Department of Electronics & Communication Engineering,

MANIT Bhopal

Major Project Report (EC-413)

Group No.: 21

Project Title: Attendance System Using Facial Recognition

S.No.	Scholar No.	Name	Signature
1	171114096	Sourabh Meena	
2	171114104	Rishit Singh	
3	171114115	Mohit Ranjan Kumar	

Under the guidance of

DR. VIJAYSHRI CHAURASIA

Department of Electronics and Communication Engineering

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY BHOPAL

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SESSION: 2020-2021

CERTIFICATE

This is to certify that the following students :

Sourabh Meena	Scholar No: UG/FT 171114096
Rishit Singh	Scholar No: UG/FT 171114104
Mohit Ranjan Kumar	Scholar No: UG/FT 171114115

of the fourth year in Electronics and Communication Engineering of the institute have completed their minor project entitled 'FACE RECOGNITION BASED ATTENDANCE MARKING SYSTEM'.

It has been submitted in fulfilment of their minor project in the Department of Electronics and Communication Engineering under the guidance of project in charge and mentor Dr. Vijayshri Chaurasia.

DR. VIJAYSHRI CHAURASIA

MAULANA AZAD NATIONAL INSTITUTE OF TECHNOLOGY BHOPAL

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We, hereby declare that the following report which is being presented in the Major Project Documentation Entitled as "ATTENDANCE SYSTEM USING FACIAL RECOGNITION" is authentic documentation of our original work and to best of our knowledge. The following project and its report, in part or whole, has not been presented or submitted by us for any purpose in any other institute or organization. Any contribution made to the research by others, with whom we have worked at Maulana Azad National Institute of Technology, Bhopal, or elsewhere, is explicitly acknowledged in the report.

Sourabh Meena	Scholar No: UG/FT 171114096
Rishit Singh	Scholar No: UG/FT 171114104
Mohit Ranjan Kumar	Scholar No: UG/FT 171114115

ACKNOWLEDGEMENT

Major project in the final year is an indispensable part of any engineering curriculum. It provides the students with an opportunity to gain experience in the practical application of their technical knowledge and to study the various theoretical aspects as well.

With due respect, we express our gratitude to Dr. Kavita Khare, HOD, Department of Electronics and Communication Engineering, MANIT. We do not doubt that this project has helped us a lot and will help us gain much practical knowledge and the experiences would be unique.

We are highly indebted to Dr. Vijayshri Chaurasia, Department of Electronics and Communication Engineering. for giving us this opportunity to work under her guidance on this project. We must mention the fact that the synopsis of a minor project could not have been accomplished without the periodic suggestions and advice of our project guide. Her technical help and goal-oriented approach have been unique and a stepping stone towards the successful completion of our project synopsis and the confirmation of the ideation of our project.

We are also thankful to all the other faculty and would like to express our deep appreciation towards our family members and batch mates for providing the much-needed support.

CONTENTS

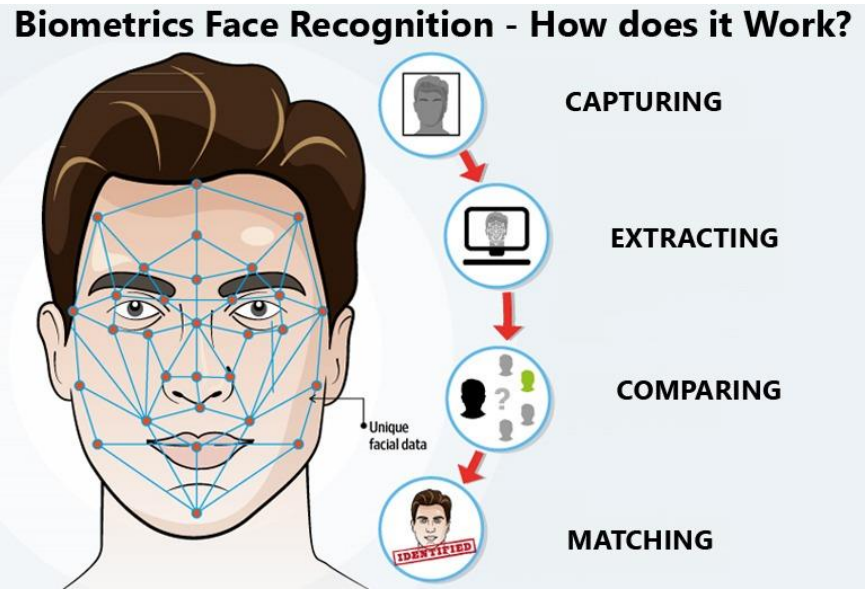
INTRODUCTION

1.1 OVERVIEW:

Have you noticed that Facebook has developed an uncanny ability to recognize your friends in your photographs? In the old days, Facebook used to make you to tag your friends in photos by clicking on them and typing in their name. Now as soon as you upload a photo, Facebook tags everyone for you *like magic*.

This technology is called face recognition. Facebook's algorithms are able to recognize your friends' faces after they have been tagged only a few times. It's pretty amazing technology — Facebook can recognize faces with 98% accuracy which is pretty much as good as humans can do!

Recognizing people by their faces in pictures and video feeds is seen everywhere starting from social media to phone cameras. A face recognition system is built for matching human faces with a digital image.



1.2 OBJECTIVE:

The objective of the project is to create an attendance marking system in order to reduce the burdensome and also a time-consuming task.

using facial recognition which would reduce the manual work of taking attendance by removing the man power required and hence creating an automated and time saving method to mark attendance.

Face recognition is really a series of several related problems:

1. First, look at a picture and find all the faces in it
2. Second, focus on each face and be able to understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person.
3. Third, be able to pick out unique features of the face that you can use to tell it apart from other people— like how big the eyes are, how long the face is, etc.
4. Finally, compare the unique features of that face to all the people you already know to determine the person's name.

LITERATURE REVIEW

2 ATTENDANCE MARKING SYSTEM:

We will develop an attendance system where the user is automatically logged when they are detected in the camera. We will store the name along with the time when they appeared. Use of face recognition for the purpose of attendance marking is the smart way of attendance management system. Face recognition is more accurate and faster technique among other techniques and reduces chance of proxy attendance. Face recognition provide passive identification that is a person which is to be identified does not to need to take any action for its identity.

A face remains the most common way to identify or authenticate a person. Today, a person's face has become the epicentre of the most fascinating and promising developing forensic technology - facial recognition.

Since the face is a unique method of identifying people, face recognition has received a lot of attention worldwide. Face recognition now has more advantages compared to other biometric systems such as palm print and fingerprint, since face recognition does not require human interaction.

In this project we design an Automated Attendance marking system with the help of facial recognition owing to the difficulty in the manual as well as other traditional means of attendance system.

The system consists of the algorithm which only detect the face of the student from the rest of the spaces and body parts and then matching this recognized image from the pre feed data, this data will be fetching while students enroll in the college or in a particular class .The system will work in three different sections first is to feed the images of the students who will be attending the classes ,it is extremely important to get all the images of the students captured . The second phase of the attendance monitoring system using face recognition is the detecting the faces this is the dynamic phase where the camera will capture the images of the faces only and with the help of the pre feed data this image which was being captured will be detected. The last phase of this system is updating the attendance sheet. The recognized faces will be marked as present and the rest of the students will be marked absent. Many face detection methods have appeared in the last two decades, while the classical methods have been very successful, their major disadvantage is that they use only 2D facial photos.

However, we know that such a representation is sensitive to changes in expression, illumination and poses. In addition, the complexities of calculation and storage as well as the procedure for generating the examples are very complicated. The requirements of these systems may include the addition of machine learning capability to enable a machine to evolve through a learning process, and perform tasks that are difficult or impossible to complete by more conventional algorithmic means. These new methods are essentially based on the notion of learning, which has been at the heart of artificial intelligence research for many years. Since face detection can be understood as a two-class model identification problem.

Why do we need it ?

In the era of modern technologies emerging at rapid pace there is no reason why a crucial event in education sector such as attendance should be done in the old boring traditional way. Attendance monitoring system will save a lot of time and energy for the both parties teaching staff as well as the students.

Attendance will be monitored by the face recognition algorithm by recognizing only the face of the students from the rest of the objects and then marking the students as present. The system will be pre feed with the images of all the students enrolled in the class and with the help of this pre feed data the algorithm will detect the students who are present and match the features with the already saved images of the students in the database.

Procedure to be followed :

The following system is based on face recognition to maintain the attendance record of students. The daily attendance of students is recorded subject wise which is stored already by the administrator.

As the time for corresponding subject arrives the system automatically starts taking snaps and then apply face detection and recognition technique to the given image and the recognize students are marked as present and their attendance update with corresponding time and subject id.

We have used deep learning techniques to develop this system, histogram of oriented gradient method is used to detect faces in images and deep learning method is used to compute and compare feature facial of students to recognize them. Our system is capable to identify multiple faces in real time.

2.2 SVM Classifier

Support Vector Machines are supervised learning models for classification and regression problems. They can solve linear and non-linear problems and work well for many practical problems. The idea of Support Vector Machines is simple: The algorithm creates a line which separates the classes in case e.g. in a classification problem. The goal of the line is to maximizing the margin between the points on either side of the so called decision line. The benefit of this process is, that after the separation, the model can easily guess the target classes (labels) for new cases.

3 SOFTWARES AND LIBRARIES

3.1 SOFTWARES:

- **Pycharm:** PyCharm is an integrated development environment used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains.

3.2 LIBRARIES:

- **NumPy**: For mathematical and scientific operations**.**

- **Cv2:** OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

- **Face_recognition:** Recognize and manipulate faces from Python or from the command line with

- **os**: The **OS module in Python** provides functions for interacting with the operating system

- **dlib:** **DLib** is an open source C++ **library** implementing a variety of machine learning algorithms, including classification, regression, clustering, data transformation, and structured prediction.

4 STEPS FOR FACIAL RECOGNITION

4.1 FINDING ALL THE FACES

To detect the faces in the image we'll be using the method invented in 2005 called Histogram of Oriented Gradients — *HOG.*

Histogram of Oriented Gradient:

To find faces in an image, we'll start by making our image black and white because we don't need color data to find faces.

Then we'll look at every single pixel in our image one at a time. For every single pixel, we want to look at the pixels that directly surrounding it.

Our goal is to figure out how dark the current pixel is compared to the pixels directly surrounding it. Then we want to draw an arrow showing in which direction the image is getting darker.

We repeat the process for every single pixel in the image and end up with every pixel being replaced by an arrow. These arrows are called *gradients* and they show the flow from light to dark across the entire image.

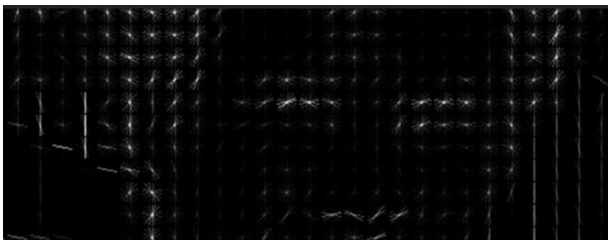
If we analyze pixels directly, really dark images and really light images of the same person will have totally different pixel values. But by only considering the *direction* that brightness changes, both really dark images and really bright images will end up with the same exact representation. That makes the problem a lot easier to solve!

But saving the gradient for every single pixel gives us way too much detail. We end up missing the forest for the trees. It would be better if we could just see the

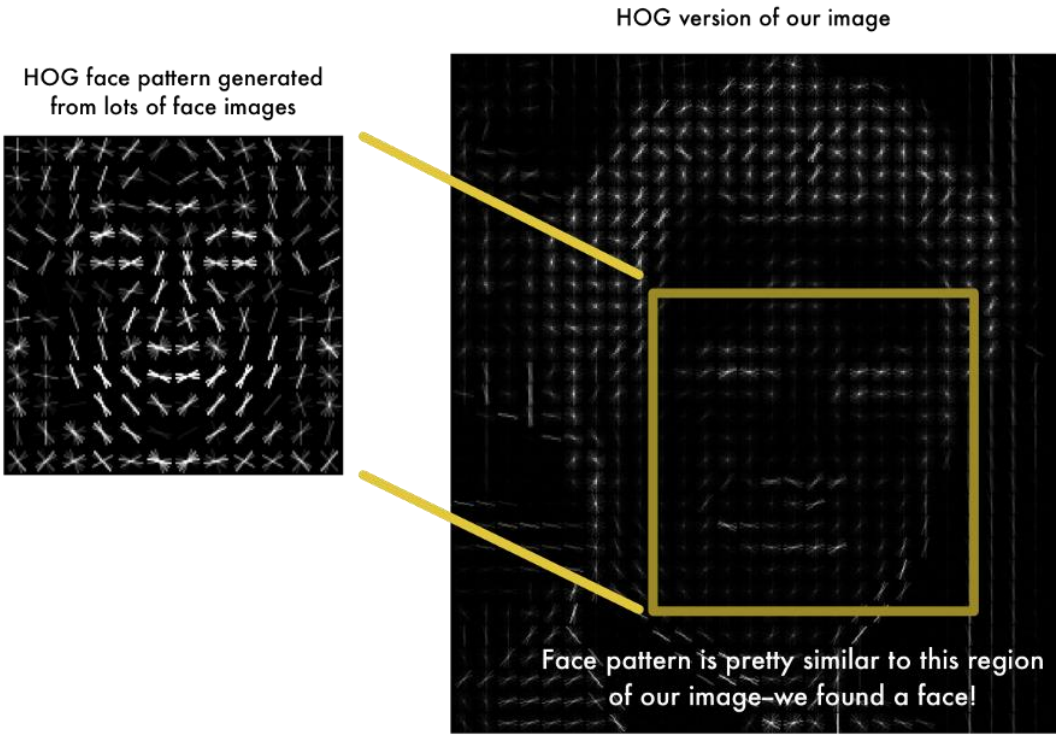
basic flow of lightness/darkness at a higher level so we could see the basic pattern of the image.

Next we'll break up the image into small squares of 16x16 pixels each. In each square, we'll count up how many gradients point in each major direction (how many point up, point up-right, point right, etc…). Then we'll replace that square in the image with the arrow directions that were the strongest.

The end result is we turn the original image into a very simple representation that captures the basic structure of a face.



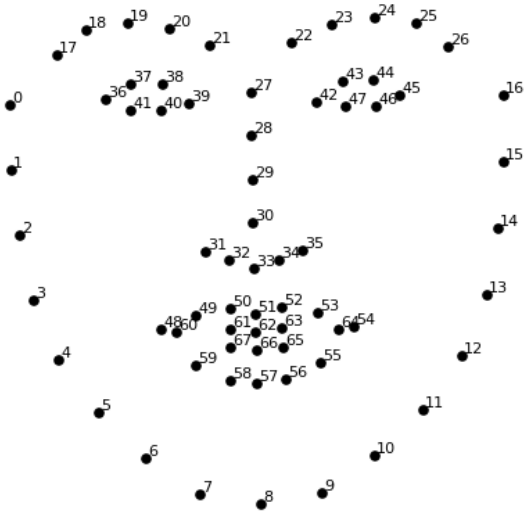
To find faces in this HOG image, all we have to do is find the part of our image that looks the most similar to a known HOG pattern that was extracted from a bunch of other training faces:



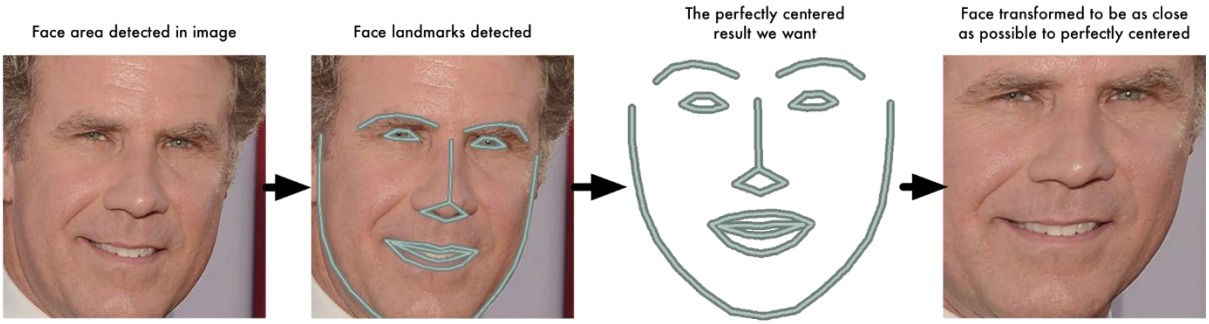
4.2 POSING AND PROJECTING FACES

Faces turned different directions look totally different to a computer.We will try to warp each picture so that the eyes and lips are always in the sample place in the image.

To do this, we are going to use an algorithm called **face landmark estimation.**



The basic idea is we will come up with 68 specific points (called *landmarks*) that exist on every face — the top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc. Then we will train a machine learning algorithm to be able to find these 68 specific points on any face.



4.3 ENCODING FACES

We need is a way to extract a few basic measurements from each face. Then we could measure our unknown face the same way and find the known face with the closest measurements. For example, we might measure the size of each ear, the spacing between the eyes, the length of the nose, etc.

Deep learning does a better job than humans at figuring out which parts of a face are important to measure.We are going to train it to generate 128 measurements for each face.

The training process works by looking at 3 face images at a time:

1. Load a training face image of a known person
2. Load another picture of the same known person
3. Load a picture of a totally different person

Then the algorithm looks at the measurements it is currently generating for each of those three images. It then tweaks the neural network slightly so that it makes sure the measurements it generates for #1 and #2 are slightly closer while making sure the measurements for #2 and #3 are slightly further apart.

After repeating this step millions of times for millions of images of thousands of different people, the neural network learns to reliably generate 128 measurements for each person. Any ten different pictures of the same person should give roughly the same measurements.

Machine learning people call the 128 measurements of each face an **embedding**.

This process of training a convolutional neural network to output face embeddings requires a lot of data and computer power.

But once the network has been trained, it can generate measurements for any face, even ones it has never seen before! So this step only needs to be done once. Lucky for us, the fine folks at OpenFace already did this and they published several trained networks which we can directly use. So all we need to do ourselves is run our face images through their pre-trained network to get the 128 measurements for each face. Here's the measurements for our test image.

Input Image

128 Measurements Generated from Image

0.097496084868908	0.045223236083984	-0.1281466782093	0.032084941864014
0.12529824674129	0.060309179127216	0.17521631717682	0.020976085215807
0.030809439718723	-0.01981477253139	0.10801389068365	-0.00052163278451189
0.036050599068403	0.065554238855839	0.0731306001544	-0.1318951100111
-0.097486883401871	0.1226262897253	-0.029626874253154	-0.0059557510539889
-0.0066401711665094	0.036750309169292	-0.15958009660244	0.043374512344599
-0.14131525158882	0.14114324748516	-0.031351584941149	-0.053343612700701
-0.048540540039539	-0.061901587992907	-0.15042643249035	0.078198105096817
-0.12567175924778	-0.10568545013666	-0.12728653848171	-0.076289616525173
-0.061418771743774	-0.074287034571171	-0.065365232527256	0.12369467318058
0.046741496771574	0.0061761881224811	0.14746543765068	0.056418422609568
-0.12113650143147	-0.21055991947651	0.0041091227903962	0.089727647602558
0.061606746166945	0.11345765739679	0.021352224051952	-0.0085843298584223
0.061989940702915	0.19372203946114	-0.086726233363152	-0.022388197481632
0.10904195904732	0.084853030741215	0.09463594853878	0.020696049556136
-0.019414527341723	0.0064811296761036	0.21180312335491	-0.050584398210049
0.15245945751667	-0.16582328081131	-0.035577941685915	-0.072376452386379
-0.12216668576002	-0.0072777755558491	-0.036901291459799	-0.034365277737379
0.083934605121613	-0.059730969369411	-0.070026844739914	-0.045013956725597
0.087945111095905	0.11478432267904	-0.089621491730213	-0.013955107890069
-0.021407851949334	0.14841195940971	0.078333757817745	-0.17898085713387
-0.018298890441656	0.049525424838066	0.13227833807468	-0.072600327432156
-0.011014151386917	-0.051016297191381	-0.14132921397686	0.0050511928275228
0.0093679334968328	-0.062812767922878	-0.13407498598099	-0.014829395338893
0.058139257133007	0.0048638740554452	-0.039491076022387	-0.043765489012003
-0.024210374802351	-0.11443792283535	0.071997955441475	-0.012062266469002
-0.057223934680223	0.014683869667351	0.05228154733777	0.012774495407939
0.023535015061498	-0.081752359867096	-0.031709920614958	0.069833360612392
-0.0098039731383324	0.037022035568953	0.11009479314089	0.11638788878918
0.020220354199409	0.12788131833076	0.18632389605045	-0.015336792916059
0.0040337680839002	-0.094398014247417	-0.11768248677254	0.10281457751989
0.051597066223621	-0.10034311562777	-0.040977258235216	-0.082041338086128

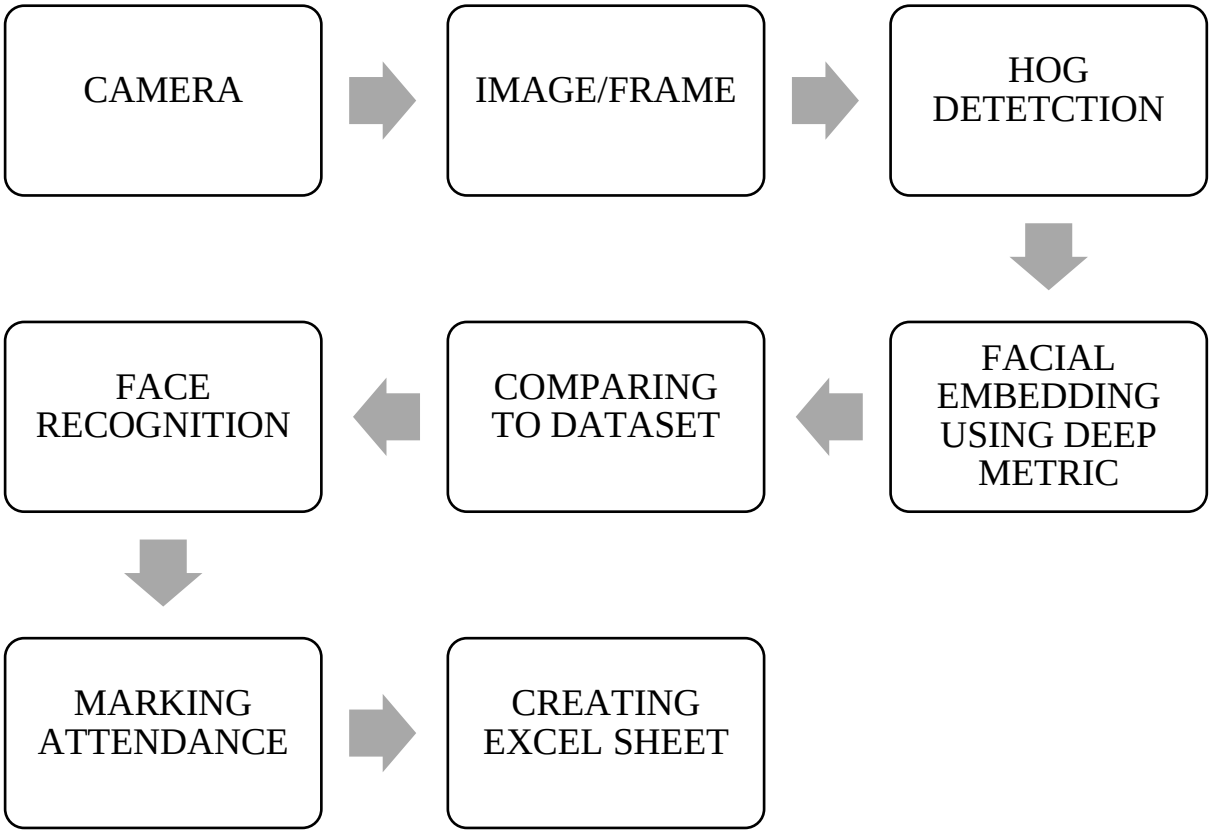
4.4 FINDING THE PERSON'S NAME FROM THE ENCODING

We have to find the person in our database of known people who has the closest measurements to our test image. For this we'll use a simple linear SVM Classifier.

We train a classifier that can take in the measurements from a new test image and tells which known person is the closest match.

All we need to do is train a classifier that can take in the measurements from a new test image and tells which known person is the closest match. Running this classifier takes milliseconds. The result of the classifier is the name of the person!

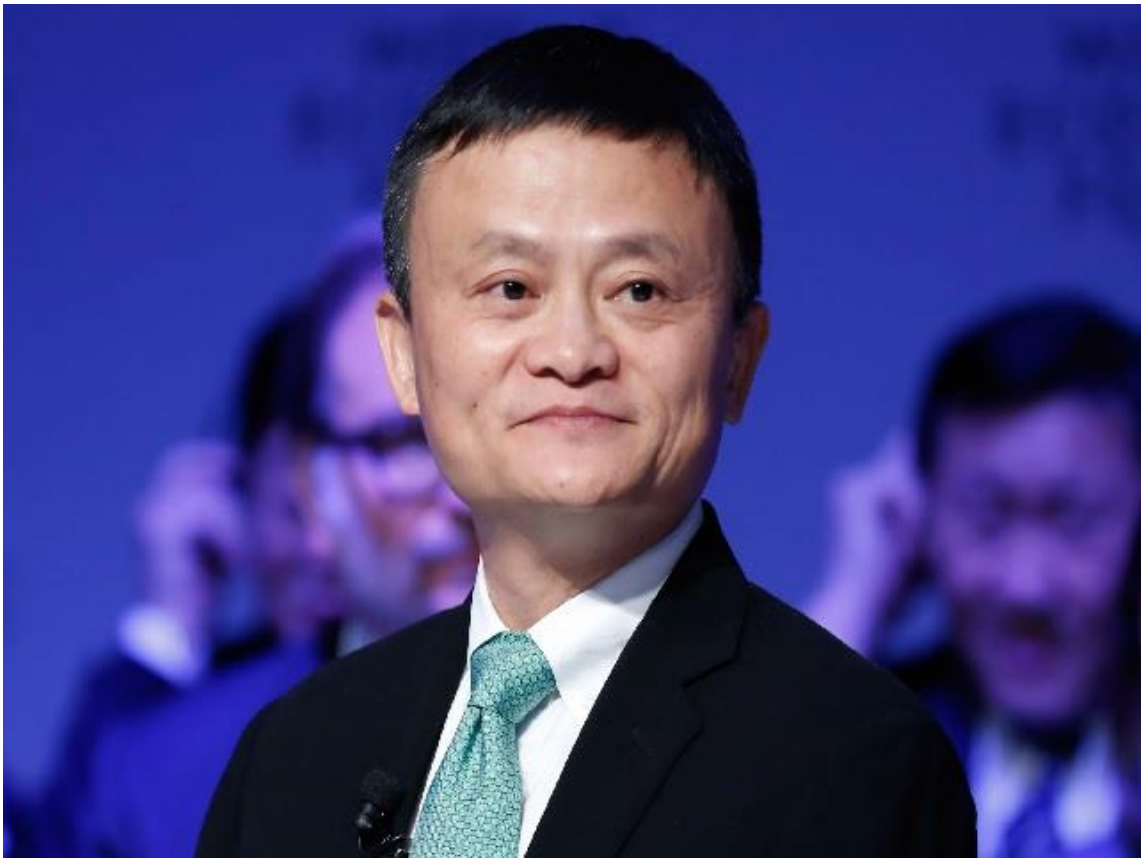
5 FLOWCHART


CAMERA
IMAGE/FRAME
HOG DETETCTION
FACIAL EMBEDDING USING DEEP METRIC
COMPARING TO DATASET
FACE RECOGNITION
MARKING ATTENDANCE
CREATING EXCEL SHEET

6 DATABASE IMAGES USED

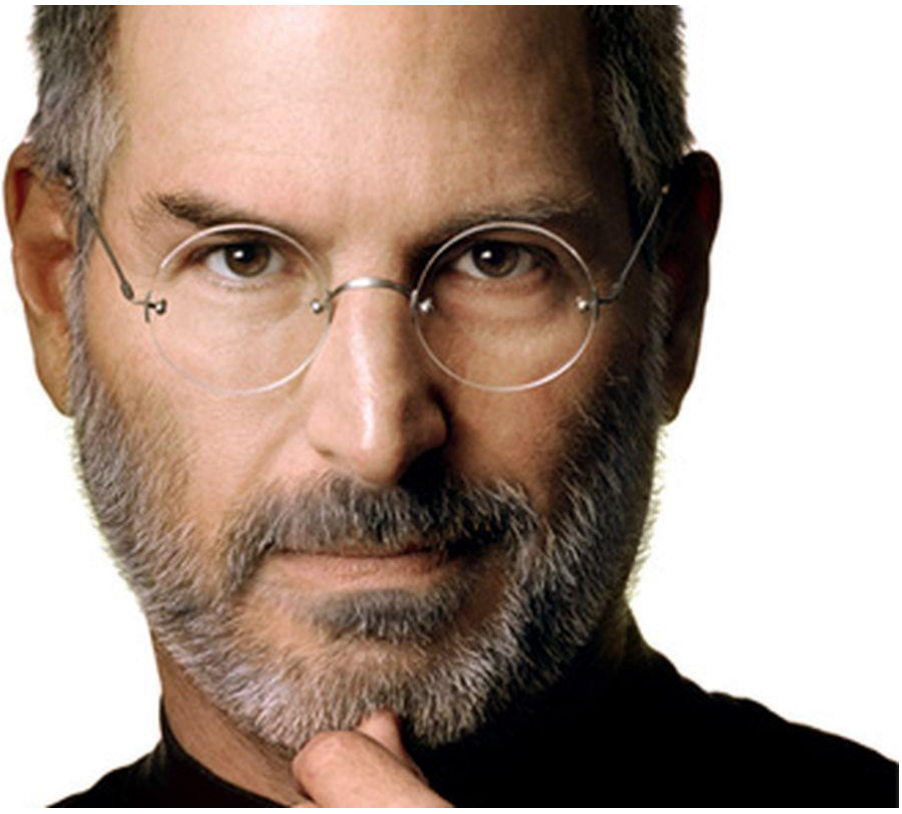
bill_gates.jpg

Elon_Musk.jpg



Jack_Ma.jpg

Steve_Jobs.jpg



7 CODE

```
#importing the different libraries Used in our code
import cv2
import face_recognition
import numpy as np
import os
from datetime import datetime

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

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

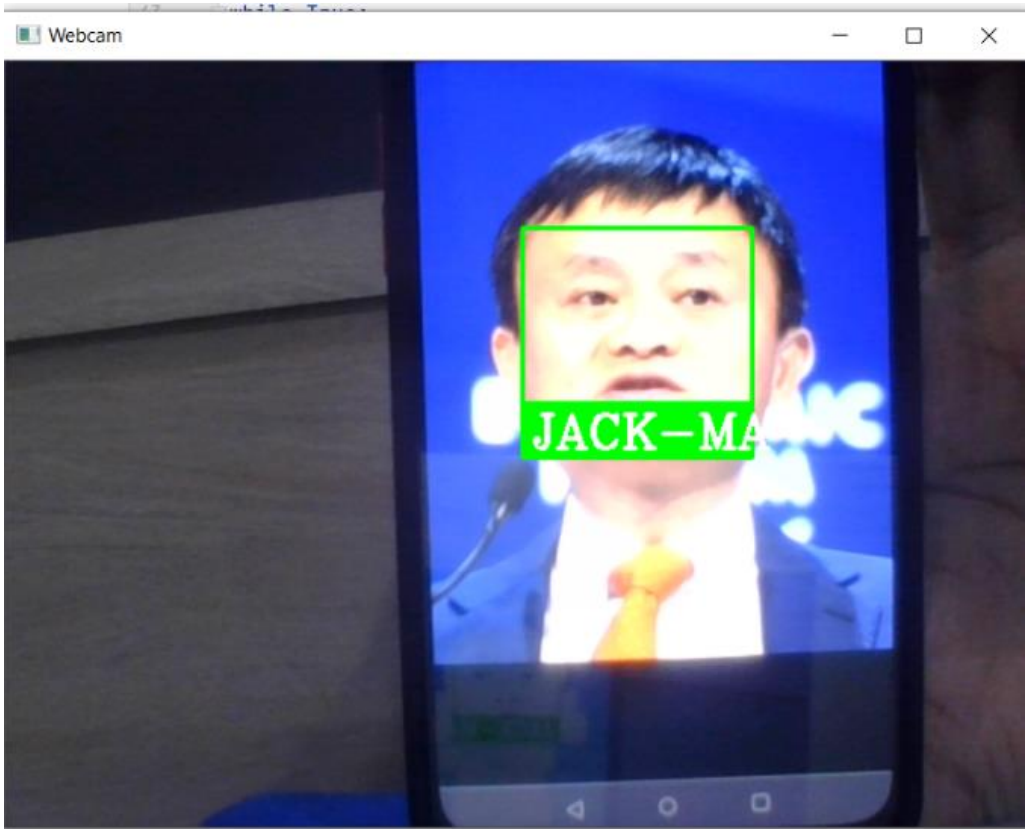
```
        encodeList.append(encode)
    return encodeList

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}')



encodeListKnown = findEncodings(images)
print('Encoding Complete')

cap = cv2.VideoCapture(0)

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

    facesCurFrame = face_recognition.face_locations(imgS)
```

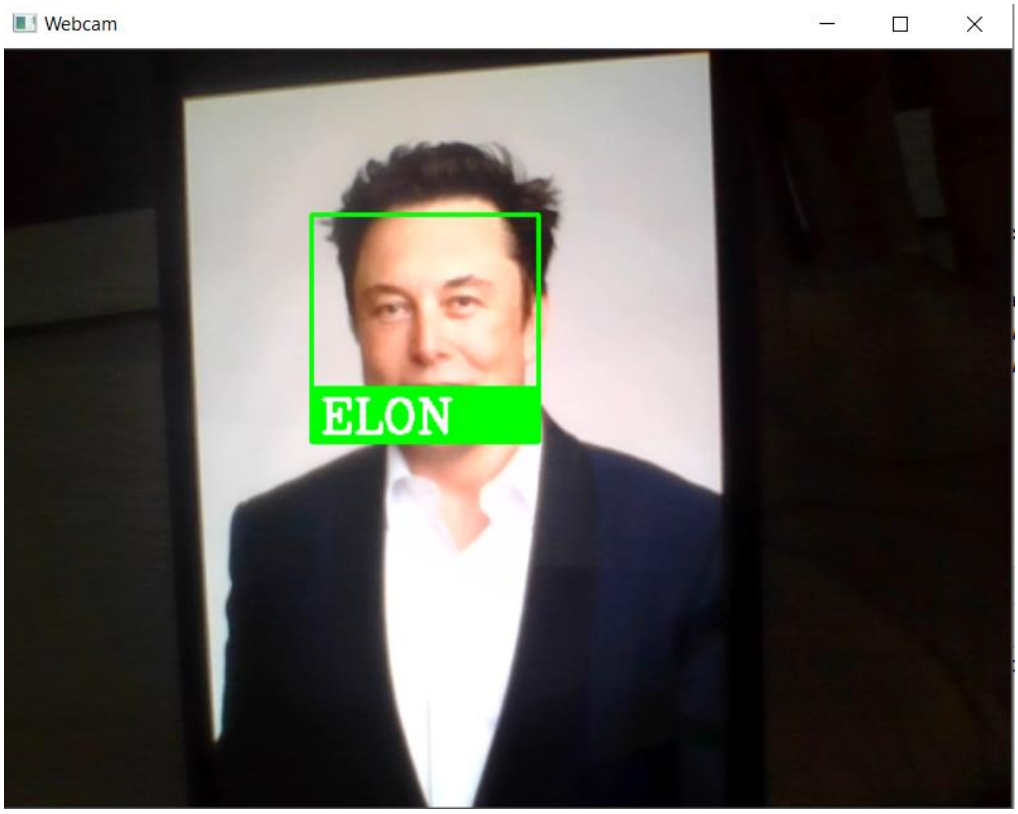
```
    encodesCurFrame = face_recognition.face_encodings(imgS,
facesCurFrame)

    for encodeFace, faceLoc in zip(encodesCurFrame, facesCurFrame):
        matches = face_recognition.compare_faces(encodeListKnown,
encodeFace)
        faceDis = face_recognition.face_distance(encodeListKnown,
encodeFace)
        # print(faceDis)
        matchIndex = np.argmin(faceDis)

        if matches[matchIndex]:
            name = classNames[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, 1, (255, 255, 255), 2)
            markAttendance(name)
else :
    name='unknown'
    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, 0, 255), 2)
    cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 0, 255), cv2.FILLED)
    cv2.putText(img, name, (x1 + 6, y2 - 6),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
    markAttendance(name)

    cv2.imshow('Webcam', img)
    cv2.waitKey(1)
```

OUTPUT

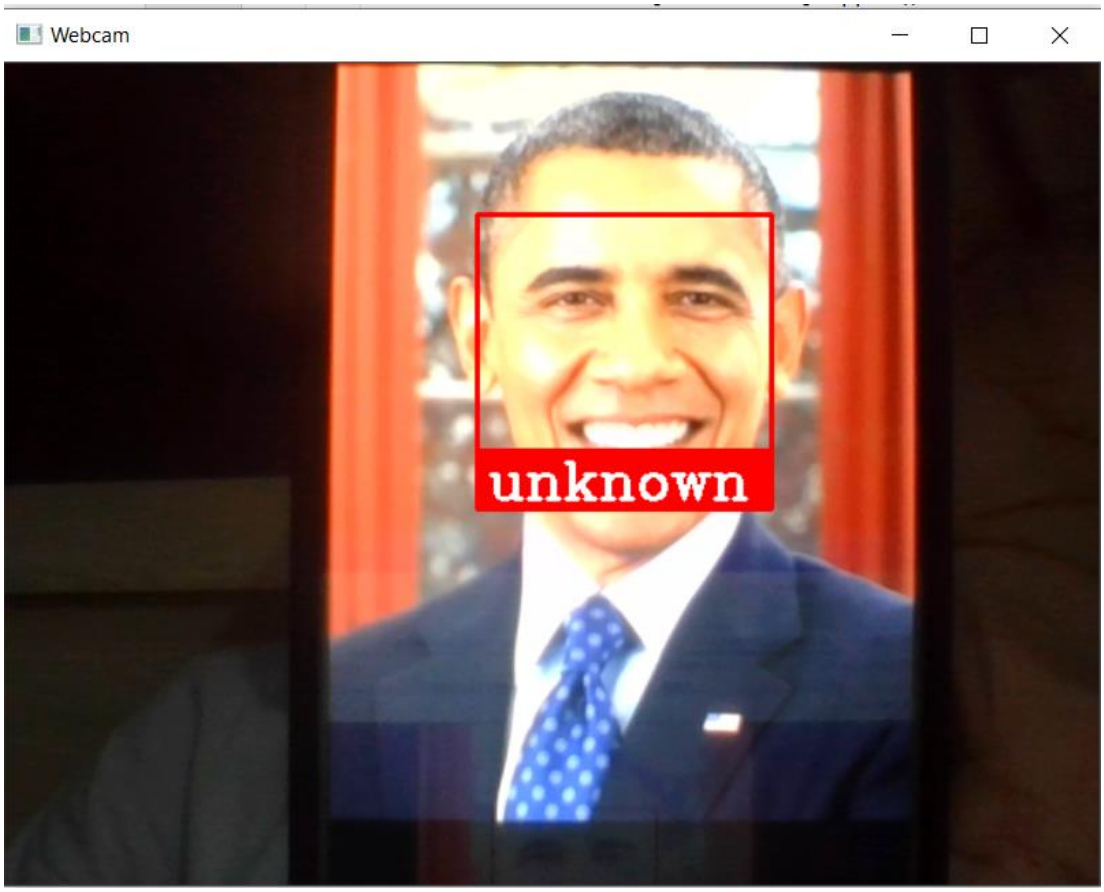
The real time webcam shows the following outputs:

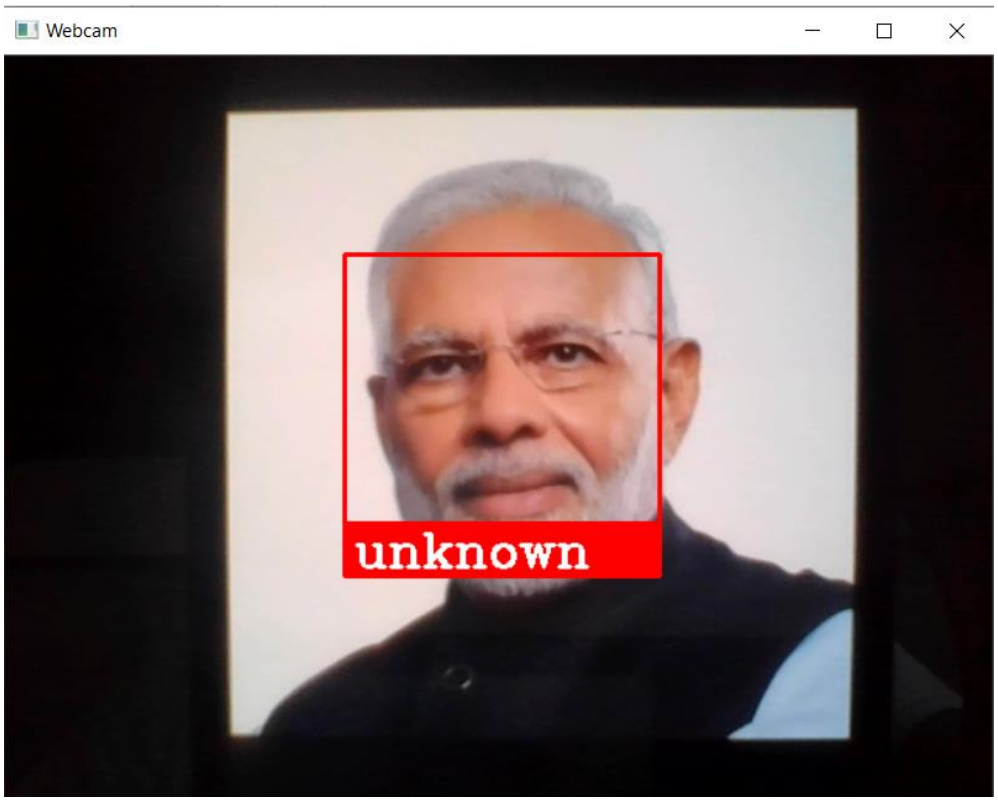
a) If the Correct Face is Detected we get the output as:



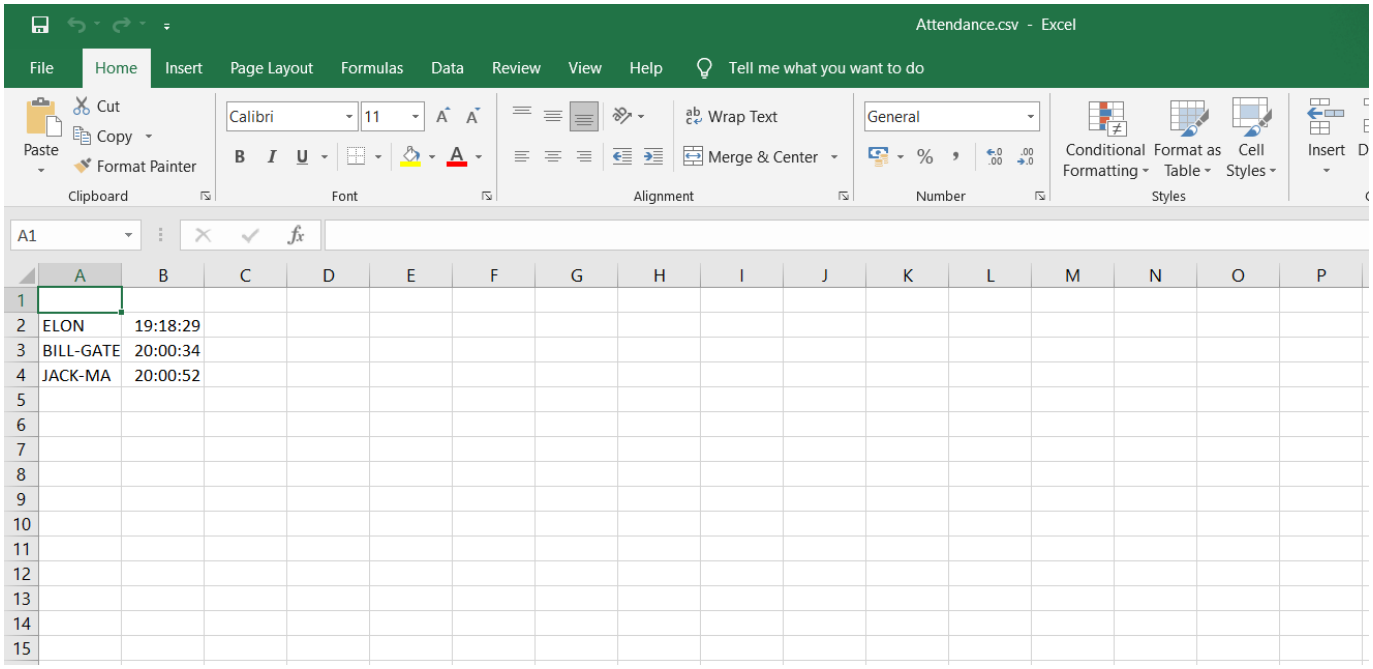


b)If any face which is not in our data base is detected then the output shows unknown:





The data will also be stored in an excel sheet having the name as well as the attendance time:



8 CONCLUSION

This paper introduces the efficient method of attendance management system in the classroom environment that can replace the old manual methods. This method is secure enough, reliable, accurate and efficient. There is no need for specialized hardware for installing the system in the classroom. It can be constructed using a camera and computer. There is a need to use some algorithms that can recognize the faces in veil to improve the system performance.

We have used Histogram of Oriented Gradients method for generating a much simpler structure of the face detected using only simple arrows (Gradients) pointing in the directing from the dark to light spots on the face.

Next, we have used an algorithm known as Face landmark Estimation which is used for projecting face to make them look in the front direction so that our Face recognizer detects the face accurately.

Then we have used a deep convolutional neural network to encode our faces into 128 markings which are compared with the detected face and hence it gives the judgement whether the faces and similar or not and the face is detected properly.

Hence, we generate a Face Recognition Based attendance marking system.

9 FUTURE SCOPE AND ADVANCEMENTS

Almost all academic institutions require attendance record of students and maintaining attendance manually can be hectic as well as time consuming task. Hence maintaining attendance automatically with the help of face recognition will be very helpful and less prone to errors as compared to manual process.

This will also reduce manipulation of attendance record done by students and it will save time as well.

The future scope of the proposed work can be, capturing multiple detailed images of the students and using any cloud technology to store these images.

The system can be configured and used in Atm machines to detect frauds.Also, the system can be used at the time of elections where the voter can be identified by recognizing the face.

10 REFERENCES:

- https://pypi.org/project/face-recognition/
- https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
- https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
- https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm