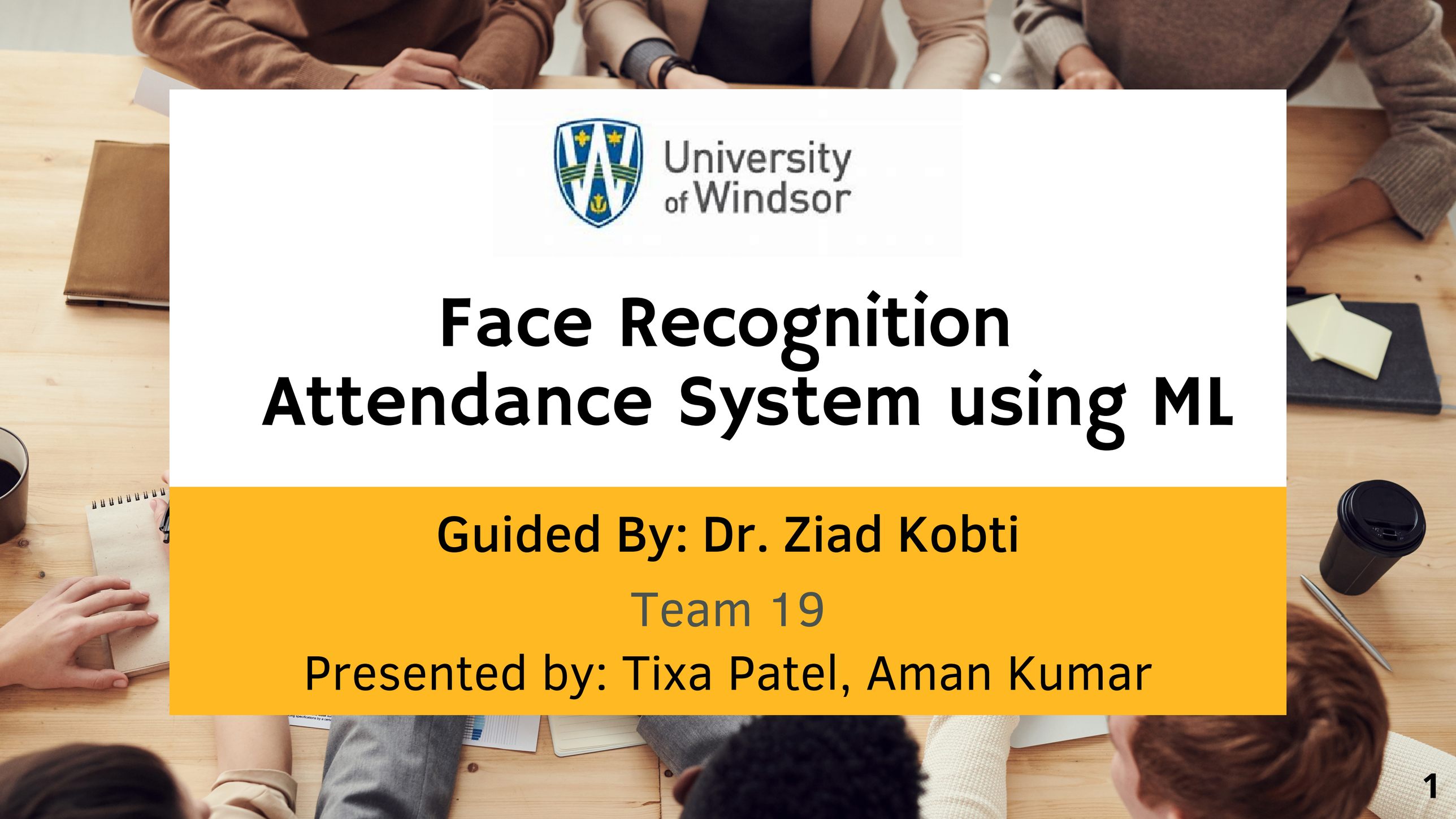


University of Windsor

Face Recognition Attendance System using ML

Guided By: Dr. Ziad Kobti

Team 19

Presented by: Tixa Patel, Aman Kumar

Agenda

- Introduction
- Problem and Motivation
- Hypothesis
- Approach
- Tools and Technologies
- Demo
- Experiments
- Results
- Conclusion
- Future Work
- References

Introduction

- In computer vision, face detection is a well-studied problem. Near frontal faces are easily detected by modern face detectors.
- Tracking students' work hours and activities is a required component of any university system. Automatic solutions have become a common choice for business processes.
- Computer vision and machine learning algorithms are heavily used in these systems. Although these solutions may be more adaptable and less prone to human error.
- Face recognition software can recognize individuals in photographs, videos, or in real-time. Face recognition maps facial features from a picture or video using biometrics.

Problem and Motivation

Problem

Maintaining students' attendance records manually is a very difficult and time-consuming task.

Motivation

Face recognition systems can recognize or validate someone by looking at their face. The recognition module can be working on images or in real-time.

Solution

Face Recognition systems can be used for taking students' attendance by using facial data and matching students' information from the database.

Hypothesis

1. Active Student Detection method (ASD) can be adopted for this system to estimate the existence of a student sitting on the seat.

2. A great number of biometric systems are available in today's society. Face recognition, on the other hand, proves to be a feasible choice due to its high accuracy and little human interaction.

3. Face detection and identification can be performed using the Raspberry Pi module. The Raspberry Pi module will be attached to the camera.

4. The attendance system can be a web-based system. HTML, PHP, and CSS can be used to build the system. XAMPP can be used as a web server and MySQL as the database for the attendance system.

Approach

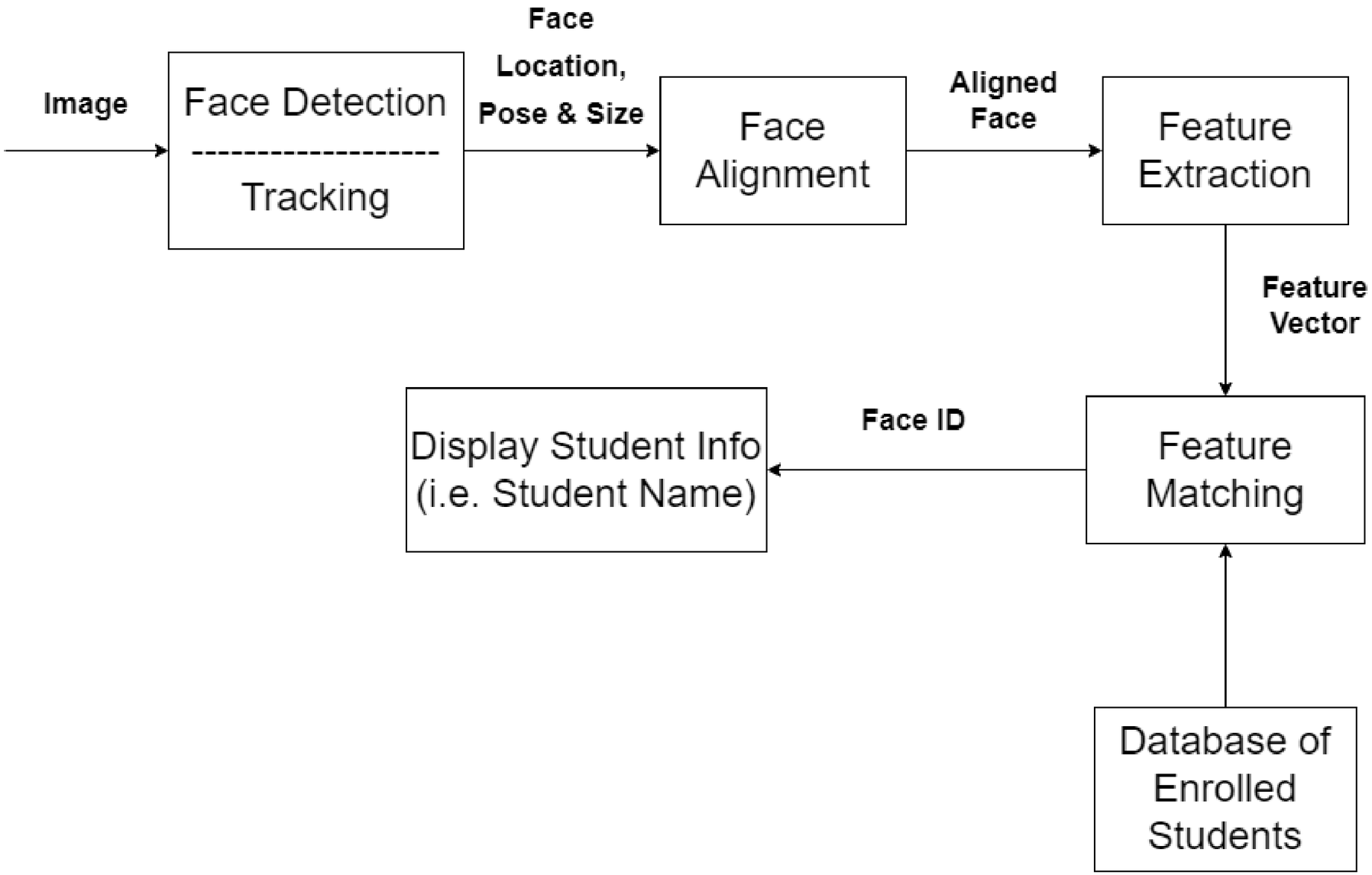


Image
Face Detection
Tracking
Face Location, Pose & Size
Face Alignment
Aligned Face
Feature Extraction
Feature Vector
Feature Matching
Face ID
Display Student Info (i.e. Student Name)
Database of Enrolled Students

Tools and Technologies

Programming Language: Python

Development Tools: Anaconda, PyCharm, Visual Studio Code

Machine Learning Libraries: NumPy, OpenCV, Face_Recognition, Dlib, Keras

Model: SVM Classifier

Datasets: Images of human faces

Launching our Face-Recognition: Attendance System Demo

Experiments

Importing NumPy, OpenCV, Face_Recognition Libraries

attendance.py | Attendance.csv | pyvenv.cfg | ob

```
1  import cv2
2  import numpy as np
3  import face_recognition
4  import os
5  from datetime import datetime
6
7  # from PIL import ImageGrab
8
9  path = 'images'
```

Experiments

Face Detection and Face Encodings of the images

attendance.py | Attendance.csv | pyvenv.cfg | obama.jpg | biden.jpg | obama2.jpg | main.py

```
import cv2
import face_recognition

img1 = face_recognition.load_image_file('images/obama.jpg')
img1 = cv2.cvtColor(img1, cv2.COLOR_BGR2RGB)
img1Test = face_recognition.load_image_file('images/obama2.jpg')
img1Test = cv2.cvtColor(img1Test, cv2.COLOR_BGR2RGB)

face = face_recognition.face_locations(img1)[0]
print(face)
encodeFace = face_recognition.face_encodings(img1)[0]
print(encodeFace)
cv2.rectangle(img1, (face[3], face[0]), (face[1], face[2]), (255, 0, 255), 2)

faceTest = face_recognition.face_locations(img1Test)[0]
encodeTestFace = face_recognition.face_encodings(img1Test)[0]
cv2.rectangle(img1Test, (faceTest[3], faceTest[0]), (faceTest[1], faceTest[2]), (255, 0, 255), 2)

```

Experiments

Find Face Encodings and Comparison of the Encoding of the Train and Test Data

attendance.py | Attendance.csv | pyvenv.cfg | obama.jpg | biden.jpg | obama2.jp

```
21  def findEncodings(images):
22      encodeList = []
23      for img in images:
24          img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
25          encode = face_recognition.face_encodings(img)[0]
26          encodeList.append(encode)
27      return encodeList
28
29
30  def markAttendance(name):
31      with open('Attendance.csv', 'r+') as f:
32          myDataList = f.readlines()
33          nameList = []
34          for line in myDataList:
35              entry = line.split(',')
36              nameList.append(entry[0])
37          if name not in nameList:
38              now = datetime.now()
```

Experiments

Function to Mark the Attendance of the Person

attendance.py | Attendance.csv | pyvenv.cfg | obama.jpg | biden.jpg | obama2.jpg | main.py

```
30  def markAttendance(name):
31      with open('Attendance.csv', 'r+') as f:
32          myDataList = f.readlines()
33          nameList = []
34          for line in myDataList:
35              entry = line.split(',')
36              nameList.append(entry[0])
37          if name not in nameList:
38              now = datetime.now()
39              dtString = now.strftime('%H:%M:%S')
40              f.writelines(f'\n{name},{dtString}')
41
42
43  #### FOR CAPTURING SCREEN RATHER THAN WEBCAM
44  # def captureScreen(bbox=(300,300,690+300,530+300)):
45  #     capScr = np.array(ImageGrab.grab(bbox))
46  #     capScr = cv2.cvtColor(capScr, cv2.COLOR_RGB2BGR)
47  #     return capScr
```

Experiments

- **Compare the face location and encodings of the webcam with the face encodings and Face location of the employee's dataset**

attendance.py | pyvenv.cfg | obama.jpg | biden.jpg | obama2.jpg | main.py

```
53
54  while True:
55      success, img = cap.read()
56      # img = captureScreen()
57      imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
58      imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)
59
60      facesCurFrame = face_recognition.face_locations(imgS)
61      encodesCurFrame = face_recognition.face_encodings(imgS, facesCurFrame)
62
63      for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
64          matches = face_recognition.compare_faces(encodeListKnown, encodeFace)
65          faceDis = face_recognition.face_distance(encodeListKnown, encodeFace)
66          # print(faceDis)
67          matchIndex = np.argmin(faceDis)
68
69          if matches[matchIndex]:
70              name = classNames[matchIndex].upper()
```

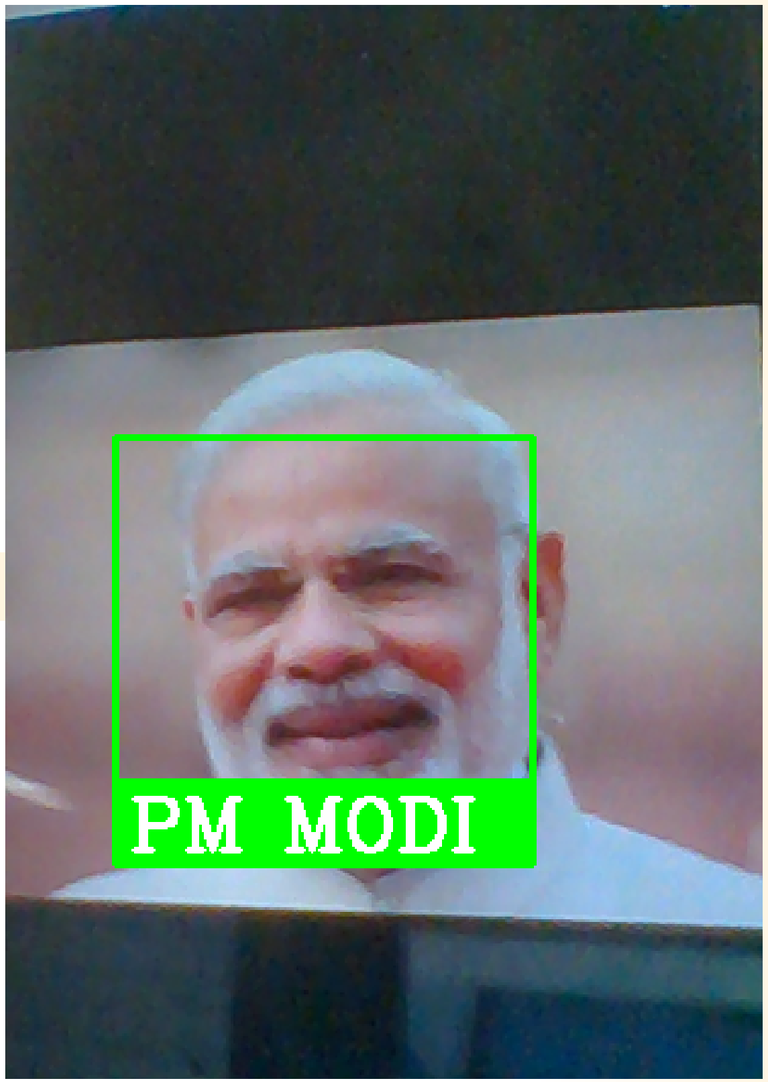
Results

I28 Face encodings values and true value after comparing test and train images in the Dataset

```
 -5.76497987e-04  7.06750378e-02  1.05678784e-02  2.74789371e-02
 -1.10814199e-01 -5.46599217e-02 -1.43718302e-01 -5.51207326e-02
  4.61458452e-02 -5.67783304e-02  4.44412343e-02  1.57387257e-01
 -2.53751606e-01  1.13612093e-01 -2.43759062e-02 -4.08405326e-02
  6.93118200e-02 -3.09808250e-03 -2.64254157e-02 -1.64143238e-02
  6.38433993e-02 -2.75689572e-01  2.61010110e-01  2.09015444e-01
  1.38789909e-02  1.82125285e-01  4.40698564e-02  4.82484512e-02
 -2.01499294e-02 -6.82892278e-04 -1.82867467e-01 -5.84296770e-02
  5.07246777e-02  8.45189467e-02  6.85785487e-02  1.59837231e-02]
[True] [0.38978046]
```

Results

Face detection



```
Name,Time
OBAMA,00:38:40
BIDEN,00:40:06
PM MODI,00:51:10
JUSTIN TRUDEAU,01:00:55
```

	Name	Time
1	Name	Time
2	OBAMA	0:38:40
3	BIDEN	0:40:06
4	PM MODI	0:51:10
5	JUSTIN TR	1:00:55
6		
7		
8		
9		

Results

Name of the Person and Time noted when he enters into the Office

Conclusion

AI-based attendance systems can be very helpful and can reduce human effort, but there is a lot of scope for improvement.

AI-based face recognition system can detect a large number of students at the same which is impossible to do manually.

A large dataset can be used to increase the accuracy of the model.

Hyper-parameter tuning can also be done to increase the accuracy of the model.

Future Work

One of the unexplored areas of this research is the analysis of additional solutions for classifying face embedding vectors

This deep learning-based solution can be implemented on GPU in runtime

Embedding CNN and RNN based models can give high accuracy for a smaller dataset

IoT modules can be introduced to the system to increase the functionality of the system and provide live student tracking

References

- Marko Arsenovic, Srdjan Sladojevic, Andras AnderlaDarko Stefanovic.FaceTime — Deep learning based face recognition attendance system In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY)

- Shreyak Sawhney, Karan Kacker,Samyak Jain, ,Shailendra Narayan Singh and Rakesh Garg Real-Time Smart Attendance System using Face Recognition Techniques In: 2019 9th International Conference on Cloud Computing, Data Science Engineering

- Yong-zhen Li,Zhi-heng Lin.Implementation of Classroom Attendance System Based on Face Recognition in Class In: 2019 International Conference on Intelligent Transportation, Big Data Smart City (ICITBS)

- Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das,Aurobinda Routray. Smart Attendance Monitoring System (SAMS): A Face Recognition Based Attendance System for Classroom Environment.In: 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT)

Thank you!

Feel free to approach us
if you have any questions.