

## FACES RECOGNITION SYSTEM BASED ATTENDANCE

<sup>1</sup> S. Amar Nadh, <sup>2</sup> Anil Kumar Gunda, <sup>3</sup> D.Sateesh, <sup>4</sup>Boya Ramanjaneyulu

<sup>1,2,3</sup>Assistant Professor, <sup>4</sup>Student, Dept. of Computer Science Engineering, Newton's Institute of Engineering, Macherla, Andhra Pradesh, India.

### ABSTRACT

Automatic face recognition (AFR) technologies have made many improvements in the changing world. Smart Attendance using Real-Time Face Recognition is a real-world solution which comes with day to day activities of handling student attendance system. Face recognition-based attendance system is a process of recognizing the students face for taking attendance by using face biometrics based on high - definition monitor video and other information technology. In my face recognition project, a computer system will be able to find and recognize human faces fast and precisely in images or videos that are being captured through a surveillance camera. Numerous algorithms and techniques have been developed for improving the performance of face recognition. Here we have used the concept of LBPH (LOCAL BINARY PATTERN HISTOGRAM) Algorithm.

**Keywords:** Face Recognition, Face Detection, LBPH Algorithm, Facial Features, OpenCV

### INTRODUCTION

The main objective of this project is to develop face recognition based automated student attendance system. In order to achieve better performance, the test images and training images of this proposed approach are limited to frontal and upright facial images that consist of a single face only. The test images and training images have to be captured by using the same device to ensure no quality difference. In addition, the students have to register in the database to be recognized. The enrolment can be done on the spot through the user-friendly interface.

### Existing problem

At present attendance marking involves manual attendance on paper sheet by professors and teachers. but it is very time consuming process and chances of proxy is also one problem that arises in such type of attendance marking. also there are attendance marking system such as RFID, Biometrics etc. but these systems are currently not so much popular in schools and classrooms for students as they have their own advantages and disadvantages.

The Problems with Current System The problem with this approach in which manually taking and maintains the attendance records is that it is very inconvenient task. Traditionally, students attendances are taken manually by using attendance sheet given by the faculty members in class, which is a time consuming event Moreover, it is very difficult to verify one by one student in a large classroom environment with distributed branches whether the authenticated students are actually responding or not the ability to compute the attendance percentage becomes a major task as manual computation produces error and also wastes a lots of time this method could easily allows for impersonation and the attendance sheet could be stolen or lost.

### Proposed problem

Traditional student attendance marking technique is often facing a lot of trouble. The face recognition student attendance system emphasis its simplicity by eliminating classical student attendance marking technique such as calling student names or checking respective identification cards. There are not only disturbing the teaching process but also causes distraction for students during exam sessions. Apart from calling names, attendance sheet is passed around the classroom during the lecture sessions. The lecture class especially the class with a large number of students might find it difficult to have the attendance sheet being passed around the class. Thus, face recognition student attendance system is proposed in order to replace the

manual signing of the presence of students which are burdensome and causes students get distracted in order to sign for their attendance. Using Local Binary Patterns Histograms (LBPH) algorithm in Open CV the student's faces are trained and recognized. Python automates the tasks by providing for the execution of the programs in Computer Vision and GUI of the system along with managing the database of the student attendance. Furthermore, the face recognition based automated student attendance system able to overcome the problem of fraudulent approach and lecturers do not have to count the number of students several times to ensure the presence of the students.

### Objectives

The objective of this project is to develop face recognition based automated student attendance system. Expected achievements in order to fulfill the objectives are:

- a. To detect the face segment from the video frame.
- b. To extract the useful features from the face detected.
- c. To classify the features in order to recognize the face detected.
- d. To record the attendance of the identified student.

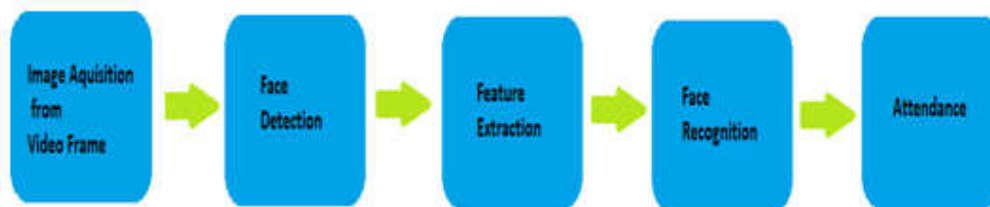


Figure.1. System flow

### RELATED WORK

#### Brief introduction of technologies used:

The technologies used in our project are as following:

#### Opencv

#### Visual studio code

#### Python

#### Opencv

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching streetview images

together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

Visual studio code:

**Visual Studio Code** is a source-code editor developed by Microsoft for Windows, Linux and macOS. It includes support for debugging, embedded Git control and GitHub, syntax highlighting, intelligent code completion, snippets, and code refactoring.

Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++,C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

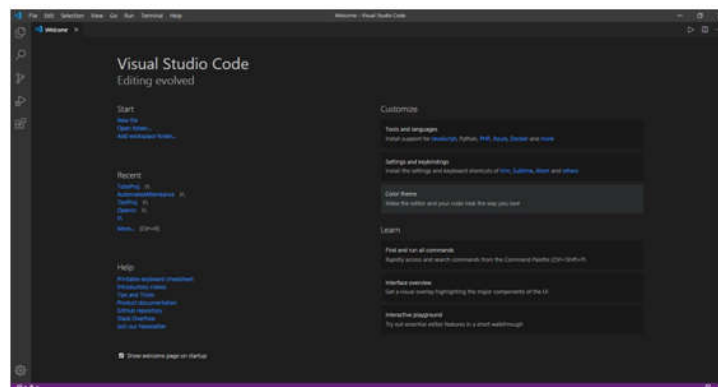


Figure 2 Virtual studio code interface

Python:

**Python** is a widely used high-level programming language for general-purpose programming, created by Guido van Rossum and first released in 1991. An interpreted language, Python has a design philosophy which emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly braces or keywords), and a syntax which allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable writing clear programs on both a small and large scale. The logo of python is shown in Figure 2.2.

Python features a dynamic type system and automatic memory management and supports object-oriented, imperative, functional programming, and procedural styles. It has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.



Figure.3. python logo

**Automated Attendance System Using Face Recognition:**

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

**Student Attendance System Using Iris Detection:**

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

Table 1: Existing system

System Type	Advantages	Disadvantages
RFID Card System	Simple	Fraudulent usage
Fingerprint system	Accurate	Time-consuming
Voice recognition system	Low Operational Costs	Less accurate compared to others
Iris recognition system	Accurate	Privacy Invasion

**Definition of Terms and History**

**Face Detection**

Face detection is the process of identifying and locating all the present faces in a single image or video regardless of their position, scale, orientation, age and expression. Furthermore, the detection should be irrespective of extraneous illumination conditions and the image and video content.

**Face Recognition**

Face Recognition is a visual pattern recognition problem, where the face, represented as a three dimensional object that is subject to varying illumination, pose and other factors, needs to be identified based on acquired images.

Face Recognition is therefore simply the task of identifying an already detected face as a known or unknown face and in more advanced cases telling exactly whose face it is

Difference between Face Detection and Face Recognition

Face detection answers the question, Where is the face? It identifies an object as a “face” and locates it in the input image. Face Recognition on the other hand answers the question who is this? Or whose face is it? It decides if the detected face is someone known or unknown based on the database of faces it uses to validate this input image. It can therefore be seen that face detections output (the detected face) is the input to the face recognizer and the face Recognition’s output is the final decision i.e. face known or face unknown.

Face Detection

A face Detector has to tell whether an image of arbitrary size contains a human face and if so, where it is. Face detection can be performed based on several cues: skin color (for faces in color images and videos, motion (for faces in videos), facial/head shape, facial appearance or a combination of these parameters. Most face detection algorithms are appearance based without using other cues. An input image is scanned at all possible locations and scales by a sub window. Face detection is posed as classifying the pattern in the sub window either as a face or a non- face. The face/non face classifier is learned from face and non-face training examples using statistical learning methods. Most modern algorithms are based on the Viola Jones object detection framework, which is based on Haar Cascades.

Table 2. Face detection methods

Face detection method	Advantages	Disadvantages
Viola Jones Algorithm	1. High detection Speed. 2. High Accuracy.	1. Long Training Time. 2. Limited Head Pose. 3. Not able to detect dark faces.
Local Binary Pattern Histogram	1. Simple computation. 2. High tolerance against the monotonic illumination changes.	Only used for binary and grey images.
Neural-Network	High accuracy only if large size of image were trained.	1. Detection process is slow and computation is complex. 2. Overall performance is weaker than Viola-Jones algorithm
Eigen faces	Recognition is Simple and Effective.	Face images tested in the experiments are taken in the uniform background, while this situation is not suitable in natural scene.
Fisher Faces	Maximizing the separability of different identities.	Sensitive to pixel misalignment, linear classes cannot adequately describe pose variations.

Local Binary Patterns Histogram:

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number.

It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further

been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images with a simple data vector.

LBPH algorithm work step by step:

LBPH algorithm work in 5 steps.

**Parameters:** the LBPH uses 4 parameters:

**Radius:** the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.

**Neighbors:** the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.

**Grid X:** the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

**Grid Y:** the number of cells in the vertical direction. The more cells, the finer the grid, the higher the

dimensionality of the resulting feature vector. It is usually set to 8.

Training the Algorithm:

First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID.

With the training set already constructed, let's see the LBPH computational steps.

Applying the LBP operation:

The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbours.

The image below shows this procedure:

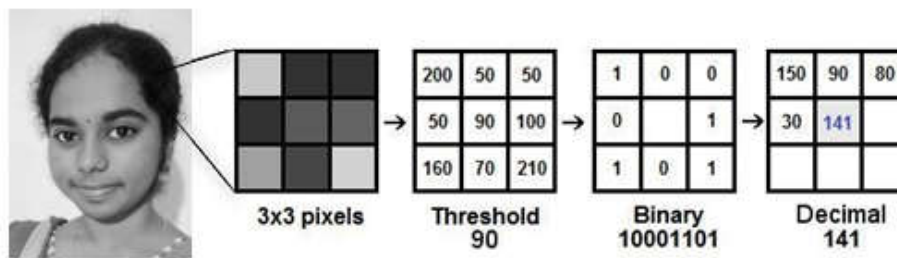


Figure 3: LBP Operation

Based on the image above, let's break it into several small steps so we can understand it easily:

- Suppose we have a facial image in grayscale.
- We can get part of this image as a window of 3x3 pixels.
- It can also be represented as a 3x3 matrix containing the intensity of each pixel (0~255).
- Then, we need to take the central value of the matrix to be used as the threshold.
- This value will be used to define the new values from the 8 neighbors.
- For each neighbor of the central value (threshold), we set a new binary value. We set 1 for values equal or higher than the threshold and 0 for values lower than the threshold
- Now, the matrix will contain only binary values (ignoring the central value). We need to concatenate each binary value from each position from the matrix line by line into a new binary value (e.g. 10001101). Note: some authors use other approaches to concatenate the binary values (e.g. clockwise direction), but the final result will be the same.
- Then, we convert this binary value to a decimal value and set it to the central value of the matrix, which is actually a pixel from the original image.
- At the end of this procedure (LBP procedure), we have a new image which represents better the characteristics of the original image

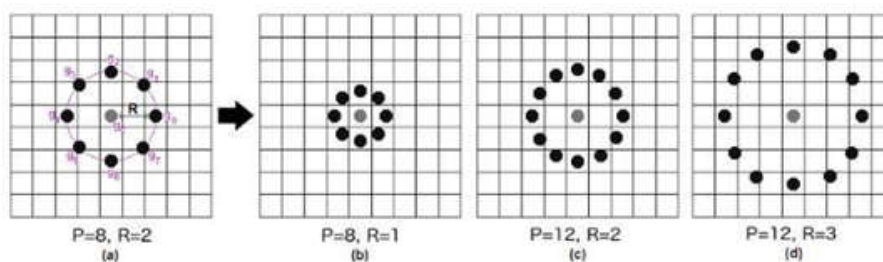


Figure 4: The LBP operation Radius Change

It can be done by using bilinear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

Extracting the Histograms:

Now, using the image generated in the last step, we can use the Grid X and Grid Y parameters to divide the image into multiple grids, as can be seen in the following image:

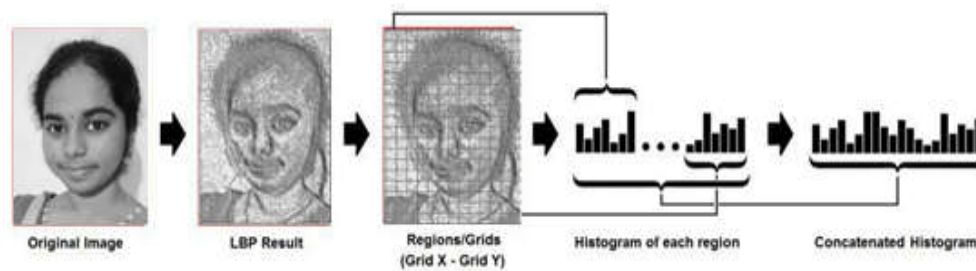


Figure.5. Extracting the Histogram

Based on the image above, we can extract the histogram of each region as follows:

As we have an image in grayscale, each histogram (from each grid) will contain only 256 positions (0~255) representing the occurrences of each pixel intensity.

Then, we need to concatenate each histogram to create a new and bigger histogram. Supposing we have 8x8 grids, we will have  $8 \times 8 \times 256 = 16,384$  positions in the final histogram. The final histogram represents the characteristics of the image original image.

**Performing the face recognition:** In this step, the algorithm is already trained. Each histogram created is used to represent each image from the training dataset. So, given an input image, we perform the steps again for this new image and creates a histogram which represents the image.

So to find the image that matches the input image we just need to compare two histograms and return the image with the closest histogram.

We can use various approaches to compare the histograms (calculate the distance between two histograms), for example: Euclidean distance, chi-square, absolute value, etc. In this example, we can use the **Euclidean distance** (which is quite known) based on the following formula:

$$D = \sqrt{\sum_{i=1}^n (hist1_i - hist2_i)^2}$$

Where D is distance in n-dimensional space

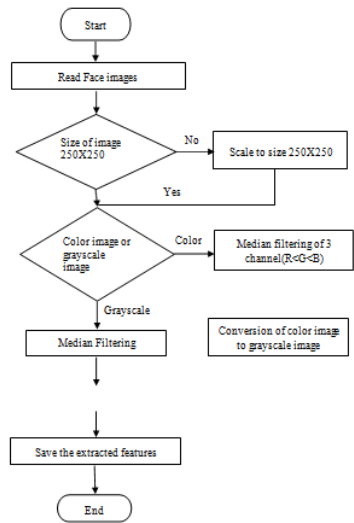
So the algorithm output is the ID from the image with the closest histogram. The algorithm should also return the calculated distance, which can be used as a 'confidence' measurement. Note: don't be fooled about the 'confidence' name, as lower confidences are better because it means the distance between the two histograms is closer.

We can then use a threshold and the 'confidence' to automatically estimate if the algorithm has correctly recognized the image. We can assume that the algorithm has successfully recognized if the confidence is lower than the threshold defined.

### Related work

The approach performs face recognition based student attendance system. The methodology flow begins with the capture of image by using simple and handy interface, followed by pre-processing of the captured facial images, then feature extraction from the facial images, subjective selection and lastly classification of the facial images to be recognized. LBP feature extraction methods are studied in detail and computed in this proposed approach in order to make comparisons. LBP is enhanced in this approach to reduce the illumination effect. The details of each stage will be discussed in the following sections.

The flow chart for the proposed system is categorized into two parts, first training of images followed by testing images (recognize the unknown input image)

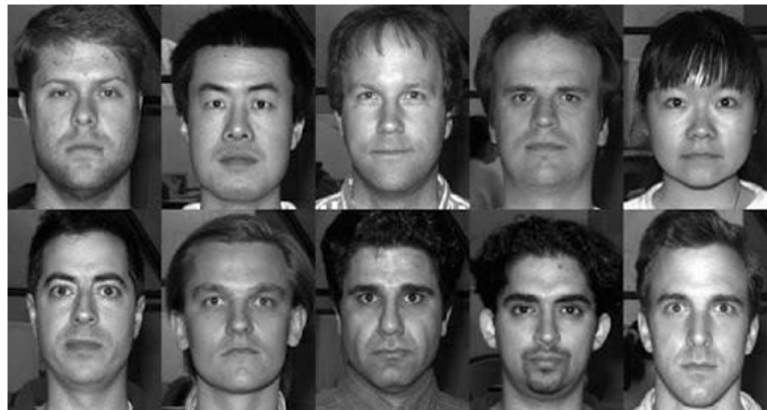




Input images

Although our own database should be used to design real time face recognition student attendance system, the databases that are provided by the previous researchers are also used to design the system more effectively, efficiently and for evaluation purposes.

Yale face database is used as both training set and testing set to evaluate the performance. Yale face database contains one hundred and sixty-five grayscale images of fifteen individuals. There are eleven images per individual; each image of the individual is in different condition. The conditions included centre-light, with glasses, happy, left-light, without glasses, normal, right- light, sad, sleepy, surprised and wink. These different variations provided by the database is able to ensure the system to be operated consistently in variety of situations and conditions.



For our own database, the images of students are captured by using laptop built in camera and mobile phone camera. Each student provided thirty to forty images, for training and for testing set. The images captured by using laptop built in camera are categorized as low quality images, whereas mobile phone camera captured images are categorized as high quality images. The high quality images consists of some students while low quality images consists of remaining students. The recognition rate of low quality images and high quality images will be compared to draw a conclusion in term of performance between image sets of different quality.

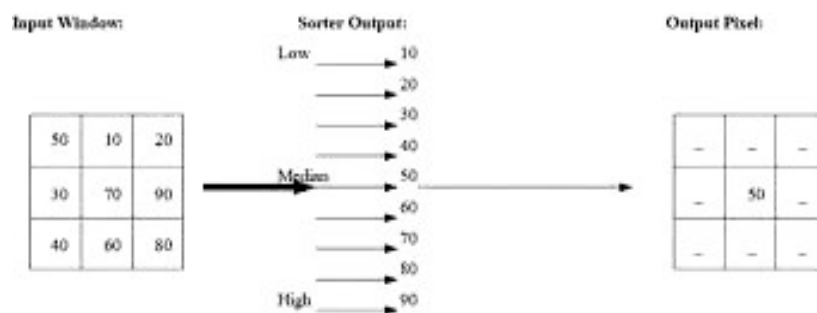


Figure 6. Pixel Matrix

The median filtering is a nonlinear kind of image filtering which is effectively used in reducing the image noise while maintaining the edges of the image. It works in a window of size  $WM \times WN$  where  $WM$  and  $WN$  are both odd. The median filter works through the image pixel by pixel, changing each value with the median value of neighboring pixels. Likewise, neighborhood refers to the window which slides pixel by pixel through the entire image. This can be computed by first sorting the entire pixel values from the identified window into numerical order and then replace it with the value of the pixel being considered with the middle (median) pixel value. If the neighborhood under consideration consists of an even number of pixels, the median value selected as the output is the average of the two middle pixel values depicted the process of median filtering. The median filter replaces a pixel by the median, instead of the average, of all pixels in a neighborhood.

Conversion to Grayscale Image:

LBP extracts the grayscale features from the contrast improved images as 8 bit texture descriptor. Therefore, color images have to be converted to grayscale images before proceeding to the later steps. By converting color images to grayscale images, the complexity of the computation can be reduced resulting in higher speed of computation.

Working Principle of LBP:

LBP is basically a texture based descriptor which it encoded local primitive into binary string. The original LBP operator works on a 3×3 mask size. 3×3 mask size contains 9 pixels. The center pixel will be used as a threshold to convert the neighboring pixels (the other 8 pixels) into binary digit. If the neighbouring pixel value is larger than the center pixel value, then it is assigned to 1, otherwise it is assigned to 0. After that, the neighborhood's pixel bits are concatenated to a binary code to form a byte value representing the center pixel.

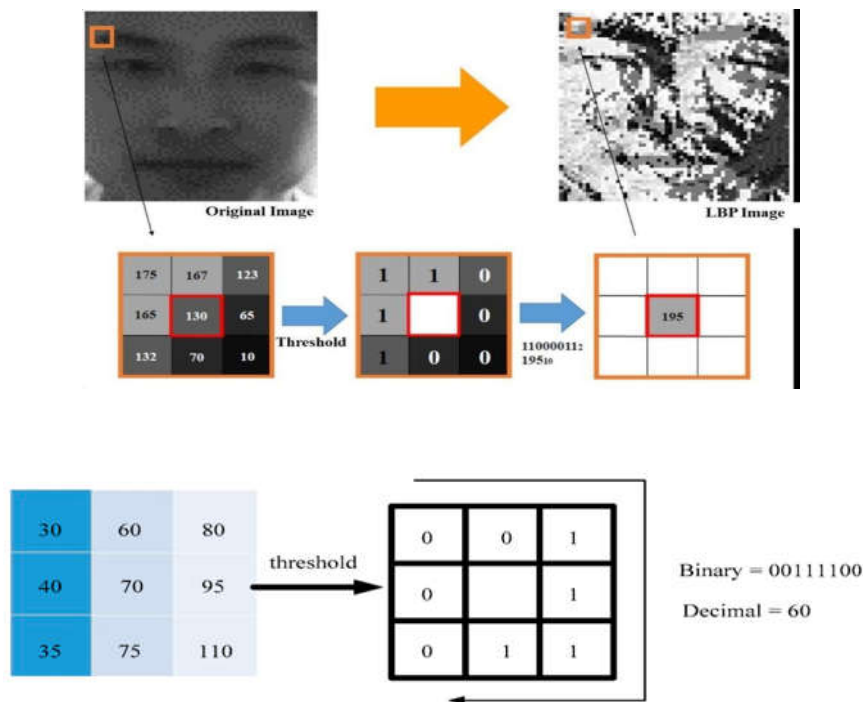


Figure 7: Working of LBP

RESULT ANALYSIS

First we input the id of the student being registered then 60 images are automatically captured as shown below:

Now as shown below the system captures

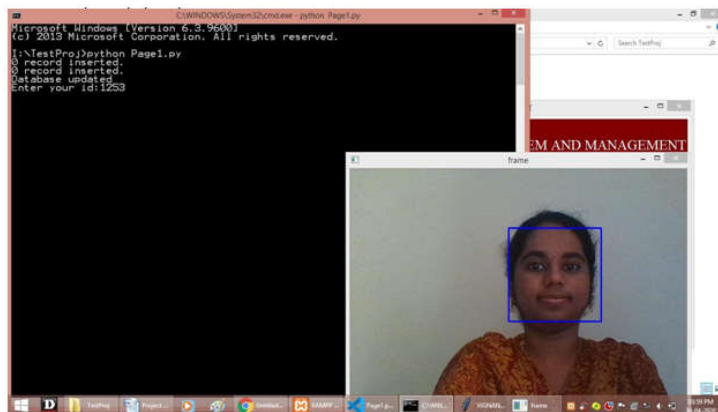
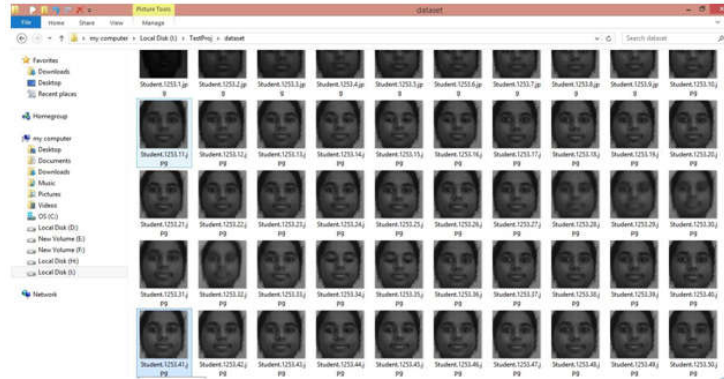


Figure 8: Create dataset

The dataset



Now we trained the model,

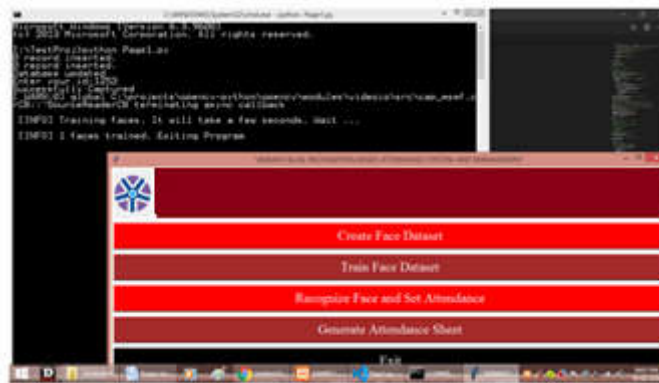
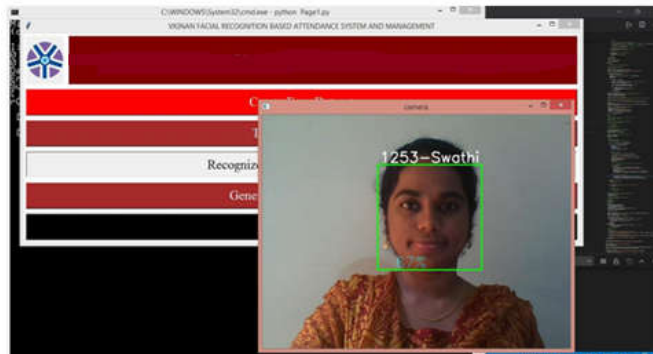
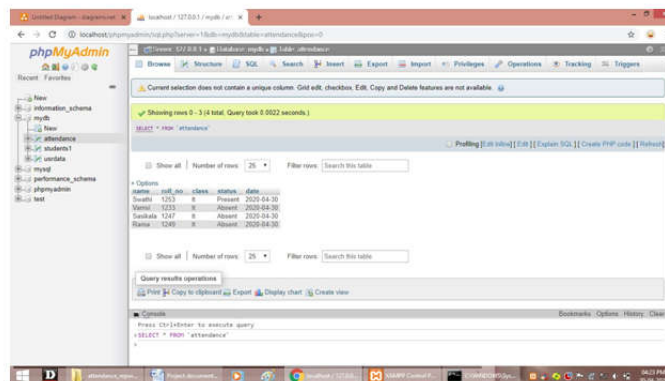


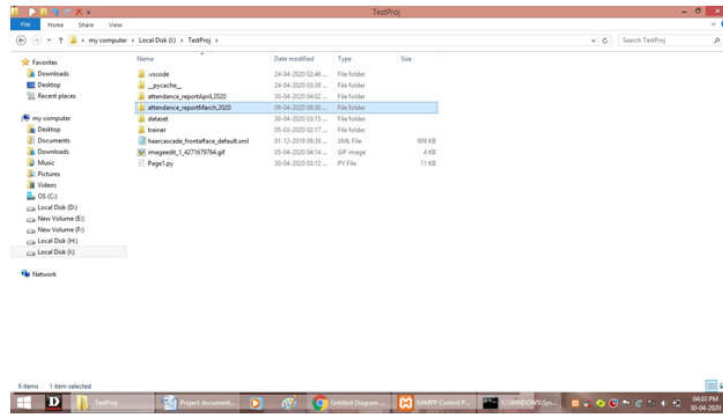
Figure 9: Training the model  
After clicking on Recognize Face and Set attendance the window is as follows:



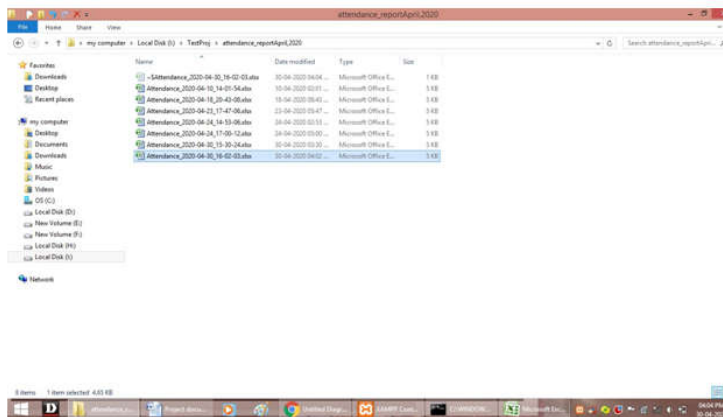
The database also gets updated according to the faces recognized,



A folder named attendance report appended with month name is created and excel sheet is stored in there.

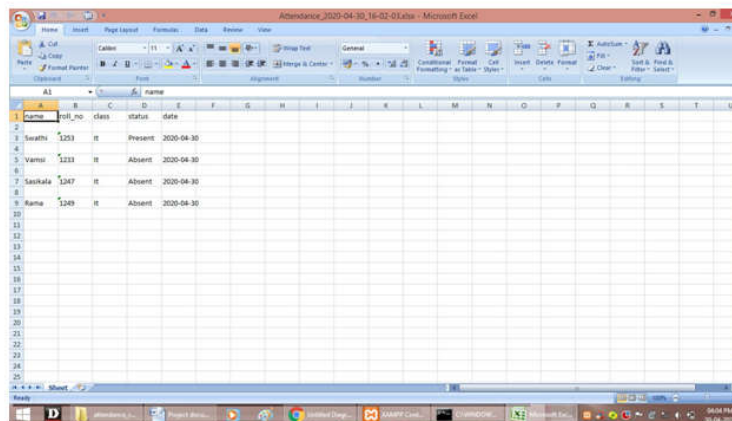


Generation of Attendance sheets



Excel sheets

The excel sheet is as follows:



Attendance report

**CONCLUSION**

The ultimate goal of our project is to automate the attendance marking system using face recognition using LBPH algorithm in python. The input is captured as a set of images for each person and the model is trained to recognize persons and mark attendance if there is id is enrolled in the stored database. Finally generates attendance report in form of excel sheets.

**REFERENCES**

1. M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.
2. T Ahonen, A Hadid and M Pietikäinen, "Face recognition with local binary patterns [M]" in Computer

- visioneccv 2004, Berlin Heidelberg:Springer, pp. 469-481, 2004.
3. Xiang, G., Qiuyu, Z., Hui, W., & Yan, C. (2016). Face recognition based on LBPH and regression of Local Binary features. 2016 International Conference on Audio, Language and Image Processing
  4. T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", Pattern Analysis and Machine Intelligence IEEE Transactions on, vol. 24, no. 7, pp. 971-987, 2002.
  5. Face Detection and Recognition by Haar Cascade Classifier , Eigen Face and LBP Histogram
  6. Aza, V., Indrabayu, & Areni, I. S. (2019). Face Recognition Using Local Binary Pattern Histogram for Visually Impaired People. 2019 International Seminar on Application for Technology of Information and Communication (iSemantic).
  7. T. Ahonen, A. Hadid and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037-2041, Dec. 2006.
  8. S. Singh, G. Mohiuddin and S. Pandit, Face Detection and Recognition by Haar Cascade Classifier Eigen Face and LBP Histogram, pp. 4, Feb. 2019.
  9. A. Satish and N. Devarajan, "Preprocessing technique for face recognition applications under varying illumination conditions", Glob. J. Comput. Sci. Technol. Graph. Vis., vol. 12, pp. 13-18, Jan. 2012.
  10. G. Zhang, X. Huang, S.Z. Li, Y. Wang and X. Wu, "Boosting Local Binary Pattern (LBP)-Based Face Recognition", Proc. Advances in Biometric Person Authentication, pp. 179-186, 2004.
  11. Nagpal, G. S., Singh, G., Singh, J., & Yadav, N. (2018). Facial Detection and Recognition using OpenCV on Raspberry Pi Zero. 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN).
  12. M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, 1991.
  13. T Ahonen, A Hadid and M Pietikäinen, "Face recognition with local binary patterns [M]" in Computer visioneccv 2004, Berlin Heidelberg:Springer, pp. 469-481, 2004.
  14. Xiang, G., Qiuyu, Z., Hui, W., & Yan, C. (2016). Face recognition based on LBPH and regression of Local Binary features. 2016 International Conference on Audio, Language and Image Processing
  15. T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns", Pattern Analysis and Machine Intelligence IEEE Transactions on, vol. 24, no. 7, pp. 971-987, 2002.
  16. Face Detection and Recognition by Haar Cascade Classifier , Eigen Face and LBP Histogram
  17. Aza, V., Indrabayu, & Areni, I. S. (2019). Face Recognition Using Local Binary Pattern Histogram for Visually Impaired People. 2019 International Seminar on Application for Technology of Information and Communication (iSemantic).
  18. T. Ahonen, A. Hadid and M. Pietikainen, "Face Description with Local Binary Patterns: Application to Face Recognition", IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037-2041, Dec. 2006.
  19. S. Singh, G. Mohiuddin and S. Pandit, Face Detection and Recognition by Haar Cascade Classifier Eigen Face and LBP Histogram, pp. 4, Feb. 2019.
  20. A. Satish and N. Devarajan, "Preprocessing technique for face recognition applications under varying illumination conditions", Glob. J. Comput. Sci. Technol. Graph. Vis., vol. 12, pp. 13-18, Jan. 2012.