AUTOMATED ATTENDANCE SYSTEM USING FACE RECOGNITION

A MINI PROJECT REPORT

Submitted by

KAMALESH S - 830119104014

MAGENDRAN P - 830119104302

SELVARAJ M - 830119104041

SRIDHAR V - 830119104046

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING-SRIRANGAM

TRICHY 620012.

MAY 2022

BONAFIDE CERTIFICATE

Certified that this project report AUTOMATED ATTENDANCE SYSTEM USING FACE RECOGNITION is the bonafide work of KAMALESH S, MAGENDRAN P, SELVARAJ M, SRIDHAR V who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SUPERVISOR **HEAD OF THE DEPARTMENT**

Submitted for Semester Mini-Project viva-voice examination held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

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. Present strategies for taking attendance are tedious and time -consuming. Attendance records can be easily manipulated by manual recording. The traditional process of making attendance and present biometric systems are vulnerable to proxies. This paper is therefore proposed to tackle all these problems. The proposed system makes the use of Haar classifiers, Local Binary pattern Histogram, Eigenfaces, Fisher faces. After face recognition attendance reports will be generated and stored in excel format. The system is tested under various conditions like illumination, head movements, the variation of distance between the student and cameras. After vigorous testing overall complexity and accuracy are calculated. The Proposed system proved to be an efficient and robust device for taking attendance in a classroom without any time consumption and manual work. The system developed is cost-efficient and need less installation.

/

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

Attendance being a very necessary side of administration may normally become an arduous, redundant activity, pushing itself to inaccuracies. The traditional approach of making roll calls proves itself to be a statute of limitations as it is very difficult to call names and maintain its record especially when the ratio of students is high. Every organization has its way of taking measures for the Attendance of students. Some organizations use document-oriented Approach and others have implemented these digital methods such as biometric fingerprinting techniques and card swapping techniques. however, these methods prove to be a statute of limitations as it subjects students to wait in a time-consuming queue. if the student fails to bring his id card then he will not be able to get attendance. evolving technologies have made many improvements in the changing world. The system of intelligent attendance is generally implemented with biometrics help. Recognition of face is one of the Biometric ways of improving this system. Face recognition proved to be a productive method for taking attendance. The normative face recognition techniques and methodologies fail to tackle challenges like scaling, pose, illumination, variations, rotation, and occlusions. The framework proposed is designed to solve the drawbacks of current systems. there has been a lot of advancement in face recognition but the vital steps are face detection, feature extraction, and face recognition. firstly, two or more cameras depend on the need, and the size of the classroom has to be installed on the ceiling of the classroom from where it covers the entire area. image captured from these cameras will be considered as an input to the system. There may be a possibility of getting image blurred due to movements of students, for better efficacy image can be upgraded using Generative Adversarial Networks. A newly generated ameliorated image will be passed to the system for face detection. process of face detection is accompanied by feature extraction and face recognition these process makes the use of Gabor filters. face recognition is done using the K-nearest neighbor algorithm, Convolutional neural networks, and SVM algorithm with their comparative studies. post-completion of face recognition, the system generates the name and identification number of the students who are present and identified in the image. then attendance is marked in front of the student names in the excel format with respective date and subject of a lecture in an institution. It requires very few hardware resources hence it is a cost-friendly system.

1.1 Problem statement

The aim of this project is to create a system that will automatically mark the attendance of students using face recognition system

1.2 Literature survey

The primary aim of this paper is to study the different approaches given by authors and to develop a real-time attendance system which overcomes the shortcomings of previous methods and to give the best solution. In Yohie Kawaguchi et.al proposed a system based on continuous observation and using face recognition. The author presented a system with an active student detecting method (ASD) having two cameras placed on the wall in which one is a sensing camera which is used for estimating seat inside the class and the other is capturing camera which is used for face detection. They have proposed a shooting plan in which one seat is estimated from the seating area obtained by ASD and then directs the capturing camera to the seat and captures an image. The existence of students is estimated using background subtraction and inters frame subtraction. The author has solved the linear sum assignment problem to give the correspondence of students and seats.

Introduced an automated system based on convolutional neural networks. The author has used the GSM module to send the generated attendance report to an authorized person. The author proposes the modified convolutional neural network by adding two normalization operations to two of the layers. This operation provides the batch normalization acceleration of the network. The face recognition system is designed using the SIFT algorithm. This system will take attendance using camera. The image will be captured and matched with the database and marked the attendance

CHAPTER 2

Requirements

2.1 Hardware requirements

RAM	: 4 GB Ram and more
Processor	: AMD Processor
Hard Disk	: 40 GB min
Camera	: 2 MP

2.2 Software requirements

Operating System	: Window10
IDE	: PyCharm
Front End Tool	: Python
Backend Tool	: MS Excel

CHAPTER 3

Architecture diagram

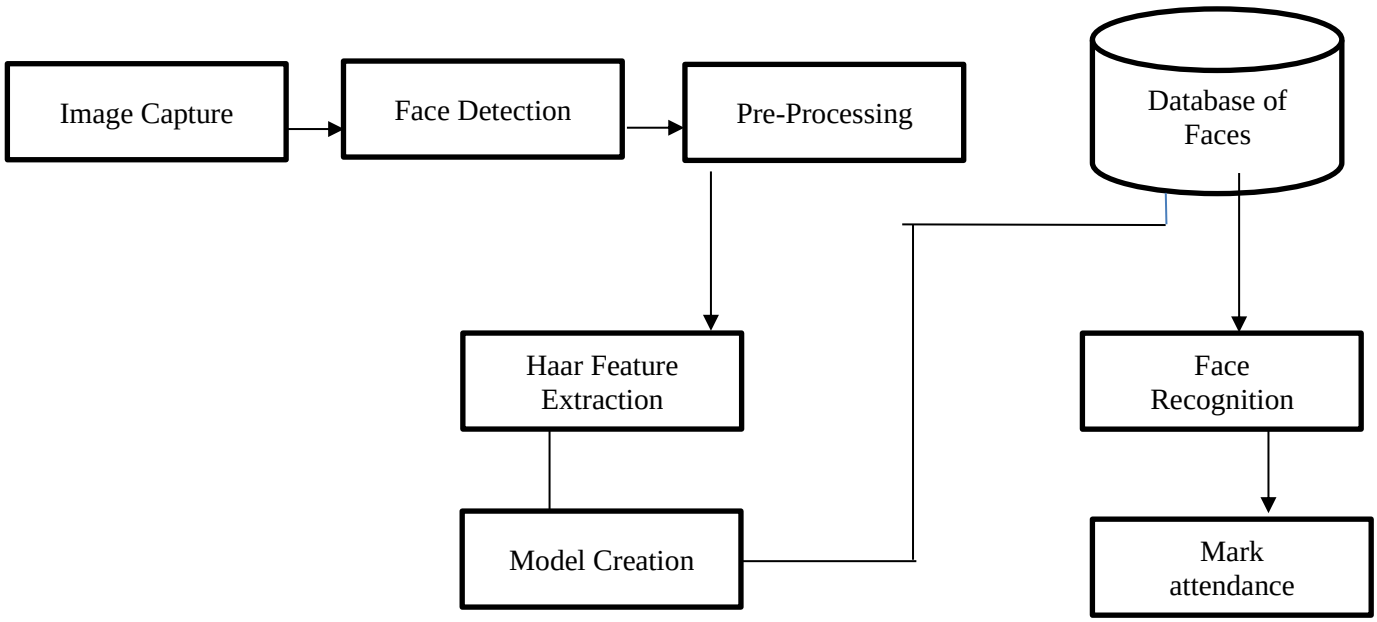


Fig 3.1

CHAPTER 4

4.1 IMAGE ACQUISITION BY WEB CAMERA

The camera capturing photographic or video type images. A camera module includes an image sensor, a lens, and an aperture and other additional components.

4.2 FACE DETECTION:

In the past few years, face recognition owned significant consideration and appreciated as one of the most promising applications in the field of image analysis. Face detection can consider a substantial part of face recognition operations. According to its strength to focus computational resources on the section of an image holding a face. The method of face detection in pictures is complicated because of variability present across human faces such as pose, expression, position and orientation, skin color, the presence of glasses or facial hair, differences in camera gain, lighting conditions, and image resolution.

Object detection is one of the computer technologies, which connected to the image processing and computer vision and it interacts with detecting instances of an object such as human faces, building, tree, car, etc. The primary aim of face detection algorithms is to determine whether there is any face in an image or not.

In recent times, a lot of study work proposed in the field of Face Recognition and Face Detection to make it more advanced and accurate, but it makes a revolution in this field when Viola-Jones comes with its Real-Time Face Detector, which is capable of detecting the faces in real-time with high accuracy.

Face Detection is the first and essential step for face recognition, and it is used to detect faces in the images. It is a part of object detection and can use in many areas such as security, bio-metrics, law enforcement, entertainment, personal safety, etc.

It is used to detect faces in real time for surveillance and tracking of person or objects. It is widely used in cameras to identify multiple appearances in the frame Ex- Mobile cameras and DSLR's. Facebook is also using face detection algorithm to detect faces in the images and recognize them.

4.3 Pre-Processing:

Once we get faces apply the pre-processing on images like noise removal, normalization etc. In this, it also converts the image into Gray Scale by taking the average of the each pixel RGB

Steps Involved in Data Preprocessing:

1. Data Cleaning:

The data can have many irrelevant and missing parts. To handle this part, data cleaning is done. It involves handling of missing data, noisy data etc.

- **(a). Missing Data:**

  This situation arises when some data is missing in the data. It can be handled in various ways.

  Some of them are:

  0. **Ignore the tuples:**

     This approach is suitable only when the dataset we have is quite large and multiple values are missing within a tuple.

  1. **Fill the Missing values:**

     There are various ways to do this task. You can choose to fill the missing values manually, by attribute mean or the most probable value.

- **(b). Noisy Data:**

  Noisy data is a meaningless data that can't be interpreted by machines. It can be generated due to faulty data collection, data entry errors etc. It can be handled in following ways:

  0. **Binning Method:**

     This method works on sorted data in order to smooth it. The whole data is divided into segments of equal size and then various methods are performed to complete the task. Each segmented is handled separately. One can replace all data in a segment by its mean or boundary values can be used to complete the

task.

1. **Regression:**
   Here data can be made smooth by fitting it to a regression function. The regression used may be linear (having one independent variable) or multiple (having multiple independent variables).

2. **Clustering:**
   This approach groups the similar data in a cluster. The outliers may be undetected or it will fall outside the clusters.

2. Data Transformation:

This step is taken in order to transform the data in appropriate forms suitable for mining process. This involves following ways:

1. **Normalization:**
   It is done in order to scale the data values in a specified range (-1.0 to 1.0 or 0.0 to 1.0)

2. **Attribute Selection:**
   In this strategy, new attributes are constructed from the given set of attributes to help the mining process.

3. **Discretization:**
   This is done to replace the raw values of numeric attribute by interval levels or conceptual levels.

4. **Concept Hierarchy Generation:**
   Here attributes are converted from lower level to higher level in hierarchy. For Example- The attribute “city” can be converted to “country”.

3. Data Reduction:

Since data mining is a technique that is used to handle huge amount of data. While working with huge volume of data, analysis became harder in such cases. In order to get rid of this, we uses data reduction technique. It aims to increase the storage efficiency and reduce data storage and analysis costs.

The various steps to data reduction are:

1. **Data Cube Aggregation:**
   Aggregation operation is applied to data for the construction of the data cube.

2. **Attribute Subset Selection:**
   The highly relevant attributes should be used, rest all can be discarded. For performing attribute selection, one can use level of significance and p- value of the attribute. The attribute having p-value greater than significance level can be discarded.

3. **Numerosity Reduction:**
   This enables to store the model of data instead of whole data, for example: Regression Models.

4. **Dimensionality Reduction:**
   This reduces the size of data by encoding mechanisms. It can be lossy or lossless. If after reconstruction from compressed data, original data can be retrieved, such reduction are called lossless reduction else it is called lossy reduction. The two effective methods of dimensionality reduction are: Wavelet transforms and PCA (Principal Component Analysis).

4.4 Face recognition:

It is a technology capable of matching a human face from digital image or video frame against a database of faces.

A facial recognition system is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.[1]

Development began on similar systems in the 1960s, beginning as a form of computer application. Since their inception, facial recognition systems have seen wider uses in recent times on smartphones and in other forms of technology, such as robotics. Because computerized facial recognition involves the measurement of a human's physiological characteristics, facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless

process.[2] Facial recognition systems have been deployed in advanced human–computer interaction, video surveillance and automatic indexing of images.[3]

Facial recognition systems are employed throughout the world today by governments and private companies. Their effectiveness varies, and some systems have previously been scrapped because of their ineffectiveness. The use of facial recognition systems has also raised controversy, with claims that the systems violate citizens' privacy, commonly make incorrect identifications, encourage gender norms and racial profiling, and do not protect important biometric data. The appearance of synthetic media such as deepfakes has also raised concerns about its security. These claims have led to the ban of facial recognition systems in several cities in the United States. As a result of growing societal concerns, Meta announced[7] that it plans to shut down Facebook facial recognition system, deleting the face scan data of more than one billion users.[8] This change will represent one of the largest shifts in facial recognition usage in the technology's history.

While humans can recognize faces without much effort, facial recognition is a challenging pattern recognition problem in computing. Facial recognition systems attempt to identify a human face, which is three-dimensional and changes in appearance with lighting and facial expression, based on its two-dimensional image. To accomplish this computational task, facial recognition systems perform four steps. First face detection is used to segment the face from the image background. In the second step the segmented face image is aligned to account for face pose, image size and photographic properties, such as illumination and grayscale. The purpose of the alignment process is to enable the accurate localization of facial features in the third step, the facial feature extraction. Features such as eyes, nose and mouth are pinpointed and measured in the image to represent the face. The so established feature vector of the face is then, in the fourth step, matched against a database of faces.

CHAPTER 5

ALOGRITHM

5.1 Eigenfaces:

Eigenfaces refers to an appearance-based approach to face recognition that seeks to capture the variation in a collection of face images and use this information to encode and compare images of individual faces in a holistic (as opposed to a parts-based or feature-based) manner. Specifically, the eigenfaces are the principal components of a distribution of faces, or equivalently, the eigenvectors of the covariance matrix of the set of face images, where an image with *N* pixels is considered a point (or vector) in *N*-dimensional space. The idea of using principal components to represent human faces was developed by Sirovich and Kirby (Sirovich and Kirby 1987) and used by Turk and Pentland (Turk and Pentland 1991) for face detection and recognition. The Eigenface approach is considered by many to be the first working facial recognition technology, and it served as the basis for one of the top commercial face recognition technology products. Since its initial development and publication, there have been many extensions to the original method and many new developments in automatic face recognition systems.

5.2 Local binary patterns

Local binary patterns (LBP) is a type of visual descriptor used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed in 1990. LBP was first described in 1994.[3][4] It has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. A comparison of several improvements of the original LBP in the field of background subtraction was made in 2015 by Silva et al. A full survey of the different versions of LBP can be found in Bouwmans et al.

The LBP feature vector, in its simplest form, is created in the following manner:

- Divide the examined window into cells (e.g. 16x16 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.

- Where the center pixel's value is greater than the neighbor's value, write "0". Otherwise, write "1". This gives an 8-digit binary number (which is usually converted to decimal for convenience).
- Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center). This histogram can be seen as a 256-dimensional feature vector.
- Optionally normalize the histogram.
- Concatenate (normalized) histograms of all cells. This gives a feature vector for the entire window.

The feature vector can now be processed using the Support vector machine, extreme learning machines, or some other machine learning algorithm to classify images. Such classifiers can be used for face recognition or texture analysis.

A useful extension to the original operator is the so-called uniform pattern,[8] which can be used to reduce the length of the feature vector and implement a simple rotation invariant descriptor. This idea is motivated by the fact that some binary patterns occur more commonly in texture images than others. A local binary pattern is called uniform if the binary pattern contains at most two 0-1 or 1-0 transitions. For example, 00010000 (2 transitions) is a uniform pattern, but 01010100 (6 transitions) is not. In the computation of the LBP histogram, the histogram has a separate bin for every uniform pattern, and all non-uniform patterns are assigned to a single bin. Using uniform patterns, the length of the feature vector for a single cell reduces from 256 to 59. The 58 uniform binary patterns correspond to the integers 0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254 and 255.

5.3 Fisher faces

A key problem in computer vision, pattern recognition and machine learning is to define an appropriate data representation for the task at hand.

One way to represent the input data is by finding a subspace which represents most of the data variance. This can be obtained with the use of Principal Components Analysis (PCA). When applied to face images, PCA yields a set of eigenfaces. These eigenfaces are the eigenvectors associated to the largest eigenvalues of the covariance matrix of the training data. The eigenvectors thus found correspond to the least-squares (LS) solution. This is indeed a powerful way to represent the data because it ensures the

data variance is maintained while eliminating unnecessary existing correlations among the original features (dimensions) in the sample vectors.

When the goal is classification rather than representation, the LS solution may not yield the most desirable results. In such cases, one wishes to find a subspace that maps the sample vectors of the same class in a single spot of the feature representation and those of different classes as far apart from each other as possible. The techniques derived to achieve this goal are known as discriminant analysis (DA).

The most known DA is Linear Discriminant Analysis (LDA), which can be derived from an idea suggested by R.A. Fisher in 1936. When LDA is used to find the subspace representation of a set of face images, the resulting basis vectors defining that space are known as Fisher faces.

5.4 HAAR-CASCADE CLASSIFIERS

Face Detection, a widely popular subject with a huge range of applications. Modern day Smartphones and Laptops come with in-built face detection software, which can authenticate the identity of the user. There are numerous apps that can capture, detect and process a face in real time, can identify the age and the gender of the user, and also can apply some really cool filters. The list is not limited to these mobile apps, as Face Detection also has a wide range of applications in Surveillance, Security and Biometrics as well. But the origin of its Success stories dates back to 2001, when Viola and Jones proposed the first ever Object Detection Framework for Real Time Face Detection in Video Footage.

This article is about taking a gentle look on the Viola-Jones Face Detection Technique, popularly known as Haar Cascades, and exploring some of the interesting concepts proposed by them. This piece of work was done long before the Deep Learning Era had even started. But it’s an excellent work in comparison to the powerful models that can be built with the modern-day Deep Learning Techniques. The algorithm is still found to be used almost everywhere. It has fully trained models available on GitHub. It’s fast. It’s pretty accurate (at least when I try it).

According to Wikipedia… Woody Bledshoe, Helen Chan Wolf, and Charles Bisson were the first ones to do the first ever Face Detection on a Computer back in the 1960s. A person had to manually pinpoint the coordinates of facial features such as the pupil centers, the inside and outside corner of eyes, and the widows peak in the hairline. The coordinates were used to calculate 20 distances, including the width of the mouth and of the eyes. A human could process about 40 pictures an hour in this manner and

so build a database of the computed distances. A computer would then automatically compare the distances for each photograph, calculate the difference between the distances and return the closed records as a possible match.

So, what is Haar Cascade? It is an Object Detection Algorithm used to identify faces in an image or a real time video. The algorithm uses edge or line detection features proposed by Viola and Jones in their research paper “Rapid Object Detection using a Boosted Cascade of Simple Features” published in 2001. The algorithm is given a lot of positive images consisting of faces, and a lot of negative images not consisting of any face to train on them. The model created from this training is available at the OpenCV GitHub repository https://github.com/opencv/opencv/tree/master/data/haarcascades.

The repository has the models stored in XML files, and can be read with the OpenCV methods. These include models for face detection, eye detection, upper body and lower body detection, license plate detection etc. Below we see some of the concepts proposed by Viola and Jones in their research.

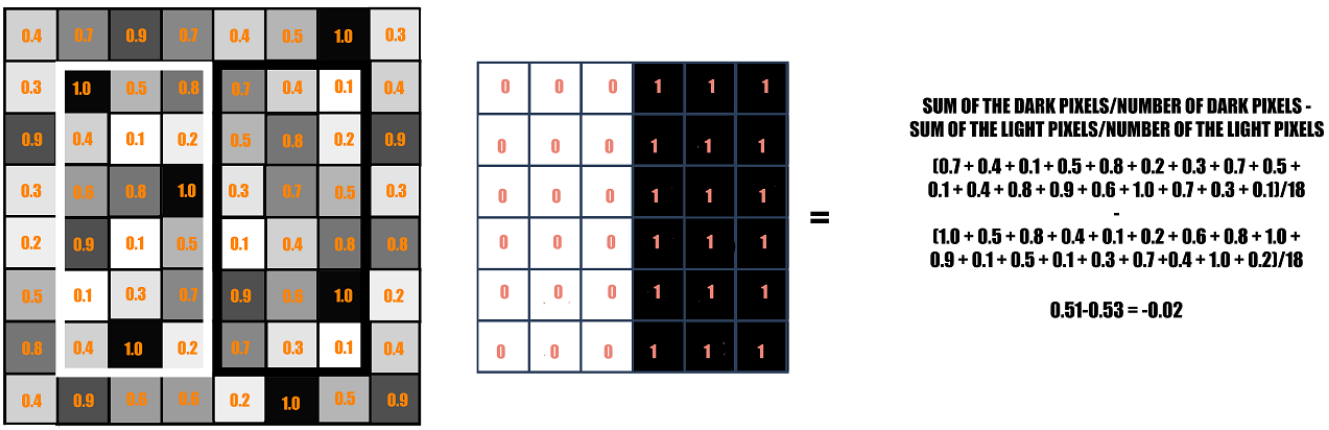


Fig 5.1

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

5.6 FACIAL FEATURES

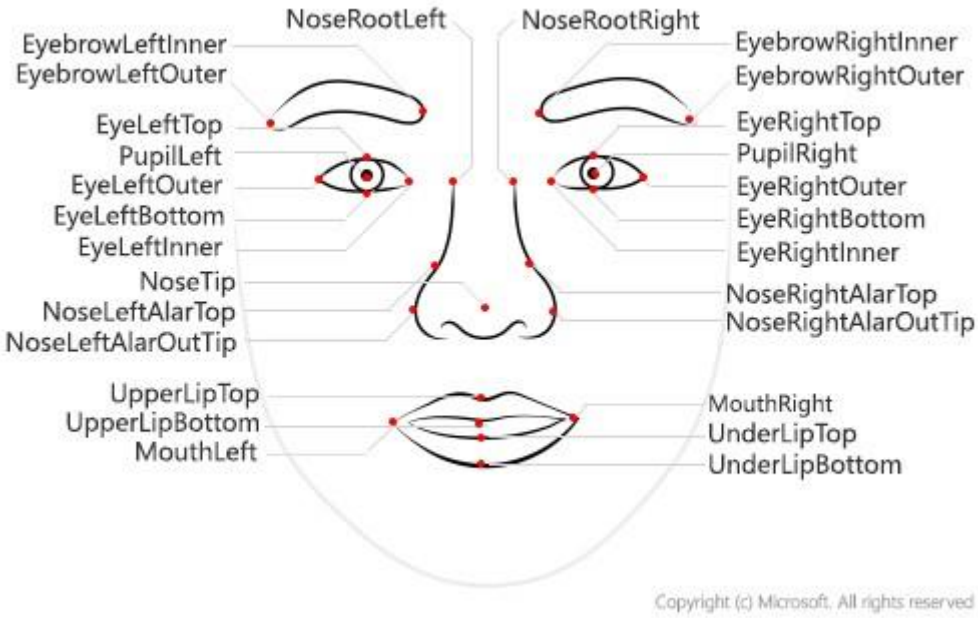


Fig 5.2

CHAPTER 6

PYTHON LIBRARIES

6.1 OPEN CV PACKAGE

OpenCV is a programming library/package that has been created especially for allowing programmers to enter the world of Computer Vision. The primary developer of the OpenCV package is Intel Corporation, and the package was released to the public during the year 1999-2000.

OpenCV stands for Open-Source Computer Vision (Library). It is the most commonly used, popular, and well-documented Computer Vision library. It is open-source, which means that one does not require a license to utilize the software.

As one may know, most Machine Learning Algorithms require inputs to be quantitative in nature, i.e., numerical. OpenCV allows us to apply Machine Learning techniques to images, however, oftentimes we are required to preprocess and prepare the raw images for them to be transformed into features (columns of data) that are useful and usable by our Machine Learning Algorithms.

6.2 NUMPY

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Num array was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Num array into Numeric package. There are many contributors to this open-source project.

6.3 DLIB

Dlib is a general purpose cross-platform software library written in the programming language C++. Its design is heavily influenced by ideas from design by contract and component-based software engineering. Thus, it is, first and foremost, a set of independent software components. It is open-source software released under a Boost Software License.

Since development began in 2002, Dlib has grown to include a wide variety of tools. As of 2016, it contains software components for dealing with networking, threads, graphical user interfaces, data structures, linear algebra, machine learning, image processing, data mining, XML and text parsing, numerical optimization, Bayesian networks, and many other tasks. In recent years, much of the development has been focused on creating a broad set of statistical machine learning tools and in 2009 Dlib was published in the Journal of Machine Learning Research.[2] Since then it has been used in a wide range of domains

6.4 FACE RECOGNITION

After extensively researching the best software for face recognition, I came to the conclusion that almost all the articles currently published are just copied and pasted from advertisements. Even worse, most of these articles recommend outdated libraries and services that are not supported anymore. Some of their suggested solutions can't even run on modern operating systems!

I promise, this is not one of those articles. I've done my best to make a comprehensive list of all the modern face recognition solutions on the market. Surprisingly, half of the promising free solutions only started their active development in 2020! I found it somewhat difficult to adequately study paid solutions for enterprises as they don't give much information for ordinary people, so I imagine this article will be most useful for small and medium-sized companies, and of course DIY enthusiasts.

Types of Facial Recognition Solutions

The first thing you should know is that there's a huge variety of facial recognition solutions. Some of them are ready to use without machine learning skills, and others need much more time and expertise.I would split facial recognition services into three types, each with its own advantages and disadvantages. Let's take a look:

Software as a Service (SaaS)-based facial recognition engines. In this case, a face recognition service provider handles everything from keeping up with machine learning technology to managing and supporting high-load servers. All you have to do is to integrate the software with your IT systems via an API. Despite their many advantages, these solutions also have lots of downsides. First of all, this is the most expensive option, as everything is handled by the provider. You also need a stable Internet

connection, as you will need to send heavy images to a server somewhere on the Internet. There could also be security issues since you have to send your photos to a third-party company and can't control what they do with them.

1. **Self-hosted REST API solutions.** Such systems can be deployed both on premises and in the cloud. They don't have the same problems as SaaS products. You store data on your own servers (or in your private cloud), so you can control where it goes, and you can even create a system that works offline. Of course, in exchange, you will have to manage the servers on your own, too. But in most cases, servers are delivered as docker containers, so it's quite easy to orchestrate them. Self-hosted solutions, while not as expensive as their SaaS counterparts, are still pretty pricey. Fortunately, free and open-source self-hosted REST API solutions are beginning to appear. They are not as mature as other solutions, but they are very promising.

2. **Open-source frameworks and libraries.** These are typically free, as many researchers are happy to publish the code from their state-of-the-art approaches. Of course, you will need at least some experience with machine learning to use this kind of software. You will also need to spend some time to restify it if you want to integrate these solutions with your custom apps. The upside is that you will get a state-of-the-art solution that you know inside and out.

As you can see, it's important to know your resources. Face recognition used to be very expensive for ordinary people. There are some services that can start at $86.40 per day or go up to $30,000 per year per camera for facial recognition from streaming video (see prices for individual solutions below). This is probably why we only hear about face recognition when the government or big companies make use of it.

But this is less of an issue than it used to be, as there is an increasing number of free options. They are at various stages of maturity, but they're not just for niche enthusiasts; they're ready for use at small and mid-sized businesses.

Before we get down to comparing the best facial recognition software, I want to clarify that I've chosen accuracy as a key parameter for my research. There are plenty of benchmarks that solutions can use to demonstrate their high standards. Just as there are new face recognition solutions coming to the market

every year, the performance benchmarks are constantly changing too. So, it's often difficult to compare even two-year-old solutions with new ones. However, there is one very old, but still popular benchmark: Labeled Faces in the Wild (LFW). Fortunately, all of the solutions on our list shared results from this benchmark, so I was able to compare their accuracy.

CHAPTER 7

DATA FLOW DIAGRAM

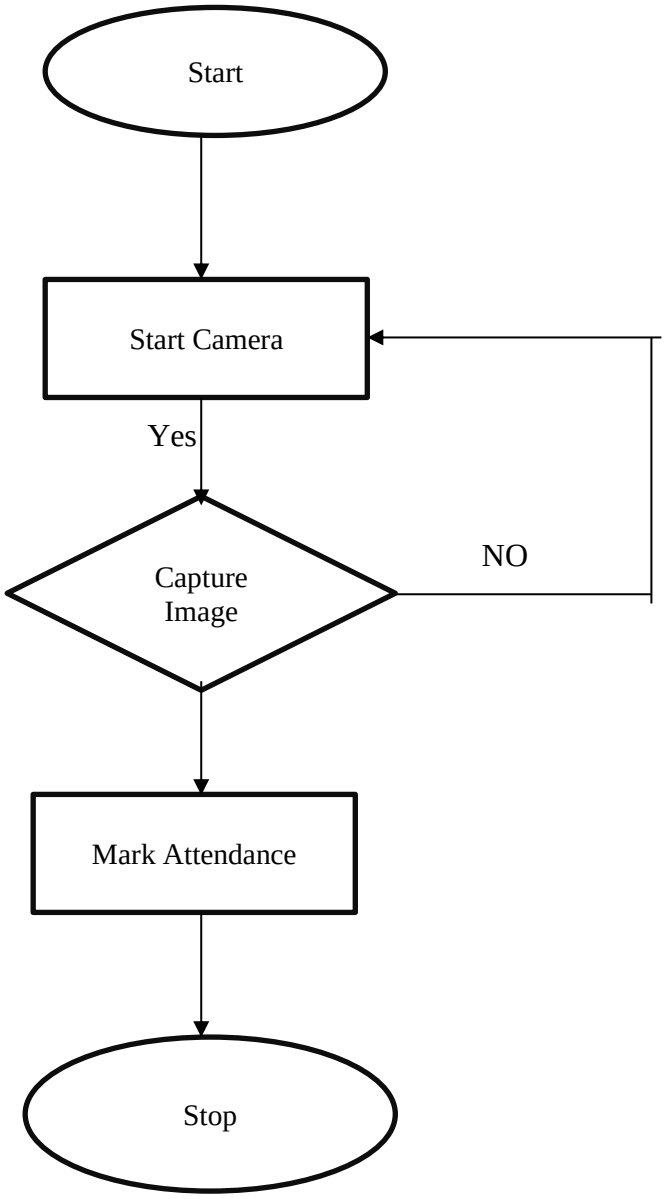


Fig 7.1

CHAPTER 8

SOURCE CODE

```
#Developed by
#P. Magendran
#S. Kamalesh
#M. Selvaraj
#v. Sridhar

#importing packages
import cv2
import numpy as np
import face_recognition
import os
from datetime import datetime



path = 'images'
images = []
personNames = []
myList = os.listdir(path)
print(myList)



for cu_img in myList:
    current_Img = cv2.imread(f'{path}/{cu_img}')
    images.append(current_Img)
    personNames.append(os.path.splitext(cu_img)[0])
print(personNames)



def faceEncodings(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 attendance(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:
            time_now = datetime.now()
            tStr = time_now.strftime('%H:%M:%S')
            dStr = time_now.strftime('%d/%m/%Y')
            f.writelines(f'\n{name},{tStr},{dStr}')


encodeListKnown = faceEncodings(images)
print('All Encodings Complete!!!')

cap = cv2.VideoCapture(1)

while True:
    ret, frame = cap.read()
    faces = cv2.resize(frame, (0, 0), None, 0.25, 0.25)
    faces = cv2.cvtColor(faces, cv2.COLOR_BGR2RGB)
```

```
    facesCurrentFrame = face_recognition.face_locations(faces)
    encodesCurrentFrame = face_recognition.face_encodings(faces, facesCurrentFrame)

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


        if matches[matchIndex]:
            name = personNames[matchIndex].upper()
            # print(name)
            y1, x2, y2, x1 = faceLoc
            y1, x2, y2, x1 = y1 * 4, x2 * 4, y2 * 4, x1 * 4
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)
            cv2.rectangle(frame, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)
            cv2.putText(frame, name, (x1 + 6, y2 - 6), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 2)
            attendance(name)


cv2.imshow('Webcam', frame)
    if cv2.waitKey(1) == 13:
        break


cap.release()
cv2.destroyAllWindows()
```

CHAPTER 9

OUTPUT

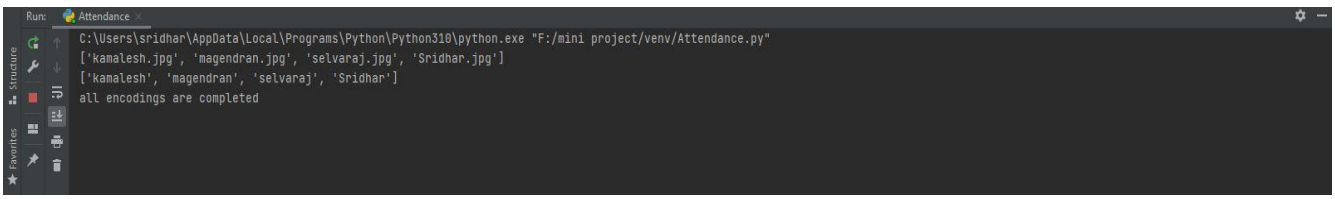


Fig 9.1

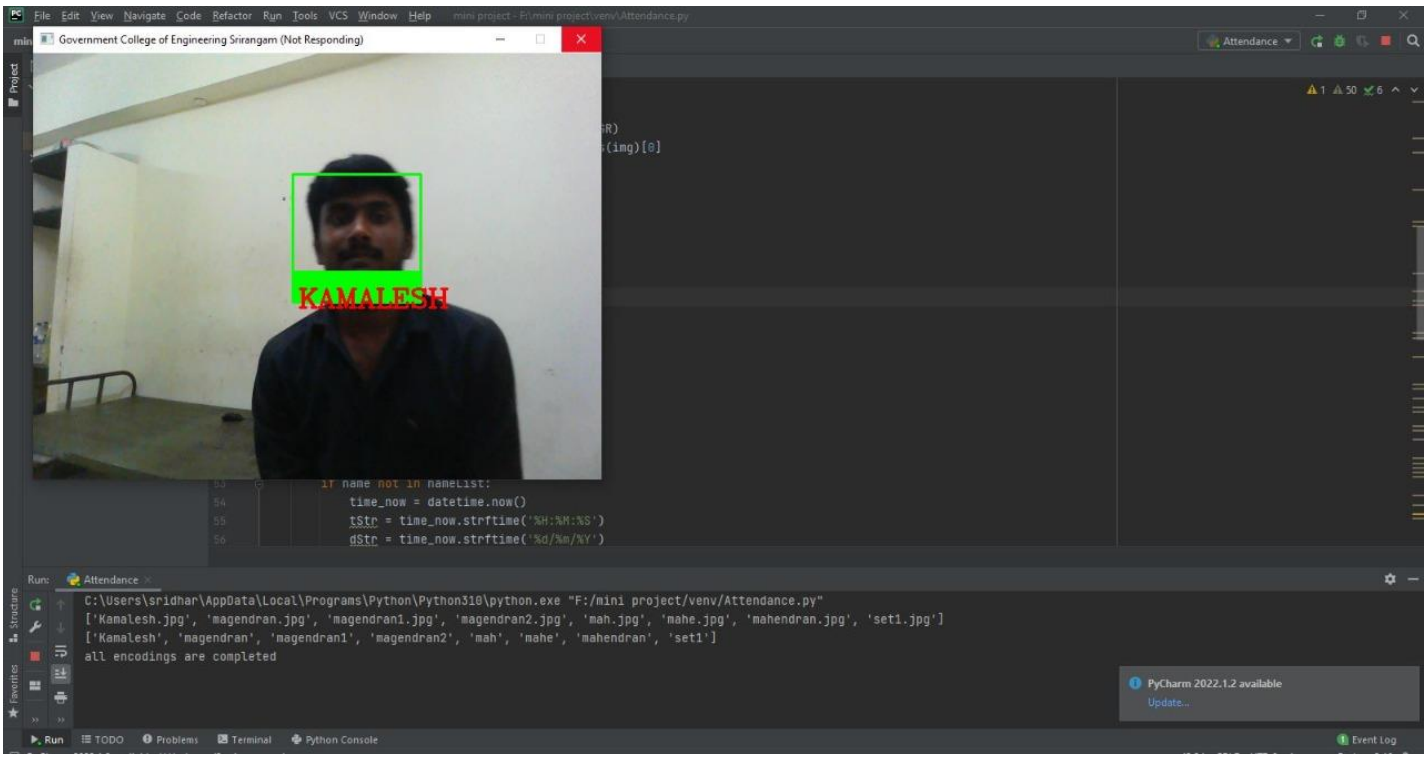

Government College of Engineering Srirangam (Not Responding)
KAMALESH
if name not in nameList:
time_now = datetime.now()
tStr = time_now.strftime('%H:%M:%S')
dStr = time_now.strftime('%d/%m/%Y')
Run: Attendance
C:\Users\sridhar\AppData\Local\Programs\Python\Python310\python.exe "F:/mini project/venv/Attendance.py"
['Kamalesh.jpg', 'magendran.jpg', 'magendran1.jpg', 'magendran2.jpg', 'mah.jpg', 'mahe.jpg', 'mahendran.jpg', 'set1.jpg']
['Kamalesh', 'magendran', 'magendran1', 'magendran2', 'mah', 'mahe', 'mahendran', 'set1']
all encodings are completed
PyCharm 2022.1.2 available
Update...
Run
TODO
Problems
Terminal
Python Console
Event Log

Fig 9.2

Fig9.3

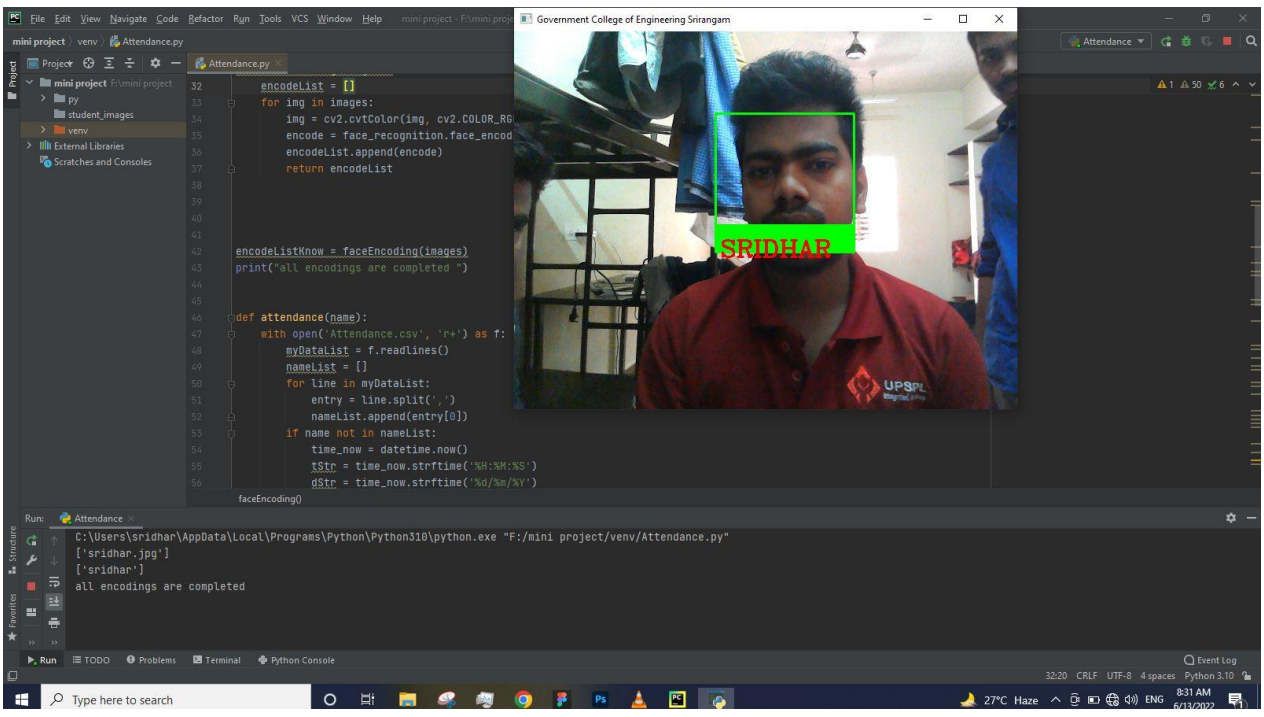

Government College of Engineering Srirangam
SRIDHAR
encodeListKnow = faceEncoding(images)
print("all encodings are completed ")
def attendance(name):
with open('Attendance.csv', 'r+') as f:
myDataList = f.readlines()
nameList = []
for line in myDataList:
entry = line.split(',')
nameList.append(entry[0])
if name not in nameList:
time_now = datetime.now()
tStr = time_now.strftime('%H:%M:%S')
dStr = time_now.strftime('%d/%m/%Y')
C:\Users\sridhar\AppData\Local\Programs\Python\Python310\python.exe "F:/mini project/venv/Attendance.py"
['sridhar.jpg']
['sridhar']
all encodings are completed

Fig 9.4

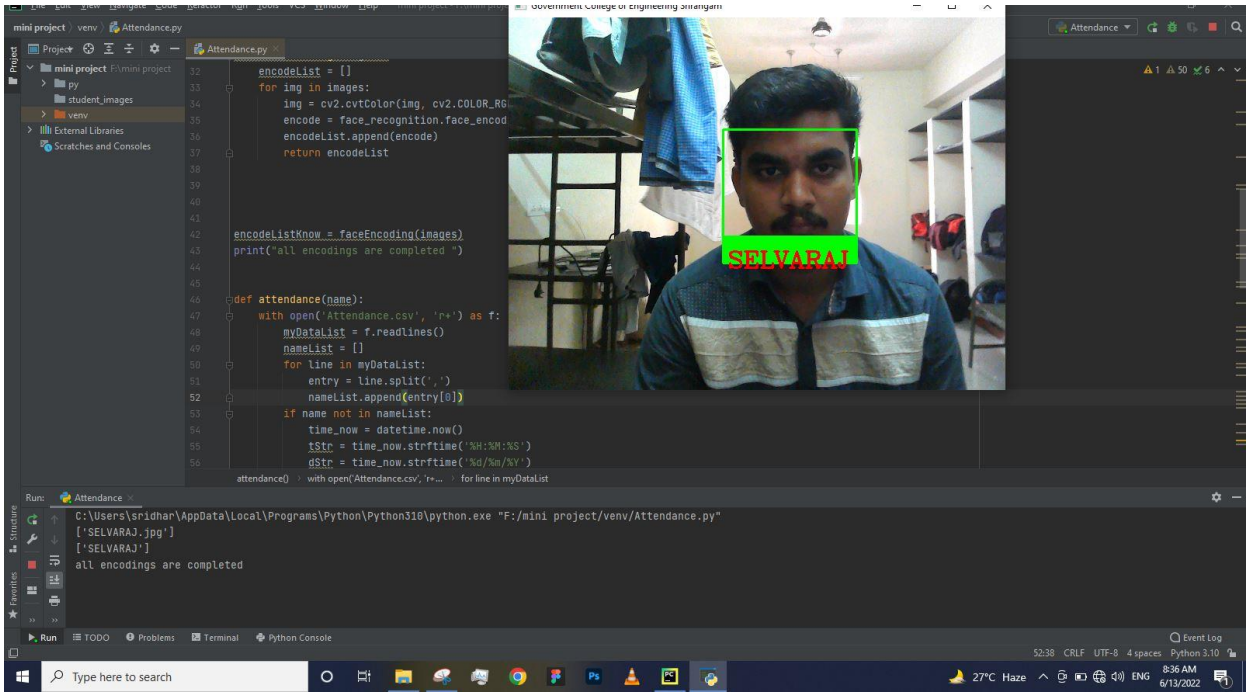
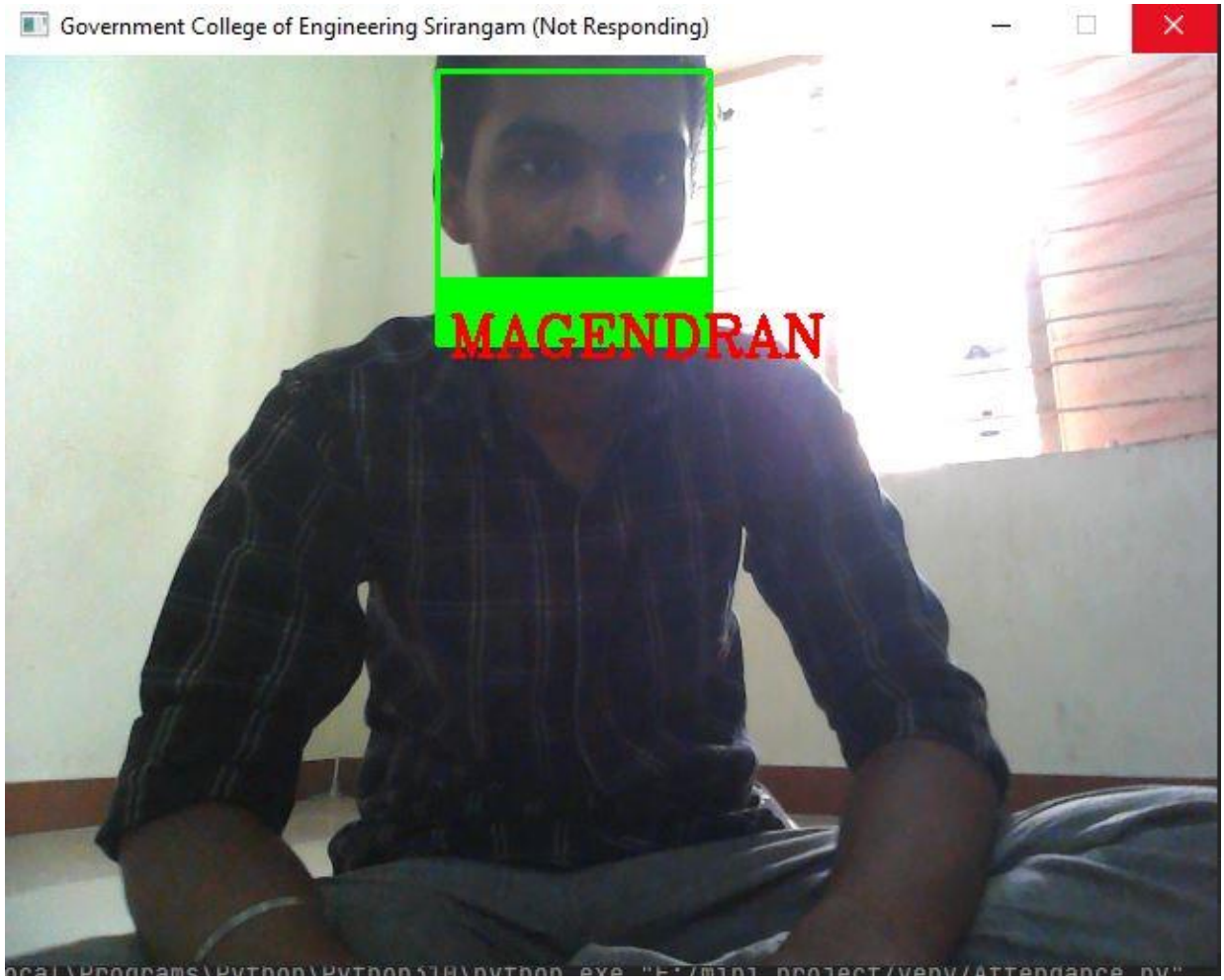
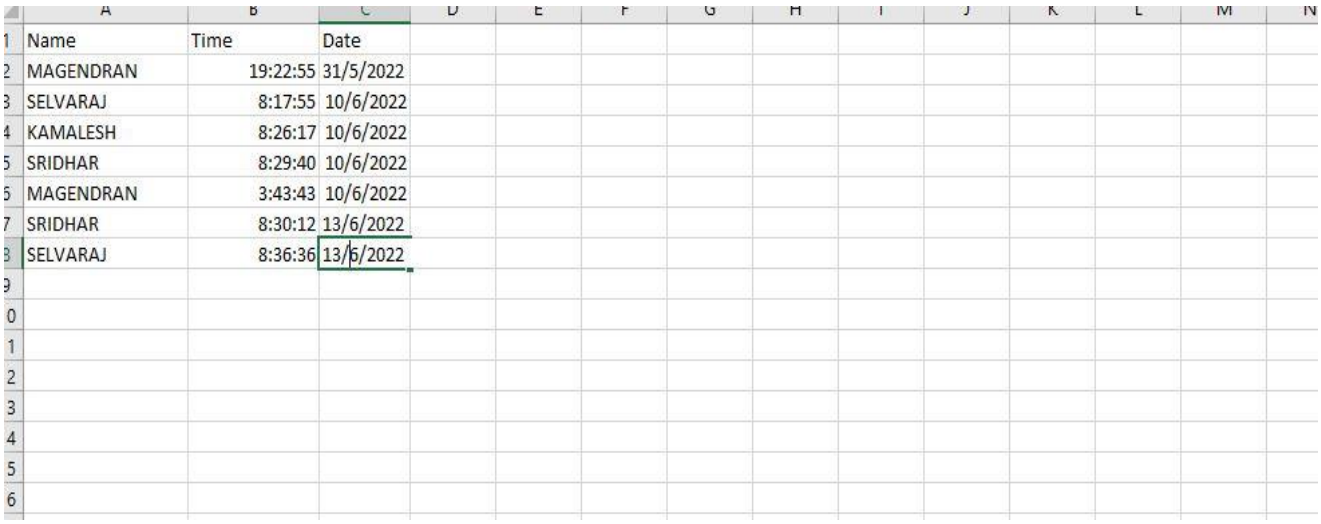

encodeList = []
for img in images:
img = cv2.cvtColor(img, cv2.COLOR_RG
encode = face_recognition.face_encod
encodeList.append(encode)
return encodeList
encodeListKnow = faceEncoding(images)
print("all encodings are completed ")
def attendance(name):
with open('Attendance.csv', 'r+') as f:
myDataList = f.readlines()
nameList = []
for line in myDataList:
entry = line.split(',')
nameList.append(entry[0])
if name not in nameList:
time_now = datetime.now()
tStr = time_now.strftime('%H:%M:%S')
dStr = time_now.strftime('%d/%m/%Y')
SELVARAJ
C:\Users\sridhar\AppData\Local\Programs\Python\Python310\python.exe "F:/mini project/venv/Attendance.py"
['SELVARAJ.jpg']
['SELVARAJ']
all encodings are completed

Fig 9.5


Government College of Engineering Srirangam (Not Responding)
MAGENDRAN



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Name	Time	Date											
2	MAGENDRAN	19:22:55	31/5/2022											
3	SELVARAJ	8:17:55	10/6/2022											
4	KAMALESH	8:26:17	10/6/2022											
5	SRIDHAR	8:29:40	10/6/2022											
6	MAGENDRAN	3:43:43	10/6/2022											
7	SRIDHAR	8:30:12	13/6/2022											
8	SELVARAJ	8:36:36	13/6/2022											
9														
10														
11														
12														
13														
14														
15														
16														

Fig 9.6

CHAPTER 10

RESULT

Figures shows the results of the automatic attendance system based on face recognition. This system is based on viola jones and principal component analysis (P.C.A).

By clicking on Mark attendance, the module will capture the image of group of students and detect their faces as shown in fig by green boxes. By clicking on Mark attendance, the module will automatically generate the attendance sheet and mark the attendance automatically for every student.

CHAPTER 11

CONCLUSION

With the help of a divergent combination of algorithms, this system helps us to achieve desired results with better accuracy and less time consumption. The database of subjects for distinct classes is cached in the backend of the system. The system is commenced at start of lecture. The subject name is fetched from the backend of the system for every hour and the attendance for the same is marked.

CHAPTER 12

LIST OF REFERENCES

NirmalyaKar, MrinalKantiDebbarma, AshimSaha, and DwijenRudra Pal," Study of Implementing Automated Attendance System Using Face
Recognition Technique",in proc. International Journal of Computer and Communication Engineering, Vol. 1, No. 2, July 2012

[2] AjinkyaPatil and MrudangShukla," Implementation of Classroom Attendance System Based on Face Recognition in Class", in proc.
International Journal of Advances in Engineering & Technology, July, 2014.

[3] Jomon Joseph and K. P. Zacharia, "Automatic Attendance Management System Using Face Recognition", in proc. International Journal of
Science and Research (IJSR).

[4] RohitChavan,Baburaophad,SankalpSawant and VinayakFutak,"Attendance Management System using face recognition",in International
journal for innovative Research in science and technology,Vol. 1,issue 11,April 2015.

[5] Naveed Khan Balcoh, M. HaroonYousaf,Waqar Ahmad and M. IraamBalg,"Algorithm for efficient attendance management:face recognition
based approach",in IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012.

[6] MrunmayeeShirodkar,VarunSinha,Urvi Jain and BhushanNemade,"Automated Attendance Management System using Face recognition",in
International Conference.

[1] Yohei Kawaguchi, Tetsuo Shoji, "Face Recognition-based Lecture
Attendance System", "3rd AERU...", 2005.

[2] B. Kavinmathi, S.Hemalatha, "Attendance System for Face Recognition
using GSM module", 4th International Conference on Signal Processing and
Integrated Networks", 2018.

[3] Ketan N. Mahajan, Nagaraj V. Dharwadkar," Classroom attendance system
using surveillance camera", International Conference on Computer Systems,
Electronics and Control",2017.

[4] Shubhobrata Bhattacharya, Gowtham Sandeep Nainala, Prosenjit Das, Aurobinda Routray “Smart Attendance Monitoring System (SAMS): A Face Recognition based Attendance System for Classroom Environment”, IEEE 18th International Conference on Advanced Learning Technologies, 2018.
[5] E.Varadharajan, R.Dharani, S.Jeevitha,., “Automatic attendance management system using face detection”, 2017.

[6] Guo, Jing-Ming, “Complexity reduced face detection using probability-based face mask prefiltering and pixel-based hierarchical-feature Ada

boosting”, Signal Processing Letters, IEEE 2011.
.
[7] K.Senthamil Selvi1, P.Chitrakala, A.Antony, Jenitha S, “face recognition based attendance marking system”, International Journal of Computer Science and Mobile Computing, 2014.
[8] Chen, Joy Iong Zong. "Smart Security System for Suspicious Activity Detection in Volatile Areas." Journal of Information Technology 2, 2020.
[9] Jacob, I. Jeena. "Capsule network based biometric recognition system." Journal of Artificial Intelligence 1,2019.
[10] Kirtiraj Kadam, Manasi Jadhav, Shivam Mulay, Tushar Indalkar, “Attendance Monitoring System Using Image Processing and Machine Learning”, International Journal of Advance Engineering and Research Development, 2017.
[11] Rajat Kumar Chauhan, Vivekanand Pandey, Lokanath M, “Smart Attendance System Using CNN”, International Journal of Pure and Applied Mathematics,2018.
[12] Mayank Yadav, Anmol Aggarwal, “Motion based attendance system in real time environment for multimedia application”, 2018.
[13] Wei Wu, Chuanchang Liu, Zhiyuan Su, “Novel Real-time Face Recognition from Video Streams”, International Conference on Computer Systems, Electronics and Control, 2017.
[14] Changxing Ding, Dacheng Tao, “Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition”, IEEE transactions on pattern analysis and machine intelligence, 2018.

[15] Aziza Ahmedi, Dr Suvarna Nandyal, “An Automatic Attendance System Using Image processing”

1. Agrawal, S., Khatri, P.: Facial expression detection techniques: based on Viola and Jones algorithm and principal component analysis. In: 2015 Fifth International Conference on Advanced Computing & Communication Technologies, pp. 108–112. IEEE (2015)

2. Ahmedi, A., Nandyal, S.: An automatic attendance system using image processing. Int. J. Eng. Sci. (IJES) 4(11), 1–8 (2015)

3. Bodhe, V.M., Bhakre, S.M., Ikhar, S.D.: Student attendance system by face detection. Int. J. Innov. Res. Comput. Commun. Eng. 5(3), 3958 (2017)

Chintalapati, S., Raghunadh, M.: Automated attendance management system based on face recognition algorithms. In: 2013 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–5. IEEE (2013)

5. Deshpande, N.T., Ravishankar, S.: Face detection and recognition using Viola-Jones algorithm and fusion of PCA and ANN. Adv. Comput. Sci. Technol. 10(5), 1173–1189 (2017)

6. D’Silva, K., Shanbhag, S., Chaudhari, A., Patil, M.P.: Spot me-a smart attendance system based on face recognition. Int. Res. J. Eng. Technol. (IRJET) 6(3), 4239 (2019)

7. Elias, S.J., Hatim, S.M., Hassan, N.A., Abd Latif, L.M., Ahmad, R.B., Darus, M.Y., Shahuddin, A.Z.: Face recognition attendance system using local binary pattern (LBP). Bull. Electr. Eng. Inf. 8(1), 239–245 (2019)

8. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Trans. Pattern Anal. Mach. Intell. 28(4), 594–611 (2006)

9. Jee, H., Lee, K., Pan, S.: Eye and face detection using SVM. In: Proceedings of the 2004 Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004, pp. 577–580. IEEE (2004)

10. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Networks 8(1), 98–113 (1997)

11. Li, S.Z., Zhang, Z.: FloatBoost learning and statistical face detection. IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1112–1123 (2004)

12. Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N.: Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. Pattern Recogn. Lett. 26(2), 181–191 (2005)

13. Shrivastava, K., Manda, S., Chavan, P., Patil, T., Sawant-Patil, S.: Conceptual model for proficient automated attendance system based on face recognition and gender classification using Haar-Cascade, LBPH algorithm along with LDA model. Int. J. Appl. Eng. Res. 13(10), 8075–8080 (2018)
14. Shriwastav, S., Jain, D.C.: A review on face recognition attendance system. Int. J. Comput. Appl. 143(8), 19–22 (2016)
15. Stelea, G.A., Gavril˘a, C., Zamfir, S., Curpen, R.: Face recognition for education in the cloud. eLearning Softw. Educ. 2, 181–188 (2017