



# DIGITAL IMAGE PROCESSING

## Automated Attendance System

Darshan M - CB.EN.U4CSE19126

Tarshit V -CB.EN.U4CSE19161

Raswanth S.R - CB.EN.U4CSE19648

Ramakrishnan - CB.EN.U4CSE19043

Group 15



# USE CASE DEFINITION :

- Attendance system using face recognition is a procedure of recognising students by using face biostatistics based on the high definition monitoring and other computer technologies.
- The system stores a database of faces.
- when a match is triggered with the database attendance is captured within a second.
- After that attendance reports will be generated and stored in excel format.

# PROBLEMS WITH EXISTING SYSTEM

**1**

**Risk of human error**  
These errors often occur in the workplace when you use manual time tracking solutions. Your personnel department can spend a lot of time correcting these errors.

**2**

**Obsolete Systems**  
The use of obsolete systems carries the risk of system crashes, security issues and loss of valuable information

**3**

**Ineffective and outdated**

Ultimately, the manual on system is ineffective and outdated. They can have a significant impact on your organizations performance. Clock errors can be costly to your organization

# OUR SOLUTION

To overcome the above solution specified, we have proposed a facial recognition based automated attendance system which is developed using Digital Image processing and Python.

## **Advantages of the proposed system :**

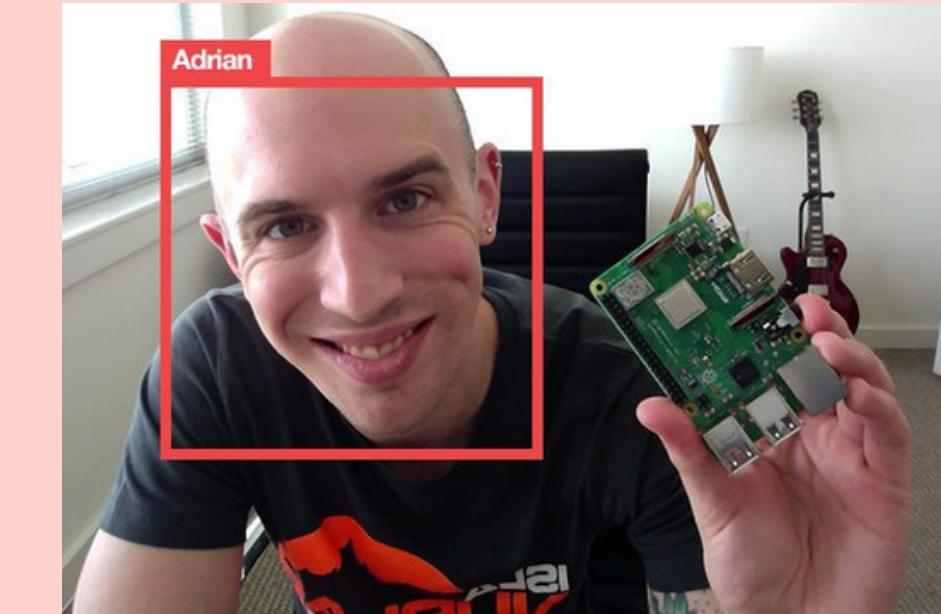
- Automated Time Tracking System
- Touchless Sign In System: A Post Pandemic Requirement.
- Accurate and improvised results.
- Increased Management and Security.
- Improvised Time management

# TOOLS/ PACKAGES USED



python

- 1.Numpy Module
- 2.OS Module



Face Recognition  
Module

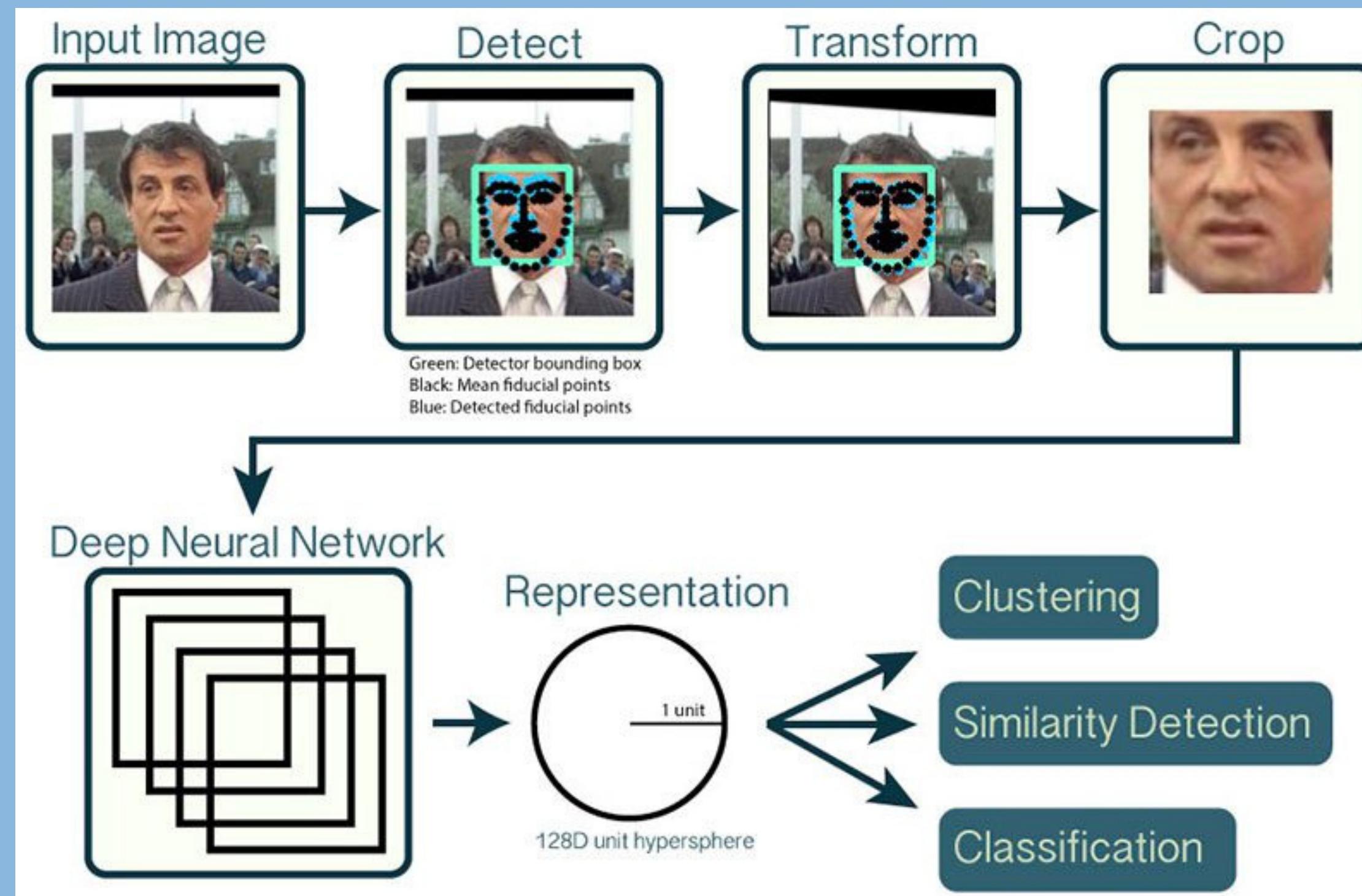


OpenCV Module  
for Image Processing



Date & Time Module

# FACE RECOGNITION WORKING



# PREPROCESSING - SHARPENING



Sharpening Input



Sharpening Output

# PREPROCESSING - SMOOTHING



Smoothing Input



Smoothing Output

# FUNCTIONS USED

## find Encodings

- this function create new list name encode list.
- we make images in image attendance file into black and white.
- using face\_recognition model we read all the images .
- send the data from above step into encode list.

# FUNCTIONS USED

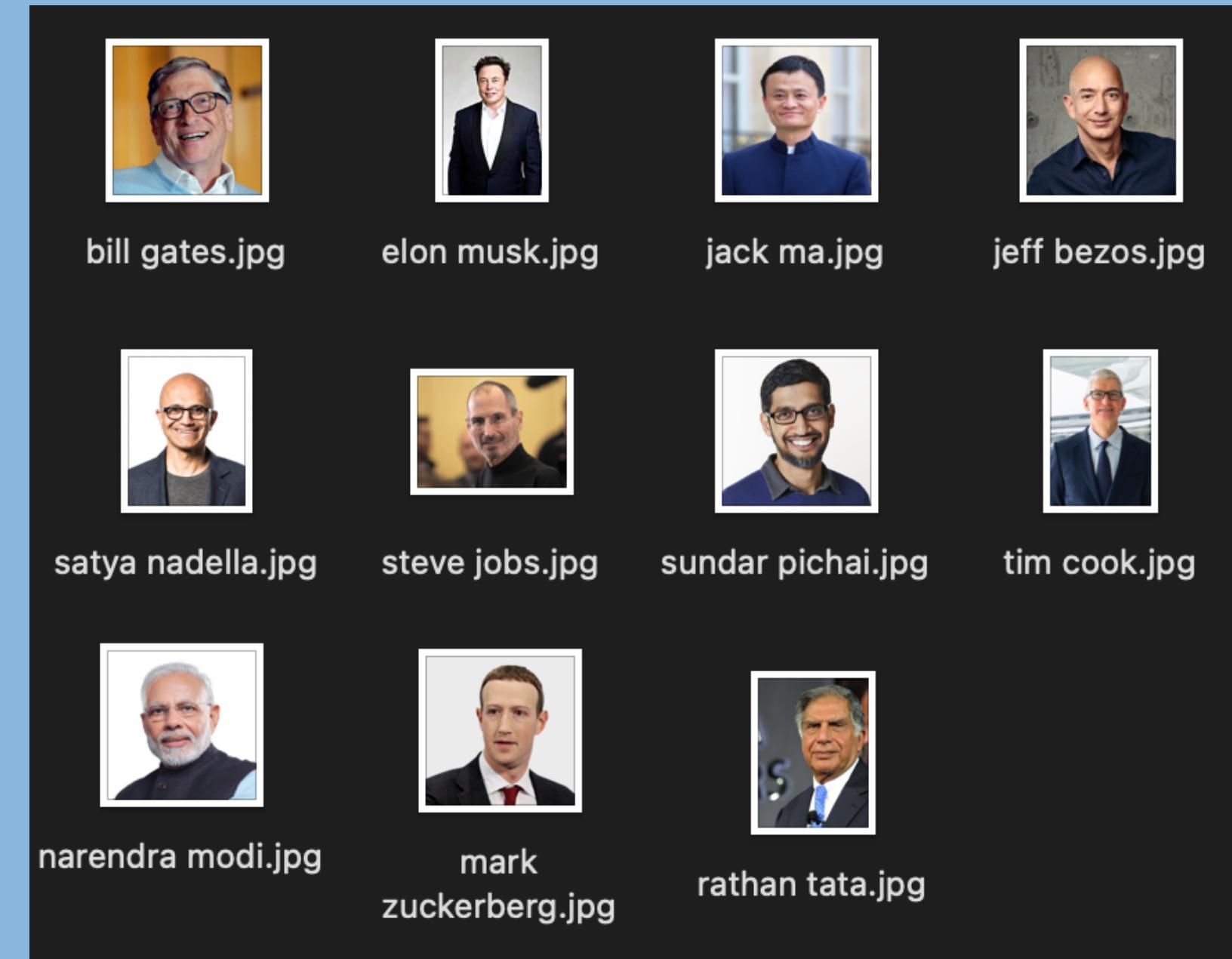
## markAttendance

- it reads the images from webcam store in a list
- it compares those images with images present in class file.
- if any images occur other than images in class file, it name it as unknown.
- otherwise it name at corresponding name in image attendance file.
- it finally update the attendance.csv file with names which it identified in web cam.

# «INPUT» FOR THE SYSTEM

## Automated Attendance System

Here we require all the attendees to stand in front of the camera for the system to analyze their faces and recognize it matching our system. This system when automated can effectively mark attendance.



Sample Attendees

# «INPUT» FOR THE SYSTEM

## Automated Attendance System

Here we require all the attendees to stand in front of the camera for the system to analyze their faces and recognize it matching our system. This system when automated can effectively mark attendance.



Webcam Input

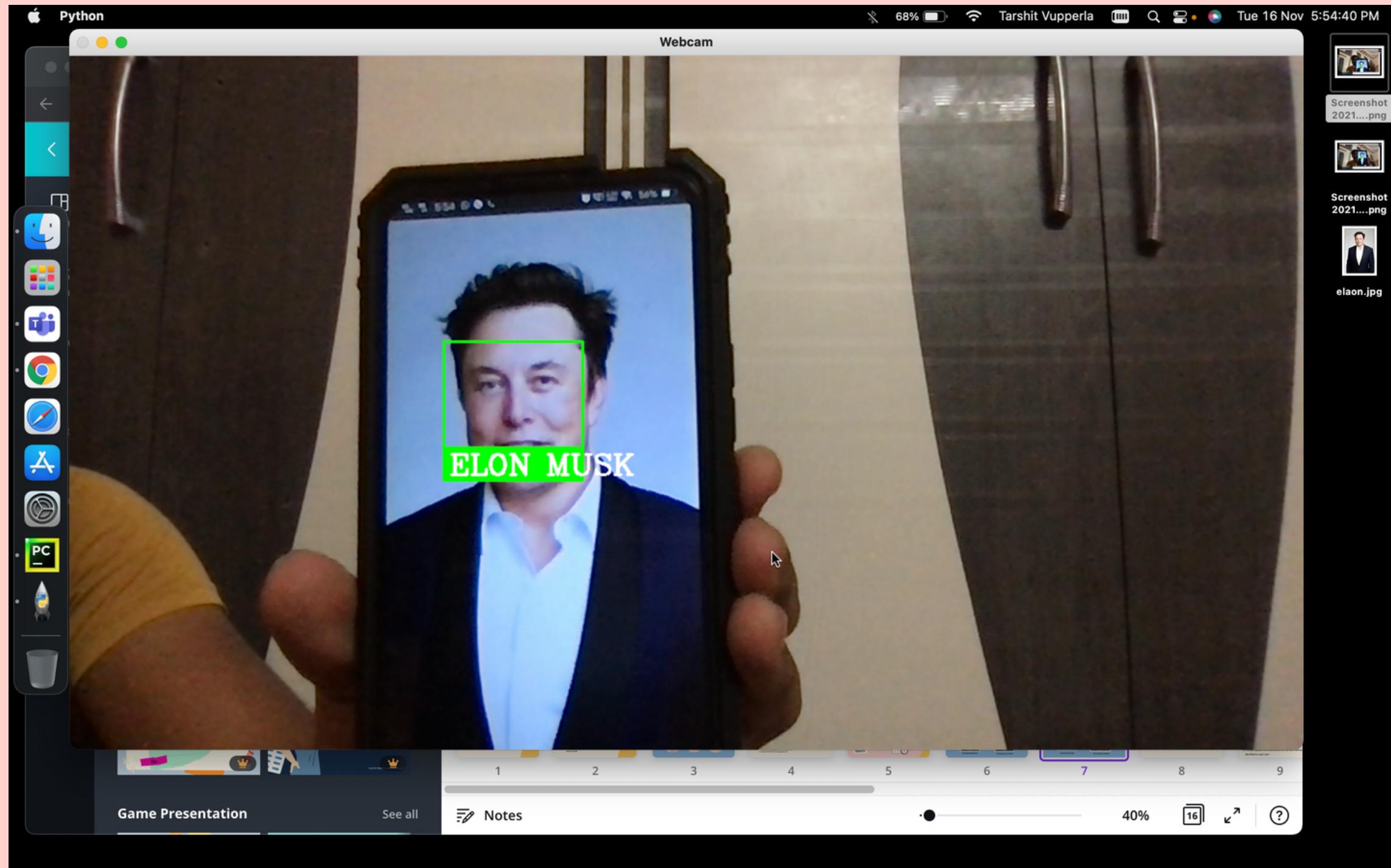
# OUTPUT

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** DIP – AttendanceProject.py
- Project Explorer (Left):** Shows the project structure under DIP ~/PycharmProjects/DIP\_Project. It includes files like Attendance.csv, bicubic.py, bilinear.py, Blurred\_Output.jpg, car.jpeg, CSE19126\_EVAL1.py, diff\_result.jpg, img11.jpg, img22.png, input.jpg, Input\_smooth.jpg, near.py, original.jpg, output\_bicubic.jpg, output\_bilinear.jpg, output\_near.jpg, Output\_smooth.jpg, plt\_smooth.png, and smoothing\_eval.py.
- Code Editor (Center):** The main editor window displays the code for AttendanceProject.py. The code imports cv2, numpy, face\_recognition, and os. It reads images from a directory and finds encodings for them.
- Run Tab (Bottom Left):** Shows the command 'AttendanceProject' and the output: 'Encoding Complete'.
- Toolbars and Status Bar (Bottom):** Includes tabs for Run, Terminal, Python Console, and Event Log. The status bar shows the time as 2:19 and the Python version as Python 3.8 (DIP).

# Compilation of the Code

# <OUTPUT>



## Face Recognition

# <OUTPUT>

Name	Time
ELON MUSK	17:12:33
NARENDRA MODI	17:12:56
Unknown	20:40:12

**Attendees List exported  
to csv**

# DEMO

# THANK YOU

## Future Improvements

- Data Analysis of the output data.
- Neural networks based image processing module for the system
- Improved accuracy in the overall system