# PROJECT REPORT

**on**

# "FACE RECOGNITION BASED ATTENDANCE SYSTEM"

**Funded by**
**TEQIP - III**

**Submitted by:**

**Neel Prakash GIN 1932001**

**UNDER THE GUIDANCE OF**

**Mr. Sunil Kumar Bansal**

Asst. Prof., EIE

**PROJECT INCHARGE**

**Submitted to**

**Department of Electrical & Instrumentation Engineering**

SANT LONGOWAL INSTITUTE OF ENGINEERING & TECHNOLOGY

Longowal - 148106, Distt. Sangrur (Punjab), India

August 2020

# DECLARATION

We hereby declare that the Project Report "**FACE RECOGNITION BASED ATTENDANCE SYSTEM**" is an authentic record of our own work for the project sanction under TEQIP - III, SLIET Longowal, Punjab under the guidance of Mr. **Sunil Kumar Bansal (Asst. Prof., EIE)**.

NEEL PRAKASH GIN 1932001

**DATE:______________**

Certified that the above statements made by the student/s are correct to the best of our knowledge and belief.

**Signatures**

**Examined by:**

**Project Incharge**
**(Sunil Kumar Bansal)**

# BONAFIDE CERTIFICATE

This is to certify that the project titled

**FACE RECOGNITION BASED ATTENDANCE SYSTEM**

is a bonafide record of the work done

by

**NEEL PRAKASH (GIN 1932001)**

In partial fulfilment of the requirements of

**TEQUIP - III**

of

**SANT LONGOWAL INSTITUTE OF ENGINEERING AND TECHNOLOGY,**

**LONGOWAL**, for the duration January, 2019-August 2020.

---------------------------------

**SUNIL KUMAR BANSAL**

**Project Viva-voce held on:** ______________________

# ACKNOWLEDEGMENT

First of all, I would like to express my gratitude to **Mr. Sunil Kumar Bansal**, Assistant Professor, Electrical and Instrumentation Engineering department SLIET Longowal, for his guidance and support throughout this project. We are really fortunate to have the opportunity to work with him. we found this guidance to be extremely valuable.

We are thankful to the Head of Department **Dr. Sanjay Marwaha** of Electrical and Instrumentation Engineering department for their encouragement and inspiration for their execution of this project.

We are also thankful to entire faculty and staff of Electrical and Instrumentation Engineering department for the help and moral support which went along the way for the successful completion of this project.

Our greatest thanks are to all who wished me success especially my parents. Above all we render my gratitude to the almighty who bestowed self-confidence, ability and strength in us to complete this work for no letting me down at time of crisis and showing us the silver lining in the dark clouds. We don’t find enough words with which we can express my feeling of thanks to our dear friends for their help, Inspiration in moral support which went a long way in successful competition of the present study.

NEEL PRAKASH (GIN 1932001)

# ABSTRACT

Face recognition is a method of identifying or verifying the identity of an individual using their face.

Face recognition systems use computer algorithms to pick out specific, distinctive details about a person’s face. These details, such as distance between the eyes or shape of the chin, are then converted into a mathematical representation and compared to data on other faces collected in a face recognition database. The data about a particular face is often called a face template and is distinct from a photograph because it’s designed to only include certain details that can be used to distinguish one face from another.

This project is an effort to use LBPH face recognizer to implement the real time face recognition based attendance system. Here once we give the start command, a camera object is initialized and few photographs of the students sitting inside a classroom will be captured. These images are then processed by a series of machine learning algorithms (faces from the photos are cropped and then fed to LBPH face recognizer) and finally a list of matches are revealed. Based upon this list attendance of all of the students present inside a class is marked.

This project can be moved on CNN and deployed on a server for real time applications.

# CONTENTS

# INTRODUCTION

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.

The rise of facial recognition system can be seen from devices like smartphones, video surveillance and modern day robotic applications. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition and fingerprint recognition, it is still widely adopted due to its contactless process.

In this project we tried to implement a facial recognition based attendance system. Here a camera will be used to capture few group photos of the students sitting inside a classroom.

These photos are then processed by a series of Machine Learning models such as caffemodel and LBPH recognizer (Faces of the students are cropped and then tested for a match inside the existing database) and finally a list of all of the students present inside the group photo is revealed.

Now based upon these results attendance of the students are marked into an excel file.

## Brief Description

In this project, we have decided to use the LBPH (local binary pattern histogram) face recognizer. It comes built in with OpenCV.

## OpenCV

OpenCV stands for open source computer vision, is a software library built to provide a common infrastructure for computer vision applications.

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 etc.

OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million.

## LBPH Algorithm

**Local Binary Pattern** (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighbourhood 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, improves the detection performance significantly. Hence in this manner LBPH came into picture.

### How LBPH Works:

Lets see the working of LBPH algorithm in stepwise manner.

### Step – 1: Parameters

The LBPH uses 4 parameters:

1. 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.
2. Neighbours: the number of sample points to build the circular local binary pattern. the more sample points you include, the higher the computational cost. It is usually set to 8.
3. 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.
4. 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.

### Step – 2: Training the Algorithm

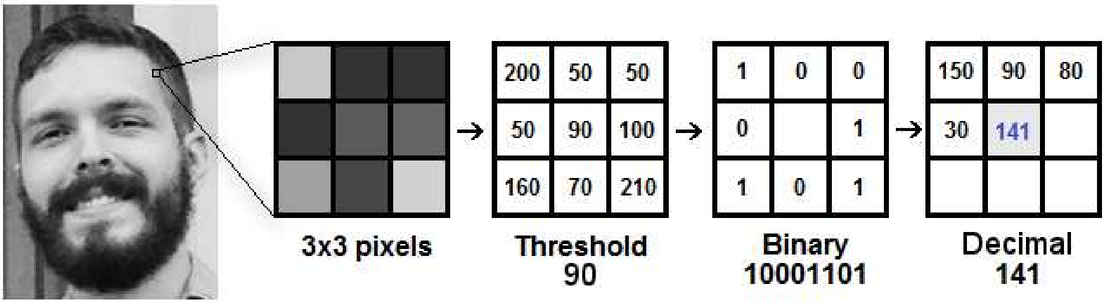
First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize.

We also need to 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.

### Step – 3: 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.

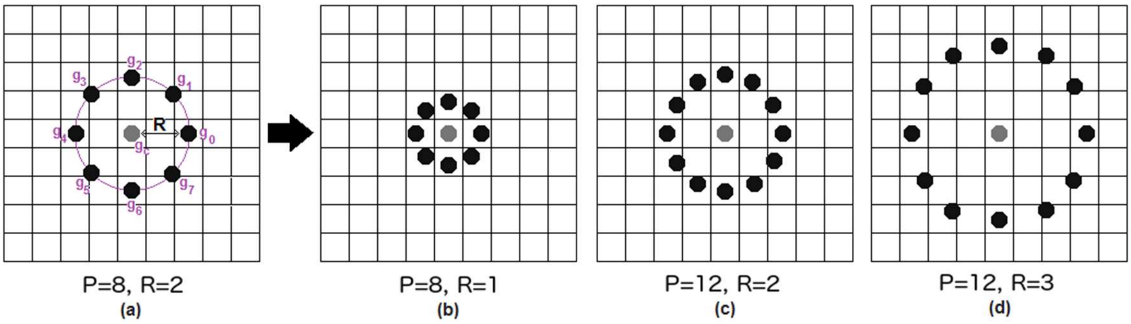
The image below shows this procedure:



Based on the image above, let’s break it into several small steps:

- 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 neighbours.
- For each neighbour 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).
- 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.

**Note**: The LBP procedure was expanded to use a different number of radius and neighbours, it is called Circular LBP.



P=8, R=2
(a)

P=8, R=1
(b)

P=12, R=2
(c)

P=12, R=3
(d)

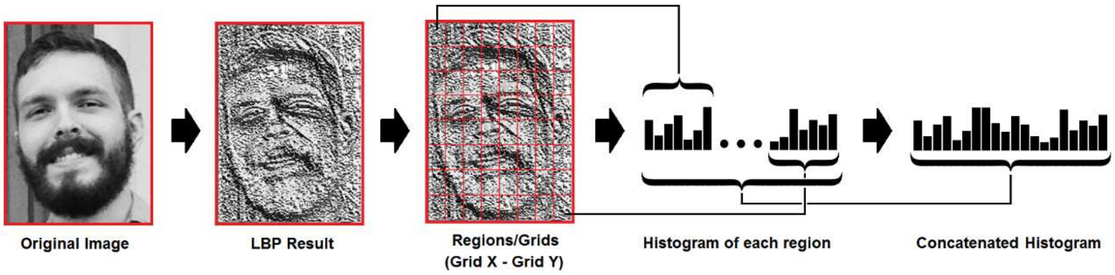
It can be done by using bilinear interpolation. If some data point is between the pixels, it uses the values from the 4 nearest pixels (2x2) to estimate the value of the new data point.

### Step – 4: 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:

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 8x8x256=16,384 positions in the final histogram. The final histogram represents the characteristics of the original image.

The LBPH algorithm is pretty much it.

### Step – 5: 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:

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

- 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 (The lower confidence, better the result is. As 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.

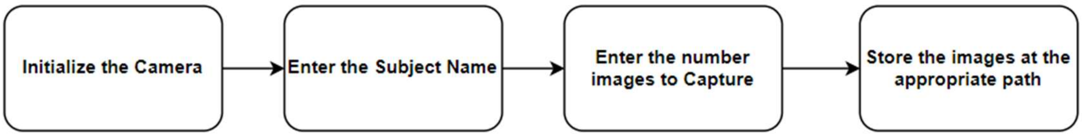
**Methodology**

The entire project working methodology can be divided into two categories:

1. Create database
2. Model Training
3. Face Recognition

## Create Database

Create_database.py script file is used for creating the database. Whenever we run this script file first a camera object is initialized, then it will ask us to Enter the subject Name (Name of the person whose photos has to be captured). If the specified subject name already exist then it will print a message “This subject already exist! Please enter a new subject name.” into the terminal and again ask for the subject name.

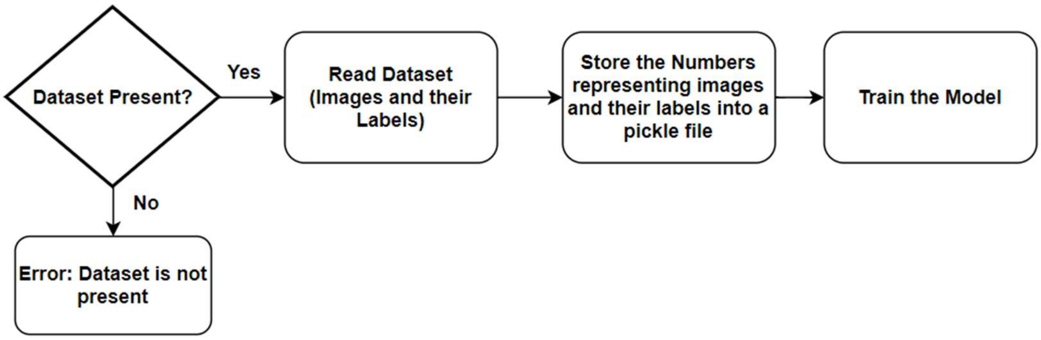

Initialize the Camera
Enter the Subject Name
Enter the number images to Capture
Store the images at the appropriate path

After accepting the Subject Name, Script will ask for the number of images to be captured, once this number is reached camera object will be automatically released.

Now the camera will start capturing the images. These images are first handled by face_extractor.py script file, it will resize and crop the faces for us, after which images are stored at the appropriate file location.

## Model Training

Figure depicts the working of model training phase. Here our model_training.py script will first check whether a dataset is present at the specified location or not. If a dataset is not present it will straight away through an error on us saying that "oop's Dataset is not present! First create the dataset".


Dataset Present?
Yes
Read Dataset (Images and their Labels)
Store the Numbers representing images and their labels into a pickle file
Train the Model
No
Error: Dataset is not present

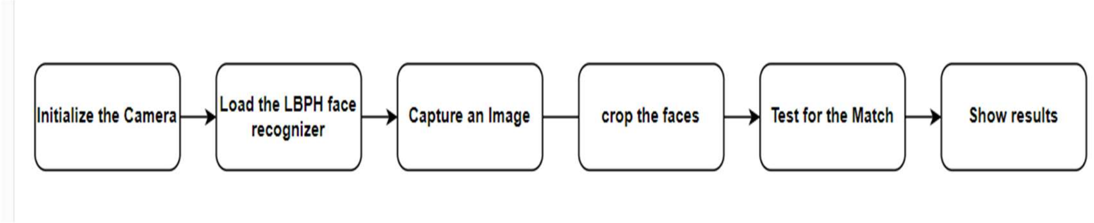
Let's say a dataset is found at the appropriate place. Now our script will read all of the image's present at the specified location along with their corresponding labels (images are by default stored in the compressed format Jpeg or png etc. this compressed data is first decompressed by OpenCV cv2.imread() function and then image files are read) and then store this data into a pickle file.

Finally, our LBPH recognizer is trained (this will take some time based upon the size of the dataset used). Once the training is done a message saying that "Training Completed" will be shown in the terminal.

## Face Recognition

Recognizer.py script file is used for face recognition purposes. Here first a camera is initialized (this camera will be used for capturing the group photo of the students sitting inside the classroom). Then the LBPH face recognizer trained in the previous step is loaded up.

Now a group photo is captured, this photo is first processed by face_extractor.py script file. The work of this script is to crop all of the faces present inside the photo and store them into a python list object.



This image list object is then analysed by our LBPH face recognizer model and based upon some internal computation predictions are made. These prediction results are then used by OpenCV library to circle down all of the faces along with their name onto our original image.

**Conclusions**

With the methodology specified above we are able to achieve 75 percent of accuracy on a dataset of 25 persons containing 250 photographs of each one. Our expectations are that with certain modulation in the dataset (such as image lightening correction and pose transformation) we can achieve nearly 80 percent accuracy in variable environment conditions.

**Future Scope**

Our plans are to shift this project on convolutional neural networks. Specifically, we have decided to use the resnet50 for face extraction and vgg19 for facial recognition purposes. The current experimental studies performed on the cat vs dog and imagenet dataset shows that we can easily achieve more then 90% accuracy under variable environment conditions.

we have also planned to deploy this project on the servers and build an entire website (using react and node.js) to support it.

We are dedicated to develop this project to its fullest so that one day it will become ready for the deployment.

## References

https://towardsdatascience.com/how-to-build-a-face-detection-and-recognition-system-f5c2cdfbeb8c

https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/

https://machinelearningmastery.com/introduction-to-deep-learning-for-face-recognition/

https://en.wikipedia.org/wiki/Facial_recognition_system

https://www.eff.org/pages/face-recognition

https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78

https://ieeexplore.ieee.org/abstract/document/9137850