DAYANANDA SAGAR UNIVERSITY

MINOR PROJECT REPORT

ON

INFORMATION RETRIEVAL SYSTEM FROM IMAGES DATABASE

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

5th SEMESTER, 2022

Under the Guidance of
Dr. Rajesh T M
Associate Professor Dept. of CSE

Reviewed By
Dr.Debanjali Bhattacharya & Prof. Vaidehi Verma

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate ,Bangalore-560068

CERTIFICATE

This is to certify that the minor project report entitled " **INFORMATION RETRIEVAL SYSTEM FROM IMAGES DATABASE***"is being submitted by Mr* **"..."** *respectively has* satisfactorily completed their Minor Project as prescribed by the University for the 5th semester B.Tech Program in Computer Science & Engineering during the academic year 2021 – 22 at the School of Engineering, Dayananda Sagar University, Bangalore.

Date: Signature of the faculty in-charge

Signature of Chairman
Department of Computer Science & Engineering

DECLARATION

We hereby declare that the work presented in this minor project entitled as ***"Facial Image information retrieval from Face Database"***, has been carried out by us and it has not been submitted for the award of any degree, diploma or the minor project of any other college or university.

Team 07

……………………………

ACKNOWLEDGEMENT

The success and final outcome of this software requirement document required a lot of guidance and assistance from many people and we are extremely privileged to have got this all through the completion of the project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank our mentor, Professors, and the Chairman for providing us an opportunity to do the Software Requirement Document and giving us all support and guidance which made me complete the project duly. We are extremely thankful to all for providing such nice support and guidance

We are especially thankful to our **Dean Dr. A Srinivas & Chairman Dr.Girisha** Department of Computer Science & Engineering who continuously helped us throughout the project and without his guidance, this project would have been an uphill task.

We are pleased to acknowledge **Dr. Rajesh T M**, Associate Prof, Department of Computer Science & Engineering for his invaluable guidance, support, motivation and patience during the course of this mini-project work..

We are thankful for and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Computer Science Engineering department, which helped us in completing our report. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support..

ABSTRACT

Face recognition is among the most productive image processing applications and has a pivotal role in the technical field. Recognition of the human face is an active issue for authentication purposes specifically in the context of attendance of students.Attendance system using face recognition is a procedure of recognizing students by using face biostatistics based on the high definition monitoring and other computer technologies. The development of this system is aimed to accomplish digitization of the traditional system of taking attendance by calling names and maintaining pen-paper records.

Table of Contents

INTRODUCTION

What is Happening in the World?

From time to time we hear about the crimes of credit card fraud, computer break-ins by hackers, or security breaches in a company or government building.In most of these crimes, the criminals were taking advantage of a fundamental flaw in the conventional access control systems: the systems do not grant access by "who we are", but by "what we have", such as ID cards, keys, passwords, PIN numbers, or mother's maiden name.

Precautions and Steps taken to prevent Fraud

Recently, technology became available to allow verification of "true" individual identity. This technology is based in a field called "biometrics".Biometric access control are automated methods of verifying or recognizing the identity of a living person on the basis of some physiological characteristics, such as fingerprints or facial features, or some aspects of the person's behavior, like his/her handwriting style or keystroke patterns.

Our Approach towards Face Recognition

Face recognition systems are part of facial image processing applications and their significance as a research area are increasing recently. We are using Viola-Jones algorithm, Ada-Boost Algorithm, Haar Cascade Classifier for Face Retrieval.Human brain can automatically and instantly detect and recognize multiple faces. But when it comes to computers, it is very difficult to do all the challenging tasks on the level of the human brain. Face recognition is an integral part of biometrics. In biometrics, basic traits of humans are matched to the existing data. Facial features are extracted and implemented through algorithms, which are efficient and some modifications are done to improve the existing algorithm models. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification, security systems, identity verification etc. The face recognition system generally involves two stages:

Face Detection – where the input image is searched to find any face, then image processing cleans up the facial image for easier recognition.
Face Recognition – where the detected and processed face is compared to the database of known faces to decide who that person is.

PROBLEM STATEMENT

Attendance is an important part of daily classroom evaluation. At the beginning and ending of class, it is usually checked by the teacher, but it may appear that a teacher may miss someone or some students answer multiple times. Face recognition-based attendance system is a problem of recognizing face for taking attendance by using face recognition technology based on high definition monitor video and other information technology.

So, we came up with a solution for the Face retrieval system from images Databases.The concept of face recognition is to give a computer system the ability of finding and recognizing human faces fast and precisely in images or videos.

LITERATURE AND SURVEY

A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques: In this paper, the idea of two technologies namely Student Attendance and Feedback system has been implemented with a machine learning approach. This system automatically detects the student performance and maintains the student's records like attendance and their feedback on the subjects like Science, English, etc. Therefore the attendance of the student can be made available by recognizing the face. On recognizing, the attendance details and details about the marks of the student is obtained as feedback.

Automated Attendance System Using Face Recognition: Automated Attendance System using Face Recognition proposes that the system is based on face detection and recognition algorithms, which is used to automatically detects the student face when he/she enters the class and the system is capable to marks the attendance by recognizing him. Viola-Jones Algorithm has been used for face detection which detects human faces using cascade classifier and PCA algorithm for feature selection and SVM for classification. When it is compared to traditional attendance marking this system saves time and also helps to monitor the students.

Student Attendance System Using Iris Detection: In this proposed system the student is requested to stand in front of the camera to detect and recognize the iris, for the system to mark attendance for the student. Some algorithms like Gray Scale Conversion, Six Segment Rectangular Filter, Skin Pixel Detection are being used to detect the iris. It helps in preventing proxy issues and it maintains the attendance of the student in an effective manner, but in one of the time-consuming processes for a student or a staff member to wait until the completion of the previous members.

Face Recognition-based Lecture Attendance System: This paper proposes that the system takes the attendance automatically recognition obtained by continuous observation. Continuous observation helps in estimating and improving the performance of the attendance. To obtain the attendance, positions and face images of the students present in the classroom are captured. Through continuous observation and recording the system estimates seating position and location of each student for attendance marking. The work is focused on the method to obtain the different weights of each focused seat according to its location. The effectiveness of the picture is also being discussed to enable the faster recognition of the image.

SOFTWARE AND HARDWARE REQUIREMENTS

- **Operating Systems:** Windows 7 and above
- Windows Server 2012 or greater.
- **Processor:** 1 GHz or greater.
- **Hard Disk Space:** 8GB and above available disk space.
- **Memory:** 2 GB (32-bit), 4 GB (64-bit).
- **Graphics:** Support for DirectX 9 graphics with minimum 128MB RAM.
- PyCharm IDE & VS Code

DOMAIN UNDERSTANDING

Firstly the Image Processing is done and Noise Removal is performed using Viola-Jones algorithm
For Classification we used the Haar Cascade Classifier.

A convolutional neural network (CNN) is a type of artificial neural network used in image recognition and processing that is specifically designed to process pixel data.

CNNs are powerful image processing, artificial intelligence (AI) that use deep learning to perform both generative and descriptive tasks, often using machine vision that includes image and video recognition, along with recommender systems and natural language processing (NLP).

k-NN is one of the foremost basic classification algorithms in machine learning. It belongs to the supervised learning class of machine learning. k-NN is usually employed in search applications wherever you're looking for “similar” things. The way we measure similarity is by making a vector illustration of the things, and then comparing the vectors using an acceptable distance metric (like the geometrician distance, for example).

It is typically utilized in data processing, pattern recognition, recommender systems and intrusion detection.

ALGORITHMS

- Viola-Jones algorithm
- Ada-Boost Algorithm
- Haar Cascade Classifier

Viola Jones is a novel approach to rapid detection of objects with running capabilities of 15 frames per second. It was the first to achieve real time object detection.

A Viola Jones detector consists of following steps :

1. Calculating Integral Image
2. Calculating Haar like features
3. AdaBoost Learning Algorithm
4. Cascade Filter

Step 1 : Start

Step 2 : Initiate Python Script

Step 3 : Implement Haar-Cascade Classifiers

Step 4 : Initiate Recognizer

Step 5 : Load Labels

Step 6 : Initialize camera for live Video Feed

Step 7 : Implement Recognizer

Step 8 : Compare face with Trained Data

Step 9 : if match found goto Step 10 or goto Step 7

Step 10 : Save file

Step 11 : STOP

What is an integral Image?

Because we have to use haar-like features in all possible sizes and locations which eventually result in around 200k features to calculate which is a really big number. The problem with novel calculation of haar features is that we have to calculate the average of a given region multiple times and the time complexity of these operations are O(n*n). We can use an integral image approach to achieve O(1) running time. A given pixel in the integral image is the sum of all the pixels on the left and all the pixels above it.

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

Input Image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

Integral Image

64	2	3	61	60	6	7	57
9	55	54	12	13	51	50	16
17	47	46	20	21	43	43	24
40	26	27	37	36	30	31	33
32	34	35	29	28	38	39	25
41	23	22	44	45	19	18	48
49	15	14	52	53	11	10	56
8	58	59	5	4	62	63	1

Original Image

64	66	69	130	190	196	203	260
73	130	187	260	333	390	446	520
90	194	297	390	484	584	683	780
130	260	390	520	650	780	910	1040
162	326	491	650	808	976	1145	1300
203	390	577	780	983	1170	1357	1560
252	454	655	910	1166	1364	1561	1820
260	520	780	1040	1300	1560	1820	2080

Integral Image

The sum of all purple boxes in the original image is equal to the sum of green boxes in the integral image subtracted by the purple boxes in the integral image.

Calculation of Haar like features with Integral Image

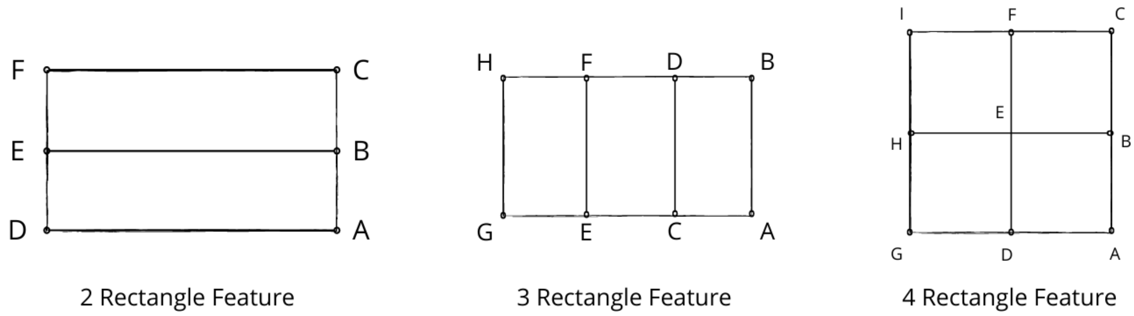
Using integral images we can achieve constant time evaluation of Haar features.

1. Edge Features or 2 Rectangular Features requires only 6 memory lookups
2. Line Features or 3 Rectangular Features requires only 8 memory lookups.
3. Diagonal Features or 4 Rectangular Features requires only 9 memory lookups.

2 Rectangle = A-2B+C-D+2E-F

3 Rectangle = A-B-2C+2D+2E-2F-G+H

4 Rectangle = A-2B+C-2D+4E-2F+H-2I+J



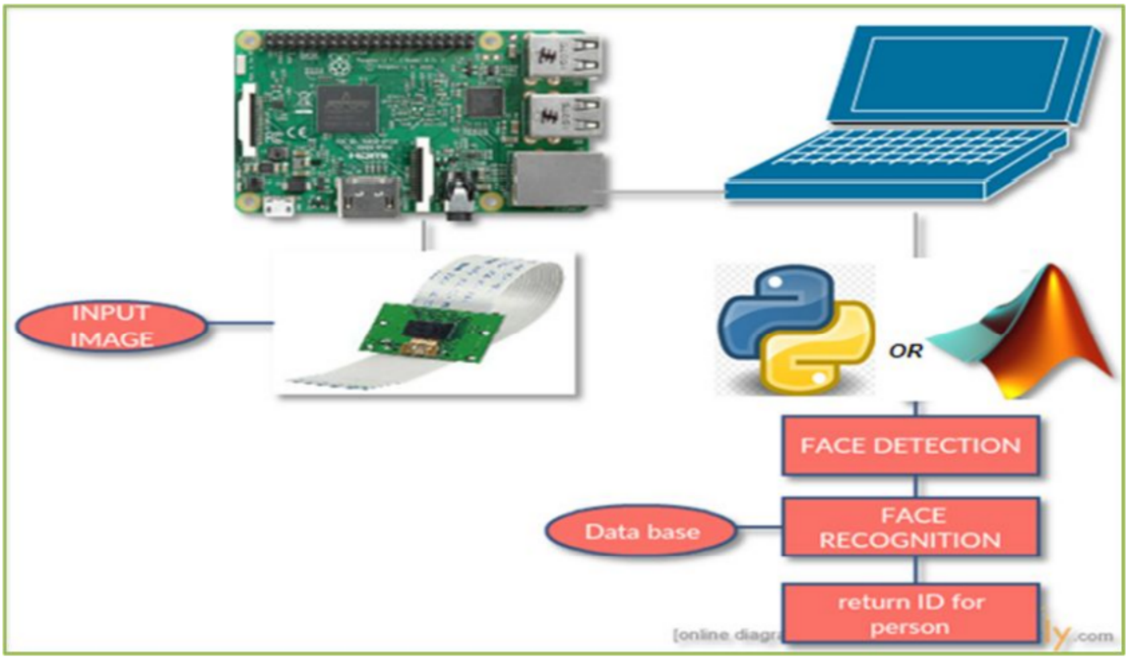
Technique for calculation of sum of regions for calculation of haar like features in a constant amount of time.

AdaBoost : One way for a new predictor to correct its predecessor is to pay a bit more attention to the training instances that the predecessor under-fitted. This results in new predictors focusing more and more on the hard cases. This is known as Adaptive Boosting. For example, to build an Adaptive Boosting classifier, a first base classifier (such a Decision Tree or SVM classifier) is trained and used to make predictions on the training set. The relative weights of the misclassified predictions are altered and increased in order to lay more emphasis on these predictions while making the next predictor. A second classifier is trained using the updated weights and again it makes predictions on the training set, weights are updated and so on. Once all the predictions are trained, the ensemble method makes predictions very much like boosting except the predictors have different weights depending on their overall accuracy on the weighted training set. The drawback of this type of algorithm is that it cannot be parallelized thereby increasing time required. Thus after successfully running AdaBoost on all the features we are left with the most relevant features required for detection. Therefore, this reduces computational time as we don't have to go through all the features and is much more efficient.

PROJECT BREAKDOWN

Functional Block Diagram

The whole system should consist of



Haar Cascade Classifier

Object detection using Haar feature-based cascade classifiers is an accurate method put ahead by Viola and Jones. It is a classifier which is used for identifying the objects from the source for which it has been trained for. This Haar Cascade is trained by superimposing these positive images over a set of negative images. Improved results are obtained by using excellent quality images and subsequently increasing the number of stages for which the classifier is trained . Haar-like features are an over-complete set of 2D Haar functions that might be utilized to encode the relative appearance of objects. They comprise more than two horizontal quadrilateral regions enclosed in a template. A Haar Cascade feature f has k rectangles given by the equation:

$$f = \sum_{i=01}^{k} w^i . \mu^i$$

where μi can be called the average pixel intensity in the z picture covered in the ith rectangle. As of convention, the weights allocated to the quadrilaterals of a Haar-like feature are set as default positive and negative numbers in a way that satisfies the given relation :

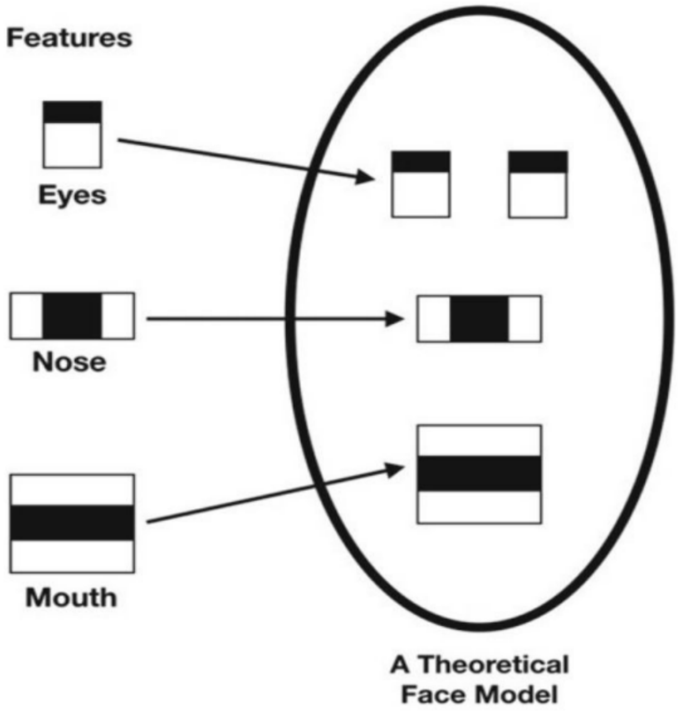
$$\sum_{i=01}^{k} w^i = 00$$

Haar Cascade is described in detail by Viola–Jones. Three main ideas behind the triumph of this detector are the integral image, the AdaBoost and the attentional cascade structure . The integral image is a stepwise procedure for a fast as well as precise calculation for the total of intensity values in the subset of an image which is rectangular in shape.

$$ii(x, y) = \sum_{x^J \leq x, y^J \leq y} i(x^J, y^J)$$

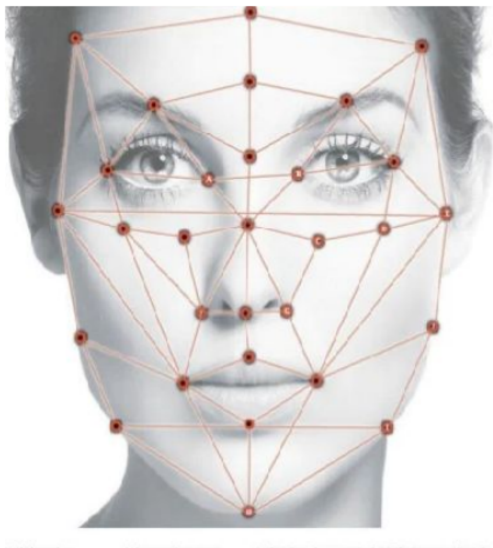
where i(x, y) is the intensity of the grey-scale image at pixel (x, y). The overall addition of the intensity pixels of any quadrilateral area ABCD can be computed with just four array references as:

$$\sum_{(x,y) \in ABCD}^{i(x,y)} = ii(D) + ii(B) - ii(C) - ii(D)$$



How Facial Recognition System Works

Every face has at least 80 distinguishable parts called nodal points.



Here are few nodal points below

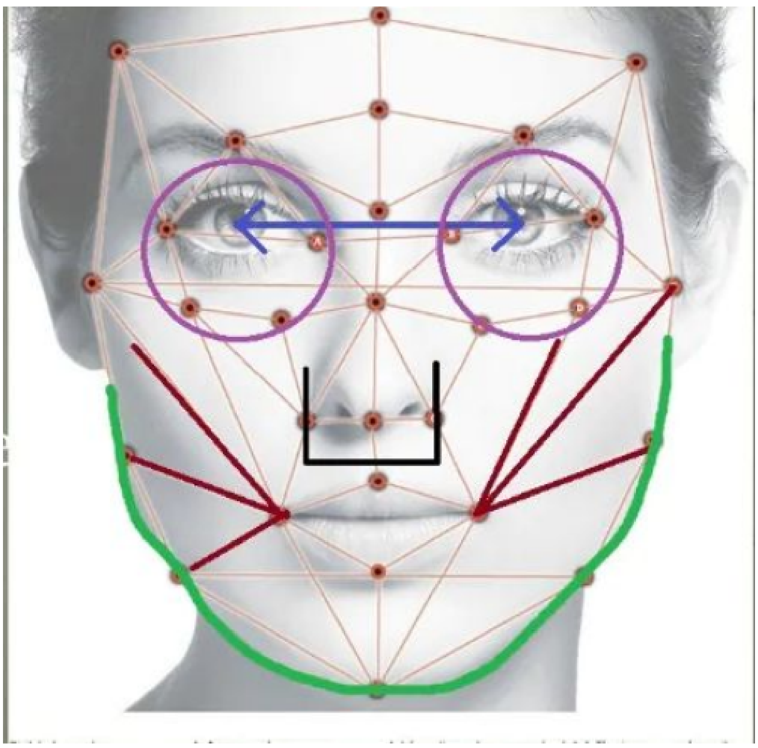
- Distance between the eyes

- Width of the nose

- Depth of eye sockets

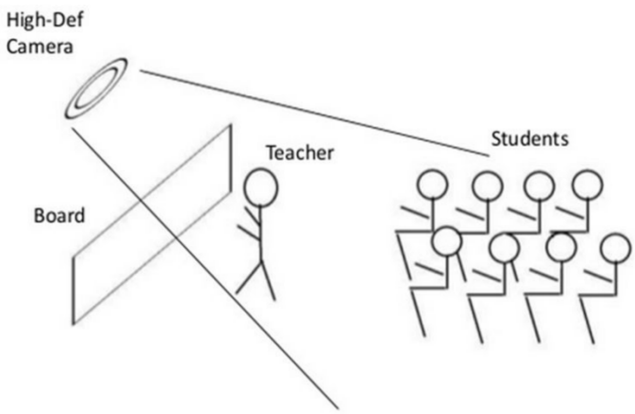
- Structure of the cheekbone

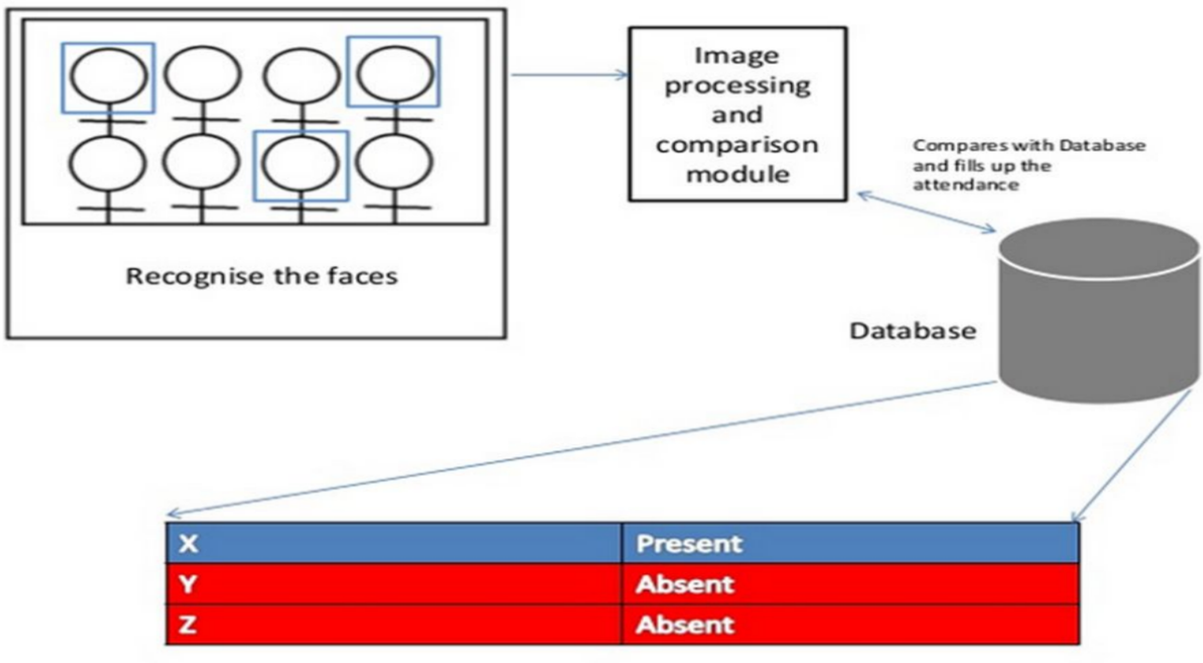
- Length of jaw line



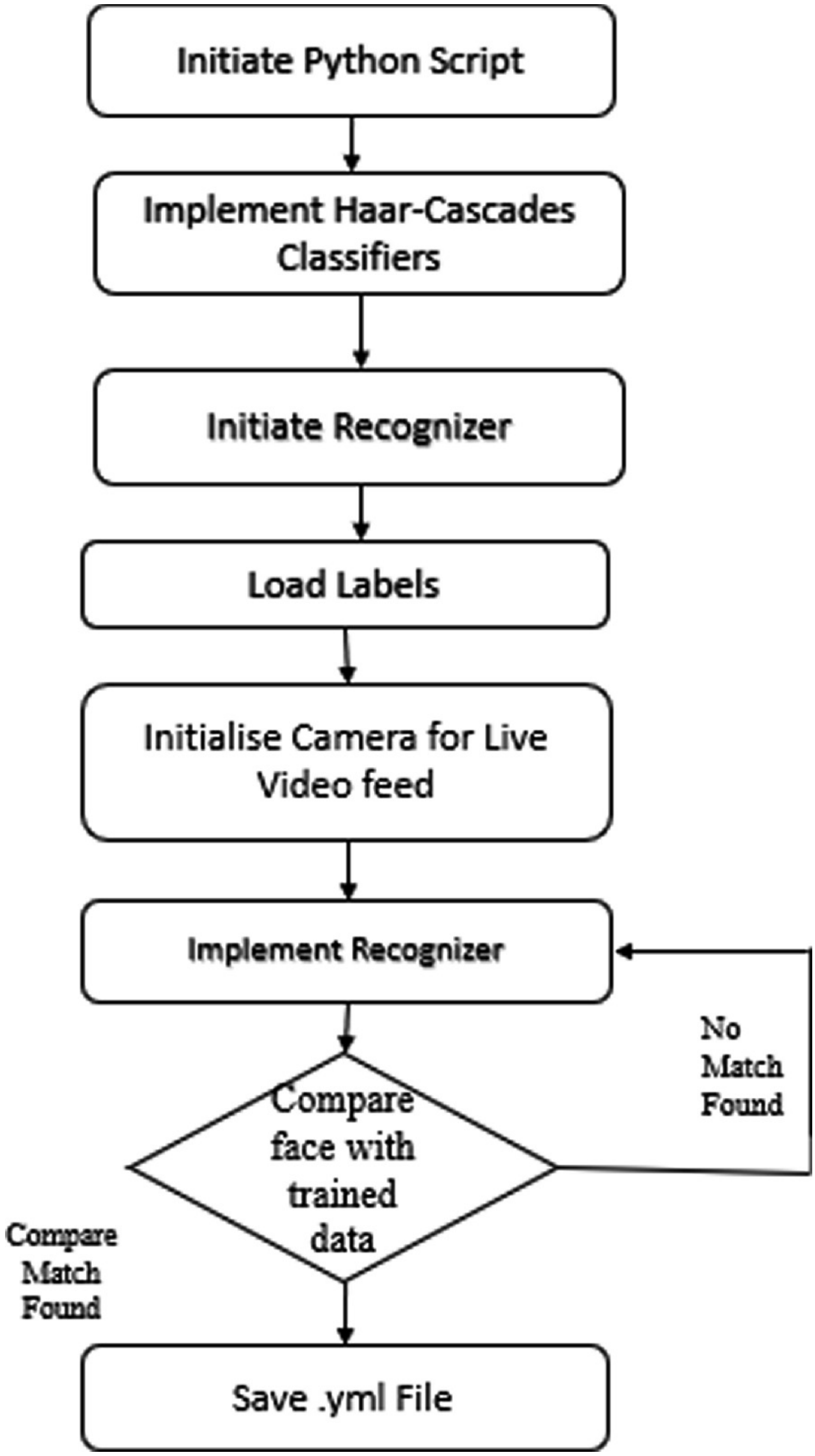
Nodal Point Representation on image

Basic Structure





FLOWCHART



Initiate Python Script
Implement Haar-Cascades Classifiers
Initiate Recognizer
Load Labels
Initialise Camera for Live Video feed
Implement Recognizer
No Match Found
Compare face with trained data
Compare Match Found
Save .yml File

IMPLEMENTATION

Step 1 : removal of noise from the Image and image processing is performed using the algorithms used and then we classify images using Haar-Cascade classifier

Face Recognition — Step by Step

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 surround 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.

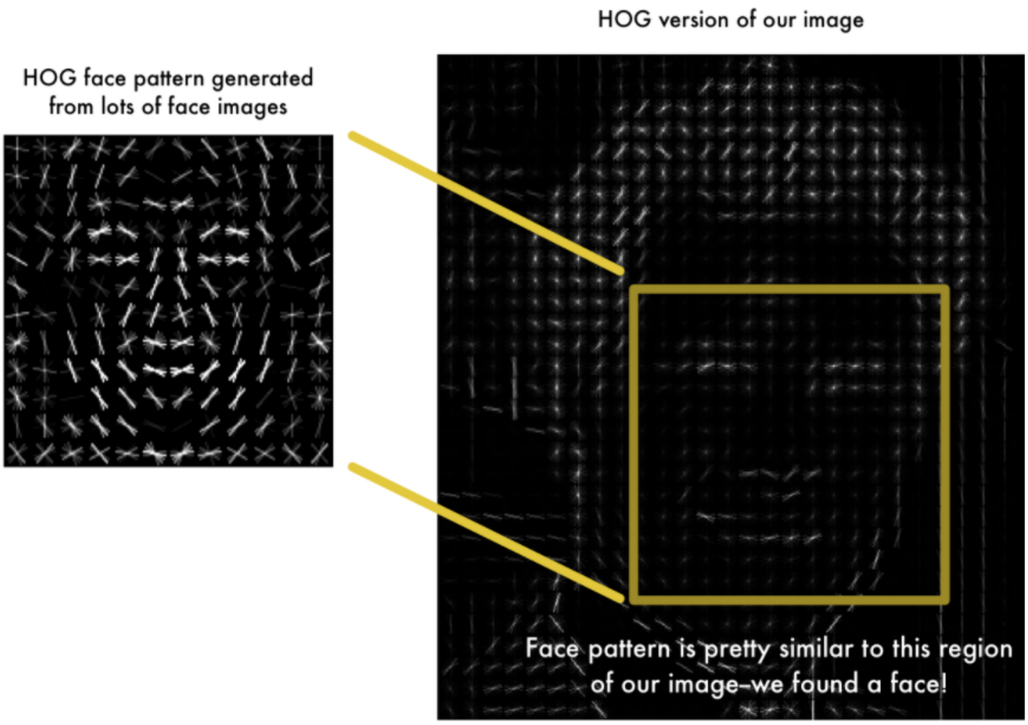
If you repeat that process for every single pixel in the image, you end up with every pixel being replaced by an arrow. These arrows are called *gradients* and they show the flow from light to dark across the entire image.

This might seem like a random thing to do, but there's a really good reason for replacing the pixels with gradients. 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!

To do this, 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 in a simple way:

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



Step 2: Posing and Projecting Faces

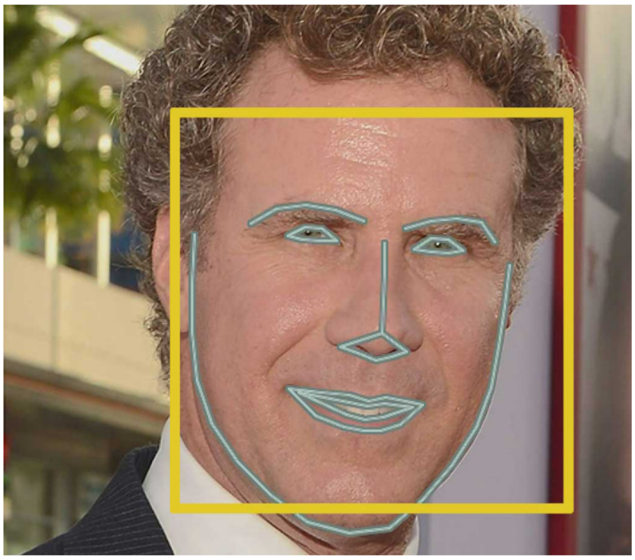
Whew, we isolated the faces in our image. But now we have to deal with the problem that faces turned in different directions look totally different to a computer.
To account for this, we will try to warp each picture so that the eyes and lips are always in the sample place in the image. This will make it a lot easier for us to compare faces in the next steps.

To do this, we are going to use an algorithm called face landmark estimation. There are lots of ways to do this, but we are going to use the approach [invented in 2014 by Vahid Kazemi and Josephine Sullivan.]

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:

The 68 landmarks we will locate on every face.

Here’s the result of locating the 68 face landmarks on our test image



68 face landmark test image

Now that we know where the eyes and mouth are, we’ll simply rotate, scale and [shear](#) the image so that the eyes and mouth are centered as best as possible. We won’t do any fancy 3d warps because that would introduce distortions into the image. We are only going to use basic image transformations like rotation and scale that preserve parallel lines (called [affine transformations](#)):

Now no matter how the face is turned, we are able to center the eyes and mouth in roughly the same position in the image. This will make our next step a lot more accurate.

Step 3: Encoding Faces

Now we are to the meat of the problem — actually telling faces apart. This is where things get really interesting!

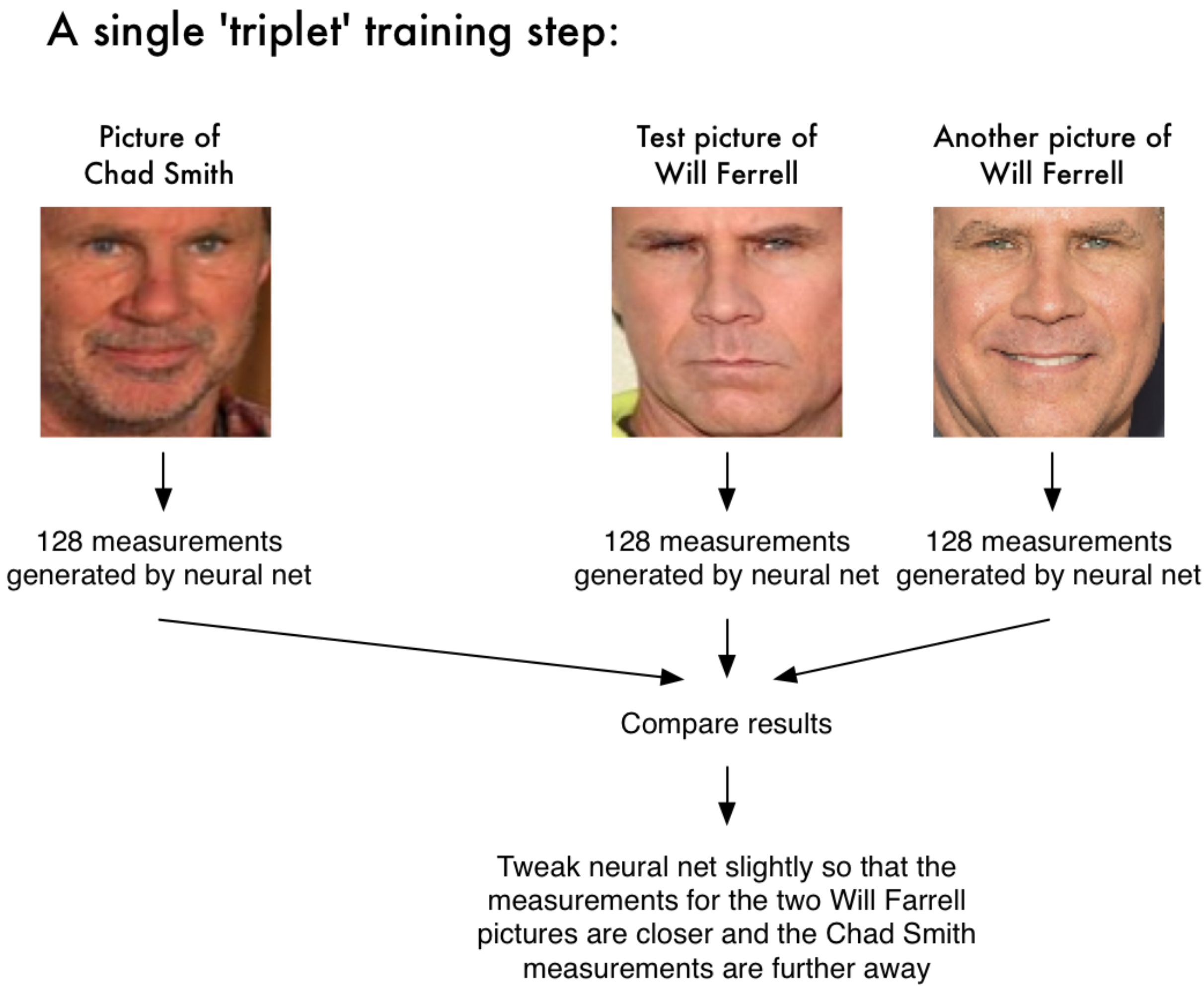
The simplest approach to face recognition is to directly compare the unknown face we found in Step 2 with all the pictures we have of people that have already been tagged. When we find a previously tagged face that looks very similar to our unknown face, it must be the same person. Seems like a pretty good idea, right?

There’s actually a huge problem with that approach. A site like Facebook with billions of users and a trillion photos can’t possibly loop through every previously-tagged face to compare it to every newly uploaded picture. That would take way too long. They need to be able to recognize faces in milliseconds, not hours.

What 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.

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



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. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning

Step 4: Finding the person’s name from the encoding

This last step is actually the easiest step in the whole process. All we have to do is find the person in our database of known people who has the closest measurements to our test image.

You can do that by using any basic machine learning classification algorithm. No fancy deep learning tricks are needed. We'll use a simple linear SVM classifier, but lots of classification algorithms could work.

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

Let's review the steps we followed:

1. Encode a picture using the HOG algorithm to create a simplified version of the image. Using this simplified image, find the part of the image that most looks like a generic HOG encoding of a face.
2. Figure out the pose of the face by finding the main landmarks in the face. Once we find those landmarks, use them to warp the image so that the eyes and mouth are centered.
3. Pass the centered face image through a neural network that knows how to measure features of the face. Save those 128 measurements.
4. Looking at all the faces we've measured in the past, see which person has the closest measurements to our face's measurements. That's our match

CODE SNIPPETS

Import Modules and Libraries :

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

Import Images and Print Names :

```
path = 'ImagesAttendance'
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)
```

Encoding Images :

```
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
```

```python
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)
        matchIndex = np.argmin(faceDis)
```

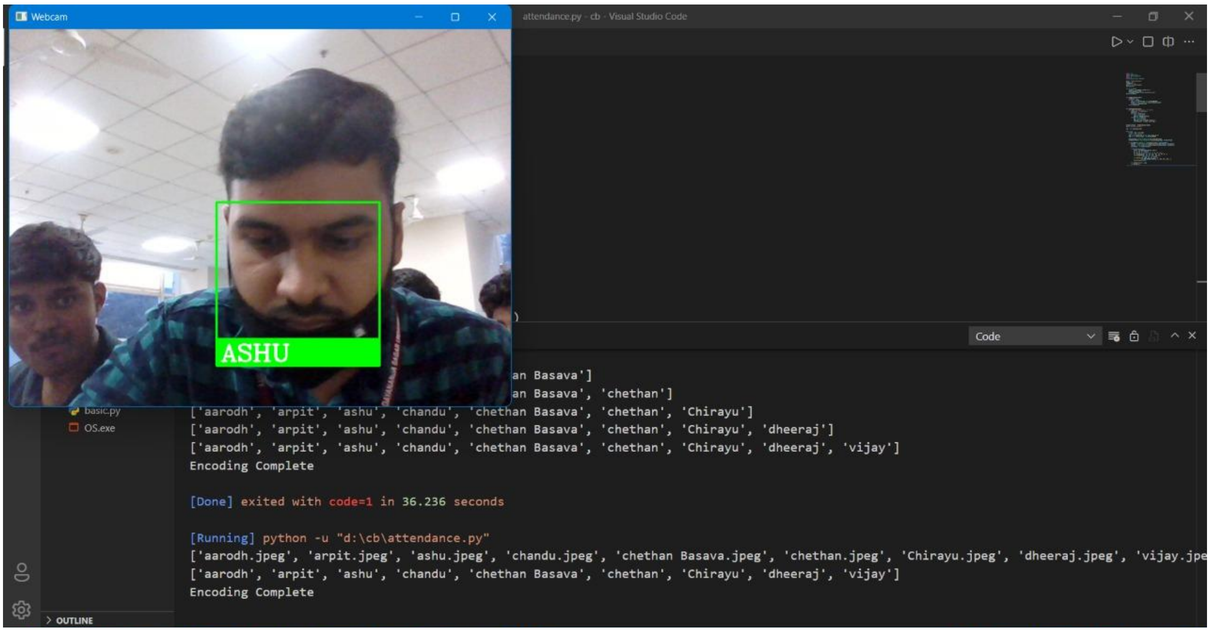
If match Found from Databases Mark Attendance :

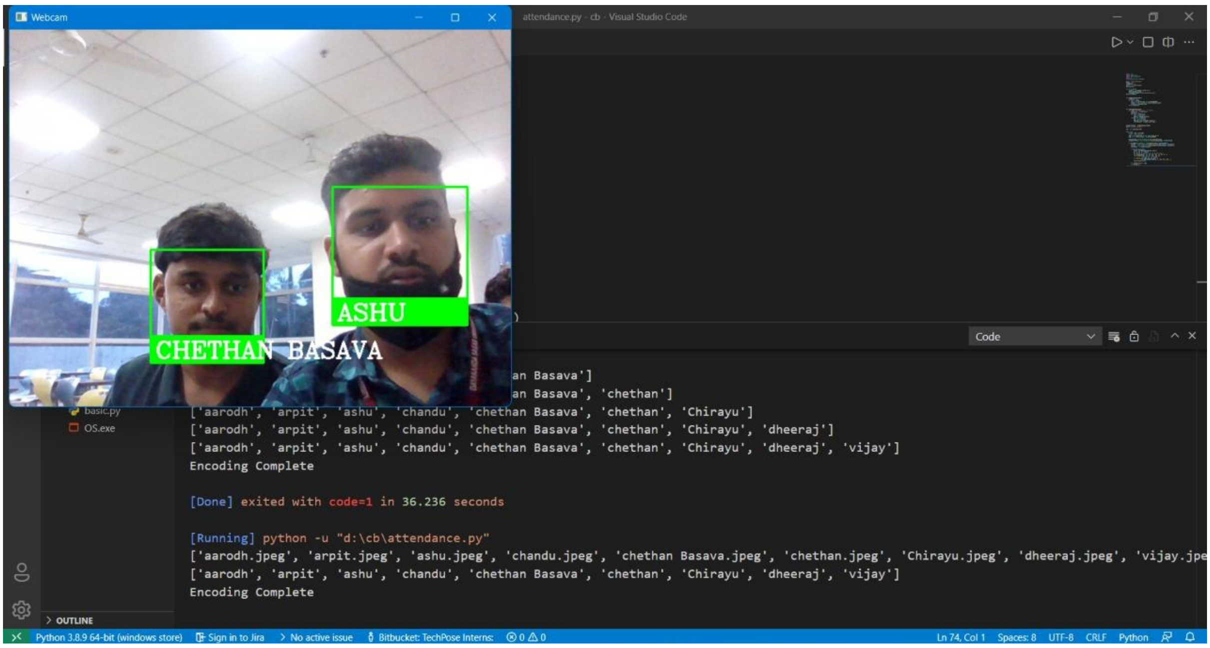
```
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(img, (x1, y2 - 35), (x2, 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)
```

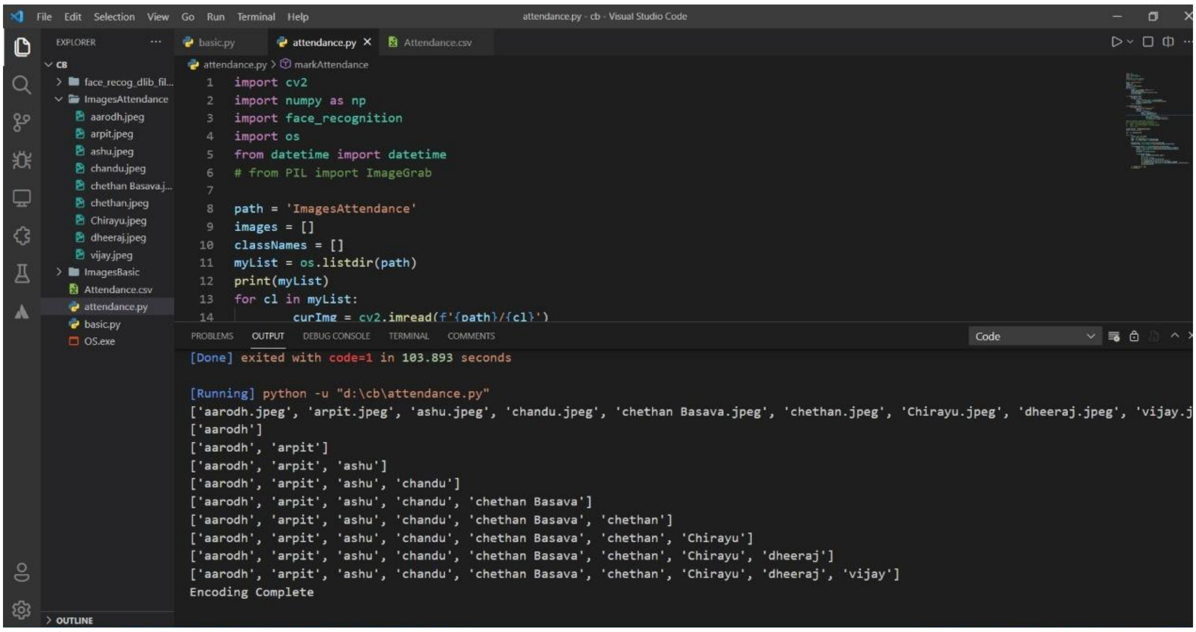
Accessing Web Camera :

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

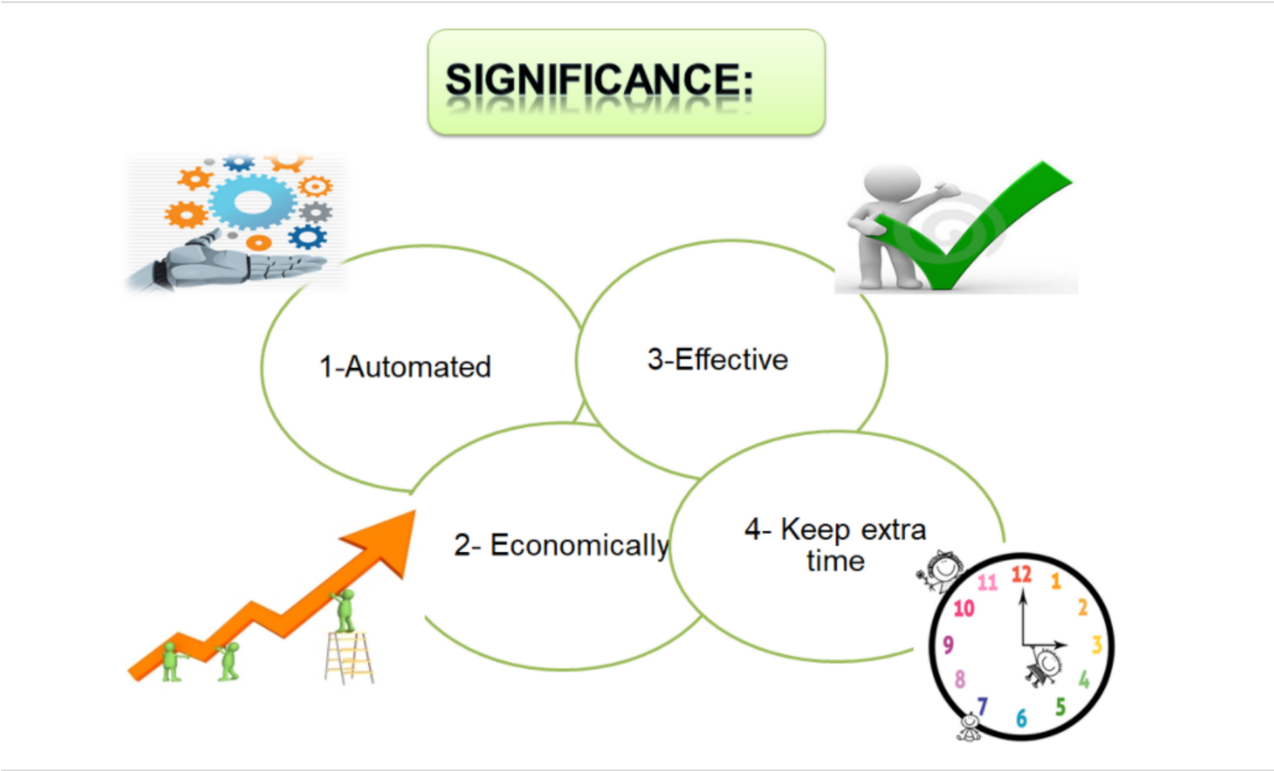
OUTPUT SCREENSHOTS







import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime
# from PIL import ImageGrab
path = 'ImagesAttendance'
images = []
classNames = []
myList = os.listdir(path)
print(myList)
for cl in myList:
[Done] exited with code=1 in 103.893 seconds
[Running] python -u "d:\cb\attendance.py"
['aarodh.jpeg', 'arpit.jpeg', 'ashu.jpeg', 'chandu.jpeg', 'chethan Basava.jpeg', 'chethan.jpeg', 'Chirayu.jpeg', 'dheeraj.jpeg', 'vijay.j
['aarodh']
['aarodh', 'arpit']
['aarodh', 'arpit', 'ashu']
['aarodh', 'arpit', 'ashu', 'chandu']
['aarodh', 'arpit', 'ashu', 'chandu', 'chethan Basava']
['aarodh', 'arpit', 'ashu', 'chandu', 'chethan Basava', 'chethan']
['aarodh', 'arpit', 'ashu', 'chandu', 'chethan Basava', 'chethan', 'Chirayu']
['aarodh', 'arpit', 'ashu', 'chandu', 'chethan Basava', 'chethan', 'Chirayu', 'dheeraj']
['aarodh', 'arpit', 'ashu', 'chandu', 'chethan Basava', 'chethan', 'Chirayu', 'dheeraj', 'vijay']
Encoding Complete



APPLICATION

1) **Primary application** being used in classrooms to take the attendance of the students.

2) Decrease the **false attendance.**

3) **Security/Counterterrorism:** Access control, comparing surveillance terrorist. images to know

4) **ATM:** The software is able to quickly verify a customer's face.

5) **Healthcare:** Minimize fraud by verifying patient

CONCLUSION

Before the development of this project. There are many loopholes in the process of taking attendance using the old method which caused many troubles to most of the institutions. Therefore, the facial recognition feature embedded in the attendance monitoring system can not only ensure attendance to be taken accurately and also eliminate the flaws in the previous system.

Using technology to conquer the defects cannot merely save resources but also reduces human intervention in the whole process by handling all the complicated tasks to the machine.

In conclusion, a better attendance monitoring system should be developed based on its portability, accessibility and the accuracy of the collected attendance information

FUTURE SCOPE

Automatic attendance system can be improved by increasing the number of features which can be extracted to increase accuracy of face recognition. Once the software is developed and tested properly, it could be improved to cover full institutions such as the faculty of engineering.

It is estimated that it will also be adopted by retailers and banking systems in coming years to keep fraud in debit/credit card purchases and payment especially the ones that are online. This technology would fill in the loopholes of the largely prevalent inadequate password system. In the long run, robots using facial recognition technology may also come to foray. They can be helpful in completing the tasks that are impractical or difficult for human beings to complete.

For security reasons, we can use a detection & recognition system. To identify culprits on bus stations, railway stations and 7 other public places, we can use this system. This will be a helping hand to the police. In this system, we will use the GSM module. Suppose if the culprit is detected, then the detected signal can be transmitted using the GSM module to the central control room of the police station. With the help of ISDN number of GSM, culprit surviving area will be recognized

REFERENCES

1. D. Gupta et al. (eds.), International Conference on Innovative Computing and Communications, AdvaD. Gupta et al. (eds.), International Conference on Innovative Computing and Communications, Advances in Intelligent Systems and Computing 1165,
2. https://doi.org/10.1007/978-981-15-5113-0_79
3. Roshan Tharanga, S. M. S.C. Samarakoon - "Smart attendance using real time face recognition,", 2013.
4. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems by Aurelien Geron
5. N.Sudhakar Reddy, M.V.Sumanth, S.Suresh Babu, "A Counterpart Approach to Attendance and Feedback System using Machine Learning Techniques",Journal of Emerging Technologies and Innovative Research (JETIR), Volume 5, Issue 12, Dec 2018.
6. Dan Wang, Rong Fu, Zuying Luo, "Classroom Attendance Auto-management Based on Deep Learning",Advances in Social Science, Education and Humanities Research, volume 123,ICESAME 2017.
7. Akshara Jadhav, Akshay Jadhav, Tushar Ladhe, Krishna Yeolekar, "Automated Attendance System Using Face Recognition", International Research Journal of Engineering and Technology (IRJET), Volume 4, Issue 1, Jan 2017.
8. B Prabhavathi, V Tanuja, V Madhu Viswanatham and M Rajashekhara Babu, "A smart technique for attendance system to recognize faces through parallelism", IOP Conf. Series: Materials Science and Engineering 263, 2017.
9. Prajakta Lad, Sonali More, Simran Parkhe, Priyanka Nikam, Dipalee Chaudhari, " Student Attendance System Using Iris Detection", IJARIIE-ISSN(O)-2395-4396, Vol-3 Issue-2 2017.
10. Samuel Lukas, Aditya Rama Mitra, Ririn Ikana Desanti, Dion Krisnadi, "Student Attendance System in Classroom Using Face Recognition Technique", Conference Paper DOI: 10.1109/ICTC.2016.7763360, Oct 2016.
11. K.Senthamil Selvi, P.Chitrakala, A.Antony Jenitha, "Face Recognition Based Attendance Marking System", IJCSMC, Vol. 3, Issue. 2, February 2014.
12. Yohei KAWAGUCHI, Tetsuo SHOJI, Weijane LIN, Koh KAKUSHO, Michihiko MINOH, "Face Recognition-based Lecture Attendance System", Oct 2014.
13. Shireesha Chintalapati, M.V. Raghunadh, "Automated Attendance Management System Based On Face Recognition Algorithms", IEEE International Conference on Computational Intelligence and Computing Research, 2013.
14. B. K. Mohamed and C. Raghu, “Fingerprint attendance system for classroom needs,” India Conference (INDICON), Annual IEEE, pp. 433–438, 2012.