F-Unlock

Sanmeet Kaur
Abhishek Handa
Abhishek Salwan
Ajay Jindal

Introduction

The project is about smart lock which can be unlocked using Face Recognition. The face is recognised using webcam in real time through web application which is connected to Raspberry Pi. In case of successful authentication, the raspberry Pi commands the relay to unlock the lock & hence the door is opened. There is a depth sensing feature which finds the depth of the face and this prevents the face recognition software not to be tricked using picture on mobiles.

The project also includes website to grant or revoke access to the users. It would graphically display recognized users along with sound alert with the name of the person. Along with this, the interface would provide the user to set a username and password to associate with an account and configure the system. Configuring includes adding an administrator and owners. The administrator has the privileges of adding owners and bypassing the security in case the software doesn't recognises the face. Along with this, the administrator has the rights to provide entry to third person on one time basis or can be the same person as an owner remotely via the mobile application.

The project also consists of mobile application for both iOS and Android platforms which will be synced with the website. Through this user can log in by giving the details set initially. These credentials will be used to bypass the face recognition system in case the owner(administrator) is not recognised. The face recognition software uses transfer learning to incorporate new images to the deep learning model in case the software can't recognise the owner due to change in face features.

This project also includes a feature to grant access rights to third person by the owner (one- time access feature). In this feature the third person when denied access can record voice message and send an access request to the owner via mobile application so that administrator grant real one-time access. This is useful to deal with emergency situations like fire when there is a need for urgent access to the room and the owner is not nearby.

Related Work

Existing Technologies

1. OpenFace

This software consists of free and open source face recognition with deep neural networks. It includes taking pictures using computer vision, face detection, feature extraction and then face recognition.

2. *Depth Sensing Cameras*

   This invention relates to a method and apparatus for sensing a three-dimensional (depth and illuminance) image of a scene. It is based on the inverse-square law relating the incident brightness on an area illuminated by a light point source, to its distance from the point source. In the preferred embodiment of the invention the scene is sequentially illuminated by more than one light point source each at a pre-calibrated location in the reference coordinate system. It is known to use various techniques to produce three- dimensional images, i.e., images containing information about the subject scene in three- dimensions. Three-dimensional imaging is of particular importance in the field of computerized imaging for sensing object shape, especially in artificial intelligence applications such as robotic object sorting, and in product dimensional inspection.

3. *Qualcomm Depth-Sensing Camera Technology Designed for Android Ecosystem*

   This module program is built on the cutting-edge technology behind the Qualcomm Spectra embedded image signal processors (ISP) family.

4. *Project Oxford*

   The Microsoft Project Oxford Face API allows developers to access and integrate the face recognition and detection functionality of Microsoft Project Oxford. Some example API methods include analysing faces, retrieving face information, and managing people and groups of people. Microsoft Project Oxford offers a collection

of APIs and SDKs for developers to add intelligent features to their applications. Detect one or more human faces in an image and get back face rectangles for where in the image the faces are, along with face attributes which contain machine learning-based predictions of facial features. The face attribute features available are: Age, Emotion, Gender, Pose, Smile and Facial

Methodology

The security system uses OpenCV's face recognition implementation. The following are the three steps through which the face recognition module works:-

1. **Data Gathering:** Gather face data (face images in this case) of the persons you want to identify.
2. **Train the Recognizer:** This step includes feeding that face data and respective names of each face to the recognizer for it to learn.
3. **Recognition:** This step includes feeding the new faces of the people and checking whether the face recognizer works or not.

After all these steps the following output image is obtained:

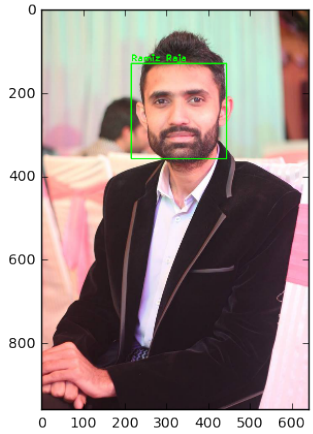

Figure 1 Output Image

For face recognition **LOCAL BINARY PATTERNS HISTOGRAMS (LBPH) FACE RECOGNIZER** is used:

THE LBPH FACE RECOGNIZER PROCESS:

Take a 3×3 window and move it across one image. At each move (each local part of the picture), compare the pixel at the centre, with its surrounding pixels. Denote the neighbours with intensity value *less than or equal to* the centre pixel by *1* and the rest by 0.

After you read these 0/1 values under the 3×3 window in a clockwise order, you will have a binary pattern like *11100011* that is local to a particular area of the picture. When you finish doing this on the whole image, you will have a list of **local binary patterns**.

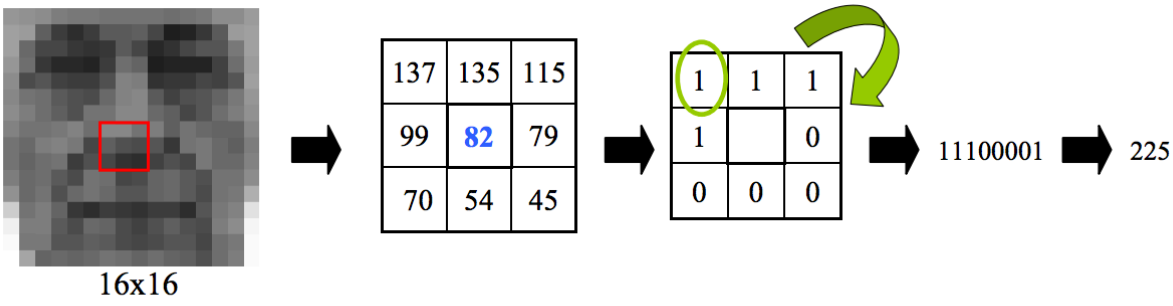

Figure 2 LBP conversion to binary. Local Binary Patterns applied to Face Detection and Recognition

Now, after you get a list of local binary patterns, you convert each one into a decimal number using **binary to decimal conversion** (as shown in above image) and then you make a **histogram** of all of those decimal values. A sample histogram looks like this:

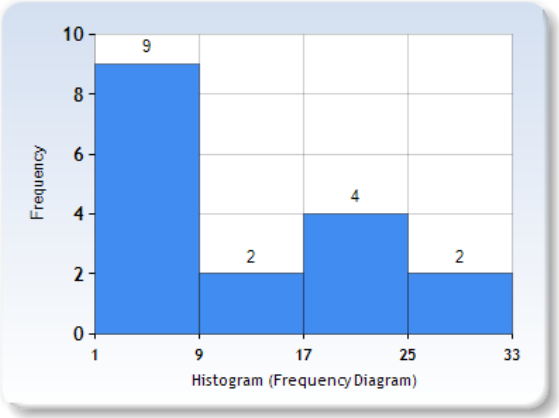

Figure 3 Histogram Sample

In the end, one histogram is obtained for each face in the training data set. That means that if there were 100 images in the training data set then LBPH will extract 100 histograms after training and store them for later recognition. Remember, the **algorithm also keeps track of which histogram belongs to which person**.

Later during recognition, the process is as follows:

1. Feed a new image to the recognizer for face recognition.
2. The recognizer generates a histogram for that new picture.
3. It then compares that histogram with the histograms it already has.
4. Finally, it finds the best match and returns the person label associated with that best match.

Below is a group of faces and their respective local binary patterns images. From the images, it can be seen that **LBP faces are not affected by changes in light conditions**:

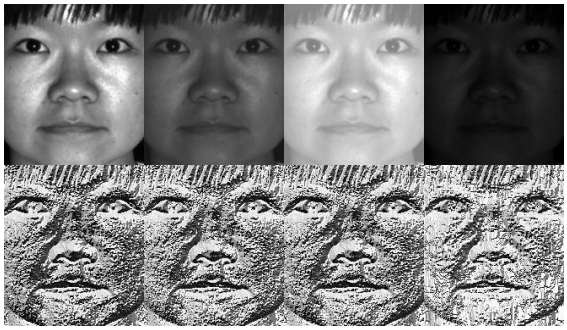

Figure 4 LBPH Face Recognizer Principal Components

Implementation Details

The implementation is divided in three steps:

1. **Prepare Training Data:** Read training images for each person/subject along with their labels, detect faces from each image and assign each detected face an **integer label** of the person it belongs.

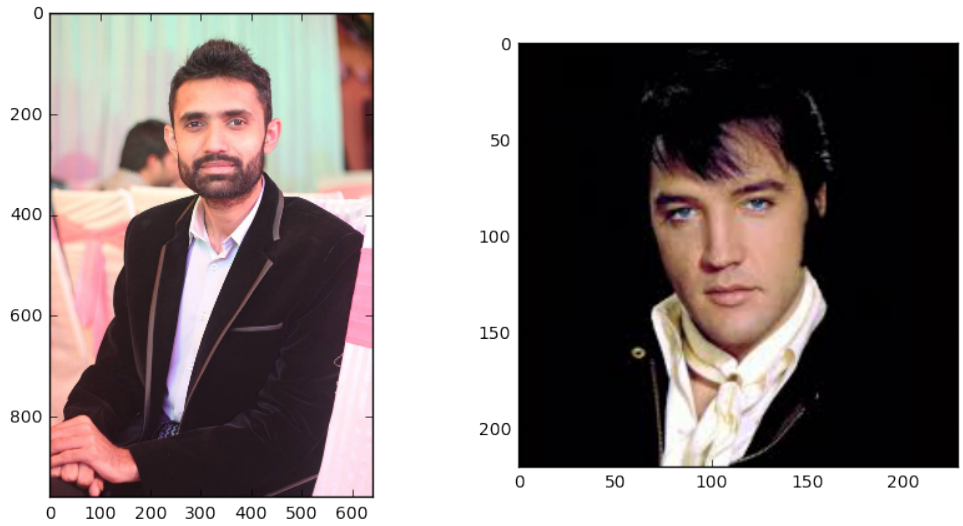

Figure 5 Original Image

2. **Train Face Recognizer:** Train OpenCV's LBPH recognizer by feeding it the data we prepared in step 1.

3. **Prediction:** Introduce some test images to face recognizer and see if it predicts them correctly.

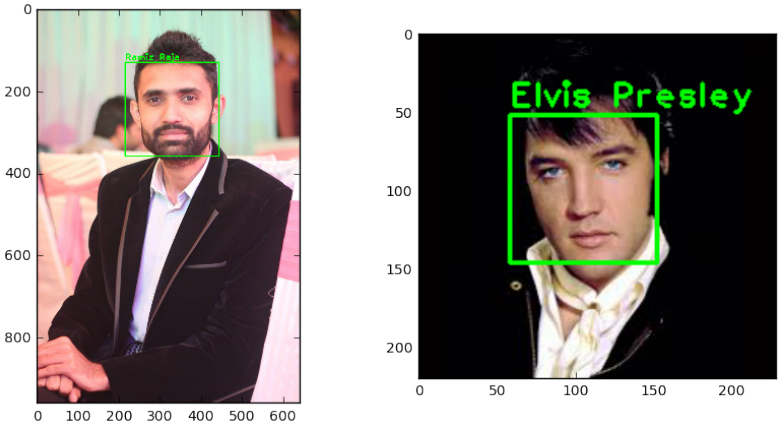

Figure 6 Predicted Image

Results & Discussion

The following results are obtained:

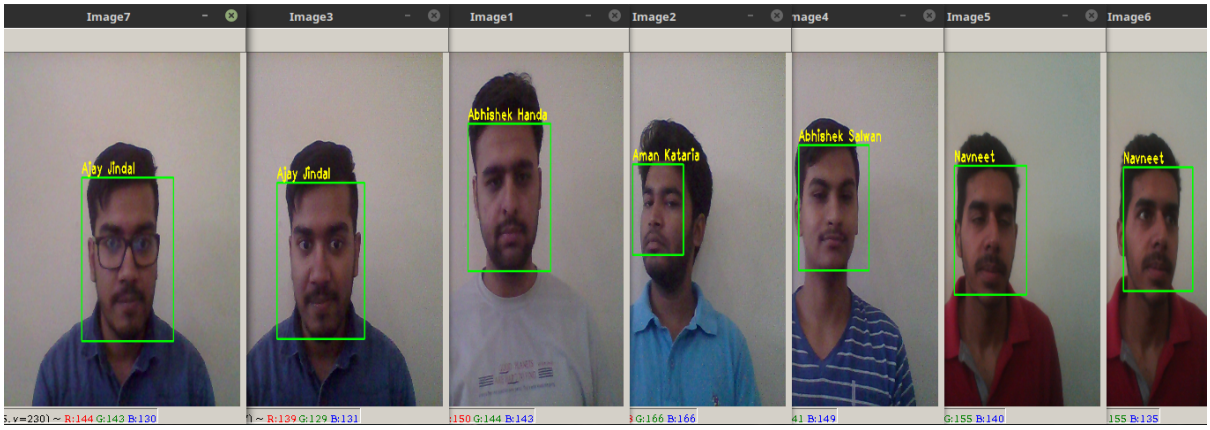

Figure 7 Test Images

For the test images as shown above, the accuracy of the results obtained is 83%. The accuracy can be increased if there are more images in the training data. Also, for further improvements in face recognition module the current approach can be comnbined with **Histogram of Oriented Gradients (HOGs)** and **Neural Networks.** But this approach requires large training dataset for the neural network to be trained and work efficiently.

Conclusion

The security system which is based on face recognition algorithm can be improved if we have considerable data available. For small dataset, OpenCV implementation is preferred. The prediction results can be improved by training the data on Neural Network with good amount of data. To overcome this limitation of collecting huge data, we can use some pre-processing on a single image to produce multiple images, thus adding to the dataset available.