Access Control with Face Recognition using Raspberry Pi

K.D.S.R.L. Surya Varma, K.Sai Kiran Reddy and P. Gunasekar

K.D.S.R.L. SURYA VARMA/R.M.K. Engineering College/ Kavaraipettai, thiruvallur district, pin:601206, India.
(dattasai.surya007@gmail.com)
K.SAI KIRAN REDDY/ R.M.K. Engineering College/ Kavaraipettai, thiruvallur district, pin:601206, India.
(kondakiran006@gmail.com)
P.GUNASEKAR/Asst. Professor R.M.K. Engineering College/Kavaraipettai, thiruvallur district, pin:601206, India.
(pgr.ece@rmkec.ac.in)

ABSTRACT

Face recognition is the identification of human by the unique characteristics of their faces. With increasing needs and with the advancement in technology, extracting information has become much simpler. This paper aims at building an application based system with face recognition using different algorithms and comparing the results. The basic idea is to identify the face and retrieve the information stored in the database. It involves two main steps. First, to identify the distinguishing factors in the image and the second is to compare it with the existing images and displaying results related to the image. The verification and identification can be performed by this method of face recognition. The Raspberry Pi 2 kit with a camera interface is used to load the pictures of individuals in the database. The Haar Cascade feature of OpenCV is used to detect the face in the image and these faces are loaded into a training set. The training is achieved by the LBPH Face Recognizer. Finally, the testing is done to give access to the individuals belonging to the database.

Keywords — Retrieve; Distinguishing; Compare; Haar Cascade; Opencv; LBPH face recognizer

1. INTRODUCTION

The face is our primary focus of attention in the social life, playing an important role in conveying identity and emotions. We can recognize a number of faces at a glance even after years of separation. Despite the large variations in visual stimulus due to the changes in condition, ageing and distractions such as beard, spectacles or changes in hair style, this skill is quite robust. Everyday actions are increasingly being handled electronically instead of pencils and paper or face to face. This growth in electronic transitions has resulted in a greater demand for fast and accurate user identification and authentication. Initially, the challenges were met by the automated surveillance to track the motion of people [1].

Access codes for buildings, bank accounts and computer systems often use PIN's for identification and security clearances. Using the proper PIN gain access, but the user of the PIN is not verified. Thus, face recognition system were developed using Eigen Faces [2].

In the initial automated surveillance [1], the motion of objects is detected. However, this paper aims at detecting the faces of the individuals and recognizing these faces with more effective binary patterns algorithm. The use of raspberry pi reduces the cost of the equipment, low power input and also the software used are open source

The Organization of the paper is as follows. First, the images are stored in the database. These are the images of the individual who can access the system. The input image is taken and this image is to be compared with all the other images in the database. Second, the Haar Cascade feature of OpenCV is initialized. This feature is used to detect the face alone in the images stored in the database [3]. These detected faces are loaded into the training set. Third, the training is done using LBPH Face Recognizer [4]. Thus, the recognizer is trained with the images that are loaded. The input image is then compared with the loaded faces and finally the result is displayed with the determined confidence value. The result displays if the individual can access the system or not.

2. CREATINGTHE DATABASE

A. Installing the software's

Initially, the SD card is loaded with noobs which is an easy operating systems installer for Raspbian. This helps the Raspberry pi to download the necessary software from the raspbian organization. This includes the installation of python also. The interfacing of pi camera is done with the help of CSI(i.e. Camera Serial Interface) on raspberry pi. The camera port is enabled and the images are captured with the camera fixed at a position and also adjusting the brightness of the surrounding. The OpenCV is also installed whose primary aim is to build a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products.

B. Grayscale Images

The images are captured as color images but are converted into gray scale images. This will reduce the time for the execution of the program. The image processing is done using grayscale pixels and it is better to have the images scored in grayscale format in the database. The images are captured using pi camera and it is then converted in grayscale image using OpenCV cvtcolor which converts an image from one image space to another [5]. Thus, the converted grayscale

images are stored in the database. It is necessary to capture the images in the same location and with the same brightness. This can be achieved with the help of flash light. Here, we consider five images for each individual.

The below figure shows the database of six individuals with five images with different expressions for each individuals. We can also consider more individuals to be authorized in the database. Each picture is assigned with a name, so that the individual can be identified. It is necessary to store the pictures with proper extensions also.

Figure 1: Database for Authorized

3. FACE DETECTION USING HAARCASCADE FEATURE

Object Detection using Haar feature-based cascade classifiers is an effective object detection method [3]. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolution kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)]$$

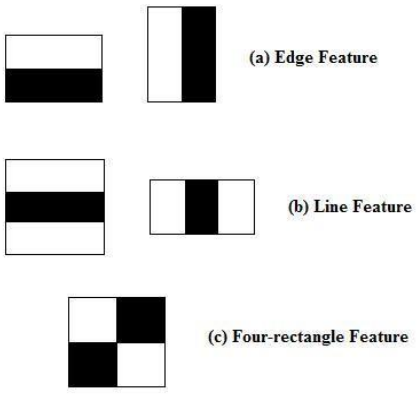

Figure 2: Haar Cascade Features

The algorithm has four stages:

a) Haar Feature Selection

b) Creating an Integral Image

c) AdaBoost Training

d) Cascading Classifiers

a. Haar Feature Selection

All human faces share some similar properties. These regularities may be matched using Haar Features. A few properties common to human faces are as follows

- The eye region is darker than the upper-cheeks.
- The nose bridge region is brighter than the eyes.

Thus, the features matched by this algorithm are then sought in the image of a face.

b. Creating an Integral Image

An image representation called the integral image evaluates rectangular features in constant time, which gives them a considerable speed advantage over more sophisticated alternative features. Because each feature's rectangular area is always adjacent to at least one other rectangle, it follows that any two-rectangle feature can be computed in six array references, any three-rectangle feature in eight, and any four rectangle feature in nine.

The integral image at location (x, y), is the sum of the pixels above and to the left of (x, y), inclusive.

c. AdaBoost Training

AdaBoost refers to a particular method of training a boosted classifier. A boost classifier is a classifier in the (1) where each f_t is a weak learner that takes an object as x input and returns a real valued result indicating the class of the object. The sign of the weak learner output identifies the predicted object class and the absolute value gives the confidence in that classification. Similarly, the T-layer classifier will be positive if the sample is believed to be in the positive class and negative otherwise.

Each weak learner produces an output, hypothesis $h(x_i)$, for each sample in the training set. At each iteration t, a weak learner is selected and assigned a coefficient αt such that the sum training error Et of the resulting t-stage boost classifier is minimized. Label for that face. For example, if the i^{th} index in the list of faces represents the 5^{th} individual in the database, then the corresponding ith location in the list of labels has value equal to 5.

We pass the function with the path of the database directory. This path has to be the absolute path. This functions returns the features (images) and labels (labels) which will be used to train the face recognizer in the next step. Thus, only the faces are loaded into the training set and the other objects which surrounds the face are eliminated with the use of algorithm.

d. Cascading Classifiers

Cascading is a particular case of ensemble learning based on the concatenation of several classifiers, using all information collected from the output from a given classifier as additional information for the next classifier in the cascade. Unlike voting or stacking ensembles which are multi-expert systems, cascading is a multistage one.

For example, a classifier (for example k-means), takes a vector of features (decision variables) and outputs for each possible classification result the probability that the vector belongs to the class. This is usually used to take a decision (classify into the class with highest probability), but cascading classifiers use this output as the input to another model (another stage). This is particularly useful for models that have highly combinatorial or counting rules (for example, class1 if exactly two features are negative, class2 otherwise), which cannot be fitted without looking at all the interaction terms. Having cascading classifiers enables the successive stage to gradually approximate the combinatorial nature of the classification, or to add interaction terms in classification algorithms that cannot express them in one stage.

Here $F_{t-1}(x)$is the boosted classifier that has been built up to the previous stage of training, E(F) is some error function and $f_t(x) = \alpha_t h(x)$ is the weak learner that is being considered for addition to the final classifier.

4. PREPARE THE TRAINING SET

The training set is the faces that the recognizer is trained for. Initially, we create a function to prepare the training set. This function takes the absolute path to the image database as input argument and returns tuple of 2 list, one containing the detected faces and the other containing the corresponding case are the images of faces and the corresponding labels assigned to these faces.

Here, we make use of LBPH Face Recognizer. The LBPH is the Local Binary Patterns Histogram. The algorithm used in this face recognizer is described as follows.

a) Load the Color Image.

b) Convert to grayscale image.

c) Calculate the LBP Mask.

d) Calculate the LBP Histogram and normalize it.

The color image is loaded and is then converted into grayscale image. Since our database consists of grayscale images, the conversion of color image to grayscale image need not be done in the process. The next step will be to calculate the LBP Mask. For each pixel in the grayscale image, a neighborhood is selected around the current pixel and then we calculate the LBP value for the pixel using the neighborhood. After calculating the LBP value of the current pixel, we update the corresponding pixel location in the LBP mask (It is of same height and width as the input image.) with the LBP value calculated as shown below. In the image, we have 8 neighboring pixels.

To calculate the LBP value for a pixel in the grayscale image, we compare the central pixel value with the neighboring pixel values. We can start from any neighboring pixel and then we can transverse either in clockwise or anticlockwise direction but we must use the same order for all the pixels. Since there are 8 neighboring pixels – for each pixel, we will perform 8 comparisons. The results of the comparisons are stored in a 8-bit binary array.

If the current pixel value is greater or equal to the neighboring pixel value, the corresponding bit in the binary array is set to 1 else if the current pixel value is less than the neighboring pixel value, the corresponding bit in the binary array is set to 0.

The whole process is shown in the image above (Figure 3). The current (central) pixel has value 7. We start comparing from the neighboring pixel where the label 0. The value of the neighboring pixel with label 0 is 2. Since it is less than the current pixel value which is 7, we reset the 0th bit location in the 8 bit binary array to 0.

Thus, Viola-Jones algorithm on object detection has extremely fast feature computation and also has efficient feature selection. Such a generic scheme can be trained for detection of other types of objects also.

5. TRAINING USING LBPH FACE RECOGNIZER

We perform the training using the FaceRecognizer. train function [6]. It requires 2 arguments, the features which in this helps

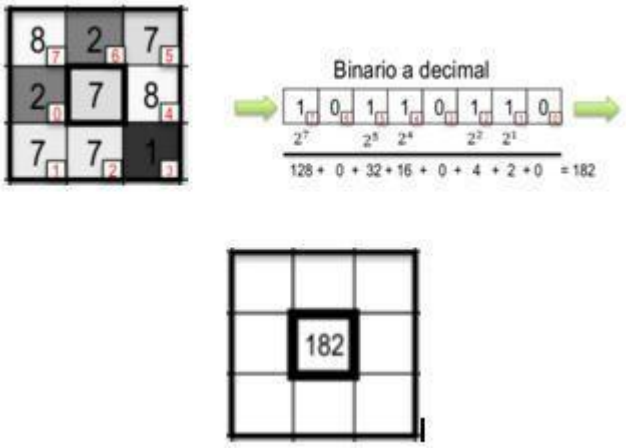

Figure 3: Calculation of LBP Mask

We then iterate in the counter-clockwise direction. The next label location 1 have value 7 which is equal to the current pixel value, so we set the 1st bit location in the 8 bit binary to 1. We then continue to move to the next neighboring pixel until we reach the 8th neighboring pixel. Then the 8-bit binary pattern is converted to a decimal number and the decimal number is then stored in the corresponding pixel location in the LBP mask.

Once we have calculated the LBP Mask, we calculate the LBP histogram. The LBP mask values range from 0 to 255, so our LBP Descriptor will be of size 1x256. We then normalize the LBP histogram.

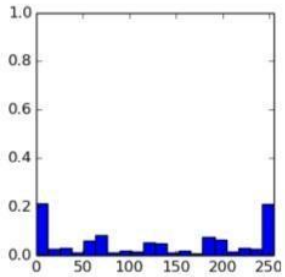

Figure 4: Normalized LBP Histogram

6. TESTING THE FACE RECOGNIZER

The input image from the pi camera should be tested with the other faces in the database. As done in the earlier function, we need to append all the images that has to tested in the path list previously created. Then for each image in the list, we read it in grayscale format and detect faces in it. Once, we have the ROI containing the faces, we pass the ROI to the FaceRecognizer. predict function [6] which will assign it a label and it will also tell us how confident it is about the recognition. The label is an integer that is one of the individual numbers we had assigned to the faces earlier. This label is stored in nbr_predicted. The more the value of confidence variable is, the less the recognizer has confidence in the recognition. A confidence value of 0.0 is a perfect recognition. We check if the recognition is correct by comparing the predicted label, nbr_predicted with the actual label, nbr_actual. The label nbr_actual is extracted using the os module and the string operations from the name of the image. We also display the confidence score for each recognition.

Thus, we set a particular confidence score as threshold. Based on this confidence value, we determine whether the individual is authorized or unauthorized. Here, we have set the confidence value as 60 with respect to the different light conditions and other aspects. When the confidence score exceeds 60, the result is displayed as unauthorized. The entry is enabled by solenoid valve/DC motor which is demonstrated here by LED. We can also have different databases in order to access different systems. In our project, we have considered two database. One database as general authorization and the other database as advanced authorization.

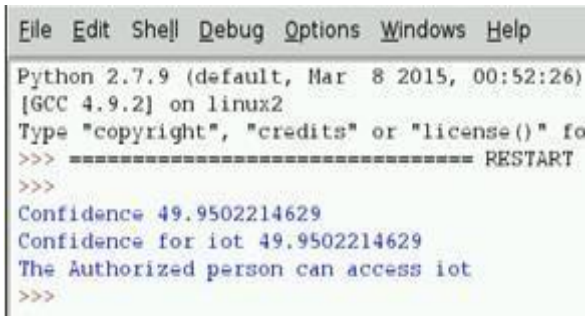

Figure 5: Result for Authorized

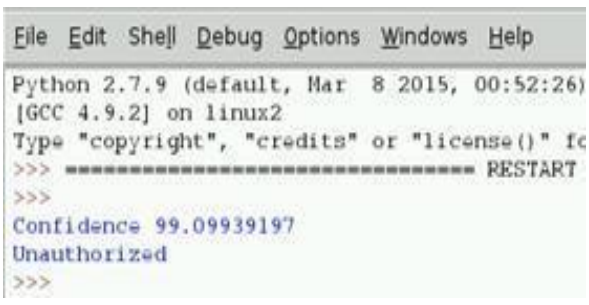

Figure 6: Result for Unauthorized

7. CONCLUSION

In this paper, we have worked towards reducing the cost of the equipment and also the power consumption. Unlike the systems which use more space, raspberry pi 2 is a credit-card sized single-board computer. It works on ARMv7 processor, which supports advanced ubuntu versions and windows 10. 1GB RAM helps the system to operate at greater speed and can be used for more powerful systems. Since the system has more convenience and social acceptability, it can be used for wide range of applications such as access control systems, identification systems, surveillance and pervasive computing.

8.ACKNOWLEGMENT

We would like to thank the Department of Electronics and Communication Engineering, R.M.K. Engineering College for extending library support and to validate our ideas successfully.

REFERENCES

[1] I. Haritaoglu, D. Harwood and L. Davis, "W4: Real time Surveillance of People and Their Activities" IEEE PAMI Vol. 22, No. 8, August 2000.

[2] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigen Faces",IEEE Computer Vision and Pattern Recognition, Pages 586-591, June 1991.

[3] Paul Viola and Michael J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", IEEE CVPR Vol. 1, 2001. [4] C. Shan, S. Gong, and P. W. McOwan, "Robust facial expression recognition using local binary patterns", IEEE International Conference on Image Processing (ICIP), volume 2, Pages 370–373, September 2005.

[5] Bradski, Gary, Kaehler, Adrian, Pisarevsky and Vadim, "LearningBased Computer Vision with Intel's Open Source Computer Vision Library", Intel Technology Journal, Vol. 9 Issue 2, Pages 119-130, May 2005.

[6] J. Howse, "Training Detectors and Recognizers in Python and OpenCV",IEEE International Symposium on Mixed and Augmented Reality, Pages 1-2, September 2014.

