FACE RECOGNITION AND ATTENDANCE REPORT USING PYTHON(HOG)

A MINI PROJECT REPORT

Submitted by

AJAY SAIRAM.N (REGNO: 212219060010)

ARJUN DINESH.S (REGNO:212219060025)

AUGUSTINE.E (REGNO:212219060034)

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

SAVEETHA ENGINEERING COLLEGE(AUTONOMOUS),CHENNAI

ANNA UNIVERSITY :: CHENNAI 600 025

JANUARY 2022

SAVEETHA ENGINEERING COLLEGE (AUTONOMOUS),CHENNAI

ANNA UNIVERSITY, CHENNAI – 600 025

BONAFIDE CERTIFICATE

Certified that this mini project report "**FACE RECOGNITION AND ATTENDANCE REPORT USING PYTHON (HOG)**" is the bonafide work of " **N. Ajay Sairam(212219060010), S. Arjun Dinesh (212219060025), E. Augustine (212219060034) "** who carried out the project work under my/our supervision.

SIGNATURE	**SIGNATURE**
Dr.Srigitha. S. Nath, M.E., Ph.D.,	**Ms.E.T.Jaba Jasphin,M.E.,(Ph.D),**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Electronics and Communication Engineering	Department of Electronics and Communication Engineering
Saveetha Engineering College,	Saveetha Engineering College,
Saveetha Nagar,Thandalam,	Saveetha Nagar,Thandalam,
Chennai-602105	Chennai-602105

Submitted for the mini project viva-voce examination held on__________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

PROJECT APPROVAL SHEET

The project sheet **“ FACE RECOGNITION AND ATTENDANCE REPORT USING PYTHON(HOG) ”** submitted by **“ N. Ajay Sairam(212219060010), S. Arjun Dinesh (212219060025), E. Augustine (212219060034) ”** is approved for submission, as partial requirement for the award of the **Degree of Bachelor of Engineering in Electronics and Communication,** Anna University during the academic year 2020- 2021.

Submitted for the “University Mini Project Viva Voce” examination held on_________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We convey our sincere thanks to **Dr. N. M. Veeraiyan -** President(SMET) and Chancellor-SIMATS, Saveetha Amaravathi University, **Dr. S. Rajesh,** Director - Saveetha Engineering College and **Dr. V. Saveetha Rajesh –** Director, Saveetha Medical College and Hospital for providing us with the facilities for the completion of our project. We are grateful to our Principal, **Dr. N. Durai Pandian** for his continuous support and encouragement in carrying out our project work. We are deeply indebted to our beloved Head of the Department, **Dr.Srigitha. S. Nath, M.E., Ph.D.,** Department of Electronics and Communication, for giving us the opportunity to display our professional skills through this project.

We are greatly thankful to our Project Guide, **Ms. E. T. Jaba Jasphin, M.E., (Ph.D),** and our co-ordinator **Dr.S.Asha , M.E., Ph.D.,** for their valuable guidance and motivation which helped to complete our project on time.

We thank all our teaching and non- teaching faculty members of the Department of Electronics and Communication for their passionate support, for helping us to identify our mistakes and also for the appreciation they gave us. We heartily thank our library staff and the management for their extensive support in providing the resources and information that helped us to complete the project successfully. Also, we would like to record our deepest gratitude to our parents for their constant encouragement and support, which motivated us a lot to complete our project work.

ABSTRACT

Advancements in machine learning have been applying deeply and widely in numerous fields. Face recognition is among the most productive image processing applications and has a pivotal role in the technical field. This work leverages machine learning methods and information systems to present a framework for student attendance combining machine learning-based face recognition algorithm and relational databases to store, recognize, and record student attendance. The proposed method is tested on various scenarios and is expected to apply in practical cases.

The investigation is done by Histogram of Oriented Gradients (HOG) for face detection and by using cosine distance to recognize faces. The purpose of this study is face recognition in real-time i.e. using a webcam, camera of the mobile device, and from a photograph or from a set of faces tracked in a video. We measured the distance between the landmarks and compared the test image with different known encoded image landmarks in the recognition stage. Face Recognition includes extracting features and then recognizing it, in any case, such as brightness, transformations as translation, rotation, and scale image. HOG algorithm is used to detect faces .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

HOG – HISTOGRAM OF ORIENTED GRADIENTS

SVM – SUPPORT VECTOR MACHINE

CNN - CONVOLUTIONAL NEURAL NETWORK

CHAPTER 1
INTRODUCTION

1.1 OVERVIEW:

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image. Face detection is an easy and simple task for humans, but not so for computers. It plays an important part in many biometric, security and surveillance systems, as well as image and video indexing systems.

1.2 BIOMETRICS

Biometrics is used in the process of authentication of a person by verifying or identifying that a user requesting a network resource is who he, she, or it claims to be, and vice versa. It uses the property that a human trait associated with a person itself like structure of data with the incoming data we can verify the identity of a particular person [1]. There are many types of biometric system like detection and recognition, iris recognition etc., these traits are used for human identification in surveillance system, criminal identification , face details etc. By comparing the existing fingerprint recognition.

1.3 FACE RECOGNITION

Human beings have recognition capabilities that are unparalleled in the modern computing era. These are mainly due to the high degree of interconnectivity, adaptive nature, learning skills and generalization capabilities of the nervous system. The human brain has numerous highly interconnected biological neurons which, on some specific tasks, can outperform super computers. A child can accurately identify a face, but for a computer it is a cumbersome task. Therefore, the main idea is to engineer a system which can emulate what a child can do. Advancements in computing capability over the past few decades have enabled comparable recognition capabilities from such engineered systems quite successfully.

Early face recognition algorithms used simple geometric models, but recently the recognition process has now matured into a science of sophisticated mathematical representations and matching processes. Major advancements and initiatives have propelled face recognition technology into the spotlight. Face recognition technology can be used in wide range of applications. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification etc. Face detection and recognition is used in many places nowadays, verifying websites hosting images and social networking sites.

Face recognition and detection can be achieved using technologies related to computer science. Features extracted from a face are processed and compared with similarly processed faces present in the database. If a face is recognized it is known or the system may show a similar face existing in database else it is unknown. In surveillance system if a unknown face appears more than one time then it is stored in database for further recognition. These steps are very useful in criminal identification. In general, face recognition techniques can be divided into two groups based on the face representation they use appearance-based, which uses holistic texture features and is applied to either whole-face or specific face image and feature-based, which uses geometric facial features (mouth, eyebrows, cheeks etc.), and geometric relationships between them. (A few example applications are shown in Fig 1.1.)

CHAPTER 2

LITERATRURE SURVEY

Several algorithms and techniques for face recognition have been developed in the past by researchers. These are discussed briefly in this section.

2.1 FACE RECOGNITION BASED ON INDEPENDENT COMPONENT ANALYSIS

A number of current face recognition algorithms use face representations found by unsupervised statistical methods. Typically these methods find a set of basis images and represent faces as a linear combination of those images. Principal component analysis (PCA) is a popular example of such methods. The basis images found by PCA depend only on pairwise relationships between pixels in the image database. In a task such as face recognition, in which important information may be contained in the high-order relationships among pixels, it seems reasonable to expect that better basis images may be found by methods sensitive to these high-order statistics. Independent component analysis (ICA), a generalization of PCA, is one such method. We used a version of ICA derived from the principle of optimal information transfer through sigmoidal neurons. ICA was performed on face images in the FERET database under two different architectures, one which treated the images as random variables and the pixels as outcomes, and a second which treated the pixels as random variables and the images as outcomes. The first architecture found spatially local basis images for the faces. The second architecture produced a factorial face code. Both ICA representations were superior to representations based on PCA for recognizing faces across days and changes in expression. A classifier that combined the two ICA representations gave the best performance.

2.2 EIGEN-SPACES

Eigenspace-based face recognition corresponds to one of the most successful methodologies for the computational recognition of faces in digital images. Starting with the EigenfaceAlgorithm, different eigenspace-based approaches for the recognition of faces have been proposed. They differ mostly in the kind of projection method used (standard, differential, or kernel eigenspace), in the projection algorithm employed, in the use of simple or differential images before/after projection, and in the similarity matching criterion or classification method employed.

The aim of this paper is to present an independent comparative study among some of the main eigenspace-based approaches. We believe that carrying out independent studies is relevant, since comparisons are normally performed using the implementations of the research groups that have proposed each method, which does not consider completely equal working conditions for the algorithms. Very often, a contest between the abilities the research groups rather than a comparison between methods is performed. This study considers theoretical aspects as well as simulations performed using the Yale Face Database, a database with few classes and several images per class, and FERET, a database with many classes and few images per class.

2.3 ELASTIC BUNCH GRAPH MATCHING

Elastic Bunch Graph Matching is one of the well-known methods proposed for face recognition. In this work, we propose several extensions to Elastic Bunch Graph Matching and its recent variant Landmark Model Matching. We used data from the FERET database for experimentations and to compare the proposed methods. We apply Particle Swarm Optimization to improve the face graph matching procedure in Elastic Bunch Graph Matching method and demonstrate its usefulness. Landmark Model Matching depends solely on Gabor wavelets for feature extraction to locate the landmarks (facial feature points). We show that improvements can be made by combining gray-level profiles with Gabor wavelet features for feature extraction. Furthermore, we achieve improved recognition rates by hybridizing Gabor wavelet with eigenface features found by Principal Component Analysis, which would provide information contained in the overall appearance of a face. We use Particle Swarm Optimization to fine tune the hybridization weights.[3] Results of both fully automatic and partially automatic versions of all methods are presented. The best-performing method improves the recognition rate up to 22.6speeds up the processing time by 8 times over the Elastic Bunch Graph Matching for the fully automatic case.

2.4 LINEAR DISCRIMINANT ANALYSIS

Both PCA and ICA do not use face class information. Linear Discriminant Analysis (LDA) finds an efficient way to represent the face vector space by exploiting the class information. It differentiates individual faces but recognizes faces of the same individual]. LDA searches for vectors in the underlying space that best discriminate among classes. For all the samples of all classes, two measures are defined.

CHAPTER 3
FACE RECOGNITION AND ATTENDANCE REPORT USING PYTHON(HOG)

Let's tackle this problem one step at a time. For each step, we'll learn about a different machine learning algorithm. I'm not going to explain every single algorithm completely to keep this from turning into a book, but you'll learn the main ideas behind each one and you'll learn how you can build your own facial recognition system in Python using "Open Face" and "dlib".

3.1 FINDING ALL THE FACES

The first step in our pipeline is face detection. Obviously we need to locate the faces in a photograph before we can try to tell them apart. If you've used any camera in the last 10 years, you've probably seen face detection in action:

Face detection went mainstream in the early 2000's when Paul Viola and Michael Jones invented a way to detect faces that was fast enough to run on cheap cameras. However, much more reliable solutions exist now. We're going to use a method called **Histogram of Oriented Gradients** — or just **HOG** for short.

3.2 HISTOGRAM OF ORIENTED GRADIENTS (HOG):

The images undergo the process where it is separated into small pixels. 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.

The process is repeated 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 of light from **lighter to darker** across the entire image.

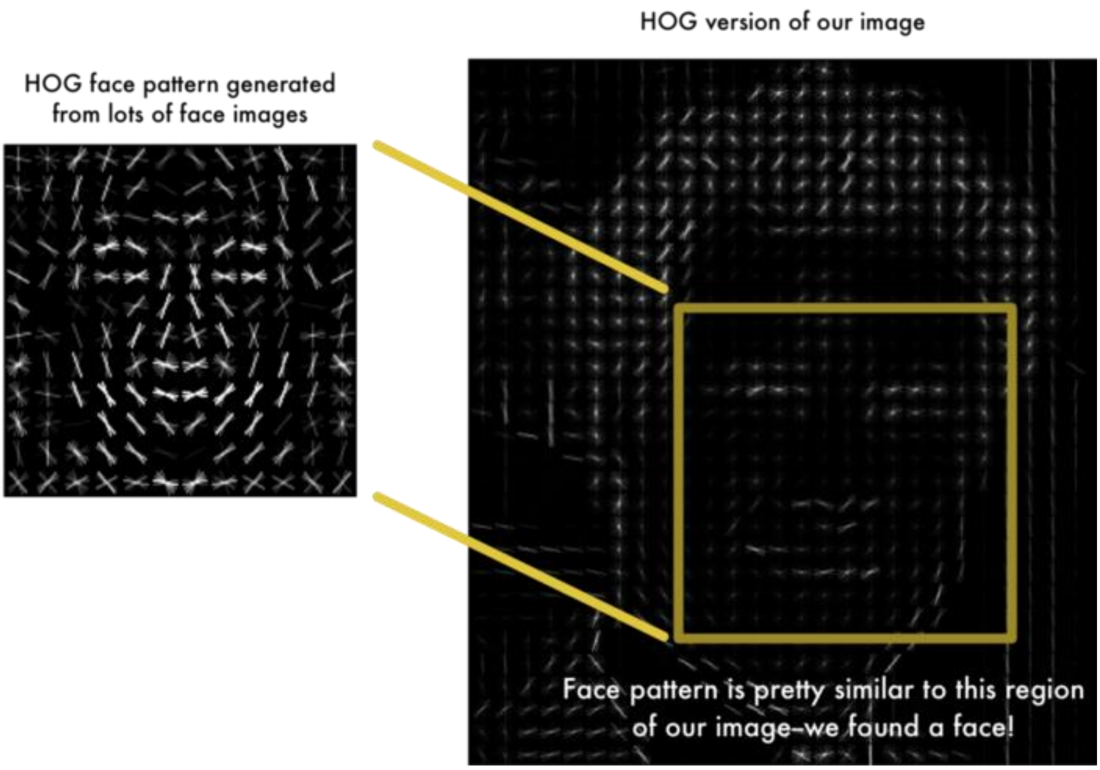

FIG 3.2.1 HOG VERSION OF THE IMAGE

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. Using this technique, we can now easily find faces in any image.

3.3 POSING AND PROJECTING FACES

Whew, we isolated the faces in our image. But now we have to deal with the problem that faces turned 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 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 specific points on any face.

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.

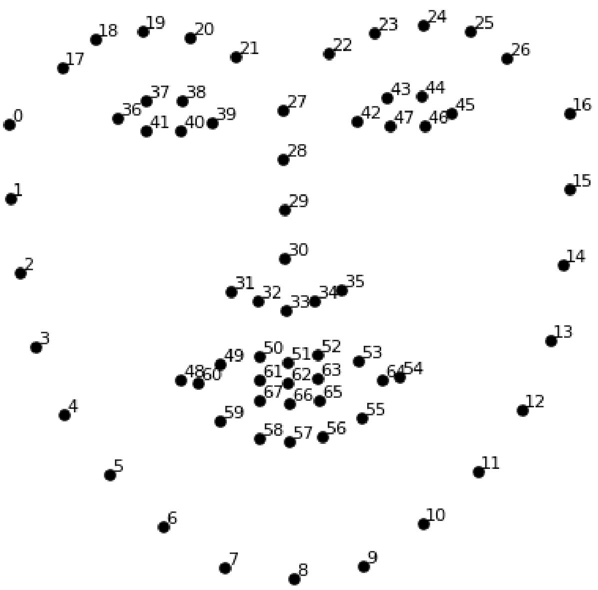

FIG 3.3.1.FACIAL LANDMARKS

3.4 FLOW CHART

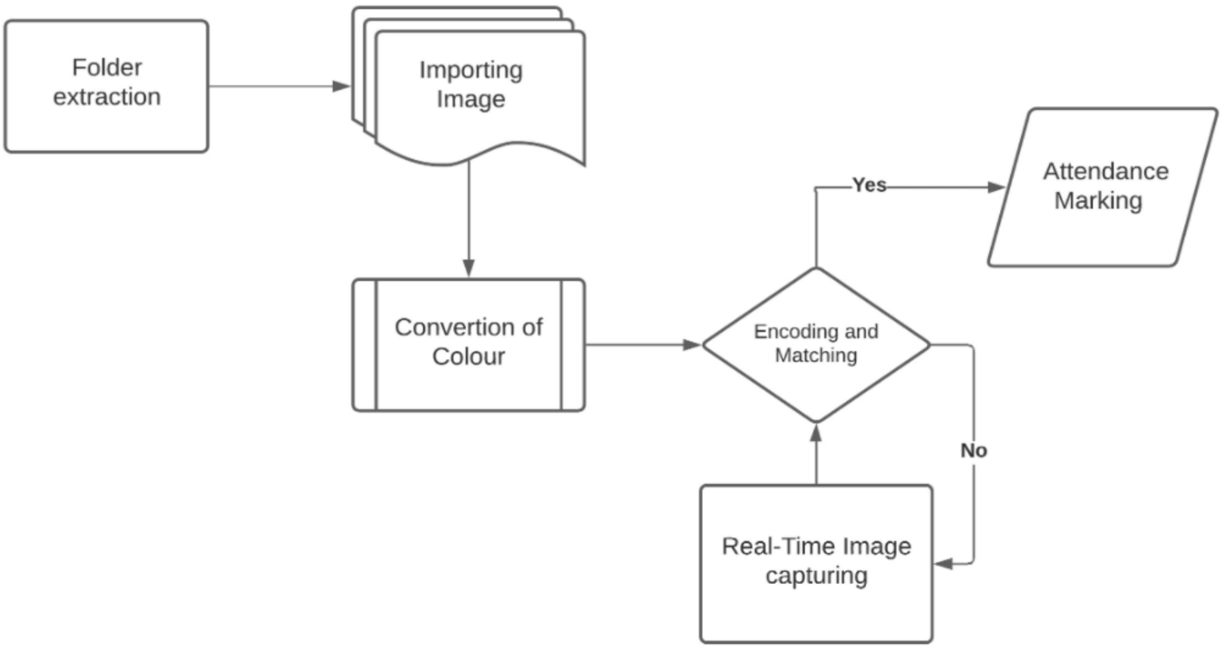

FIG 3.4.1 FLOW CHART

3.5 BLOCK DIAGRAM OF ENCODING :

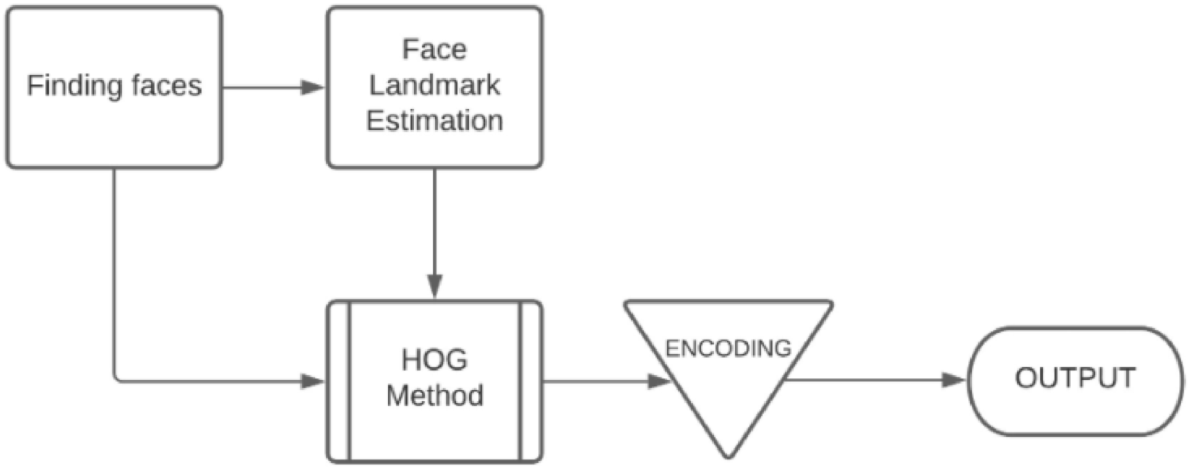

FIG 3.5.1 BLOCK DIAGRAM

3.6.1 ENCODING FACES:

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 of people that have already been tagged already.

The HOG-descriptor from the source image is calculated as follows:
In the first step, color normalization and gamma correction are performed. Next, the gradient values are calculated vertically and horizontally. After that, the image is divided into a uniform grid of cells.
The basic unit of a HOG- descriptor is a block — a rectangular area of image pixels of a given size. A block consists of cells that are assigned a histogram of directions (inclination relative to the horizontal) of gradients. The HOG-descriptor is the vector of the components of the normalized histograms of cells from all areas of the block. As a rule, blocks overlap, that is, each cell is included in more than one final descriptor.

The solution to the classification problem is to determine the class of an object based on its input characteristics. A training set is a set of n-dimensional vectors, and it is known to which class each vector belongs. When training a classifier, an approximate function is constructed that associates with an arbitrary object the class to which this object belongs. In the case of the SVM classifier, when training in a training set, a hyperplane is constructed, dividing the object space into classes. It is given by the following equation:

The solution to the classification problem is to determine the class of an object based on its input characteristics. A training set is a set of n-dimensional vectors, and it is known to which class each vector belongs. When training a classifier, an approximate function is constructed that associates with an arbitrary object the class to which this object belongs. In the case of the SVM classifier, when training in a training set, a hyperplane is constructed, dividing the object space into classes.

It is given by the following equation: $<w, x>= b$, where w is perpendicular to the separating hyperplane, b is a scalar threshold. In this case, the vector w is a linear combination of the support vectors of the elements of the training sample, which are closest to the separating hyperplane. In the case of using HOG-descriptors as objects, the reference vectors are some set of HOG-descriptors from the training set. Therefore, to obtain SVM classifiers for windows of new sizes from the original classifier, its support vectors can be scaled, treating them as HOG-descriptors.

The identification task can be solved by directly comparing the unknown person found in the classification step with all those already marked. If an unknown person is found that looks like a tagged one, then it’s the same person. The HOG- descriptor allows you to extract all areas on the original image that correspond to the found faces. For each person, it is necessary to find some basic characteristics for comparison with the characteristics of known persons.

The solution is to train the convolutional neural network (CNN) . But instead of learning the network to recognize graphic objects, it is now necessary to teach it to create 128 characteristics for each person. The learning process is valid when considering 3 face images at the same time: loading the learning face images of known persons; uploading another image of the same person’s face; loading the image of the face of some other person.

The algorithm considers the characteristics that it currently creates for each of these three images and adjusts the neural network so that the characteristics created by it for images 1 and 2 are a little closer to each other, and for images 2 and 3 - a little further. In case a trained neural network already exists, the rest of the task is to transfer the original images from the database through the previously trained network and obtain 128 characteristics for each person.

The final stage of identification consists in comparing the newly received image with those already existing in the database. In this paper, the Euclidean metric is used as a metric for estimating the distance between two vectors consisting of 128 face characteristics, which is the geometric distance between points x and y in n-dimensional space and is calculated using the Pythagorean theorem as the root of the sum of differences of squares of coordinates of points.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2} \quad \mathbf{<= 0.6}$$

Thus, the Euclidean distance is a similarity metric for a 128- dimensional vector.

3.6.2 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 tells which known person is the closest match. Running this classifier takes milliseconds. The result of the classifier is the name of the person.

3.7 FACE RECOGNITION SYSTEM IMPLEMENTATION:

The library files are imported initially. The library files such as “opencv, numpy,face_recognition,os,and datetime” are imported in the first place.

The images need to be trained are stored in a folder called “Images Attendance” and then the images extracted and stored in list called “Images”.

CHAPTER 4
PYTHON CODE:

4.1 FOR ENCODING PROCESS:

```
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
```

The given images are converted into BGR from RGB because the open cv function will work in this format only.

The command line

`img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)` is used the above process.

4.2 FOR CREATING A CSV FILE :

The main purpose of the system is to mark attendance based on the images shown in real-time.For this to happen a csv file should be created to store the name and time of the person entering.

```
with open('Attendance.csv','w') as f:
    f.writelines(f'Name             Time')
```

The file will be used for the entire process so it is then converted into write and readable file.

```
with open('Attendance.csv','r+') as f:
    myDataList = f.readlines()
```

4.3 FOR REAL-TIME IMAGE EXTRACTION:

The real-time image extraction process involves in which the face shown in the web-cam or the camera of the mobile phones will be extracted to the encoding process.

```
cap = cv2.VideoCapture(0)
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)
```

4.4 FOR ATTENDANCE MARKING:

For attendance marking , An **csv** file is created initially to store the list of persons. The created file has to be in write, read and append mode.

```
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}')
```

4.5 FOR MATCHING AND IDENTIFICATION:

If the image matches the next will come under execution. Creating a rectangle over the face in real-time.

```
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,0.70,(255,255,255),2)
    markAttendance(name)
```

If the image not matches the "UNKNOWN" text will be shown over it.

For this to happen the below code will be used.

```
else:
    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, 'Unknown', (x1 + 6, y2 - 6),
cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
    markAttendance('Unknown')
```

CHAPTER 5

WORKING

5.1 INSTALLATION:

Install **PYTHON** complier such as Pycharm community, PyDev, Spyder, Atom, Visual Studio Code, Jupyter Notebook and add python 3.9 or higher version of python interpreter. Here Pycharm community is used.

Create a new project and name it as say (Face recognition project). Select the virtual environment and type the code.

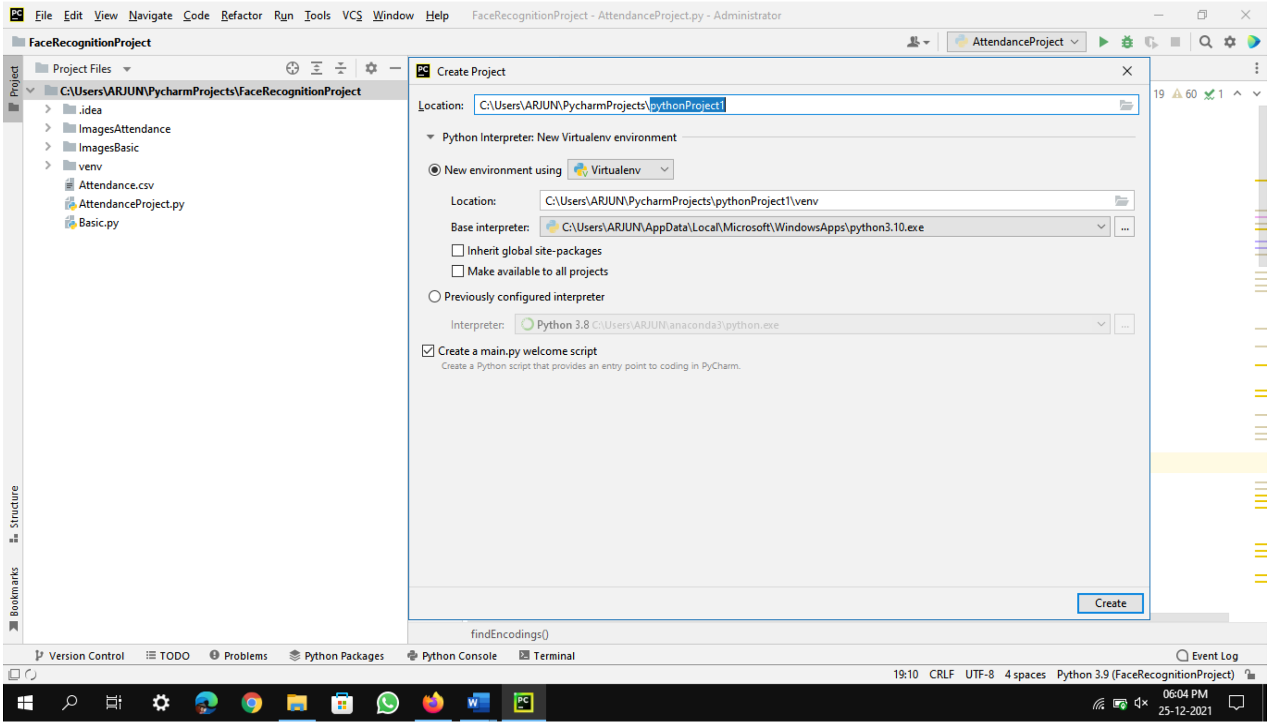

FIG 5.1.1 PYCHARM INTERFACE

Import the library functions such as numpy, cv2, face_recognition, os, and datetime functions.

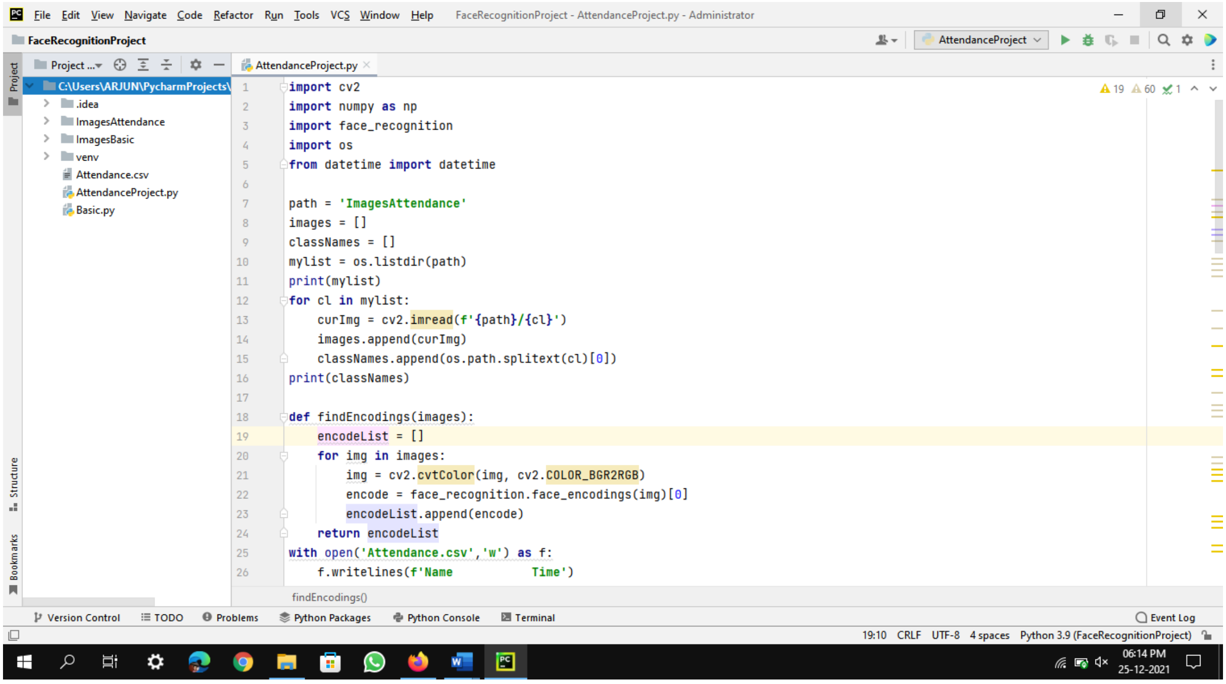

FIG 5.1.2 CODING WINDOW

5.2 OUTPUT:

The web-cam turns on and the face of the person is captured in real-time. A green colored rectangle appears over the face of person. The face of the person is tracked in live motion with the help of the eye movement.

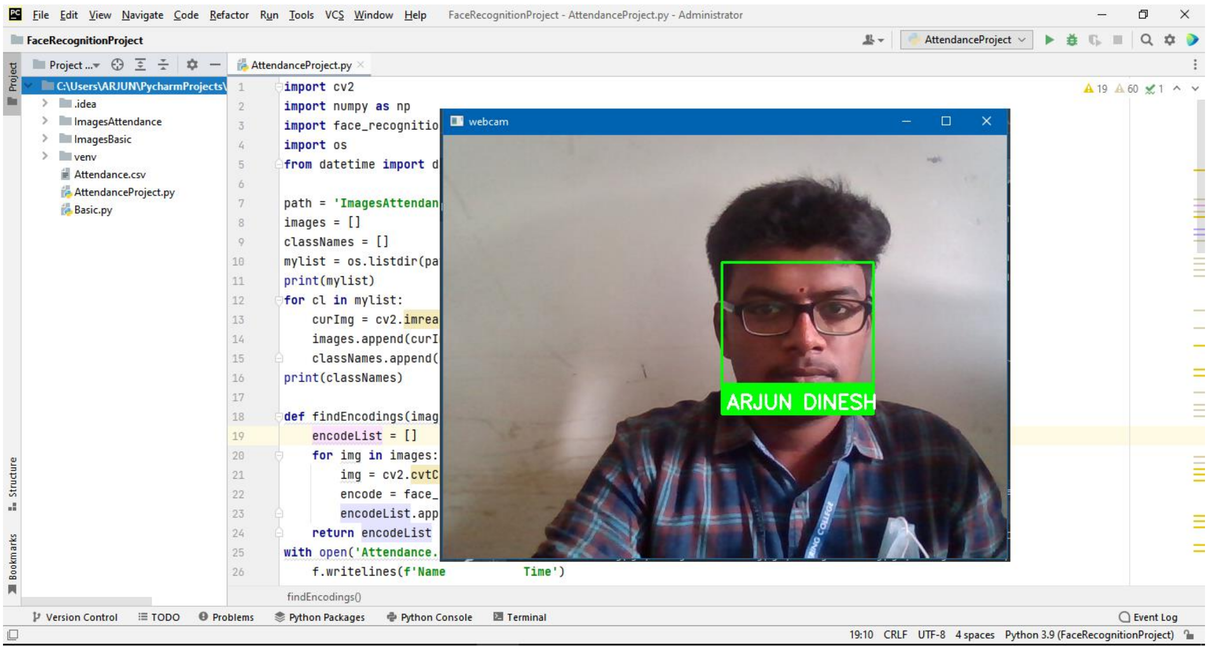

FIG 5.2.1 DETECTION OF KNOWN FACES

The proposed system can detect multiple person at a time.

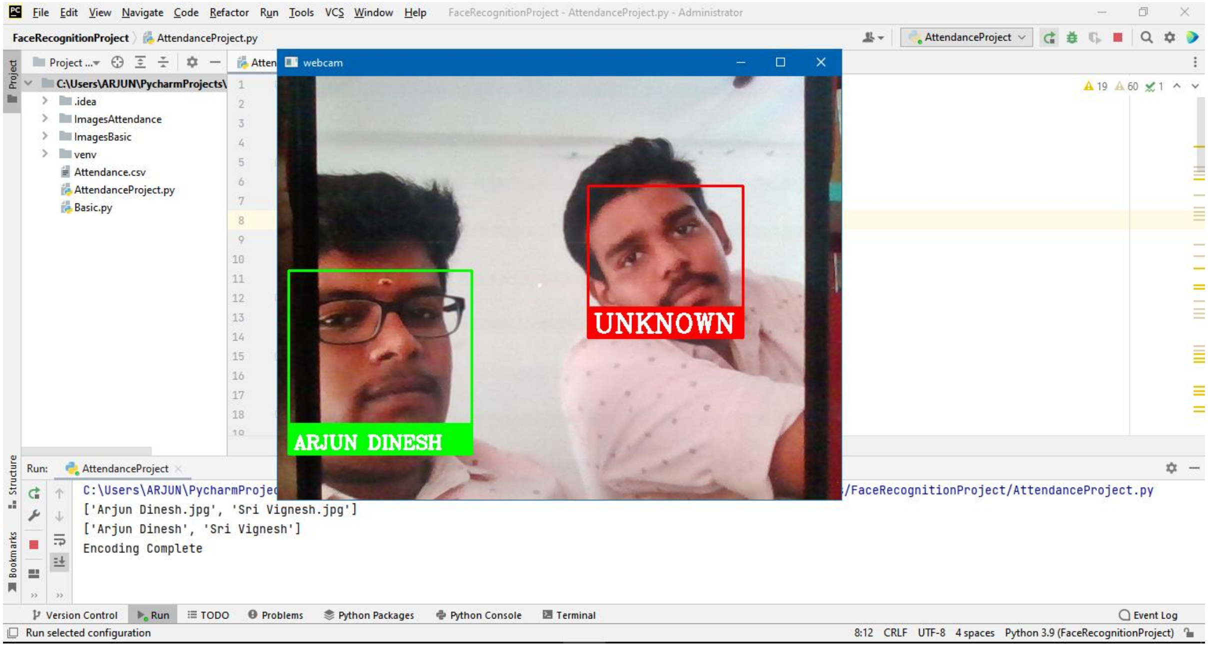

FIG 5.2.2. DETECTION OF KNOWN AND UNKNOWN FACES

The below image show how the attendance report is marked.

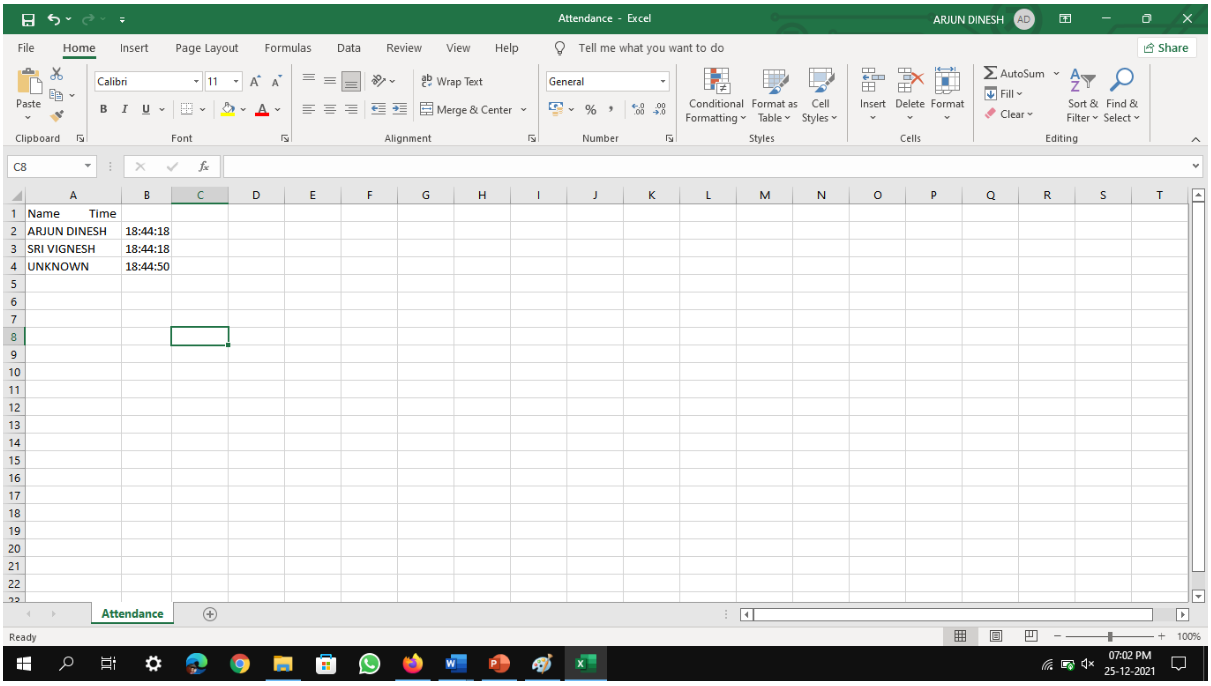

FIG 5.2.3 ATTENDANCE REPORT

5.3 PROBLEMS

The problem of face recognition can be stated as follows :

Face Recognition human facial features like the mouth, nose and eyes in a full frontal face image. We will be adapting a multi-step process in order to achieve the goal. To detect the face region we will be using Facial Landmark estimation method. Facial features can be located in the interior of the face contour. We will use several different facial-images to test our method

1. Trying to find a face within a large database of faces. In this approach the system returns a possible list of faces from the database. The most useful applications contain crowd surveillance, video content indexing, personal identification (example: driver's license), mug shots matching, etc.

2. Real time face recognition: Here, face recognition is used to identify a person on the spot and grant access to a building or a compound, thus avoiding security hassles. In this case the face is compared against a multiple training samples of a person.

CHAPTER 6

APPLICATIONS, MERITS AND DEMERITS

6.1 APPLICATIONS

S.NO	AREAS	TASK PERFORMED
1.	Security and Surveillance	• Airport • ATM Machines • Border Cross Point • Network Security
2.	Indexing of Videos	• Surveillance • Mug shot / Ticket Booking • Criminal Justice System
3.	Investigation Of Image database	• To find the Missing Children's • Witness face reconstruction • To Manage the Driving license
4.	Verification for identify	• Banking field • Electronic Commerce

TABLE 6.1 AREAS AND TASK PERFORMED

6.2 MERITS

Face recognition can be used to unlock devices such as Mobile phones, Laptop, etc., It can be used at airports to find unknown person. It can help us improve our overall safety levels. Facial recognition helps in identifying diseases. May help to identify fake passports. Can prevent many kinds of fraud.

6.3 DEMERITS

Facial recognition can be expensive. Sensitive data may get stolen by hackers. Storage of data can be problematic. Rules and regulation against facial recognition in some regions. General public may not accept this technology. Can be misused by governments and media companies.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION:

An automated face recognition system is made which gives accurate results. Human face recognition plays a very important role as part of modern surveillance and security applications. By applying the methods described in this paper, accurate algorithms and quality face recognition results can be obtained. Moreover, with the help of HOG models, one can achieve high-performance levels in recognizing human faces and analyzing facial features, even in scenes containing complex backgrounds.

7.2 FUTURE WORK:

This system can be deployed for verification and attendance tracking at various government offices and corporates. For access control verification and identification of authentic users it can also be installed in bank lockers and vaults. For identification of criminals the system can be used by police force also.

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.

7.3 SUGGESTED IMPROVEMENTS :

There are multiple things which can impact the overall performance of the automated attendance system such as distance of face being recognized from the camera and excessive images in different angles of the same person which might result in hanging of the system. An equal number of images taken from every student will definitely improve the overall performance.

100 % accuracy of this setup is not guaranteed as it might result in failure sometimes due to environmental, noise, internet and live web camera issues. It might not work with people such as twins, or siblings studying in the same class with similarity in features. Our computers does not contain GPU/CUDA cores in our system that is why we preferred HOG algorithms for better accuracy and results of our system.

7.4 REFERENCES

1. Sareen P. Biometrics—introduction, characteristics, basic technique, its types and various performance measures. Int J Emerg Res Manag Technol. 2014.3.
2. Bobak AK, Dowsett AJ, Bate S. Solving the border control problem: Evidence of enhanced face matching in individuals with extraordinary face recognition skills. PLoS One. 2016.11.

3. Research on face recognition based on deep learning – 2018 Sixth International Conference. IEEE.

4. Faces 2D-Recognition and Identification Using the HOG Descriptors Method - Vyacheslav Voronov, Vladimir Strelnikov, Liliya Voronova, Artyom Trunov, Andrey Vovi. - 15 April 2019.

5. A Deep facial recognition system using computational intelligent algorithms- Diaa Salama AbdElminaam, Abdulrhman M.Almansori, Mohamed Taha, Elsayed Badr. – December3, 2020.