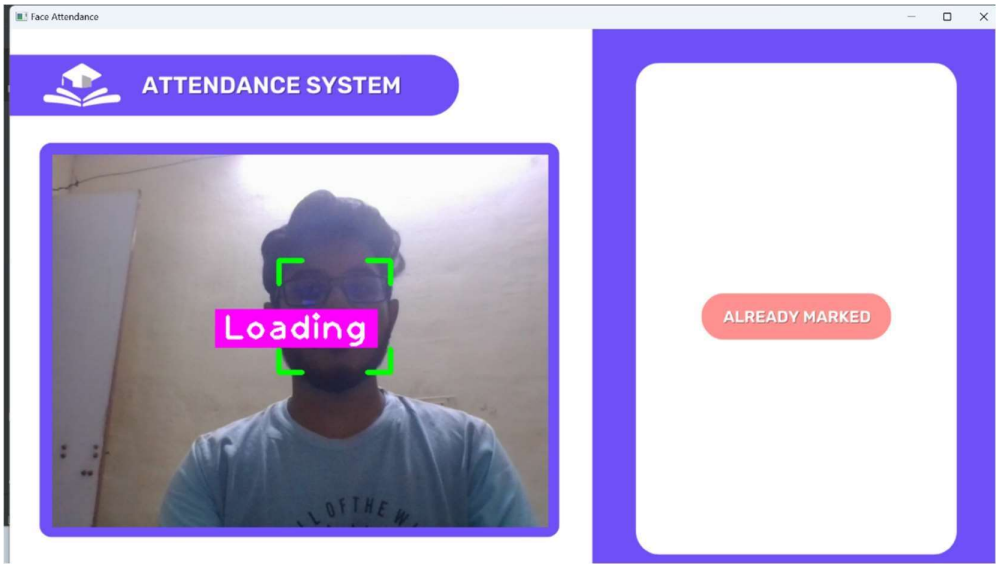

Figure 6.3

This shows that the attendance for the person is already marked.

6.2 Encodings

Encode File Loaded

{'last_attendance_time': '2023-04-18 19:19:08', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 27, 'year': 3}

1040767.593754

{'last_attendance_time': '2023-04-30 20:25:15', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 28, 'year': 3}

7.870666

{'last_attendance_time': '2023-04-30 20:25:15', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 28, 'year': 3}

10.446147

{'last_attendance_time': '2023-04-30 20:25:25', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 29, 'year': 3}

4.458016

{'last_attendance_time': '2023-04-30 20:25:25', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 29, 'year': 3}

6.146325

{'last_attendance_time': '2023-04-30 20:25:25', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 29, 'year': 3}

7.995608

{'last_attendance_time': '2023-04-30 20:25:25', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 29, 'year': 3}

9.875719

{'last_attendance_time': '2023-04-30 20:25:25', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 29, 'year': 3}

11.445519

{'last_attendance_time': '2023-04-30 20:25:36', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 30, 'year': 3}

10.36131

{'last_attendance_time': '2023-04-30 20:25:46', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 31, 'year': 3}

6.011584

{'last_attendance_time': '2023-04-30 20:25:46', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 31, 'year': 3}

7.573946

{'last_attendance_time': '2023-04-30 20:25:46', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 31, 'year': 3}

9.318095

{'last_attendance_time': '2023-04-30 20:25:46', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 31, 'year': 3}

10.860311

{'last_attendance_time': '2023-04-30 20:25:57', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 32, 'year': 3}

5.519417

{'last_attendance_time': '2023-04-30 20:25:57', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 32, 'year': 3}

8.852246

{'last_attendance_time': '2023-04-30 20:25:57', 'major': 'Engineering', 'name': 'Shiva', 'standing': 'B', 'starting_year': 2020, 'total_attendance': 32, 'year': 3}

10.802676

Database Recordings:

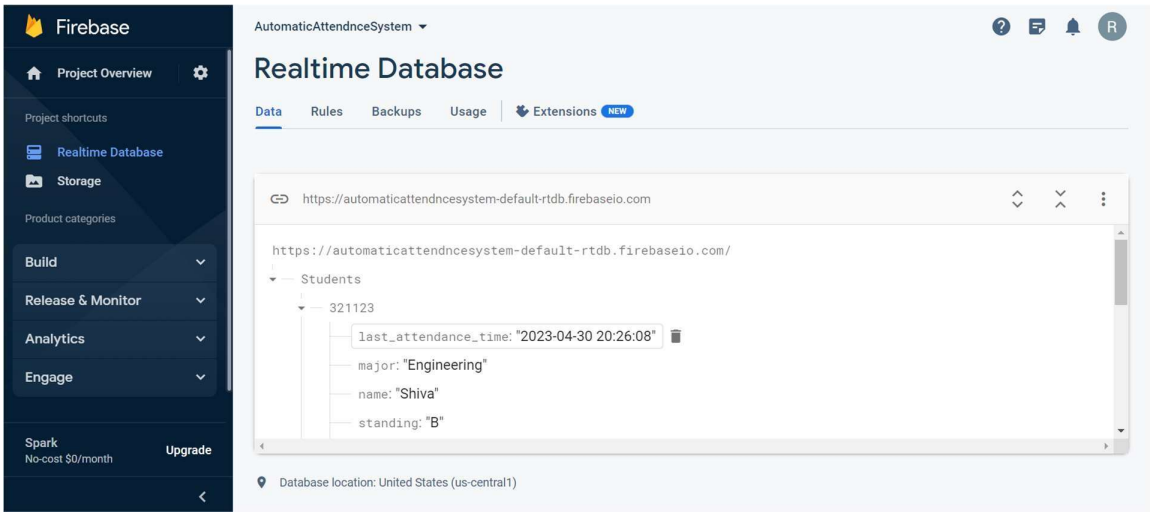

Figure 6.4

The above picture shows the real time database recordings.

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

In conclusion, the proposed automatic attendance system using CNN-based facial recognition technology is a fast and accurate way to record attendance. It overcomes the limitations of traditional attendance systems and previous facial recognition techniques.With the added features of in-time and out-time recording and liveness detection measures, this system provides an enhanced user experience and improved security. It has the potential to revolutionize attendance recording processes in educational institutions and other organizations.With facial recognition technology, predictive analytics, mobile app integration, and robust security measures, this system is paving the way for a more efficient and effective approach to attendance tracking. As technology continues to advance, we can expect even more exciting developments in this field.

7.2Future Enhancement

Improved accuracy: Although CNN-based face recognition systems are highly accurate, improvements can still be made to increase the accuracy of the system. This can be achieved by using larger datasets for training, fine-tuning the hyperparameters, and implementing advanced techniques like ensemble learning.

Integration with other biometric technologies: In addition to face recognition, the system can be enhanced to integrate with other biometric technologies such as fingerprint recognition or iris recognition for even more accurate identification.

Cloud-based implementation: Implementing the system on the cloud can improve scalability and reduce the computational requirements on local devices. This can be achieved by using cloud services like Amazon Web Services or Microsoft Azure.

REFERENCES

[1] D. Joseph, M. Mathew, T. Mathew, V. Vasappan and B. S Mony. “Automatic Attendance System using Face Recognition”, International Journal for Research in Applied Science and Engineering Technology. vol. 8, pp. 769-773, 2020. doi: 10.22214/ijraset.2020.30309.

[2] Dr.R.S Sabeenian, S. Aravind, P. Arunkumar and P.Joshua, and G. Eswarraj. “Smart Attendance System Using Face Recognition”, Journal of Advanced Research in Dynamical and Control Systems. vol. 12, pp. 1079-1084, 2020. doi: 10.5373/JARDCS/V12SP5/20201860.

[3] G. Al-Muhaidhri and J. Hussain, “Smart Attendance System using Face Recognition”, International Journal of Engineering Research & Technology (IJERT), vol 8, pp. 51-54, 2017. doi: 10.17577/IJERTV8IS120046.

[4] P. Chakraborty, C.Muzammel, M. Khatun, Sk. Islam and S. Rahman, “Automatic Student Attendance System Using Face Recognition”, International Journal of Engineering and Advanced Technology (IJEAT), vol. 9, pp. 93-99, 2020. doi: 10.35940/ijeat.B4207.029320.

[5] S. Saypadith and S. Aramvith, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System", 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, pp. 1318-1324, 2018. doi: 10.23919/APSIPA.2018.8659751.

[6] K. Bertók and A. Fazekas, "Face recognition on mobile platforms", 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, pp. 000037-000042, 2016. doi: 10.1109/CogInfoCom.2016.7804521.

[7] R. Padilha et al., "Two-tiered face verification with low-memory footprint for mobile devices", IET Biometrics, vol. 9, no. 5, pp. 205- 215, 2020. doi: 10.1049/iet-bmt.2020.0031.

[8] W. Jiang and W. Wang, "Face detection and recognition for home service robots with end-to-end deep neural networks", 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 2017, pp. 2232-2236. doi: 10.1109/ICASSP.2017.7952553.

[9] W. S. M. Sanjaya, D. Anggraeni, K. Zakaria, A. Juwardi and M. Munawwaroh, "The design of face recognition and tracking for humanrobot interaction," 2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE), Yogyakarta, pp. 315-320, 2017. doi: 10.1109/ICITISEE.2017.8285519.

[10] D . Mery, I. Mackenney and E. Villalobos, "Student Attendance System in Crowded Classrooms Using a Smartphone Camera", 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa Village, HI, USA, pp. 857-866, 2019. doi: 10.1109/WACV.2019.00096.

[11] M. A. Rahman and S. Wasista, “Sistem Pengenalan Wajah Menggunakan Webcam Untuk Absensi Dengan Metode Template Matching,” Jur. Tek. Elektron. Politek. Elektron. Negeri Surabaya, pp. 1–6.

[12] E. Indra et al., "Design and Implementation of Student Attendance System Based on Face Recognition by Haar-Like Features Methods," 2020 3rd International Conference on Mechanical, Electronics, Computer, and Industrial Technology (MECnIT), Medan, Indonesia, pp. 336-342, 2020. doi: 10.1109/MECnIT48290.2020.9166595.

[13] S. Sawhney, K. Kacker, S. Jain, S. N. Singh and R. Garg, "Real-Time Smart Attendance System using Face Recognition Techniques," 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, pp. 522-525, 2019. doi: 10.1109/CONFLUENCE.2019.8776934.

[14] P. Viola and M. J. Jones, "Robust real-time face detection", International journal of computer vision 57.2 (2004), pp. 137–154, 2004.

[15] P. Kowsalya,J. Pavithra, G. Sowmiya and C.K. Shankar. “Attendance monitoring system using face detection & face recognition”, International Research Journal of Engineering and Technology (IRJET),vol. 06, no. 03, pp. 6629-6632, March 2019. ISSN: 2395- 0072.