AUTOMATIC ATTENDANCE SYSTEM USING FACIAL RECOGNITION

Submitted in partial fulfilment of the requirements for the degree of

Bachelor of Technology

in

Computer Science Engineering with specialization in Data Science

by

RASHI MAHESHWARI (19BDS0006)

ADARSH AGARWAL (20BCE2446)

SAUMYA MAHESHWARI (20BCE0196)

Under the guidance of

Prof. Gunavathi C

Scope

VIT, Vellore

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

DECEMBER 2021

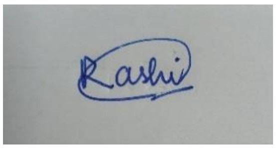
DECLARATION

We hereby declare that the thesis entitled "*Automatic Attendance System using Facial Recognition*" submitted by us, for the award of the degree of *Bachelor of Technology* in Computer Science Engineering with specialization in Data Science to VIT is a record of bonafide work carried out by us under the supervision of *Prof. Gunavathi C.*

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place: Vellore

Date: 01-12-21



Signature of the candidate

<u>CERTIFICATE</u>

This is to certify that the thesis entitled "*Automatic Attendance System using Facial Recognition*" submitted by ***Rashi Maheshwari (19BDS0006), Adarsh Agarwal (20BCE2446), Saumya Maheshwari (20BCE0196)***, VIT, for the award of the degree of *Bachelor of Technology in Programme*, is a record of bonafide work carried out by them under my supervision during the period, 01. 12. 2021 to 10.12.21, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfils the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place: Vellore

Date: 01-12-21

Signature of the Guide

Internal Examiner **External Examiner**

CONTENTS

9. REFERENCE

<u>ACKNOWLEDGEMENTS</u>

It is a great relief and satisfaction whenever we accomplish a college project. Whenever a project was completed then there will definitely be the support and guidance of our teachers, principal, friends, and family members. So it is always a good practice to acknowledge all the members who supported in completing your project.

I would like to express my sincere thanks to **<u>Prof. Gunavathi C</u>**, *for her valuable guidance and support in completing our project.*

I would also like to express my gratitude towards our **<u>INSTITUTION VIT</u>** *for giving us this great opportunity to do a project on* **<u>AUTOMATIC ATTENDANCE SYSTEM USING FACIAL RECOGNITION</u>**.

Without their support and suggestions, this project would not have been completed.

Executive Summary

This software project uses some basic Dot net API‘s to interact and get the output of local camera. It maybe a webcam or any other attached camera. We use these API’s to get the camera video input to our system.

We then use the video data to manipulate and recognize faces in real time. Our system works as follows:

- The user needs to start the system in visual studio.
- Now the system accepts camera input with the help of Dot net API’s.
- We now set some parameters and start live detection using the video input.
- Our system works and manipulates live video data to identify faces in it.
- Next step is to store these faces based on a matrix form.
- Once faces are recognized and stored we can now detect them.
- Whenever the face reappears it is recognized by name in a real time video and an attendance is marked for that particular person in the database.

The project is developed on python#.net platform

INTRODUCTION

- **OBJECTIVE:**

Over the past decade, taking down students' attendance process had been developed and changed. The driven force of this development is the desire to automate, facilitate, speed up and save time and efforts.

Our project targets the students of different academic levels and faculty members. It provides facility for the automated attendance of students and uses live face recognition to recognize each individual and mark their attendance automatically.

It also utilizes video and image processing to provide inputs to the system. This will also reduce manipulation of attendance record done by students and it will save time as well.

Face recognition is a computer technology that determines the location and size of a human face in a digital image.

- **MOTIVATION:**

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person.

Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities.

Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes.

Furthermore, Intel Company allows the users to use face recognition to get access to their online account.

Apple allows the users to unlock their mobile phone, iPhone X by using face recognition.

- **BACKGROUND:**

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision.

G., & Robinson-Riegler, B. (2008) mentioned that after visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyze the information. The analyzed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities; there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems.

The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person.

The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair colour and lip thickness to automate the recognition. In 1988, Kirby and Sirovich first suggested principle component analysis (PCA) to solve face recognition problem. Many studies on face recognition were then conducted continuously until today.

PROJECT DESCRIPTION AND GOALS

DESCRIPTION:

- Face recognition is a computer technology that determines the location and size of a human face in a digital image.
- It is a biometric approach that employs automated methods to verify or recognize the identity of a living person based on his/her physiological characteristics.
- In general, a biometric identification system makes use of either physiological characteristics or behavior patterns to identify a person.
- Our system uses this technology to record the attendance through a high resolution digital camera that detects and recognizes faces and compare the recognize faces with students' faces images stored in faces database.
- Once the recognized face matches a stored image, attendance is marked in attendance database for that person. The process will repeat if there are missed faces.
- For example, if there are 4 faces missed for a bad position while the detecting phase, then this phase will start again to detect the missed faces and recognize them and continue the attending process.
- Hence maintaining attendance automatically with the help of face recognition is very helpful and less prone to errors as compared to manual process.

METHODOLOGY:

- First, we get the location of where exactly the face is in the image .

- Then face encodings (markings of eyes, nose, mouth, jaws which remain the same for different images of the same person) are taken.
- Both these two steps are followed for the original and test image. Then a comparison between these two returned lists is done.
- The face distance function gets the value of that by how much the two images differ. The lower the distance the better the matching and vice versa.
- Now, a real-time face attendance system is built wherein webcam captured frames will be matched against the existing database images and if the match is found then it'll store it in a CSV file called 'Attendance Register' along with name and time of capture.
- Only once the file will store the matched image's details, if the same image is received again then it'll not update.

PROJECT GOAL:

- Taking and tracking students' attendance manually, losing attendance sheets, dishonesty, wasted time and high error scales are problems facing the lecturers use the existing attendance system.
- It is a hard process, take time and cause a lot of paper-based work.
- As a result, in order to solve these problems and avoid errors we suggest to computerize this process by providing a system that record and manage students' attendance automatically without needing to lecturers' interference.
- Face Recognition technology can be utilized to build an automated attendance system that makes counting and identifying students much easier and convenient.
- Face occlusions, face scaling, and posture are still important problems in such systems.

TECHNICAL SPECIFICATION

Software Requirements:

- Windows 10, Windows 7(ultimate, enterprise)
- Sql 2019
- Visual studio 2020

Hardware Components:

- Processor – i10
- Hard Disk – 5 GB
- Memory – 2GB RAM

DESIGN APPROACH AND DETAILS

➢ Comparison Among Algorithms

- **HAAR ALGORITHM:**

It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper “Rapid Object Detection using a Boosted Cascade of Simple Features” published in 2001. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them.

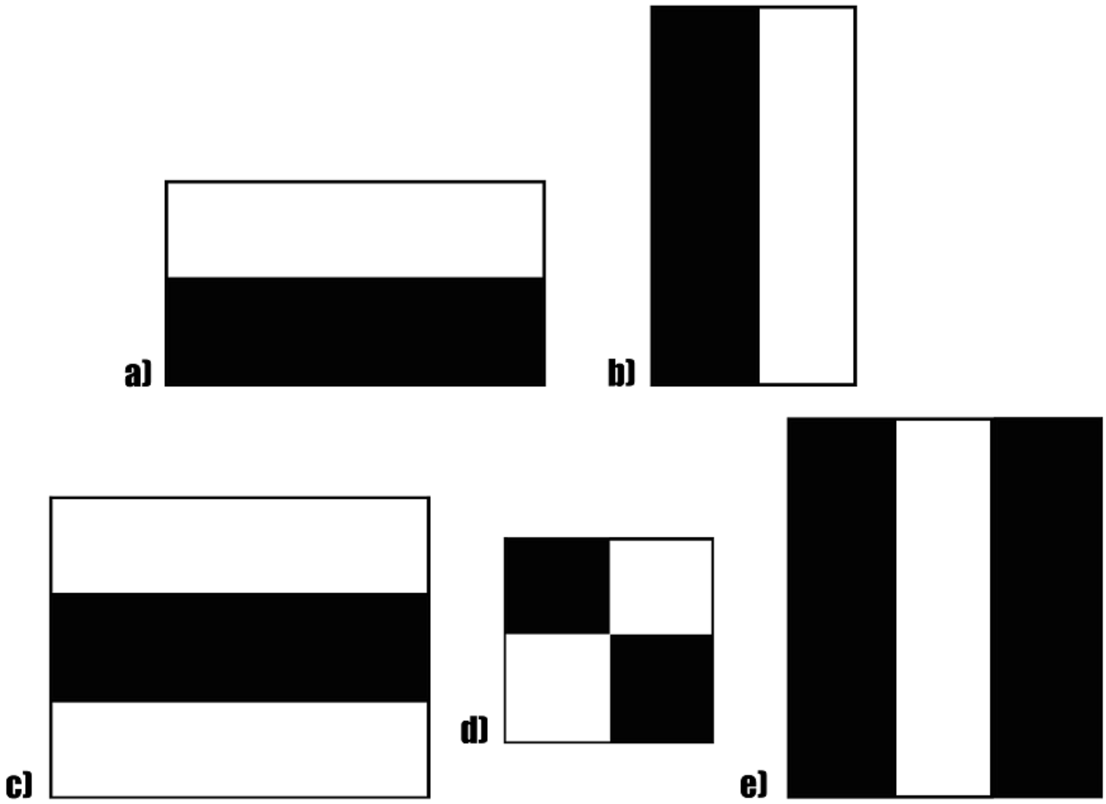


FIG.1

The first contribution to the research was the introduction of the haar features shown above. These features on the image makes it easy to find out the edges or the lines in the image, or to pick areas where there is a sudden change in the intensities of the pixels.

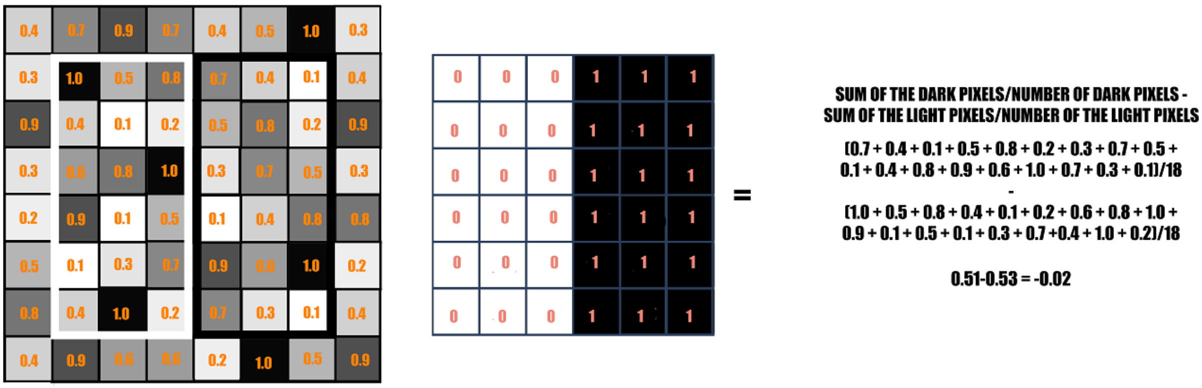


FIG.2

A sample calculation of Haar value from a rectangular image section has been shown here. The darker areas in the haar feature are pixels with values 1, and the lighter areas are pixels with values 0. Each of these is responsible for finding out one particular feature in the image. Such as an edge, a line or any structure in the image where there is a sudden change of intensities. For ex. in the image above, the haar feature can detect a vertical edge with darker pixels at its right and lighter pixels at its left.

The objective here is to find out *the sum of all the image pixels lying in the darker area of the haar feature* and *the sum of all the image pixels lying in the lighter area of the haar feature*. And then find out their difference. Now if the image has an edge separating dark pixels on the right and light pixels on the left, then the haar value will be closer to 1. That means, we say that there is an edge detected if the haar value is closer to 1. In the example above, there is no edge as the haar value is far from 1.

This is just one representation of a particular haar feature separating a vertical edge. Now there are other haar features as well, which will detect edges in other directions and any other image structures. To detect an edge anywhere in the image, the haar feature needs to traverse the whole image.

The haar feature continuously traverses from the top left of the image to the bottom right to search for the particular feature. This is just a representation of the whole concept of the haar feature traversal. In its actual work, the haar feature would traverse pixel by pixel in the image. Also all possible sizes of the haar features will be applied.

Depending on the feature each one is looking for, these are broadly classified into three categories. The first set of *two rectangle features* are responsible for finding out the edges in a horizontal or in a vertical direction (as shown above). The second set of *three rectangle features* are responsible for finding out if there is a lighter region surrounded by darker regions on either side or vice-versa. The third set of *four rectangle features* are responsible for finding out change of pixel intensities across diagonals.

Now, the haar features traversal on an image would involve a lot of mathematical calculations. As we can see for a single rectangle on either side, it involves 18 pixel value additions (for a rectangle enclosing 18 pixels). Imagine doing this for the whole image with all sizes of the haar features. This would be a hectic operation even for a high performance machine.

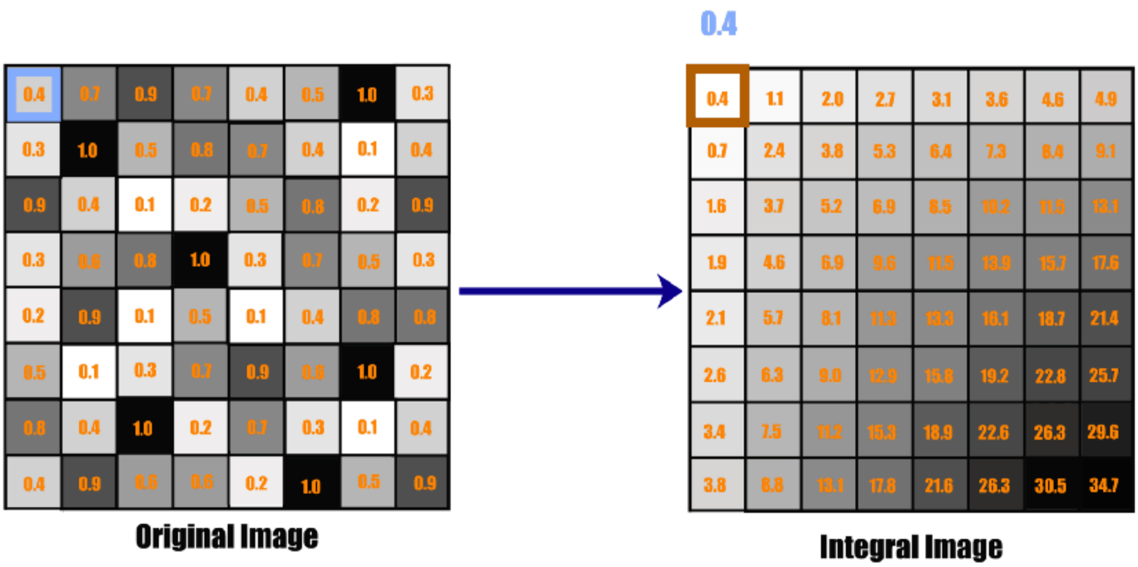


FIG. 3

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

Using the LBP combined with histograms we can represent the face images with a simple data vector.

As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following step-by-step explanation.

Step-by-Step

Now that we know a little more about face recognition and the LBPH, let's go further and see the steps of the algorithm:

1. Parameters: the LBPH uses 4 parameters:

- Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.
- Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.
- Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

Don't worry about the parameters right now, you will understand them after reading the next steps.

Training the Algorithm: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

<u>3. Applying the LBP operation</u>: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

The image below shows this procedure:

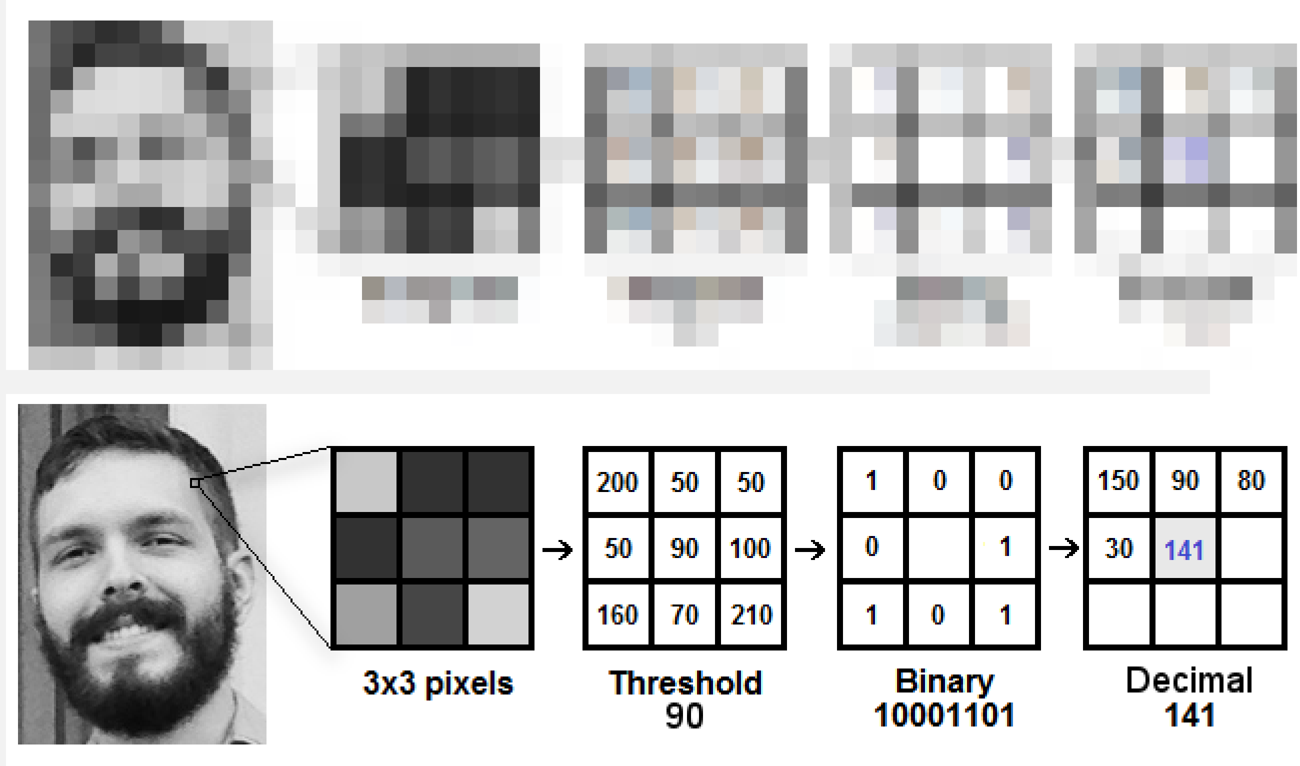


FIG. 4

Based on the image above, let's break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.

- This value will be used to define the new values from the 8 neighbors.

- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold.

- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.

- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.

- At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image.

- Note: The LBP procedure was expanded to use a different number of radius and neighbors, it is called Circular LBP.

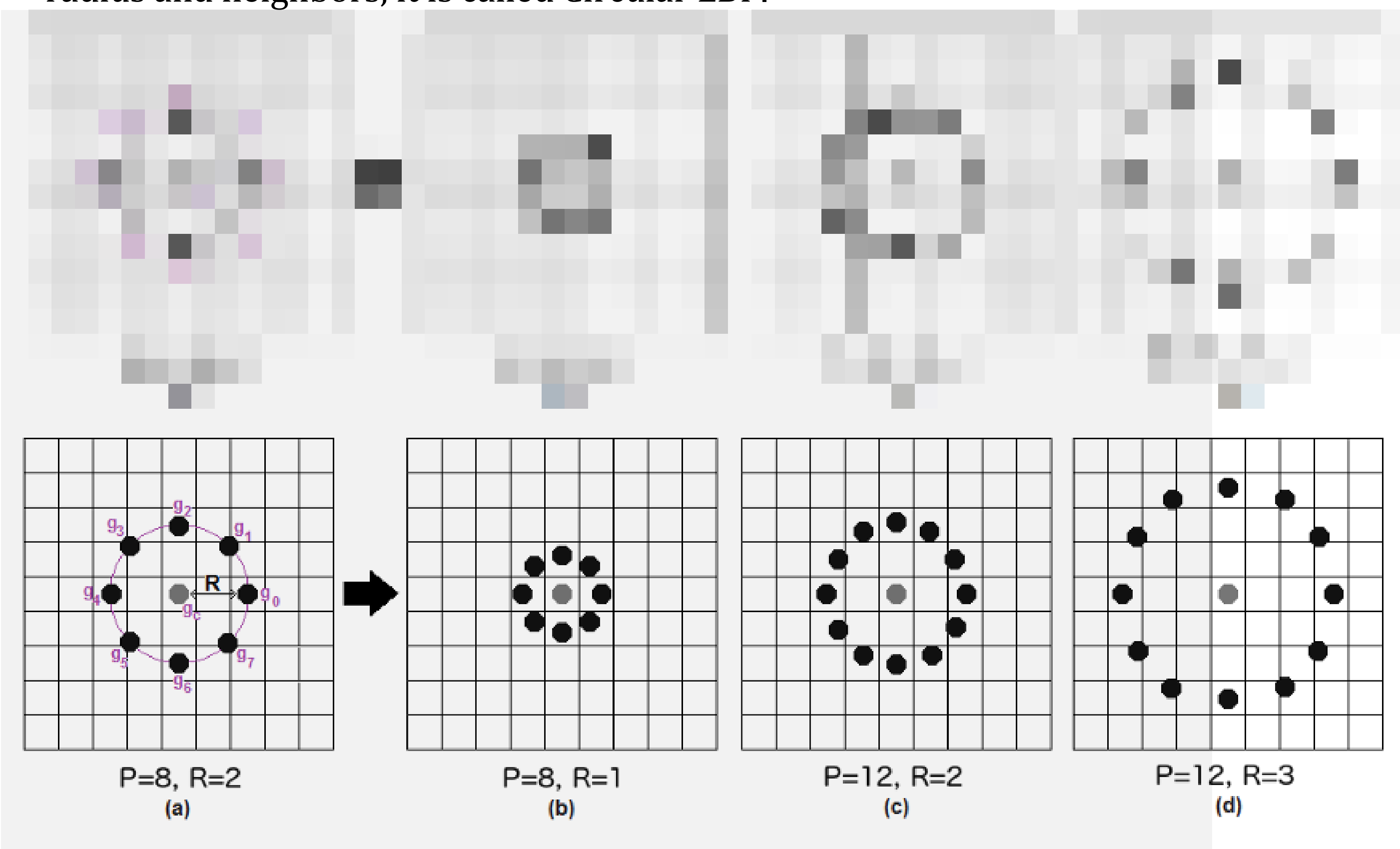


FIG. 5

It can be done by using bilinear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

4. Extracting the Histograms: Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image:

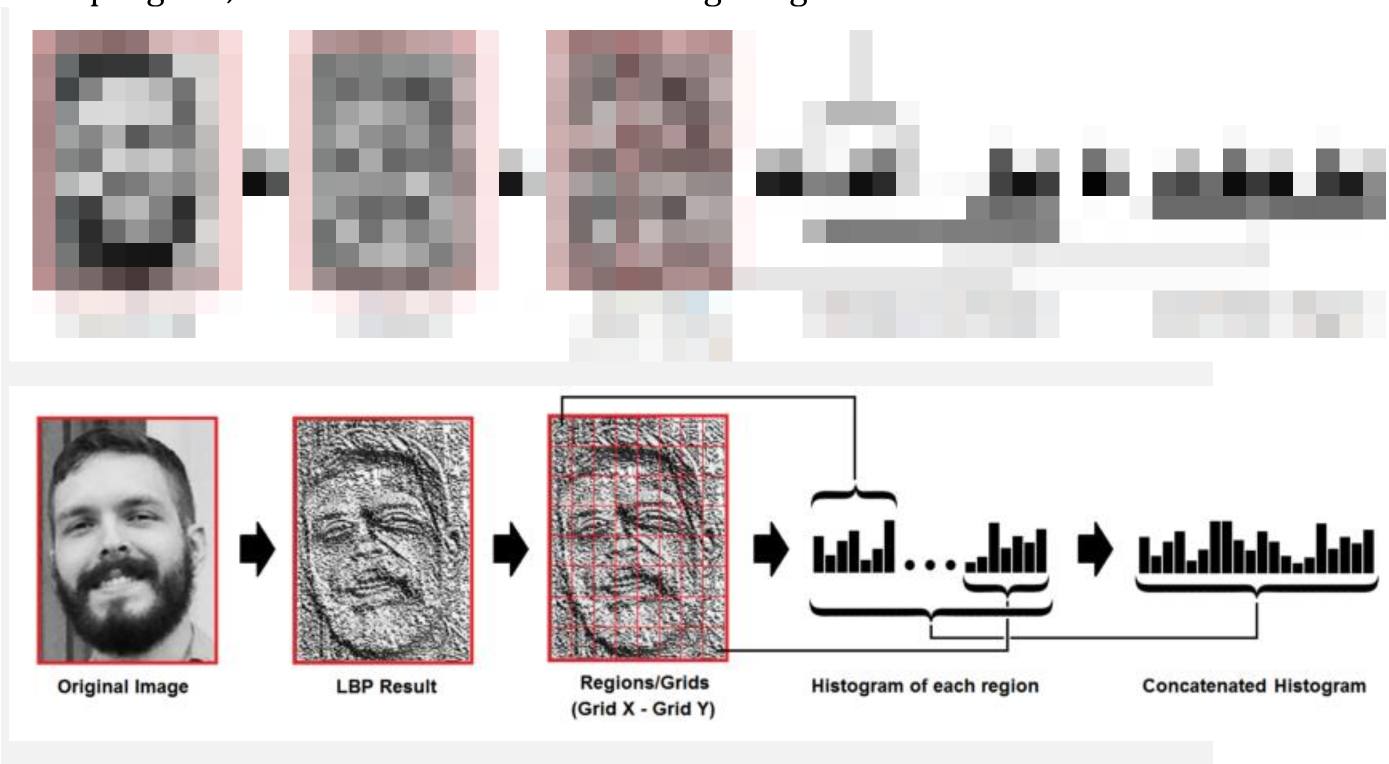


FIG. 6

Based on the image above, we can extract the histogram of each region as follows:

- As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.
- Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have 8x8x256=16.384 positions in the final histogram. The final histogram represents the characteristics of the image original image.

The LBPH algorithm is pretty much it.

5. Performing the face recognition: In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

- So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.
- We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: euclidean distance, chi-square, absolute value, etc. In this example, we can use the Euclidean distance (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^{n} (hist1_i - hist2_i)^2}$$

FIG. 7

- So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.
- We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

Conclusions

- LBPH is one of the easiest face recognition algorithms.
- It can represent local features in the images.
- It is possible to get great results (mainly in a controlled environment).
- It is robust against monotonic gray scale transformations.

- It is provided by the OpenCV library (Open Source Computer Vision Library).

• PROPOSED METHOD:

In our procedure comprises camera is used catching of the pictures of lecture hall image then directing pictures to picture enlargement segment. Spitting image rises identification and appreciation of the modules after theimage improvement and being present noticeable at particular time to the recorder system.

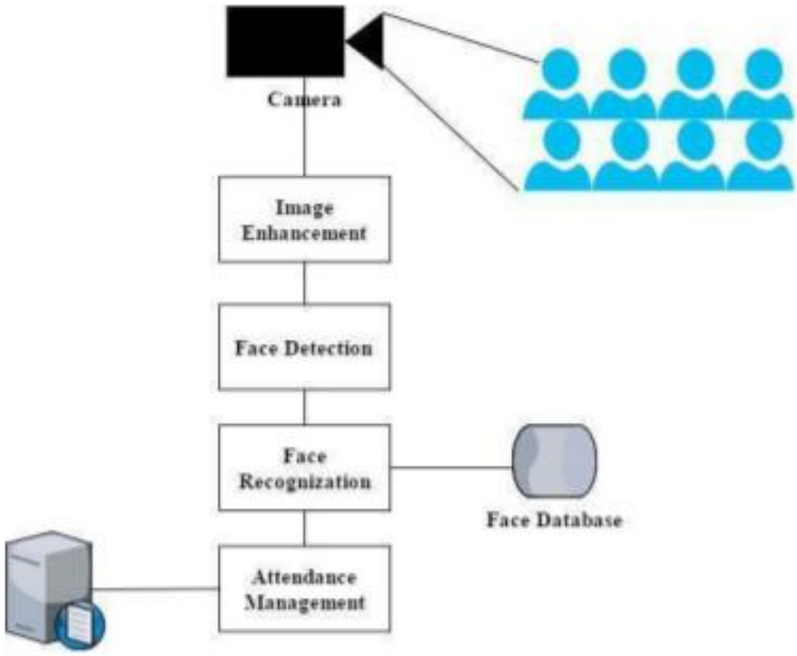


FIG. 8

5.1 Algorithm flowchart

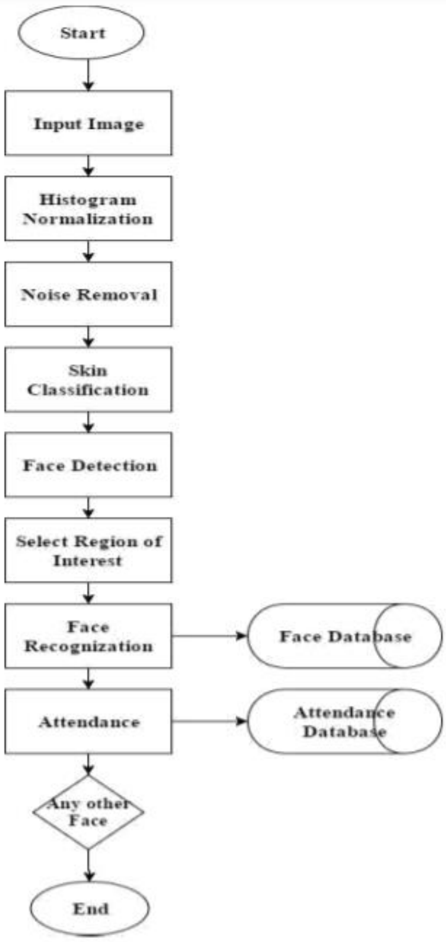


FIG. 9

In our work, we acknowledge the fact that capability of the LBP face recognition algorithm highly depends on the accuracy performance of feature extraction and comparison stage, which also highly depends on the quality of both the input face images and the training/reference images participating in the face comparison process. In order to improve the face recognition accuracy of the LBP algorithm, we exploited the following image quality properties of our input and reference face images: illumination, sharp, noise, resolution, scale and pose, so as to obtain the best quality images that will expose better details of image features for more accurate feature extraction and comparison.

Our first improvement approach is to apply the Contrast Adjustment method on our input face images as defined in equation (1). We tested this method with different values of alpha and beta to select the one that gives the best detection and recognition accuracy result, which are 1.5 (α) value and 0.0 (β) value.(1)$g(x,y)=\alpha*f(x,y)+\beta$

We selected the filter that gives the best result in our case, which is the bilateral filter as defined in equation (2).(2)F(x,y)=∑x=−NN∑y=−NNI(x,y)W(x,y)∑x=−NN∑y=−NNW(x,y)where W(x,y) is the filter weighting function, I(x,y) is the input face image neighborhood pixel and the denominator is the normalization of the weighting function, and F(x,y) is the result of the bilateral filter applied on a 2N+1 neighborhood. Now we defined CF(x,y) in equation (3) as the function to reduce noise and control contrast effects in the input images, where g(x,y) in equation (3) is the contrasted image and F(x,y) is the applied filter.(3)CF(x,y)=g(x,y)∗F(x,y)

The resultant image pixels derived from the above equation are equalized using the image histogram equalization method defined in equation (4) to finally address the global lightning issues in the processed facial images.(4)Eq=H’(CF(x,y))where H’ is the normalized cumulative distribution with a maximum value of 255 as in Ref. [28], finally, we applied the LBP algorithm on our detected face images for feature extraction & comparison.

A more formal description of the LBP operator can be given as follows in equation (5):(5)LBPp,r(Xc,Yc)=∑p=0p−12pS(ip−ic)where (Xc,Yc) is gray-level value of the center pixel with ip and ic being the intensity of the neighbor pixel and *p* the surrounding pixels in the circle neighborhood with a radius *r*, and S(X) is the sign function defined in equation (6), it’s used to threshold the fixed 3X3 neighborhood as in Ref. [29].(6)S(X)={1ifx≥00ifx<0
In our approach, we demonstrated the effect of using equations (1), (2), (3), (4) to improve the overall quality of our input face images so as to boost the accuracy performance of the LBP face recognition algorithm as shown in Fig. below.

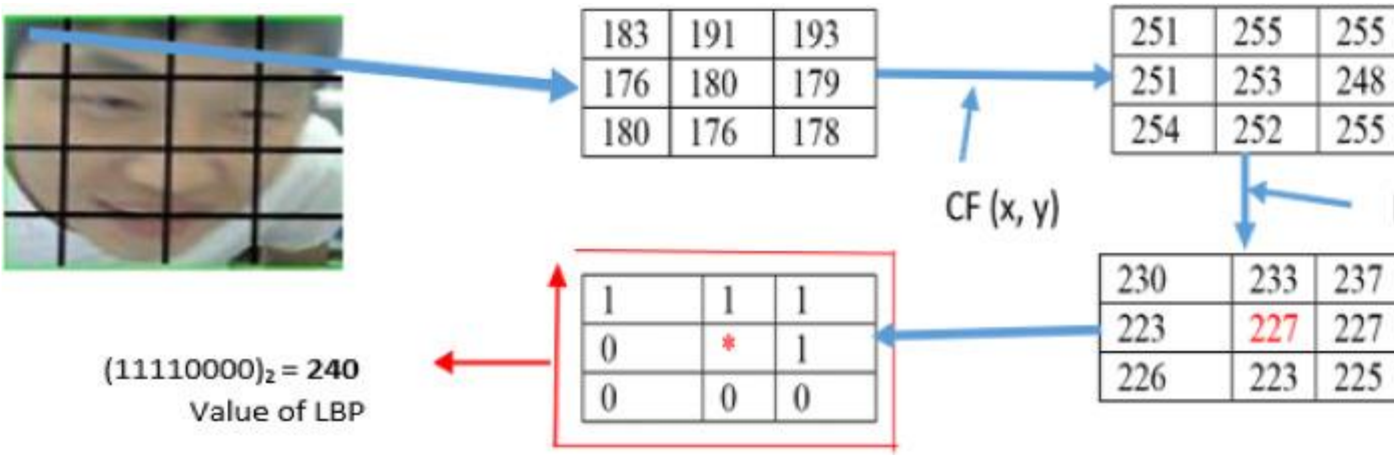


Fig. 10. The modified LBP operator.

Examining each window of the 3 × 3 neighborhood pixels of the resultant images from equations (3), (4)), the issues of noise, illumination, sharp, and resolution were highly mitigated. We obtained higher pixel and threshold values as shown in Fig. 2, after applying equations (1)–(4) to the original input images, we obtained an improved high quality images using our method, which implies better image quality that will expose better image features and enabled more accurate image histograms for facial image comparison as shown in section 4, Fig. 3 (a) to (d), thus resulted in an improved LBP codes, that will stimulate the overall face recognition accuracy.

(a) Original Image (b) Contrasted Image (c) Filtered Image (d) Equalized Image

Fig.11 Shows the original image and processed images defined in above equations

	Total Faces	Haar	LBP	Proposed Method
True Positive	226	208	206	**215**
False Positive	226	18	20	**11**
False Negative	226	55	41	**33**
Detection Accuracy Rate		**92%**	**91%**	**95%**

Comparing the two methods in FIG., it is shown that our proposed method outperformed, the Haar Classifier algorithm for Face Detection and also in the number of detected faces, represented as true positive or false positive, represented as features that are incorrectly detected as faces or false negative, represented as faces that are not detected as faces in an image. In Table, for face detection, the Haar algorithm, the original LBP and the improved LBP using our method are compared using a dataset of 226 images, which our method perform better in all cases.

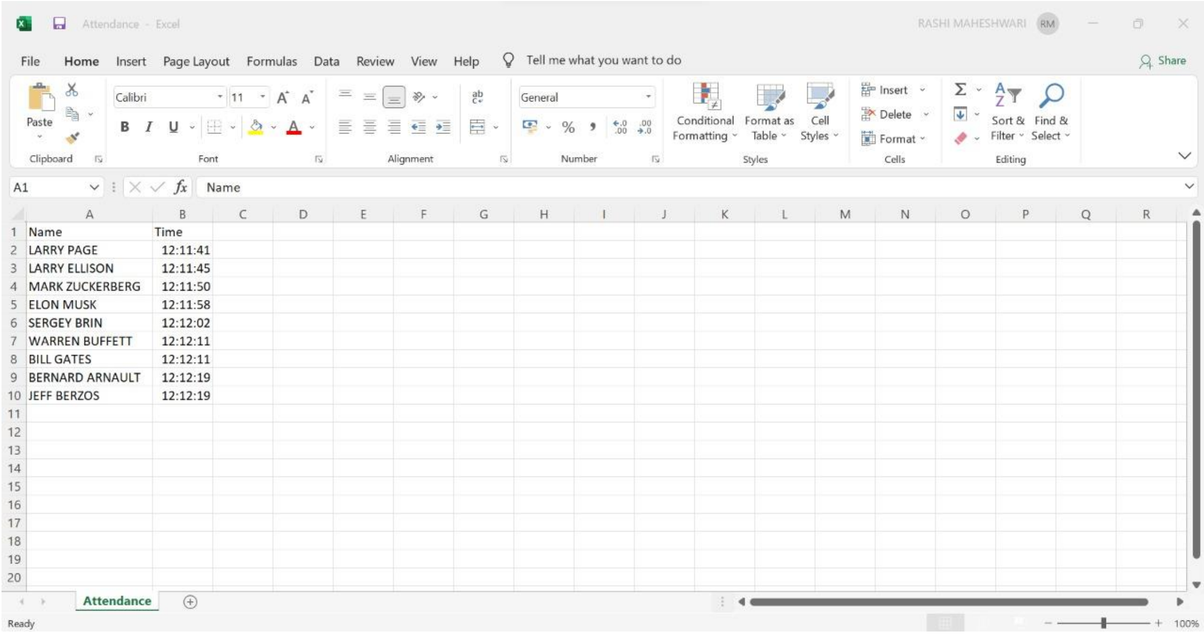
➢ Codes and Standards

```
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime


path = 'Train_Images'
images = []
className = []
myList = os.listdir(path)
print(myList)
for c1 in myList:
    curImg = cv2.imread(f'{path}/{c1}')
    images.append(curImg)
    className.append(os.path.splitext(c1)[0])
print(className)


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

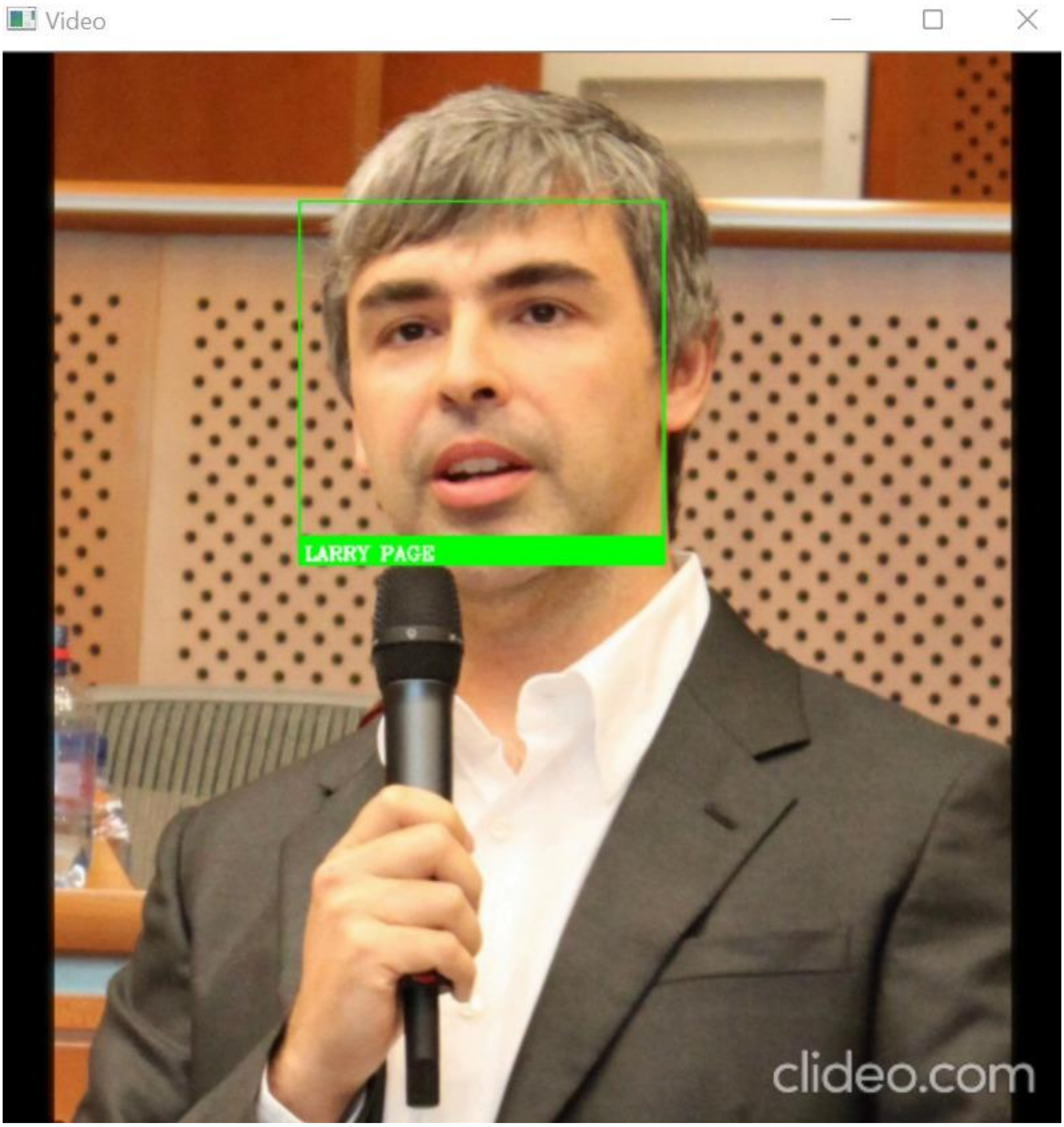
```
def markAttendance(name):
    with open('Attendance.csv','r+') as f:
        myDataList = f.readlines()
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            dtString = now.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dtString}')

print('Starting Encoding...')
encodeListKnow = findEncodings(images)
print('Encoding Completed')

cap = cv2.VideoCapture("Test_Images/Test5.mp4")

while True:
    success, img = cap.read()
    imgS = cv2.resize(img, (0, 0), None, 0.25, 0.25)
    imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

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

    for encodeFace, faceLoc in zip(encodeCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnow, encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnow, encodeFace)
```

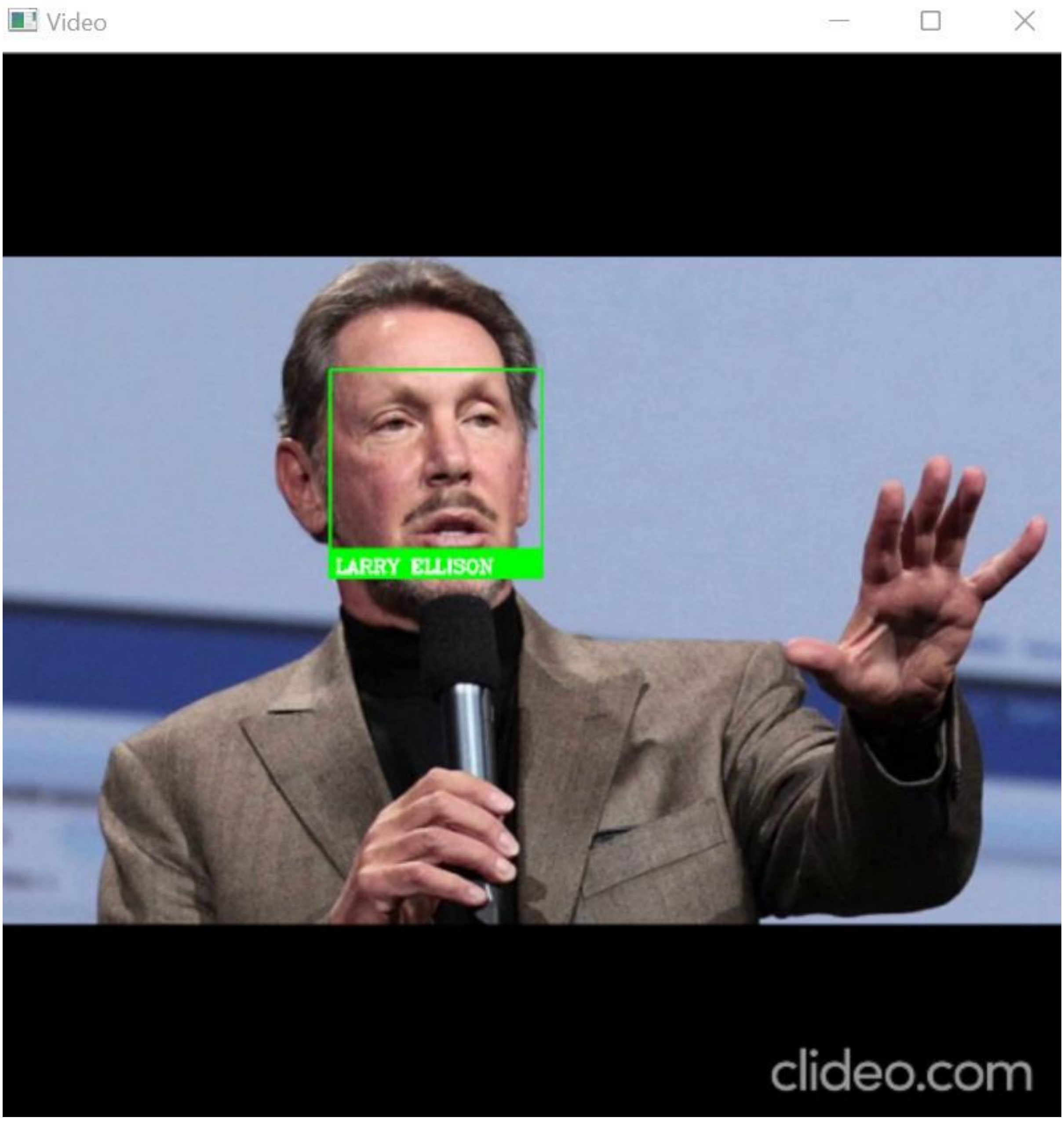
```
        #print(faceDis)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = className[matchIndex].upper()
            #print(name)
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2-35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 0.75,
(255,255,255), 2)
            markAttendance(name)

    im = cv2.resize(img, (540, 540))
    cv2.imshow('Video', im)
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break
```

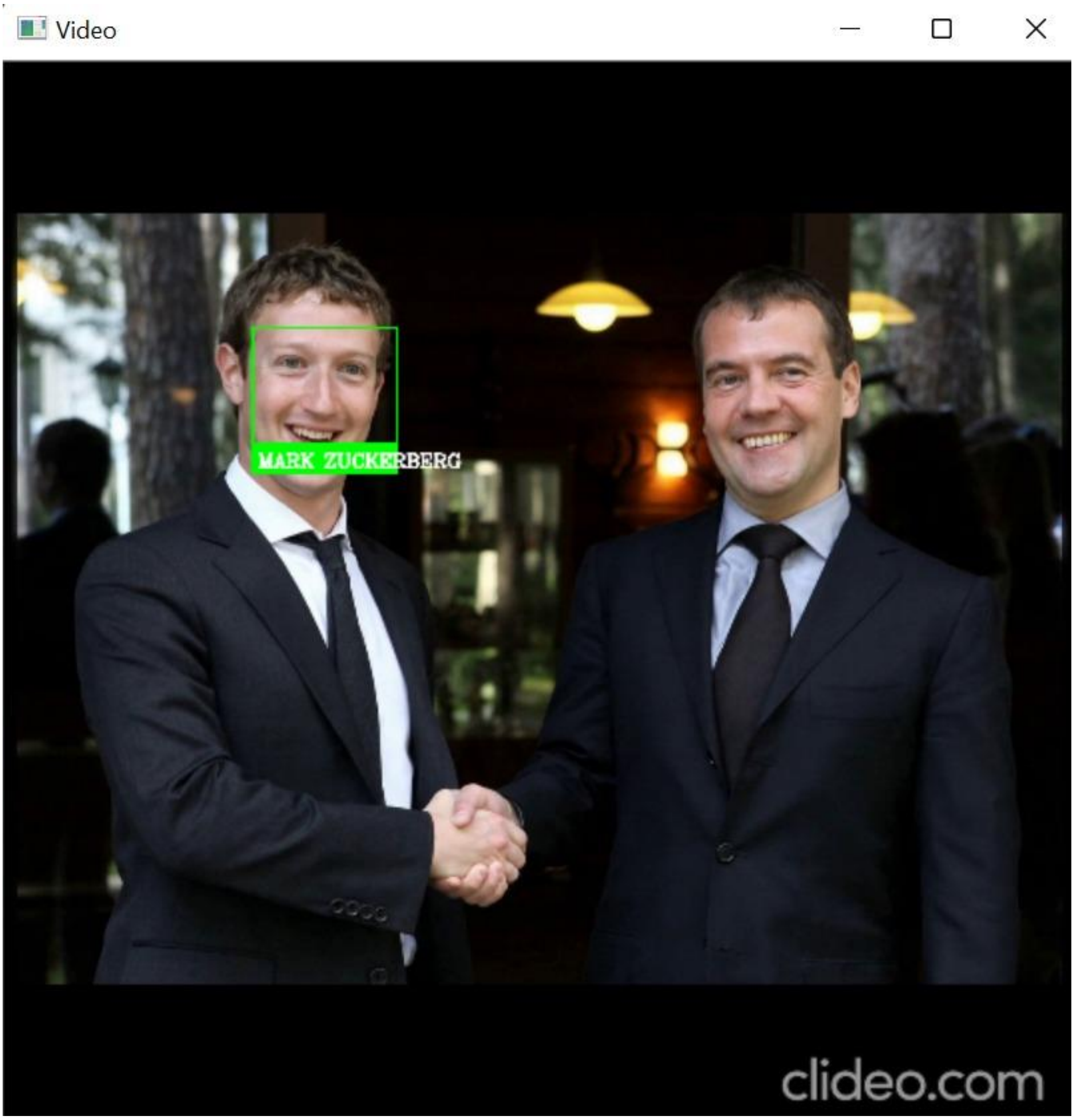
Sample Images:

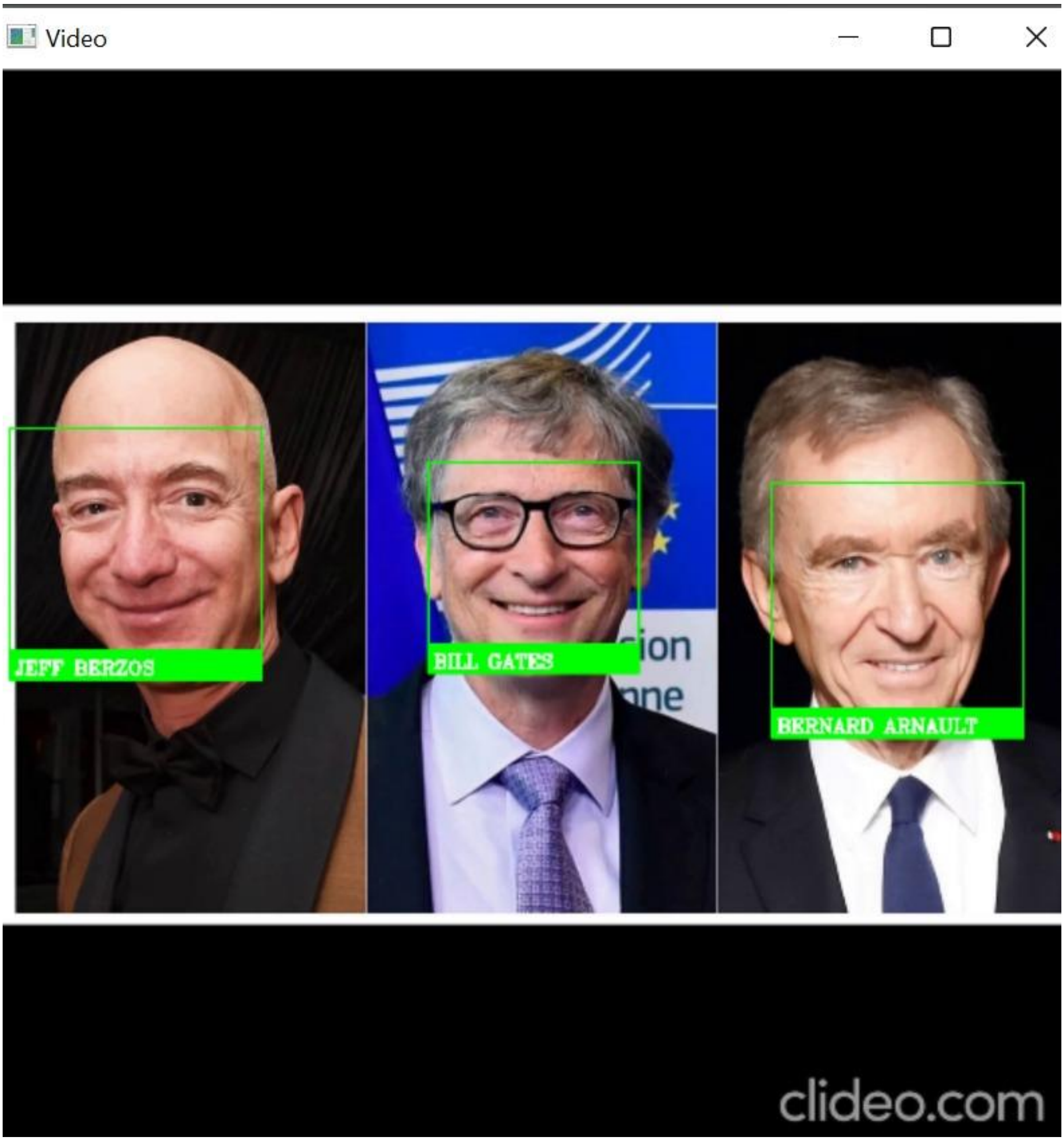


Name	Time
LARRY PAGE	12:11:41
LARRY ELLISON	12:11:45
MARK ZUCKERBERG	12:11:50
ELON MUSK	12:11:58
SERGEY BRIN	12:12:02
WARREN BUFFETT	12:12:11
BILL GATES	12:12:11
BERNARD ARNAULT	12:12:19
JEFF BERZOS	12:12:19

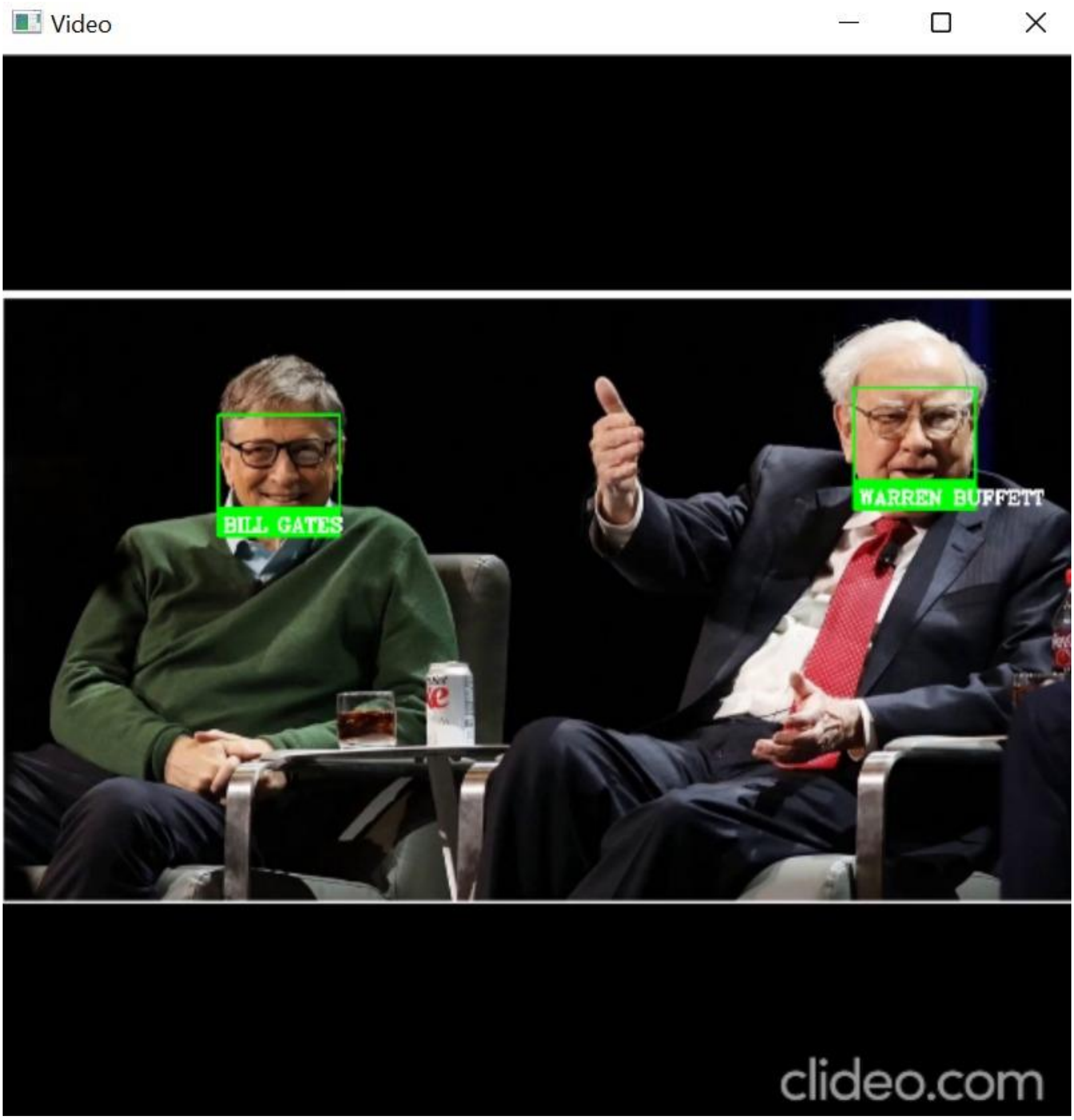

Video
LARRY PAGE
clideo.com

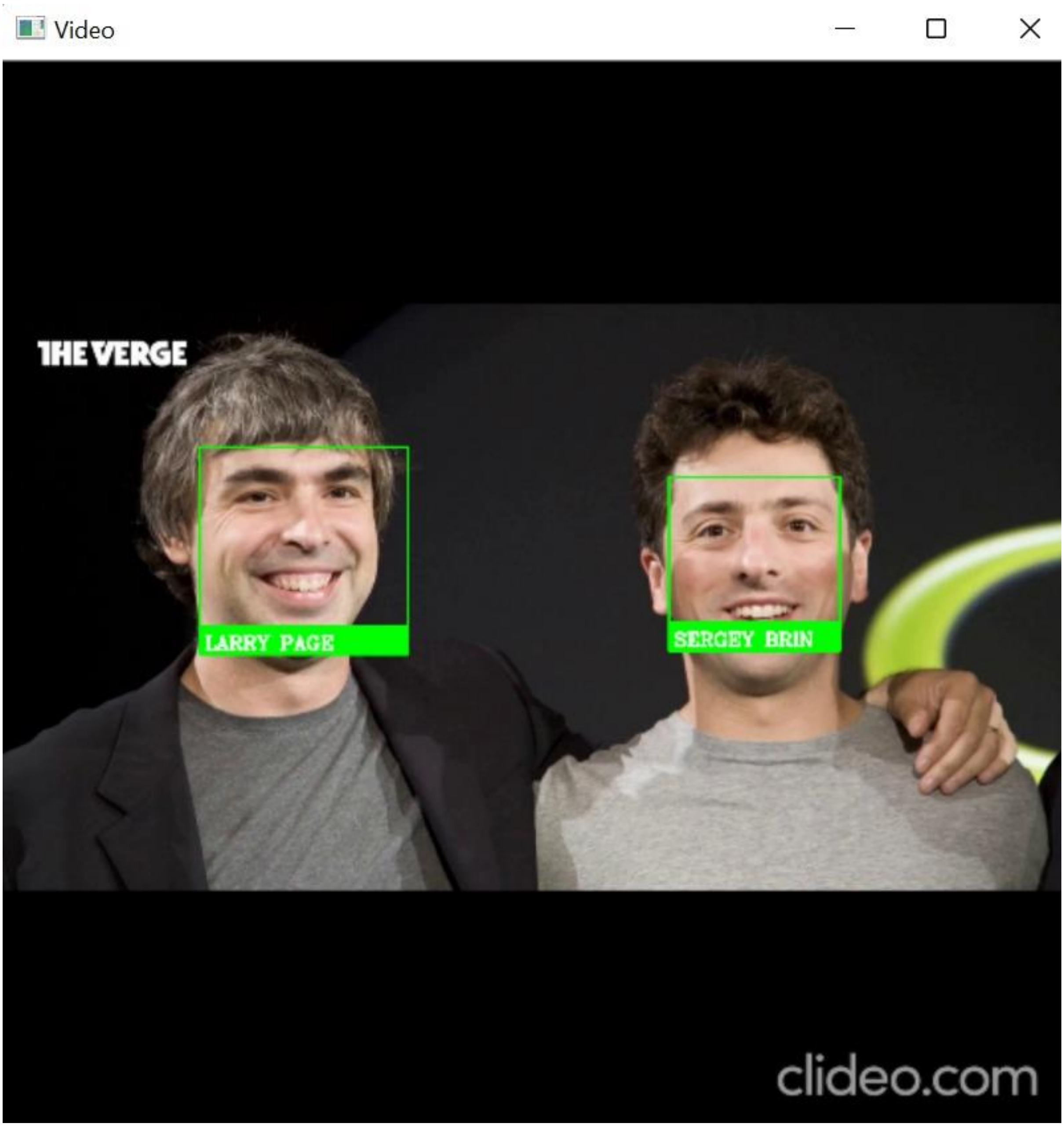

Video
LARRY ELLISON
clideo.com

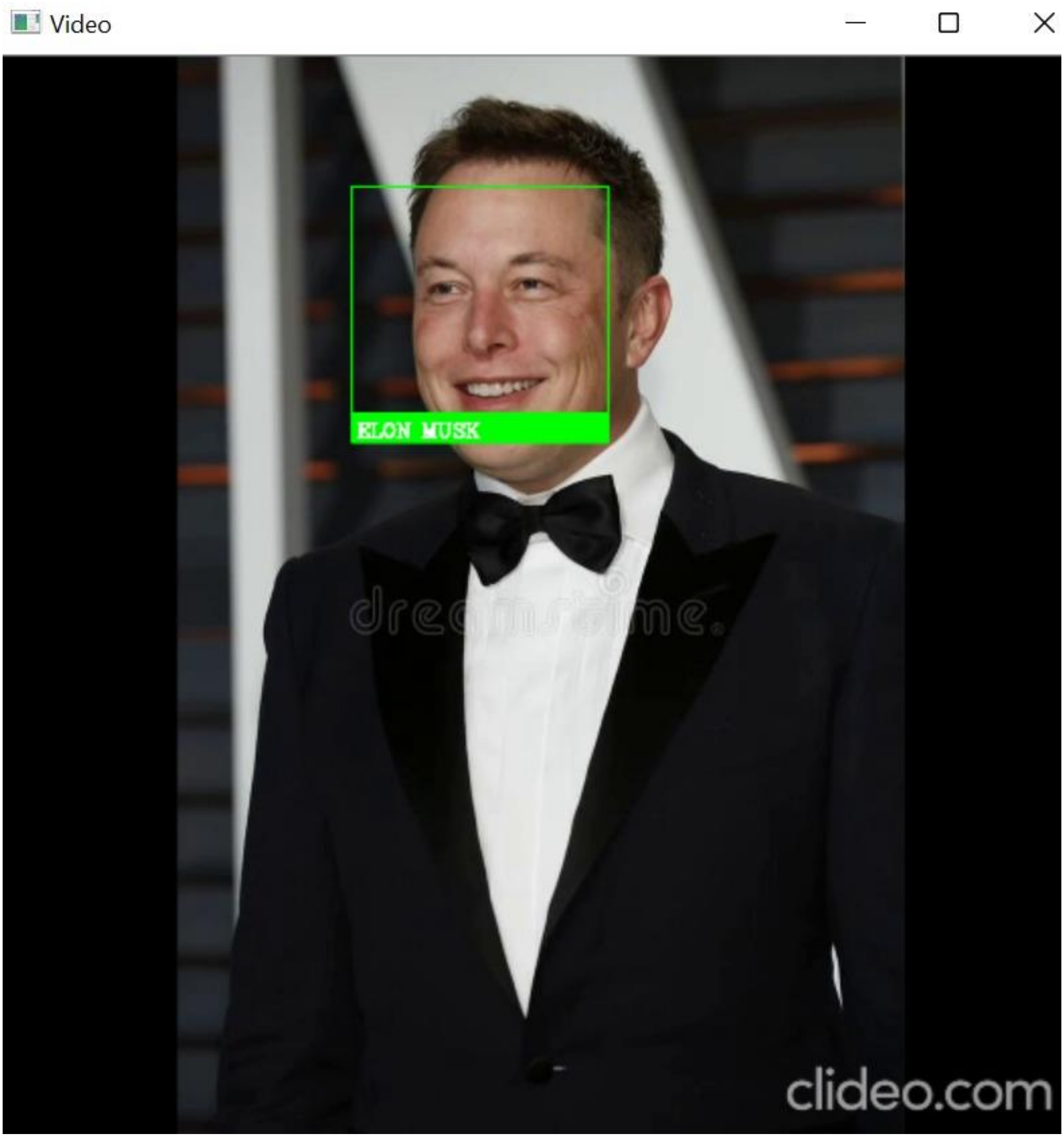

Video
MARK ZUCKERBERG
clideo.com



Video
JEFF BERZOS
BILL GATES
BERNARD ARNAULT
clideo.com


Video
BILL GATES
WARREN BUFFETT
clideo.com


Video
THE VERGE
LARRY PAGE
SERGEY BRIN
clideo.com


Video
ELON MUSK
dreamstime.
clideo.com

SCHEDULE & TASKS

Week No.	Proposed Work
Week-1	Project Proposal Report and Presentation
Week-1	Study related works
Week-1	Study in Python
Week-2	Study related works using OpenCV
Week-2	Study related works using Bluetooth
Week-3	Study related works using processing
Week-3	Study image processing
Week-3	Study image processing
Week-4	Sketching basic structure
Week-4	Prototype design
Week-4	Finalize Prototype design
Week-4	Flexible Box
Week-5	Runnable with basic commands(Input, Output, Turn on, Turn Off)
Week-5	Designing Lookahead table
Week-5	Designing Lookahead table
Week-6	Creating environment for image processing
Week-6	Creating environment for image processing
Week-7	Integrating all together
Week-7	Start coding
Week-8	Coding for basic instructions (Compare, Result, Accuracy measure etc.)

Week-8	**Coding for single face detection**
Week-9	**Single face detection and Compare with database**
Week-9	**Multiple Face detection and Compare**
Week-10	**Detecting Multiple face, store and compare with database**
Week-10	**Attendance collection**
Week-10	**File Generate base on collective data**
Week-10	**Daily file generation of attendance**

Table: Work Plan

SUMMARY

In this long yet useful chapter we managed to cover the entire structure of how the system has been developed and how it functions to give the best outcome.

To conclude, we discussed the scheduling processes of developing this system. Additionally we have also identified how feasible the system is through the lens of evaluating using various feasibility studies

In order to develop this system, we gave enormous importance to scheduling because we believed if we want to provide the best of quality in a given period of time then we must give due importance to scheduling which also helped us to achieve a better results.
The figure below focuses the weekly work we had accomplished.

REFERENCES