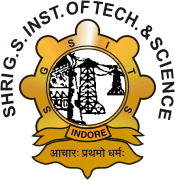
STUDENT ATTENDANCE AND INFORMATION MANAGEMENT SYSTEM



DEPARTMENT OF ELECTRICAL ENGINEERING

SESSION 2022-2023

##### A Dissertation submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal (M.P.)

Towards the Partial Fulfillment of Requirements for Awarding the Degree

of

##### Bachelor of Engineering (B.E.) in Electrical Engineering

Submitted To :- SUBMITTED BY :-

Dr. Arun Parakh Sir 0801EE191031 – Isha Das

Ms. Dipti Rai Ma’am 080EE191043- Parag Saxena 0801EE191093- Tinkle Rajawat 0801EE191071 – Sakshi Tomar 0801EE191057 – Raghvendra Raghuwanshi

Department of Electrical Engineering Shri G.S. Institute of Technology and Science,

Indore-452003 (M.P.) -India

©Shri G.S. Institute of Technology and Science, (S.G.S.I.T.S.), Indore, 2022-2023



2022-2023

# RECOMMENDATION

We are pleased to recommend that the dissertation work entitled “Student Attendance and Information Management System” carried out by Raghvendra Raghuwanshi, Isha Das , Parag Saxena , Tinkle Rajawat , Sakshi Tomar in partial fulfillment for degree of Bachelor of Engineering (B.E.) in Electrical Engineering of Shri G.S. Institute of Technology and Science , Indore during the year 2022-2023 . The project report has been approved as it satisfies the academic requirement in respect of project work prescribed for the Bachelor of Engineering degree.

Supervisor:

Dr. Arun Parakh Sir

Department of Electrical Engineering,

Shri G.S. Institute of Technology and Science, Indore

Co-Supervisor:

Ms. Dipti Rai Ma’am

Department of Electrical Engineering,

Shri G.S. Institute of Technology and Science, Indore



2022-2023

# CERTIFICATE

We are pleased to certify that the dissertation work entitled “Student Attendance and Information Management System” carried out by Raghvendra Raghuwanshi , Isha Das , Parag Saxena , Tinkle Rajawat , Sakshi Tomar is accepted in partial fulfillment for the award of the degree of Bachelor of Engineering in Electrical Engineering of Shri G.S. Institute of Technology and Science , Indore during the year 2021 – 2022 .

Internal Examiner External Examiner

Date: Date:



2022-2023

# DECLARATION

We Raghvendra Raghuwanshi , Isha Das , Parag Saxena , Tinkle Rajawat , Sakshi Tomar student of Bachelor of Engineering (B.E.) in Electrical Engineering, hereby declare that we have worked on project with title “ Student Attendance and Information Management System ” here under the supervision of Dr. Arun Parakh Sir , Head of Department , Department of Electrical Engineering and Co-Supervisor Ms. Dipti Rai Ma’am , Department of Electrical Engineering , Shri G.S. Institute of Technology and Science, Indore and all the material used in the report are properly referenced and all references are taken into account to the best of my belief.

|  |  |  |
| --- | --- | --- |
| Isha Das | Parag Saxena | Tinkle Rajawat |
| 0801EE191031 | 0801EE191043 | 0801EE191093 |

Sakshi Tomar Raghvendra Raghuwanshi 0801EE191071 0801EE191057

*This thesis is dedicated to .. ..*

Family*,* Teachers *&*

Friends



2022-2023

# ACKNOWLEDGEMENT

First of all, we would like to express my gratitude towards my supervisor Dr. Arun Parakh, Asst. Prof., Department of Electrical Engineering, S.G.S.ITS, Indore. We take pride in saying that we have successfully completed our project work under his able guidance. He is a major support to us throughout our project being available all the time with his ideas, inspiration and encouragement. It is through his competent guidance that we have been able to complete this project.

Furthermore, we would also like to acknowledge with much appreciation the crucial role of Ms. Dipti Rai ma’am, Department of Electrical Engineering, S.G.S.ITS, Indore who helped us in every step and guided us from the completion of our project.

Then, we would like to express our sincere thanks to Dr. H. K. Verma, Prof & Head, Department of Electrical Engineering, SGSITS, for his co-operation and providing all required facilities to complete this course.

We owe this moment of satisfaction, with a deep sense of gratitude to Dr. Rakesh Saxena. Director, and the administration of SGSITS for helping us to complete the project work relatively easier and better.

Then we would also like to thank all the faculty members of Electrical Engineering Department SGSITS, Indore for their suggestions and constructive denigration that led us to improve our project to its perfection:

Lastly, we would like to acknowledge the support of our friends and family who have helped us throughout for the completion of project and for their unconditional support.



2022-2023

# ABSTRACT

The face is the identity of a person. Attendance marking in a classroom during a lecture is not only tough task but also time consuming due to the large number of students in the classes. The presently used attendance system has been an area of challenge and takes too much time for the student’s responses

i.e. attendance through pen paper and calling each student’s name then marking the attendance and it may works for small no. of students. But this method is not efficient when the strength of students is higher. This work comprises of an automatic real time face detection system attendance system and can be used for automatic attendance system at different places like schools, institutes, industries, etc. In recent years, the problem of automatic attendance marking has been widely addressed through the use of standard biometrics like fingerprint and RFID tags etc., However, these techniques lack the element of reliability. In this project an automated attendance marking and management system is proposed by making use of face detection and recognition algorithms. The working of this system is divided into some main steps, at first, the capture the images in the classroom In the next step, the system processes these images and detect faces of the students using some algorithm and face detection will be implemented automatically to generate a list of detected student faces. In the last step, these images are compared with the stored images of students. and finally updating of attendance in database is processed. The main objective of this work is to make the attendance marking and management system efficient, time saving, simple and easy. Compared to existing system traditional attendance marking system, this system reduces the workload of people and also saves the time.



2022-2023

# CONTENTS

1. Introduction
2. Related work and existing system
3. Requirement analysis
4. Cases to be handled
5. System design 5.1.Hardware development 5.2.Sorfware development
6. Algorithms
7. Attendance system framework
   1. Primary database creation and training 7.2Image capturing

7.3Image detection 7.4Face recognition 7.5Image matching 7.6Proposed architecture

1. Creation of database
2. Higher level single block diagram 10. Methodology

11.Impotant code used 12.Experiment and results 13.Impact and significance

14.Future scope and project direction 15.Conclusions

1. Bibliography



2022-2023

# INTRODUCTION

Attendance is a necessary parameter which is required in most of the schools, college’s, institutes, etc. for evaluating the performance of students.. On an average this attendance is carried out to have accurate count of people or students seating in a particular classroom or any practice area. Traditional method of taking this attendance carried out by humans where the lecturer or teacher manually counts each and every student with their required data like candidate’s name, serial number, status etc. This process is very much time consuming and maintenance of collected data is difficult. Now a days, smart class rooms are becoming popular. The main objective of having smart classrooms is to save the precious time of teacher as well as student. Thus, we can digitize the process of taking attendance and make it simple and accurate. Our project will digitize the process of attendance collection by using image processing technique and will automatically update real time data of faculty’s data base via IOT or some wireless technique. This work consists of many stages: face detection; feature extraction; and face recognition, attendance marking. Feature extraction involves obtaining relevant facial features from the data. These features could be certain face regions, variations, angles, or measures. Since this project is software based. we have few hardware components. Initially cameras are used in the classroom with different angles so that we can scan each and every student’s face without any interruption. Camera only provides captured real time image as an input to computer node 1 which is acting as a master processor. Using face detection technique, faces are extracted and processed using standard algorithms which is reliable, secure and fast. Once faces of all the students are stored in database then our program compares it with existing database using a suitable algorithm Attendance of all the students which are present in the class room are now marked in a spreadsheet. As the whole algorithm is processed on

modern computers hence it is faster than any other previous method and eliminates the manual work involved in attendance.

This project uses python software which aid us to make our system realizable. The input device we are using is web cam but it can be replaced by CCTV cameras which captures images at regular time intervals. This work uses different algorithms for face detection and face matching to make attendance system. There are many issues with the practical implementation of this system and some of most challenging tasks is to detect the faces with required accuracy. This system has following advantages like reduced paperwork and replace the traditional methods used in today’s world, easy to operate, proper monitoring of the student’s presence and their information management.

Department of Electrical Engineering Shri G.S. Institute of Technology and Science,

Indore-452003 (M.P.) –India



2022-2023

# RELATED WORK

There are a number of existing systems closely related to the proposed idea of marking attendance in a class by making use of facial recognition techniques and algorithms. To analyze these systems a literature survey of the proposed systems was done. The proposed case study was designed around certain relevant sources related to facial recognition and image processing. A descriptive framework was designed using the other design approaches.

Several researches have been done in this area. To make automated attendance system there are many technologies which are available today and some of them are as follows

* Facial recognition
  + Voice recognition
  + Signature recognition
  + Retinal scanning
  + Iris recognition

All these methods are for object detection and identification of human faces.

Many systems are being proposed for attendance management. One of the systems generates a smart attendance system which uses Quick Response (QR) code to track & record the attendance. Students and professors are given a unique QR code, at the beginning of the course, they are required to scan their QR code using a QR reading device. Attendance of students whose QR code is scanned will be recorded. This system is responsive to mobile phones and different computer systems.

A reliable attendance monitoring system based on biometric is developed, which is used to monitor the presence of students in a more effective way. It reduces the chances of marking proxy attendance and also reduces the problems like missing papers of attendance, which occur during marking attendance manually. Teachers have a small fingerprint scanner with them and students will press their finger on it to mark their attendance. Attendance management systems using Iris recognition [3], are more reliable and accurate because of the inner characteristics of iris like uniqueness, time invariance, immovability etc. The Iris pattern of each student is used for attendance. By using the camera live images of student iris are captured and stored in a database. Gray coding algorithm is used for measuring radius of iris and then that radius is matched with the radius of each student in the database and attendance of that student will be marked.

# EXISTING SYSTEM

Traditional attendance marking techniques i.e. pen and paper or signing attendance sheets are easy to bypass and trick as giving proxies or false signatures is a common practice among students nowadays, students take an unfair advantage of this at most times. But a facial recognition system is unassailable and cannot be fooled as each person has a set of unique and individual features common to that person and cannot be replicated or changed, it all comes down to one simple truth that is, unless you are physically present in the lecture your attendance will not get marked.

### Existing system

Pen and paper RFID tags Biometric

### Limitations

False signatures and proxies Can be used by anybody

Is a costlier approach

Table shows Existing systems and their limitations

Now the attendance systems that we have gone through in the literature survey have one aspect crucial to attendance marking that they do not take into account and that is what if a student’s attendance is not marked, because his face is not recognized or processed, in such a case the unrecognized image will be stored in a secondary database and an alert will be generated for the admin who then can manually add the attendance for said students.



2022-2023

# REQUIREMENT ANALYSIS

* To ensure the speed of the attendance recording process is faster than the previous system which can go as fast as approximately 3 second for each student.
* Have enough memory space to store the database.
* Able to recognize the face of an individual accurately based on the face database.
* Develop a database for the attendance management system.
* Provide a user-friendly interface for admins to access the attendance database
* Able to show an indication to the user whether the face- recognition process is successful or not.



2022-2023

# CASES TO BE HANDLED

* + The main problem to be handled is to maintain transparency along with not acquainting complete procedure. The time at which attendance is to be taken or the camera has to capture the images should not be known to the student.
  + To maintain accuracy, images with different angles has to be taken to record different profiles and to ensure complete coverage of number of students.
  + To increase efficiency by keeping a standby manual supervision arrangement.
  + To link the real time information created by the camera to a database which can compare and store attendance simultaneously and separately.
  + The face recognition system creates several challenges for any system such as the similarity between different faces, and the large number of angles from which a face can be viewed in common situations.
  + This system has disadvantages like detecting large number of faces.
  + The camera plays a crucial role in the working of the system hence the image quality and performance of the camera must be tested thoroughly before actual implementation.
  + Sometimes the poor lighting condition of the classroom may affect image quality which affects the system’s performance.



2022-2023

# SYSTEM DESIGN

The design part of the attendance monitoring system is divided into two sections which consist of the hardware and the software part. Before the software. The design part can be developed, the hardware part is first completed to provide a platform for the software to work. Before the software part we need to install some libraries for effective working of the application.

## HARDWARE DEVELOPMENT

* + - Camera Module with good mega pixels.
    - Power Supply Cable
    - 16Gb Micro SD Card

# SOFTWARE DEVELOPMENT

There are two major system flows in the software development section as shown below:

* + The creation of the face database
  + The process of attendance taking

Both processes mentioned above are essential because they made up the backbone of the attendance management system .



2022-2023

# ALGORITHMS

This section describes the software algorithm for the system. The algorithm consists of the following steps:

* + Creation of database
  + Video acquisition
  + Frame generation
  + Pre processing
  + Face detection
  + Face recognition
  + Attendance



2022-2023

# ATTENDANCE SYSTEM FRAMEWORK

This section deals primarily with proposed techniques, methodologies and concepts relevant to facial recognition and image processing which is more specific and to a single process which uses facial recognition algorithms image processing techniques. The proposed project includes four sequential phases; namely capture ,detection, image matching and attendance marking.

# Primary database creation and training

The original database containing the images of the students is created by taking a live real time video of the students, and splitting the video into thirty frames, converting them to gray scale and storing only the faces of the students as images, then we will be training the respective images using the LBPH algorithm all the while storing their respective histogram value’s and then comparing the stored and trained images against the captured images to mark the attendance. The software used for splitting the video into frames is Open-CV.

Images of students are captured using a web cam. Multiple images of single student will be acquired with varied gestures and angles. These images undergo pre-processing. The images are cropped to obtain the Region of Interest (ROI) which will be further used in recognition process. Next step is to resize the cropped images to particular pixel position. Then these images will be converted from RGB to gray scale images. And then these images will be saved as the names of respective student in a folder.

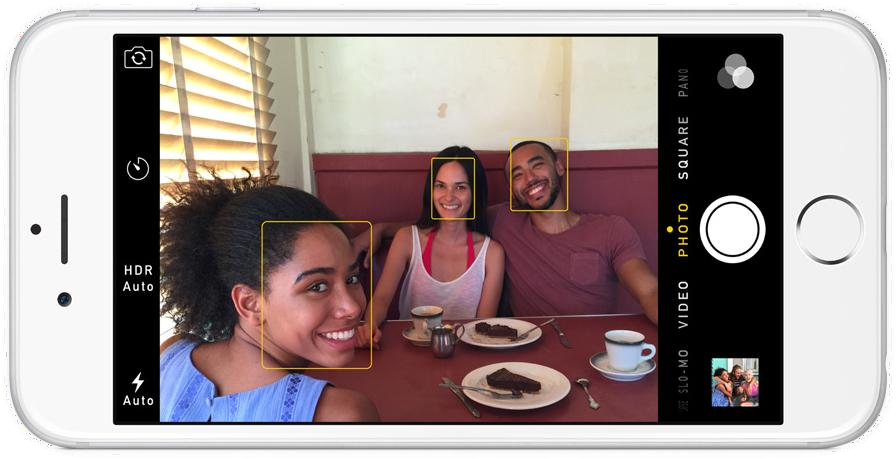
# Image Capturing

In this section the image will be captured a of each student present in the classroom. Then the captured images will be saved in a separate folder with the respective student’s name and enrolment numbers.

# Face Detection

In this this section once the video has begun capturing, simultaneously the algorithm is applied to the video to get individual faces of the students and obtaining the distinct features of their face (eyes, nose, ear and lips) by making use of line features and edge features. Once the faces are detected they are extracted and stored. This is required to create a rectangle around the faces in an image.

If you’ve used any camera in the last 10 years, you’ve probably seen face detection in action:

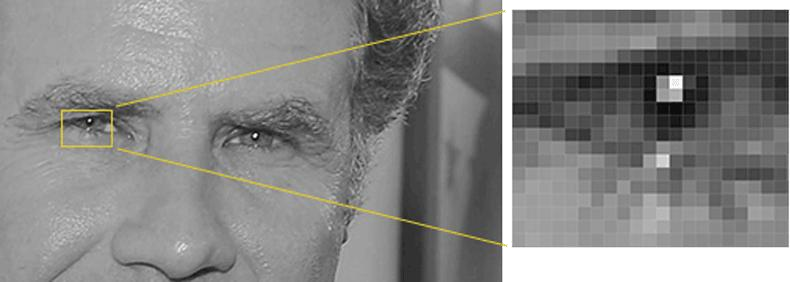


Face detection is a great feature for cameras. When the camera can automatically pick out faces, it can make sure that all the faces are in focus before it takes the picture. But we’ll use it for a different purpose — finding the areas of the image we want to pass on to the next step in our pipeline.

Face detection went mainstream in the early 2000's when Paul Viola and Michael Jones invented a [way to detect faces](https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework) that was fast enough to run on cheap cameras. However, much more reliable solutions exist now. We’re going to use [a method invented in 2005](http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf) called Histogram of Oriented Gradients — or just ***HOG*** for short.

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



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

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



Looking at just this one pixel and the pixels touching it, the image is getting darker towards the upper right.

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!

But saving the gradient for every single pixel gives us way too much detail. We end up [missing the forest for the trees](https://en.wiktionary.org/wiki/see_the_forest_for_the_trees). It would be better if we could just see the basic flow of lightness/darkness at a higher level so we could see the basic pattern of the image.

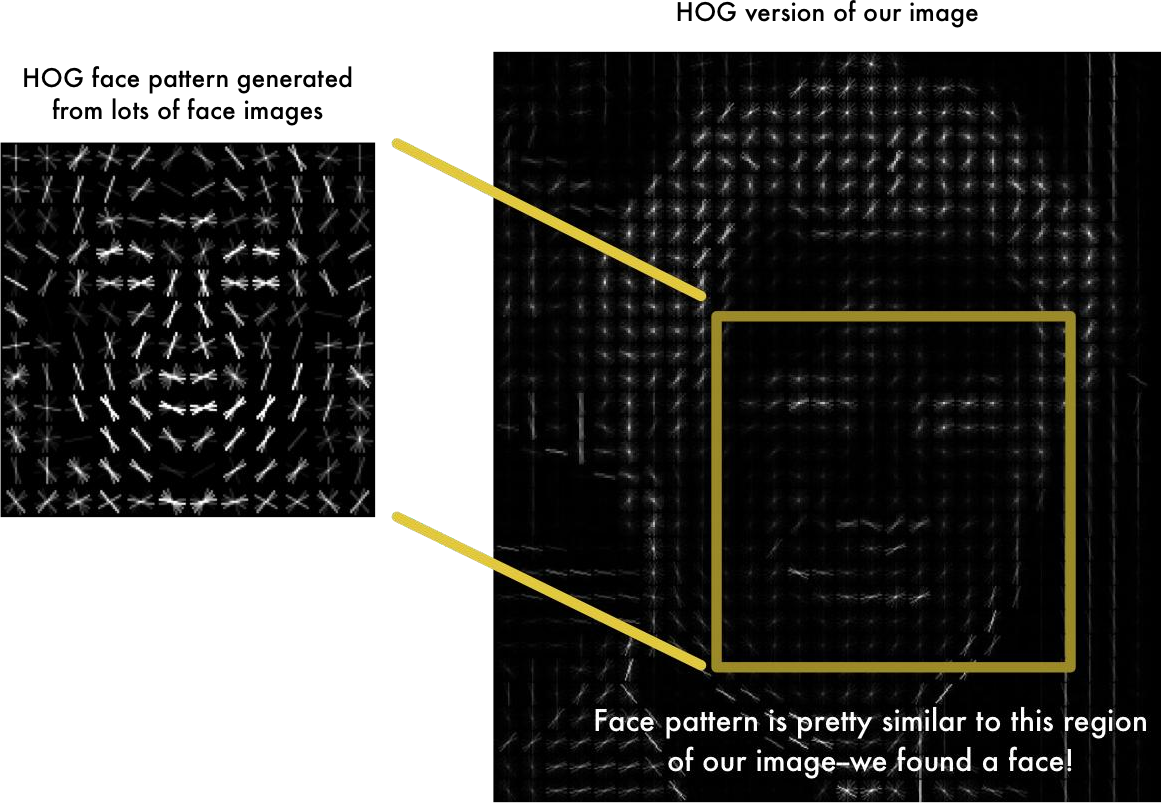
To do this, we’ll break up the image into small squares of 16x16 pixels each. In each square, we’ll count up how many gradients point in each major direction (how many point up, point up-right, point right, etc…). Then we’ll replace that square in the image with the arrow directions that were the strongest.

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:

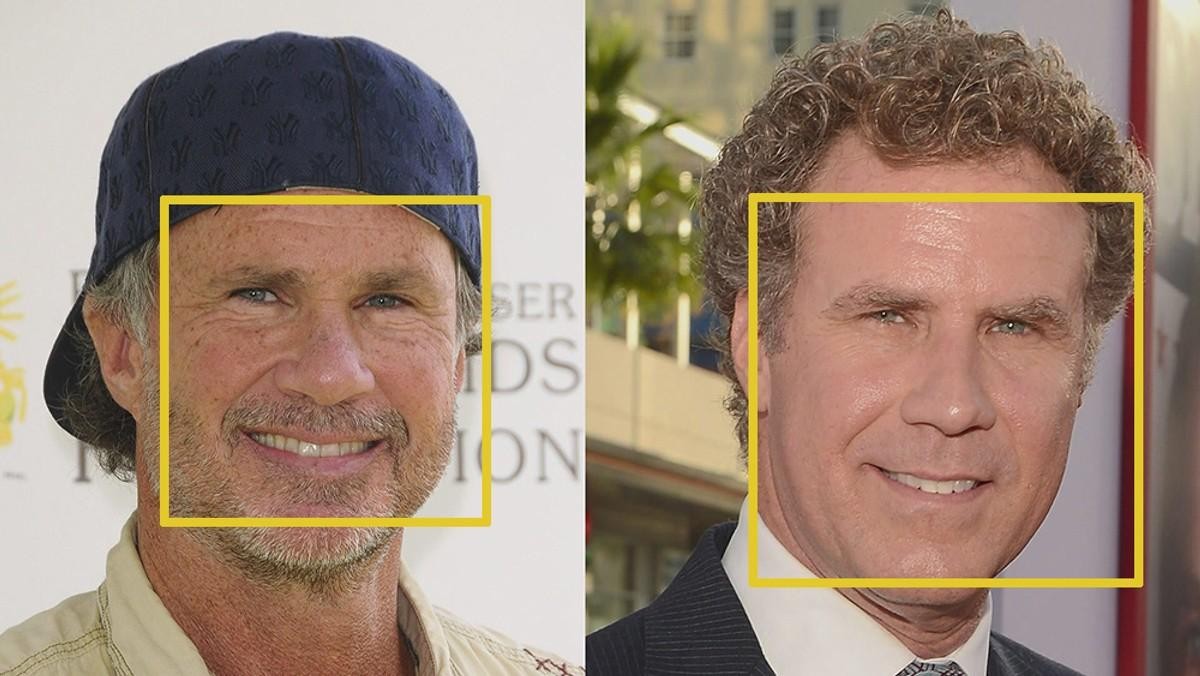


The original image is turned into a HOG representation that captures the major features of the image regardless of image brightnesss.

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:



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



If you want to try this step out yourself using Python and dlib, [here’s](https://gist.github.com/ageitgey/1c1cb1c60ace321868f7410d48c228e1) [code](https://gist.github.com/ageitgey/1c1cb1c60ace321868f7410d48c228e1) showing how to generate and view HOG representations of images.

When, 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:

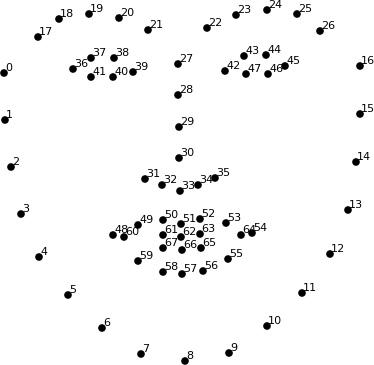


Humans can easily recognize that both images are of Will Ferrell, but computers would see these pictures as two completely different people.

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](http://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf) [in 2014 by Vahid Kazemi and Josephine Sullivan.](http://www.csc.kth.se/~vahidk/papers/KazemiCVPR14.pdf)

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. This image was created by [Brandon Amos](http://bamos.github.io/) of CMU who works on [Open Face](https://github.com/cmusatyalab/openface).

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



**PROTIP**: You can also use this same technique to implement your own version of Snapchat’s real-time 3d face filters!

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](https://en.wikipedia.org/wiki/Affine_transformation) [transformations](https://en.wikipedia.org/wiki/Affine_transformation)):



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

If you want to try this step out yourself using Python and dlib, here’s the [code](https://gist.github.com/ageitgey/ae340db3e493530d5e1f9c15292e5c74) [for finding face landmarks](https://gist.github.com/ageitgey/ae340db3e493530d5e1f9c15292e5c74) and here’s the [code for transforming the image](https://gist.github.com/ageitgey/82d0ea0fdb56dc93cb9b716e7ceb364b) using those landmarks.

# Face Recognition

Face recognition process can be divided into three steps- prepare training data, train face recognizer, prediction. Here training data will be the images present in the dataset. They will be assigned with a integer label of the student it belongs to. These images are then used for face recognition

Initially, the list of local binary patterns (LBP) of entire face is obtained. Later, during recognition process histogram of the faces to be recognized is

calculated and then compared with the already computed histograms and returns the best matched label associated with the student it belongs to.

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 previous- 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. If you’ve ever watched a bad crime show like CSI, you know what I am talking about:



Just like TV! So real! #science

### The most reliable way to measure a face

Ok, so which measurements should we collect from each face to build our known face database? Ear size? Nose length? Eye color? Something else?

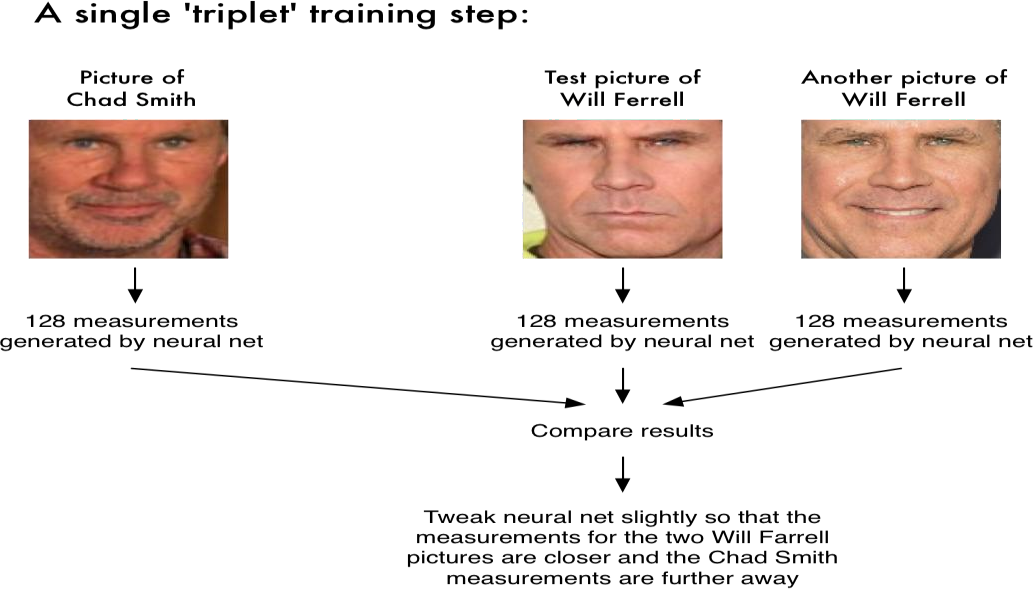
It turns out that the measurements that seem obvious to us humans (like eye color) don’t really make sense to a computer looking at individual pixels in an image. Researchers have discovered that the most accurate approach is to let the computer figure out the measurements to collect itself. Deep learning does a better job than humans at figuring out which parts of a face are important to measure.

The solution is to train a Deep Convolutional Neural Network ([just like we did in](https://medium.com/%40ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721) [Part 3](https://medium.com/%40ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721)). But instead of training the network to recognize pictures objects like we did last time, we are going to train it to generate 128 measurements for each face.

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

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

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



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

Machine learning people call the 128 measurements of each face an **embedding**. The idea of reducing complicated raw data like a picture into a list of computer-generated numbers comes up a lot in machine learning (especially in language translation). The exact approach for faces we are using [was invented in 2015 by researchers at Google](http://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf) but many similar approaches exist.

### Encoding our face image

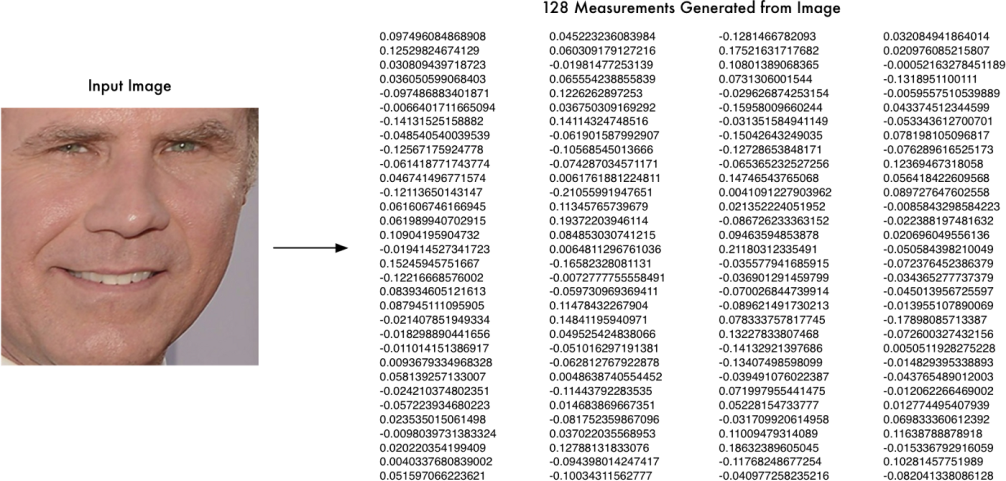
This process of training a convolutional neural network to output face embeddings requires a lot of data and computer power. Even with an

expensive [NVidia Telsa video card](http://www.nvidia.com/object/tesla-supercomputing-solutions.html), it takes [about 24 hours](https://twitter.com/brandondamos/status/757959518433243136) of continuous training to get good accuracy.

But once the network has been trained, it can generate measurements for any face, even ones it has never seen before! So this step only needs to be done

once. Lucky for us, the fine folks at [Open Face](https://cmusatyalab.github.io/openface/) already did this and they [published several trained networks](https://github.com/cmusatyalab/openface/tree/master/models/openface) which we can directly use. Thanks [Brandon Amos](http://bamos.github.io/) and team!

So, all we need to do ourselves is run our face images through their pre-trained network to get the 128 measurements for each face. Here’s the measurements for our test image:



So, what parts of the face are these 128 numbers measuring exactly? It turns out that we have no idea. It doesn’t really matter to us. All that we care is that the network generates nearly the same numbers when looking at two different pictures of the same person.

If you want to try this step yourself, Open Face [provides a script](https://github.com/cmusatyalab/openface/blob/master/batch-represent/batch-represent.lua) that will generate embeddings all images in a folder and write them to a csv file. You [run](https://gist.github.com/ageitgey/ddbae3b209b6344a458fa41a3cf75719) [it like this](https://gist.github.com/ageitgey/ddbae3b209b6344a458fa41a3cf75719).

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](https://en.wikipedia.org/wiki/Support_vector_machine), 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!

# Face Matching

In this the most crucial section of recognition the student, that is comparing captured image against the stored images in the database, this method is done by algorithm each image stored in the database has it’s histogram value calculated and is cross checked against the calculated Histogram value of the images extracted from the captured video feed. The name of the student appears above. As we can see, the system not only detects just one face of a single student, but of multiple students or faces.

# Proposed Architecture

The architecture for the proposed system has been designed to keep it pretty straightforward and easy to understand. The steps that have to be undertaken to reach the final end step of the system which is making sure the attendance of the student is updated correctly and timely. The system can easily be accessed by anyone, where attendance of the students can easily be checked and maintained by the faculty as when required. The Droid-

Cam will allow easy use for capturing live video feeds of the class and simultaneously perform recognition for the students. OpenCV-Python will be used to access the algorithms and their libraries that are required for training, recognition and matching of the captured images against the stored images available in the previously acquired data sets.



2022-2023

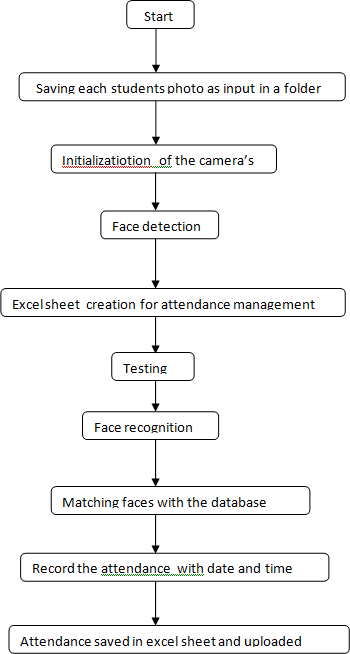
# CREATION OF DATABASE

The face database is an important step to be done before any further process can be initiated. This is because the face database acts as a comparison factor during the recognition process which will be discussed in later section. In the process above, a csv file is created to aid the process of image labelling because there will be more than one portrait stored for each student, thus, in order to group their portraits under the name of the same person, labels are used to distinguish them. After that, those images will be inserted into a recognizer to do its training. Since the training process is very time consuming as the face database grew larger, the training is only done right after there is a batch of new addition of student’s portraits to ensure the training is done as minimum as possible.



2022-2023

# Higher level single block diagram of the system





2022-2023

METHODOLOGY

* + 1. Flow chart:

Start

Saving images of each students as input in a folder

Image processing

Face detection

End

Attendance marking

* + 1. **SOFTWARE DOWNLOAD**

PyCharm is an integrated development environment used in computer programming. It is downloaded for windows 10 from the official website.

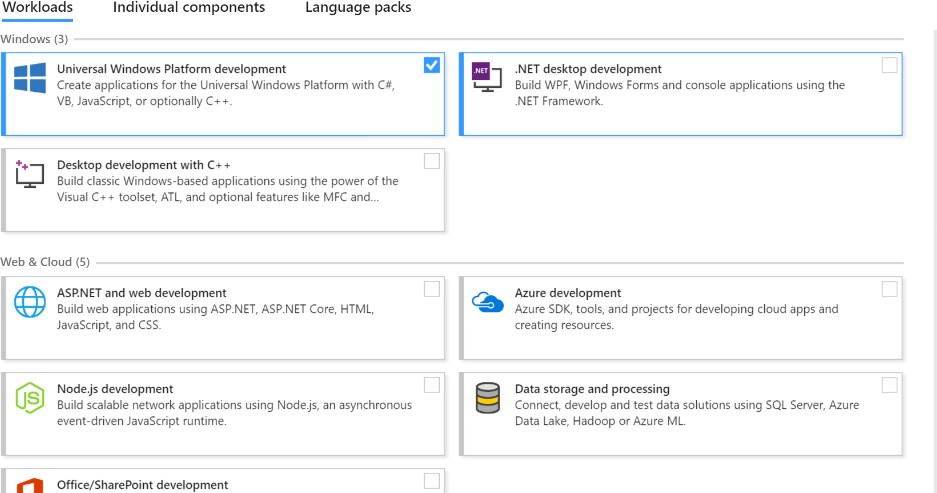
<https://www.jetbrains.com/pycharm/download/>



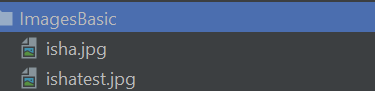
## INSTALLATION OF PACKAGES

* + - * CMake
      * dlib
      * Face-recognition
      * OpenCV-python
      * NumPy

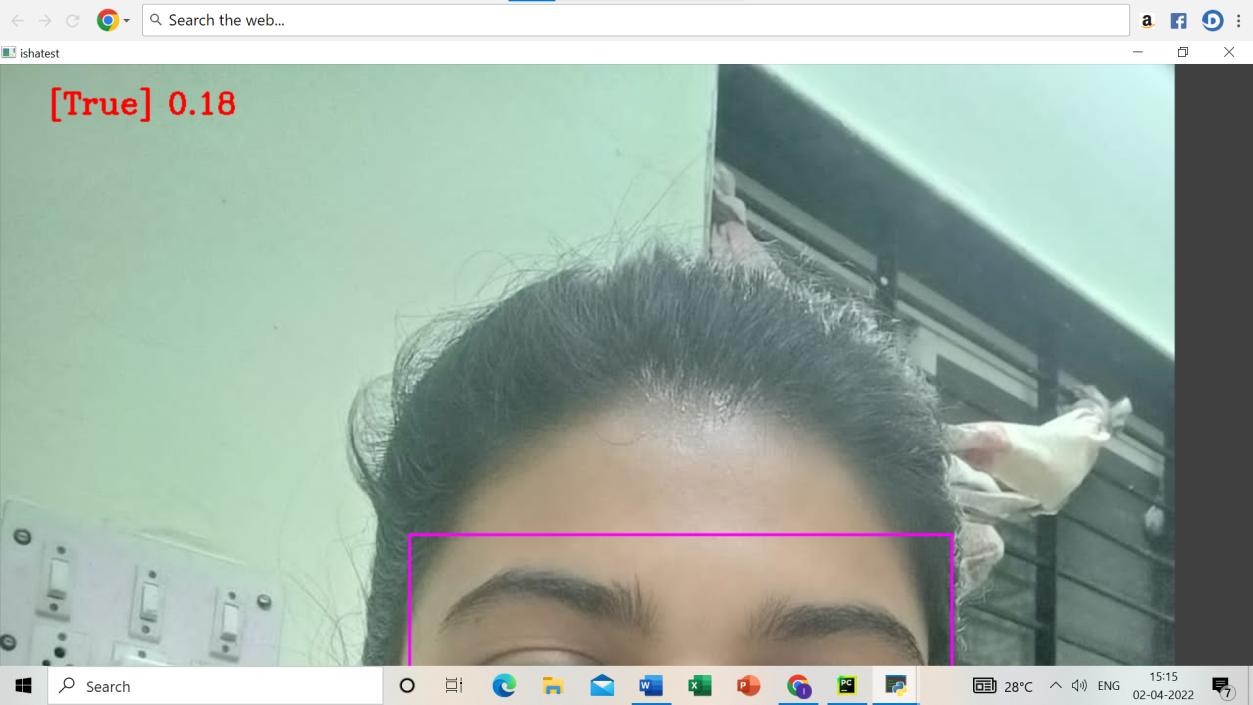
All these libraries are installed along with visual studio with desktop development with C++.



## CREATION OF FOLDER OF TEST IMAGES



* The images are converted into RGB.
* Face encodings and locations are detected using face recognition library.
* Face distance is calculated and image is compared with test images.
* Result is true if the image has least distance values as compared to the test images.

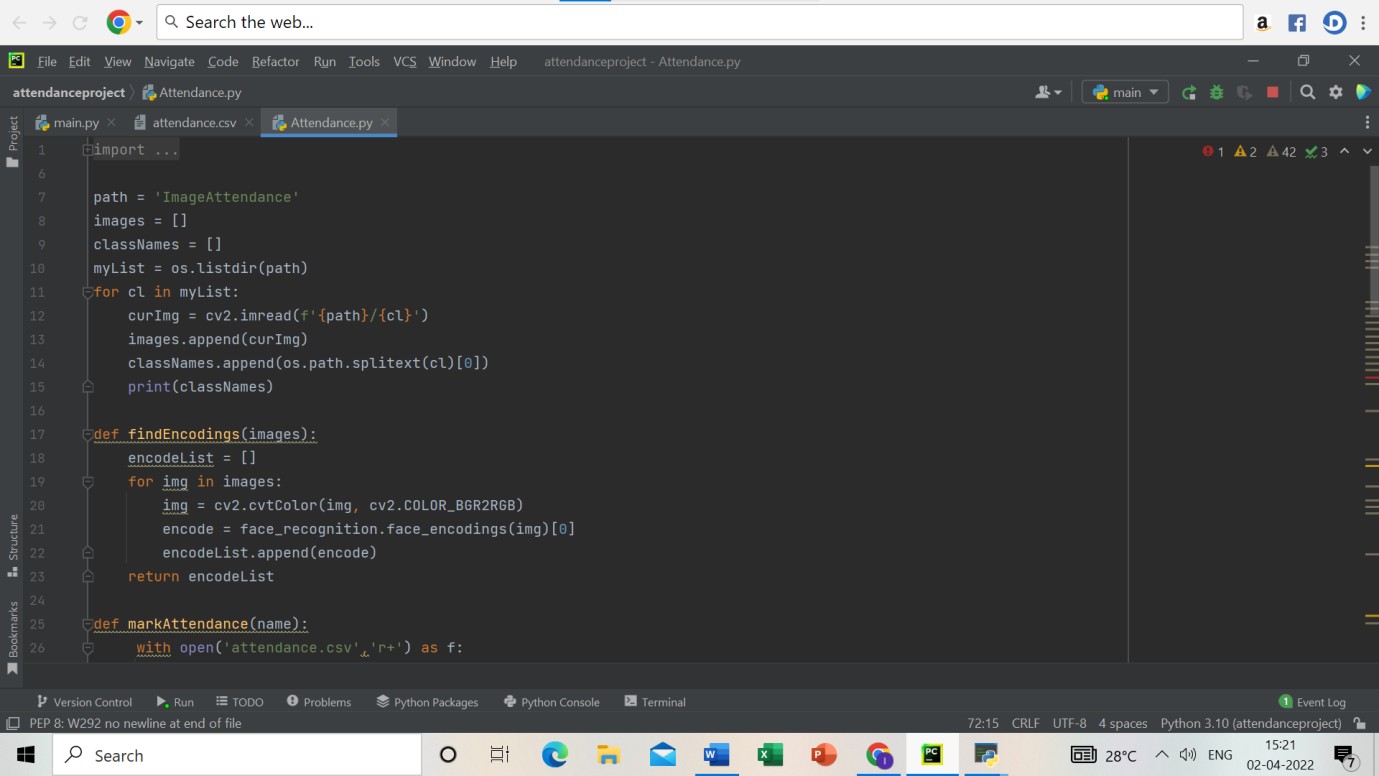


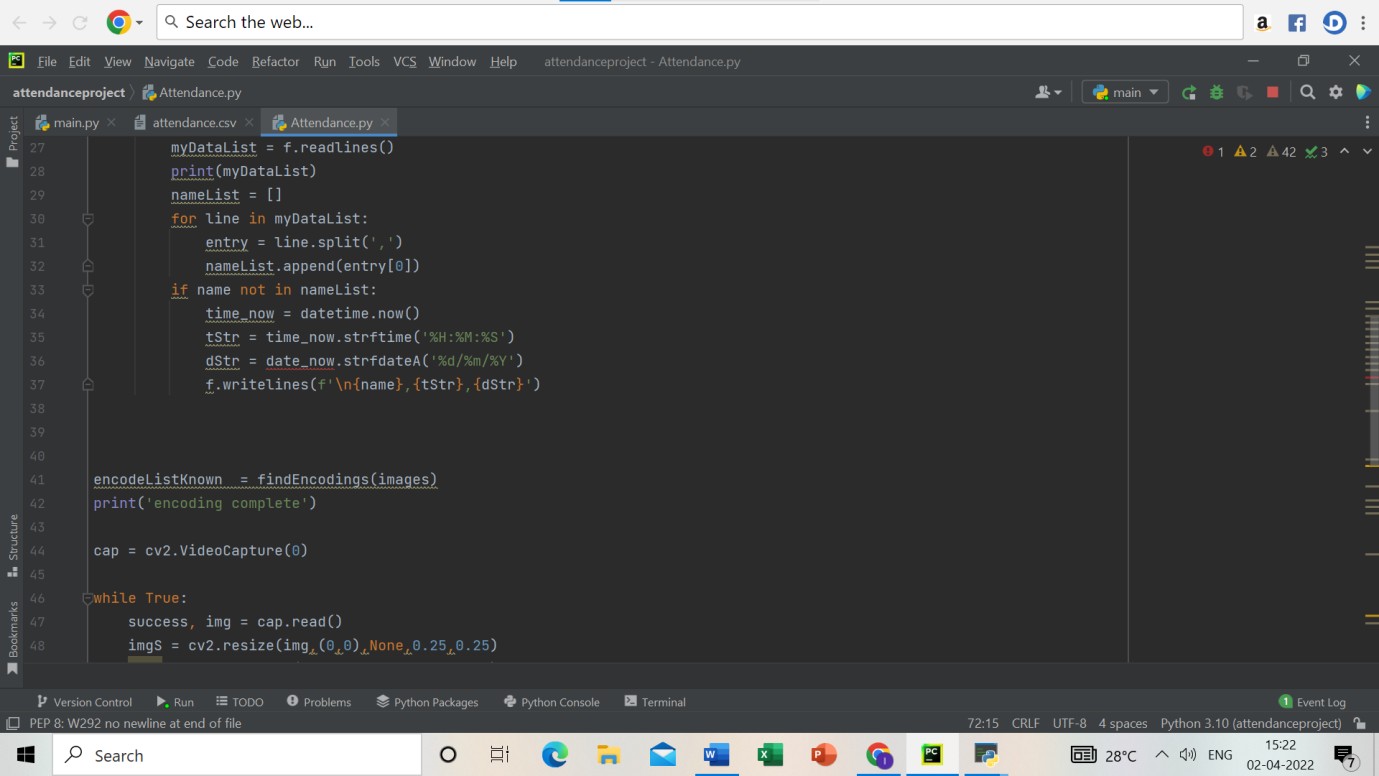
# CREATION OF attendance.py:

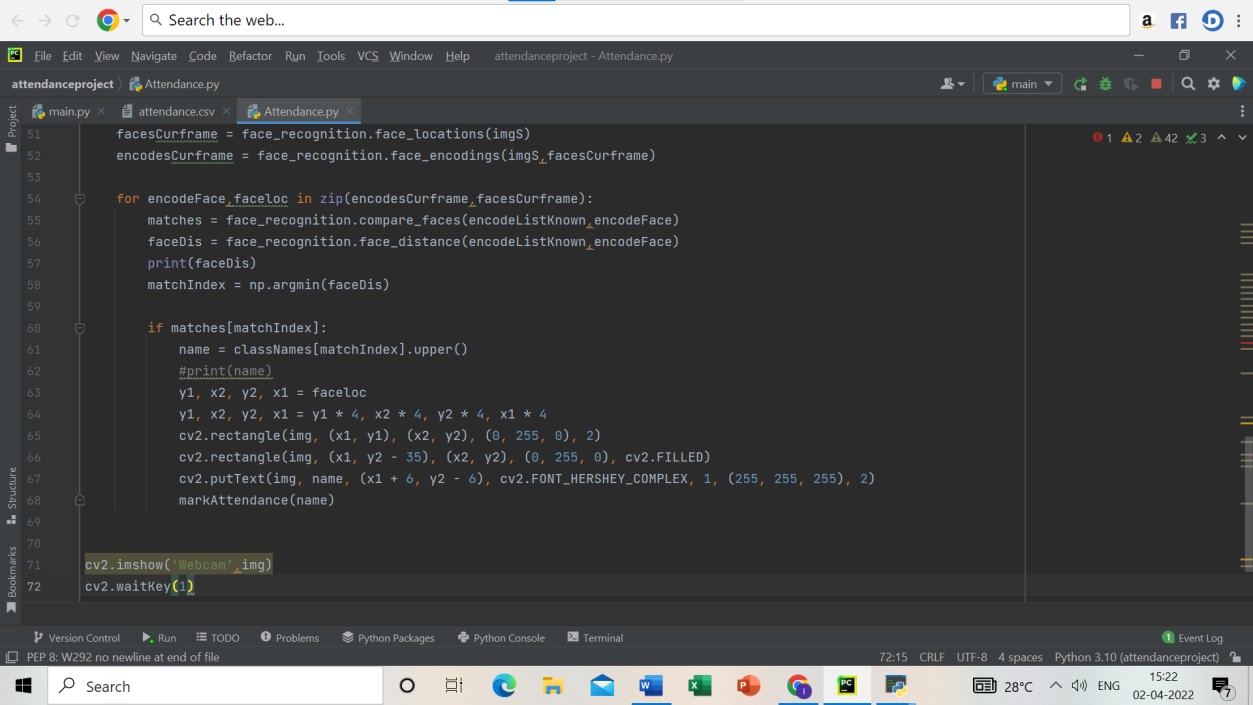
The same process of recognition and comparison is repeated for multiple images. The folder created has recently 5 images of our group members as shown.

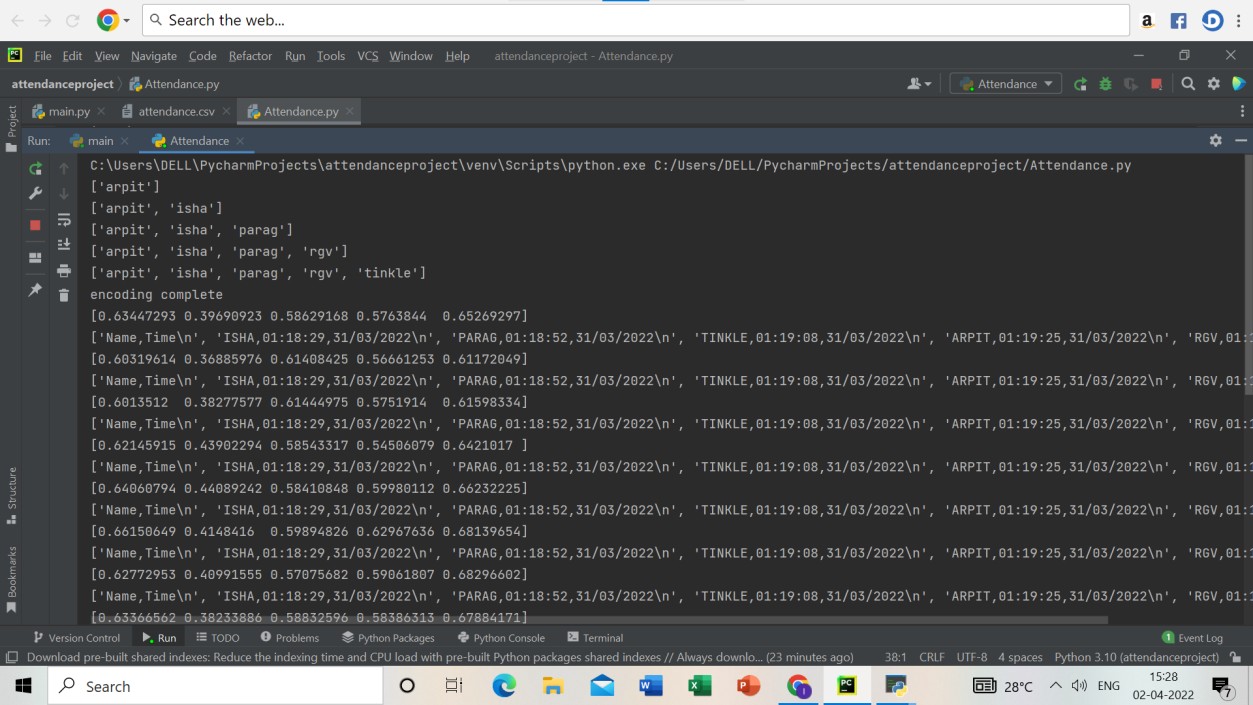


The code for performing face recognition, generation of encodings and distance, face detection



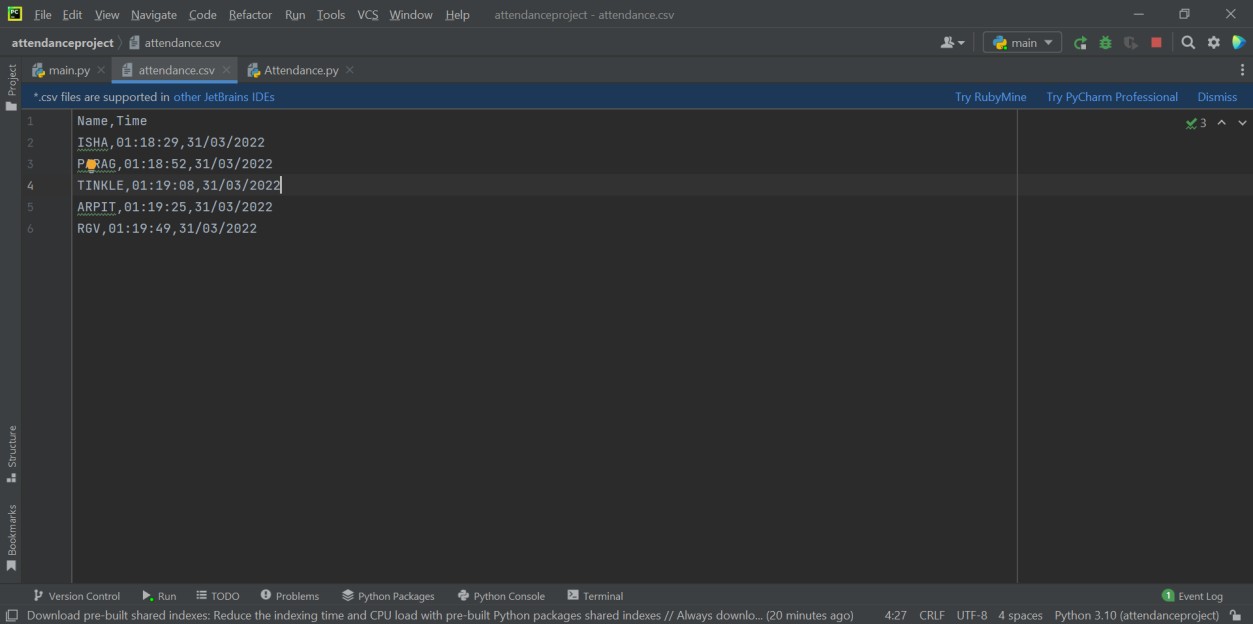


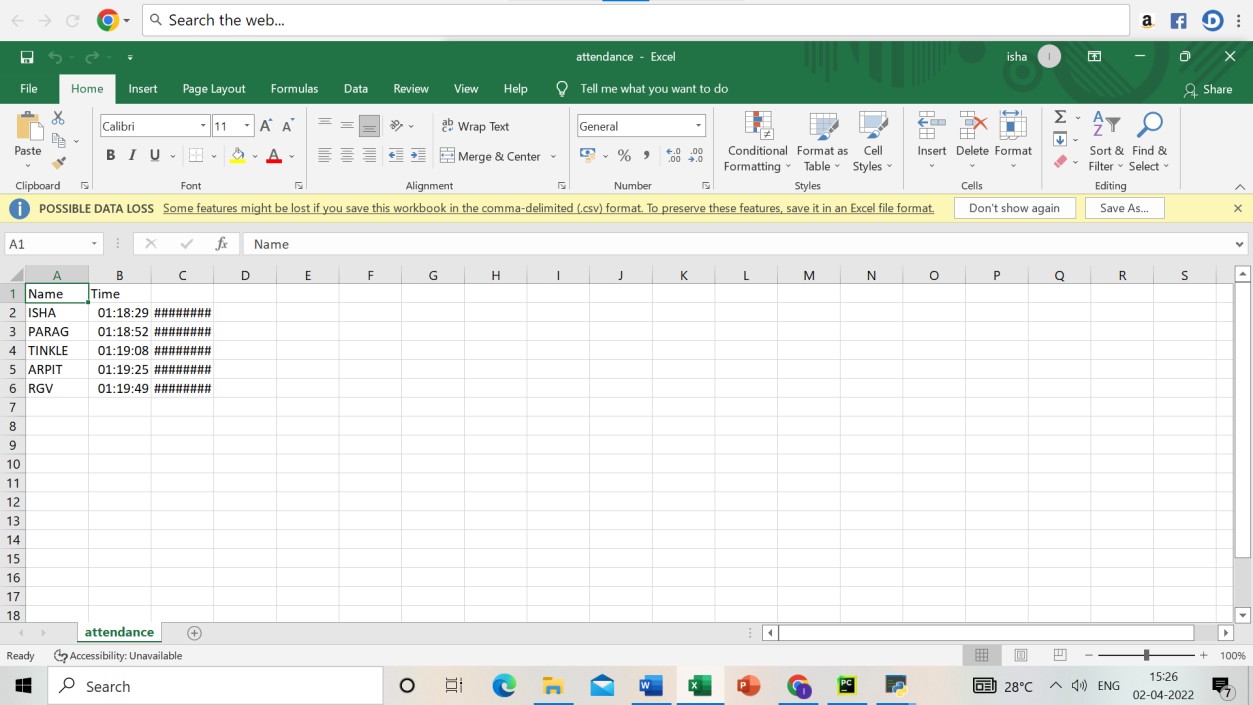




## CREATION OF CSV FILE:

This file stores attendance of students in real date and time. The data of this file can be transferred to an excel sheet for educational or other purposes.







2022-2023

# EXPERIMENT AND RESULTS

In this proposed system, results of different experiments performed are shown. Image samples are collected using webcam. For capturing the image samples, we resized the frame such that it captures the face region effectively, the image samples are stored in a separate folder which further helps in training the model. When face is matched with the list of faces available in the database, it recognizes the person successfully and displays the name of the person and if in case they were not registered it would have shown. The system is also capable of distinguishing between different faces . After recognizing the faces successfully, the system updates the attendance in Excel sheet with student details including date and time and also shows the results in the terminal. The system updates the attendance which we can import and export it into Excel sheet. Finally results obtained from experiments done in a live classroom are shown.



2022-2023

# CODES USED IN THIS PROJECT

## BASIC CODE:

import cv2

import NumPy as np import face\_recognition

imgisha = face\_recognition.load\_image\_file('ImagesBasic/isha.jpg') imgisha = cv2.cvtColor(imgisha,cv2.COLOR\_BGR2RGB)

imgishatest = face\_recognition.load\_image\_file('ImagesBasic/ishatest.jpg') imgishatest = cv2.cvtColor(imgishatest,cv2.COLOR\_BGR2RGB)

faceLoc = face\_recognition.face\_locations(imgisha)[0] encodeisha = face\_recognition.face\_encodings(imgisha)[0]

cv2.rectangle(imgisha,(faceLoc[3],faceLoc[0]),(faceLoc[1],faceLoc[2]),(255,0,25 5),2)

faceLoctest = face\_recognition.face\_locations(imgishatest)[0] encodeishatest = face\_recognition.face\_encodings(imgishatest)[0]

cv2.rectangle(imgishatest,(faceLoctest[3],faceLoctest[0]),(faceLoctest[1],faceL octest[2]),(255,0,255),2)

results = face\_recognition.compare\_faces([encodeisha],encodeishatest) faceDis = face\_recognition.face\_distance([encodeisha],encodeishatest) print(results,faceDis)

cv2.putText(imgishatest,f'{results}

{round(faceDis[0],2)}',(50,50),cv2.FONT\_HERSHEY\_COMPLEX,1,(0,0,255),2)

cv2.imshow('isha',imgisha) cv2.imshow('ishatest',imgishatest) cv2.waitKey(0)

## MAIN CODE :

import cv2

import numpy as np import face\_recognition import os

from datetime import datetime

path = 'ImageAttendance' images = []

classNames = []

myList = os.listdir(path) for cl in myList:

curImg = cv2.imread(f'{path}/{cl}') images.append(curImg) classNames.append(os.path.splitext(cl)[0])

print(classNames)

def findEncodings(images):

encodeList = [] for img in images:

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB) encode = face\_recognition.face\_encodings(img)[0] encodeList.append(encode)

return encodeList

def markAttendance(name):

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

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 = date\_now.strfdateA('%d/%m/%Y') f.writelines(f'\n{name},{tStr},{dStr}')

encodeListKnown = findEncodings(images) print('encoding complete')

cap = cv2.VideoCapture(0) while True:

success, img = cap.read()

imgS = cv2.resize(img,(0,0),None,0.25,0.25) imgS = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

facesCurframe = face\_recognition.face\_locations(imgS)

encodesCurframe = face\_recognition.face\_encodings(imgS,facesCurframe)

for encodeFace,faceloc in zip(encodesCurframe,facesCurframe):

matches = face\_recognition.compare\_faces(encodeListKnown,encodeFace) faceDis = face\_recognition.face\_distance(encodeListKnown,encodeFace) print(faceDis)

matchIndex = np.argmin(faceDis)

if matches[matchIndex]:

name = classNames[matchIndex].upper() #print(name)

y1, x2, y2, x1 = faceloc

y1, x2, y2, x1 = y1 \* 4, x2 \* 4, y2 \* 4, x1 \* 4 cv2.rectangle(img, (x1, y1), (x2, y2), (0, 255, 0), 2)

cv2.rectangle(img, (x1, y2 - 35), (x2, y2), (0, 255, 0), cv2.FILLED)

cv2.putText(img, name, (x1 + 6, y2 - 6), cv2.FONT\_HERSHEY\_COMPLEX, 1, (255,

255, 255), 2)

markAttendance(name) cv2.imshow('Webcam',img) cv2.waitKey(1)



2022-2023

# IMPACT AND SIGNIFICANCE

* Students will be more punctual on attending classes. This is due to the attendance of a student can only be taken personally where any absentees will be noticed by the system. This can not only train the student to be punctual as well as avoids any immoral ethics such as signing the attendance for their friends.
* The institution can save a lot of resources as enforcement are now done by means of technology rather than human supervision which will waste a lot of human resource for an insignificant process.
* The application can operate on any device at any location as long as there is Wi-Fi coverage or Ethernet connection which makes the attendance system to be portable to be placed at any intended location.
* It saves a lot of cost in the sense that it had eliminated the paperwork completely.
* The system is also time effective because all calculations are all automated. In short, the project is developed to solve the existing issues in the old attendance system.



2022-2023

# FUTURE SCOPE AND PROJECT DIRECTION

In future, to increase the reliability of the results of this work and to make it more user accessible and friendly some extra work can be done and implemented to this project, such as

* + To increase the accuracy of the face recognition and face detection, this work can use machine learning algorithms, artificial intelligence, computer vision and new feature software can be used.
  + The stored data can be shared and uploaded online on the institute portal.
  + Some software or an android application can be built to provide easy access to the users.
  + The System will have to separate between recognized and unrecognized faces, faces that go unrecognized can be stored in a secondary database.
  + The future scope of the project can be integrated with the hardware components for example GSM through which a monthly list of the defaulter students can be sent to the mentor.

The aim is to automate and make a system that is useful to the organization such as an institute, schools, industries . The main intention of this project is to solve the issues encountered in the traditional methods of attendance system which consumes too much time while reproducing a brand new innovative smart system that can provide convenience to the institution and saves the time and workload of faculties or management system.

In this project, an application will be developed which is capable of recognizing the identity of each individual and eventually record down the data into a database system. Apart from that, an excel sheet is created which shows the students attendance and can be shared to the respected faculty.



2022-2023

# CONCLUSIONS

This work presents a simple yet efficient approach to calculate the attendance in a class by implementing facial recognition techniques.

Face detection has many applications in different fields. The proposed system used the concept of face recognition and extraction to make an automatic attendance system. Many problems faced during the development of this system, but the most crucial one is to have good accuracy. Face matching is another problem which is greatly affected by the light intensity, face orientation and some other practical situations. Different experiments has been performed using different matching algorithms. Multiple images at different angles are taken for enhancing the accuracy of the system. The system not only detects just one face of a single student, but successfully detects multiple students or faces. The system also successfully recognizes and marks the attendance of the detected student’s. This project is easy to operate and also it saves time. Moreover, the attendance data can be accessed anytime and also can be saved in the database as required.



2022-2023

# BIBLIOGRAPGHY

* + Prince Tiwari and Arun Parakh. Automatic Attendance System using Face Detection. In MAT Journals, Journal of Instrumentation and Innovation Sciences, Vol 2 Issue 2, pages 28 37, 2016
  + [https://medium.com/@ageitgey/machine-learning-is-fun-part-4-](https://medium.com/%40ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78) [modern-face-recognition-with-deep-learning-c3cffc121d78](https://medium.com/%40ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78)
  + E. Varadharajan, R. Dharani, S. Jeevitha, B. Kavinmathi and S. Hemalatha, "Automatic attendance management system using face detection," *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, 2016, pp. 1-3, doi: 10.1109/GET.2016.7916753.
  + [https://www.researchgate.net/publication/343283987\_Class\_Attendanc](https://www.researchgate.net/publication/343283987_Class_Attendance_Management_System_using_Facial_Recognition) [e\_Management\_System\_using\_Facial\_Recognition](https://www.researchgate.net/publication/343283987_Class_Attendance_Management_System_using_Facial_Recognition)
  + Salim, Omar Abdul Rehman, Rashidah Funke Olanrewaju, and Wasiu Adebayo Balogun. "Class attendance management system using face recognition." 2018 7th International Conference on Computer and Communication Engineering (ICCCE). IEEE, 2018.
  + https:[//www.r](http://www.researchgate.net/publication/326261079_Face_detection_)es[earchgate.net/publication/326261079\_Face\_detection\_](http://www.researchgate.net/publication/326261079_Face_detection_) system\_for\_attendance\_of\_class\_student
  + Lukas, Samuel, et al. "Student attendance system in classroom using face recognition technique." 2016 International Conference on Information and Communication Technology Convergence (ICTC). IEEE, 2016.
  + Some contents from YouTube
  + Introduction to visual studio.com/Wikipedia.com
  + <https://www.jetbrains.com/pycharm/download/>
  + https:[//www.sup](http://www.superdatascience.com/blogs/opencv-face-recognition)er[datascience.com/blogs/opencv-face-recognition](http://www.superdatascience.com/blogs/opencv-face-recognition)