DOI: http://dx.doi.org/10.26483/ijarcs.v9i2.5815

ISSN No. 0976-5697

Volume 9, No. 2, March-April 2018

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

FACE DETECTION AND RECOGNITION USING HAAR CLASSIFIER AND LBP HISTOGRAM

Indrasom Gangopadhyay
Department of Information Technology
Meghnad Saha Institute of Technology
Kolkata, India

Anulekha Chatterjee
Department of Information Technology
Meghnad Saha Institute of Technology
Kolkata, India

Indrajit Das
Department of Information Technology
Meghnad Saha Institute of Technology
Kolkata, India

Abstract: Facial recognition technology is the process for identifying or verifying a face from digital images. The need for face recognition has been of real importance with the development of modern society. Detection and recognition of faces has been on the rise worldwide owing the requirement for security for economic transactions, authorization, national safety and security and other important factors. The technology comprises of face detection, database creation and face recognition. This paper presents a new approach of face identification using LBP method and Haar-like features. The first step is face detection which is done using Haar cascade classifier. After detection, a face is saved in the database. Then the faces from the database are passed through the face recognition algorithm. The Local Binary Pattern Histogram (LBPH) method is used for face recognition. The performance of face detection can be seen to produce maximum error of 1.6%, 2.1% and 0.8% in case of Real-Time video, image file and video file respectively which may be considered accurate. The recognition algorithm produces maximum error of 0.4% which may be considered accurate as well.

Keywords: Face Detection; Haar cascade; Local Binary Pattern Histogram; Face recognition; Haar Features; Integral image

I. INTRODUCTION

Facial recognition technologies have grown fast in the last few years. Effective facial detection and identification is important today in the fields of social development, public safety, national security and others [1]. The two parts of facial recognition technology are face detection and face recognition. The most complicated and hence challenging step of face recognition is face detection which needs to be highly accurate as well as efficient performance-wise. Face detection comprises of gathering of information such as size and position of the face from image or video stream which is noted. Position of the nose, space between the eyes and a few other things are the rules governing the process of face detection. There is a pre-defined manual standard sample which facilitates the process of temple matching [2] and then the search window is used to explore the other possible face areas. The factors like direction, size and rotation of the face can affect this otherwise simple method which we just mentioned. In a video stream or an image, face detection is implemented for finding a human face and its poses. It is one of the most important information processing approaches. Few of the issues which are encountered while performing the most common types of face detection are as follows: Complicated imaging circumstances may cause difficulty to distinguish between foreground and background. Using bigger and better samples while training a good template can be a solution. But, since this method is based on single RGB (Red Green Blue) camera without depth information, it might lead to another problem. The face object on the camera could be a real face or a picture of a face printed on a magazine or a paper [3]. Hence, in this method traditional camera is not valid to ascertain the real face.

The next part is the face recognition process which consists of gathering of information from the face. The main advantages of face recognition and biometric identification process lies in its simplicity. Collection of data can be easily done by a camera which does not need to be of very high quality. Thus the collection procedure is highly convenient, budgeted and equipment is easily available. Other techniques like Iris recognition, fingerprint identification require consent of people whereas in face detection and recognition we do not need to disturb others. Thus we can say this method is not only more direct, user-friendly and convenient but are also popular in day to day life. Based on geometric information, skin color and statistical form the current face recognition algorithm, system can detect the target face under any lighting condition. In case of skin color based detection, background of the image can cause interference with the normal processing and recognition of the faces. Post-processing of the image maybe adversely affected if the face is not correctly extracted from the background image. Skin color has several advantages in comparison with other features as skin color is not affected by posture and size; regardless of whether the face is spinning and the expression changes, skin color is always detectable and also it is not sensitive to direction. This paper proposes a method of face detection using Haar cascade classifier [4] . The classifier is trained with positive images (images with faces) and negative images (images without faces). Then, an image is scanned using the classifier. If Haar-like features are found, then a face is detected in the image. The next step is database creation using a yml file. Though skin color method is used extensively, a non-human face with similar skin color is difficult to differentiate. Therefore, after the use of Haar-like feature for face detection, the local feature recognition, LBP (Local Binary Patterns) [5] method is utilized for face recognition. LBP reflects the details of the face characteristics, focusing on the description of texture features. As we have mentioned before, though face recognition technique is a useful identification tool but this technique is also vulnerable to several kinds of attacks. These afore mentioned features can be used to cross check and give more precise results. Yet, the most demanding work is detection of concealed face in image processing field because of the factors like low resolution of image, illumination changes and movement of abject, etc.

In today’s world, face detection and recognition is widely used in identifying a person and surveillance systems. In face recognition, for example, face detection occupies most of the recognition time and leads to a time constraint of the recognition algorithms. Hence for real-time detection system, the important issue here is speed enhancement and time-complexity reduction.

The remaining part of the paper is organized in the following manner: Section II denotes all the related works conducted in the field. Section III describes the proposed method that we have followed. Section IV is a description of the experiments performed with the experimental results. Section V contains a brief conclusion.

II. LITERATURE SURVEY

Researchers all around the globe have worked on face detection and recognition and have put forward various methods for the same. In the following section we present some existent works along with their performance analysis.

A Hardware Architecture of Face Detection for Human-robot Interaction and its Implementation [6]

Researchers Sang-Seol Lee et al. have developed a hardware based method for human-robot interaction using LBP. The Face Detection is done by first resizing the image followed by generation of LBP for detecting performance and finding particular case of texture spectrum in the image and then using AdaBoost-based 4 cascade classifications to decide FD regions with an associated confidence value.

The hardware core of FD has been used by the LBP-based algorithm with a variety of scale spaces, cascade train method. The main features include support of up to 24 scale spaces with variable scale step size. The minimum face detection can be carried out on 22x24 pixel size. Another feature involves the implementation of shared memory structure and calculation mode. The FD hardware needs a lot of memory to store the gray images. Thus the image dataset needs an external memory. The images are stored in FIFO order. The FD basically uses LBP generation and AdaBoost based Haar Classifier methods to detect faces. The FD hardware IP needs massive memories for the grey image and parameters for confidence calculation. For a 24-bit VGA image its grey image requires a memory size of 2.5 MB and for that external memory is needed.

Face Detection and Authentication using LBP and BSIF [7]

Researchers Naveen S et al. have developed a method for recognizing and authenticating a face to distinguish between a real face and a mask. The process first detects the face region from 2D images and the corresponding depth map is captured with a Kinect Camera which is extracted with cascade object detector. The face is detected using Viola-Jones algorithm for detection and trained datasets for detection. Local and global features from the face are detected using a cascade detector. The global features are utilized for face recognition. Local features are used for identifying a 3D mask from the actual face. Spectral detection from the eye region and calculating the surface area of the nose gives a clue about the presence of a mask. The features obtained from 2D and depth classifier are input to a classifier for classification. A well-known Eucledian distance is used for calculating the distance between the test and reference data. This data is then used for authentication. The procedure followed in this paper uses global features for recognition and local features for authentication. The combination of these two features provides a low HTER of 7.65%.

LBP_T and BSIF_T indicate LBP and BSIF applied on texture map or 2D image respectively. LBP_D and BSIF_D indicate LBP and BSIF applied on depth map respectively. Now, coming to accuracies obtained for different features, LBP_T is 84.90%, LBP_D is 47.84%, BSIF_T is 77.45%, BSIF_D is 86.47% accurate.

Face Detection Based on LBP [8]

Researchers Zheng Jun et al. carried out face detection using LBP. The experiment was carried out on FERET’s database consisting of 11 images each of 15 people. The result of the experiment carried out using LBP was compared with the face detection methods using skin color detection. The former showed better and more accurate results compared to experiment carried out using skin color detection. Using normalized RGB has shown problems in removing luminance as there is luminance information left in remaining components. YUV color space is used by compressing the chroma information to save storage space. Similar to the YUV color model is the YC_bC_r (a hardware oriented) color model, which is widely used in digital video. Here Y represents the luminance, and C_b and C_r represent the blue component and the red component, respectively. In YC_bC_r space, the distribution of C_b and C_r components of facial color tends to be consistent. It can effectively remove the illumination brightness effect of Y, and it also has a good clustering feature. Based on the above characteristics, many researchers have carried out the study of human face detection. This color scheme consists of three channels. It is widely used in the coding process of digital video. The results show detection rate of 79.80% to 99.20% for the method proposed in the paper and a rate of 65.40% to 91.90% for skin color detection method when carried out on FERET database. Carrying the experiment out on YALE-B database, a recognition rate of80.6% to 99.8% was seen for the proposed method and 60.4 to 92.4% for skin color based detection.

Development of face Recognition System Based on PCA and LBP for Intelligent Anti-Theft Doors [9]

Researchers Zhengzheng Liu et al. have devised a method for secure anti-theft doors using facial recognition based on PCA (Principal Component Analysis) and LBP. The experiment performed uses embedded processor S3C6410 platform USB camera to capture the face data. It uses AdaBoost algorithm for detecting and classifying face region in Opencv face database. Then LBP is used to describe the texture of local area which can extract facial feature rapidly. Finally, PCA method is used to reduce the facial feature matrix dimensionality, which reduces the amount of calculation and data quantity and speeds up the recognition process. It has been seen that combining PCA and LBP, the influence of different expressions, angles and sizes can be overcome. The face detection is very fast and has 90% recognition efficiency, although efficiency of identification is affected when changes of illumination are strong.

Real-time and Low-memory Multi-face Detection System Design based on Naive Bayes Classification Using FGPA [10]

Yon-Ping Chen et al. proposes a real-time face detection system based on Naive Bayesian classifier using FGPA. The process is divided into 3 phases namely, Feature extraction, Face detection and false candidate elimination. The image is first downscaled and then from each of the downscaled images, the facial features are extracted. The features then go through a Naive Bayesian classifier to identify candidate faces. Then skin color and face overlapping elimination is done to eliminate false positives. The result of detection is an image that appears on the monitor in VGA format. The face detection system implemented in the paper is done using FGPA. As a result of FGPA, parallel processing, in 640x480 resolutions, it has been seen that the face detection of any image takes place within 16.7 milliseconds. The improved binary features, compared to Harr features, save around 140 times the amount of memory. The experimental results show that the accuracy rate is 95% higher in face detection.

Optimization Detection of Smiling and Opening Eyes in Faces with Algorithm LBP [11]

Islomov Shahboz Zokir ugli et al. have devised a method to detect smiling and opening eyes in faces using LBP. The paper describes a technique for optimizing real time human face detection using LBP algorithm based on colour and edges. In the open eyes and smiling faces, key points are detected in the image using LBP algorithm. The classification between smiling and neutral faces is done by a novel neural architecture that combines fixed, directional filters and adaptive filters in our face. This method has proven to capture more information about image structure than any other traditional algorithm. Also lesser features of the completed feature set is seen to make the training process easier.

The next section provides a description and a summary of the algorithms for face detection and recognition which have been used in this paper.

III. PROPOSED METHOD

The method that has been proposed in this paper consists of 3 parts:

- Face Detection
- Database Creation
- Face Recognition

Face detection

The face detection process is carried out using Haar cascade classifiers. The reason for choosing Haar classifier is because of its speed. It may be considered as a funnel wherein every region of an image is analyzed using a set of classifiers called Haar Features that act as a funnel called the Haar Cascade. The classifiers at the top of the cascade are extremely fast and have extremely low false negative rate that immediately remove regions of an image that do not contain a face. The Haar features become more and more complex further down the cascade as an optimization. The images are rejected as soon as possible if the features do not resemble a face. The Haar detector calculates the integral of a greyscale image. Every pixel of the integral image contains the sum of intensities of every pixel above it and to the left in the original image. This allows the calculation of the average intensity of any rectangular portion of the image to be obtained by accessing only 4 pixel values, rather than hundreds at a time.

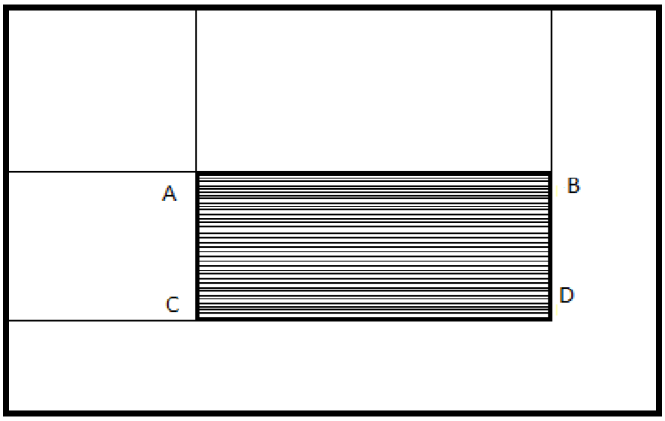

Figure 1. Integral image.

The Haar cascade features are computed using the following Fast-Computing methodology:
Sum=I(C) +I (A)-I (B)-I (D),
where points A, B,C,D belong to the integral image I as shown in Fig. 1. Each Haar-like feature may need more than four lookups, depending on how it was defined.

A Haar-object detector takes advantage of three different types of rectangular features; one at a time: Edges, lines and combinations of four rectangles in order to detect an object. A Haar-object detector finds thousands of rectangular features into regions and defines these rectangular features into regions to define an object and detect them as described in Fig. 2.

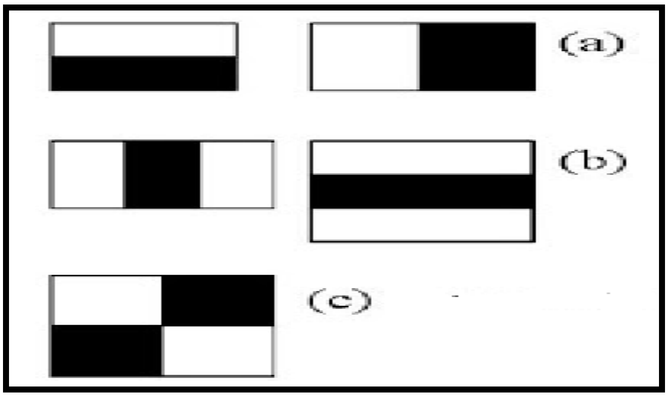

Figure 2. (a) Edge Features (b) Line Features (c) Rectangular Features

Each of these features is recognized extremely quickly in an integral image; thousands of features are verified thousands of times for every single frame of video. The integral image of a frame allows the software to be efficient by effectively calculating the images' intensities in advance.

For the purpose of face detection, we used a .xml file containing a cascade of thousands of rectangular features that are present in the face. The cascade contains features like the cheeks and nose can be brighter than the eye sockets. The forehead is the brightest region of the face. Thousands of these comparisons are conducted in real time to make sure that a single frame of video contains a face.

Database creation

The database creation is the most important part of the process. The Haar cascade face detector is used to detect a face in an image or a video frame. Once a face is detected, the image is saved in a folder. The images saved by the face detector, which are used to create a .yml file by the trainer, are then consulted by the face recognition algorithm to recognize a face from a real time video or an image. The names of the individuals are stored in the database with an id. The corresponding ids are stored with the images of the faces in the file. The images are resized and converted to grayscale images. Conversion into grayscale images makes the intensities a lot more prominent for the following stage to work properly.

Face Recognization

In this paper, we have used Local Binary Patterns Histogram (LBPH) algorithm to perform facial recognition. The LBPH algorithm looks at 9 pixels at a time: a 3x3 pixels (fig. 3) to construct a histogram. The algorithm uses the following parameters:

- **Radius:** It is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.
- **Neighbours:** Sample points required to build the circular local binary pattern. It is usually set to 8.
- **Grid X:** 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:** 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.

LBP is defined as follows (fig. 8):

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p\ s(i_p - i_c);$$

where (x_c , y_c) is the center pixel and its brightness is i_c, and the brightness of adjacent pixels are i_p.

s(.) is a sign function defined as:

$s(x)=1$ if $x \geq 0$ and
$s(x)=0$ otherwise

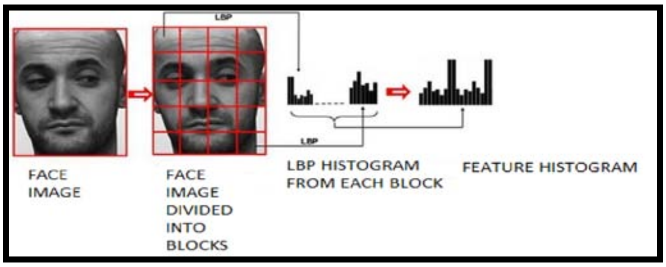

Figure 3. Extraction of feature histogram from a face image.

In case of LBPH, the pixel in the center is compared with all the other pixel values. If the pixel value is greater than the pixel value in the center it is denoted by 1, else it is denoted by 0(fig. 4). The transition from 1's to 0's and 0's to 1's signifies the presence of edges which may be considered as essentially the change of illumination. The algorithm is also illumination invariant in case of non-uniform illumination, i.e., all the pixel values increase by the same amount. Hence, the difference between any pixel value and the pixel value in the center remains the same. The 1 and 0 representation does not need to be altered. One downside to this algorithm is the existence of shadows which can cause non-uniform illumination. However, this problem can be eliminated by choosing smaller pixel matrices.

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 **neighbors**.

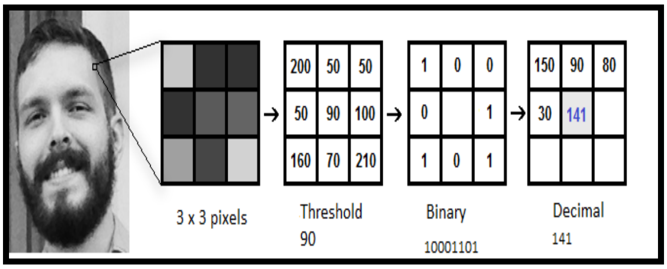

Figure 4. Conversion of grayscale image to decimal.

Using the image generated in the last step, the Grid X and Grid Y parameters may be used to divide the image into multiple grids.

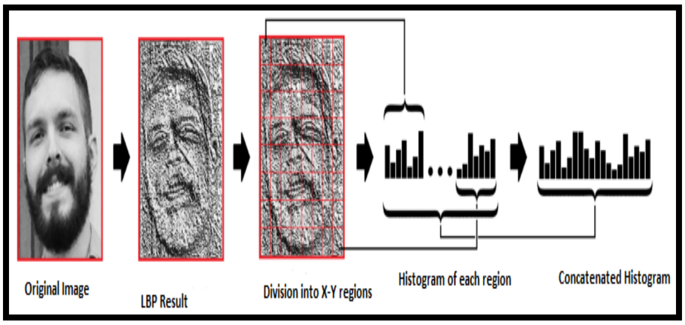

Figure 5. Obtaining Histogram of Image. (leka gulo)

Based on the image above, we can extract the histogram of each region as follows (fig. 5):

- 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.

This principle can be used in image recognition in the following way:

When an image or a video frame is given to the LBPH operator, the LBPH algorithm matches the input face with all the faces in the database using the histogram as a texture descriptor. On finding a match, it produces the identity of the face with the name at the bottom.

In the following section we present the experimental results performed based on the methods proposed in the above section, along with statistics on the performance of each algorithm.

IV. EXPERIMENTAL RESULTS

The experiment was performed using Python in Python IDLE 3.6.4 on Windows 8.1 Operating System with Intel core i3 running at 2.6 GHz. Real time video was captured using the webcam of the computer (CyberLink YouCam 6).

The database that we used was sqlite3. The face detection procedure was first implemented which was done by the help of the face detection algorithm that we have used. The input for faces was given using the following sources: real time video, images and video files.

In the first row (fig. 6), 5 images of a face are presented in different postures. The following row displays that the faces have been detected properly and accurately in those respective postures.

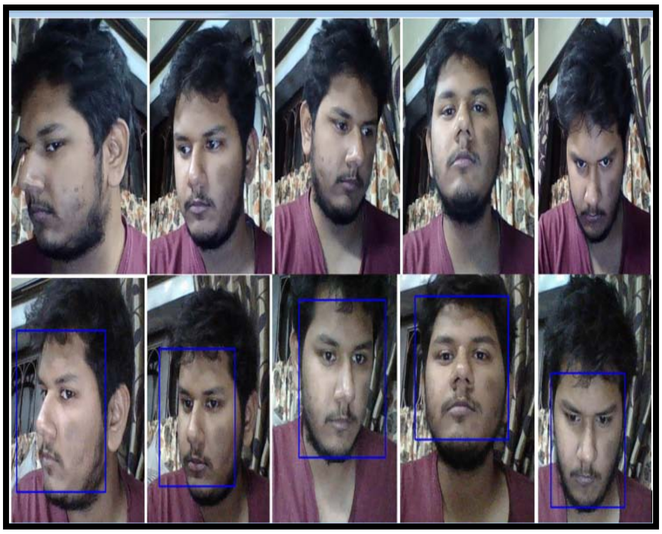

Figure 6. Face Detection in various poses.

The process of face detection was carried out with images from real-time video via webcam, image and video file. The results for face detection performance rate and the time taken by the program for an input from Webcam (Real-Time video) is shown in table I. The same for input image file and video file are shown in table II and III respectively. The performance graphs are shown in fig. 7.The results are as follows:

Table I. Performance for Real Time Video Image

No. of Faces	*Detection Accuracy (%)*	*Execution Time (seconds)*
1	Real Time	0.05
2	100	0.05
3	100	0.05
4	100	0.07
5	100	0.07
6	100	0.08
7	98.7	0.1
8	98.4	0.1
9	99.5	0.12
10	98.5	0.12

Table II. Performance for Image File.

No. of Faces	*Detection Accuracy (%)*	*Execution Time (seconds)*
1	100	0.02
2	100	0.02
3	100	0.02
4	100	0.02
5	100	0.02
6	98.3	0.02
7	98.7	0.02
8	99.2	0.02

9	98	0.02
10	97.9	0.03

Table III. Performance for Video File.

No. of Faces	*Detection Accuracy (%)*	*Execution Time (seconds)*
1	100	0.05
2	100	0.05
3	100	0.06
4	100	0.07
5	100	0.09
6	99.8	0.09
7	99.8	0.1
8	99.5	0.12
9	99.2	0.14
10	99.2	0.15

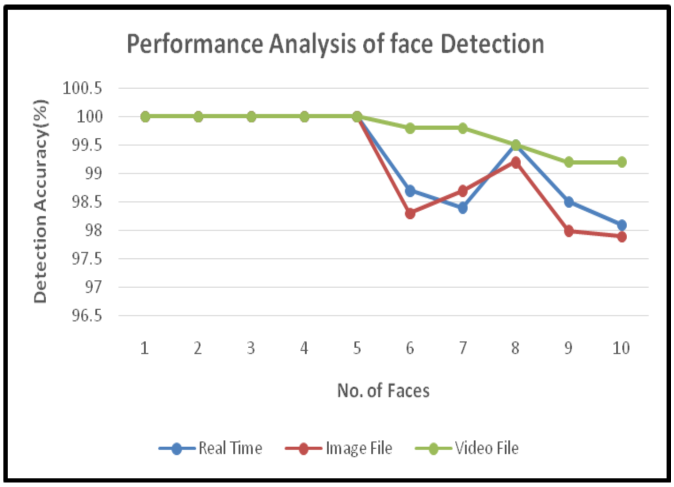

Figure 7. Performance graph for face detection.

The results from various sources show that in video file, a very good and consistent performance is noticed. Real time video produces good results. Image files produce somewhat less consistent and inaccurate results as compared to the other two sources.

The detected face is then cropped and is stored in a file in grayscale as shown in fig. 8.

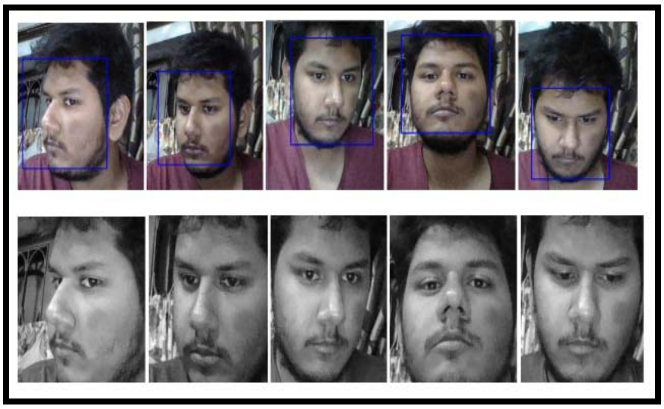

Figure 8. Cropping detected faces.

Now, we proceed to the method of face recognition. The portion of the face is cropped out by the Haar Classifier algorithm for performance reasons. On running the Training algorithm, the images are trained to the algorithm to make it ready for recognition.

After the training is complete, the LBPH algorithm is used to detect the face and to recognize it. The results are shown in Table IV.

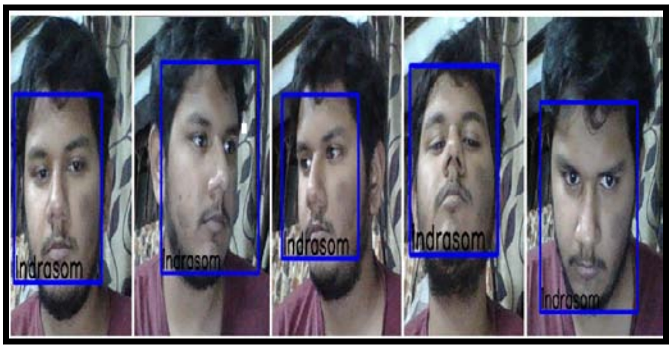

Figure 9. Recognized Faces.

Fig. 9 shows that the images show that the faces are being recognized from various angles.

Table IV. Performance of Facial Recognition

No. of Samples	*Recognition Accuracy (%)*	*Execution Time (seconds)*
20	100	0.35
40	100	0.42
60	100	0.5
80	100	0.52
100	100	0.53
120	100	0.53
140	99.9	0.55
160	99.7	0.56
180	99.6	0.57
200	99.6	0.57

The performance graph for face recognition is shown in fig. 10.

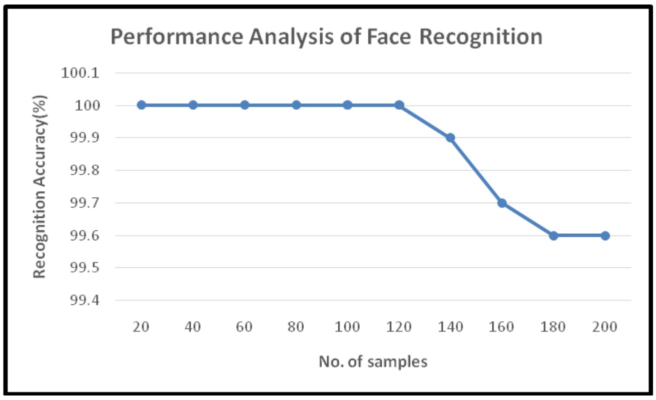

Figure 10. Performance graph for face recognition.

It can be seen that with increase in sample size, the recognition rate decreases somewhat. However, not much change in detection rate is noticed.

The following section provides a brief conclusion to our paper.

V. CONCLUSION

A face detection and recognition method is discussed and implemented in this paper. To detect the faces, Haar classifier has been used which produces maximum error rate of 1.6%, 2.1% and 0.8% in case of Real-Time video, image file and video file respectively which may be considered accurate. For face recognition, LBPH algorithm has been used which has a very high recognition rate and maximum error of 0.4% which shows that the algorithm is efficient.

VI. REFERENCES

[1] Z.Wang and F.Gao, “An Embedded Parallel Face Detection System Based on Multicore Processor”, IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2017, pp. 2684-2688, ISBN: 978-1-4673-8979-2.

[2] P.Ji, Y. Kim, Y. Yang and Y.S.Kim, “Face Occlusion Detection Using Skin Color Ratio and LBP Features for Intelligent Video Surveillance Systems”, Federated Conference on Computer Science and Information Systems (FedCSIS), 2016, pp. 253–259, ISBN: 978-8-3608-1090-3.

.

[3] T.Li, W.Hou, F.Lyu, Y.Lei and C.Xiao, “Face Detection based-on Depth Information Using HOG-LBP”, 6th International Conference on Instrumentation & Measurement, Computer, Communication and Control, 2016, pp.779-784, ISBN: 978-1-5090-1195-7.

[4] B.V.Thiyagarajan, A.Mayur, B.Ravina, and G.Akilesh, “LBP-Haar Multi-Feature Pedestrian Detection for Auto-Braking and Steering Control System”, International Conference on Computational Intelligence and Communication Networks (CICN), 2015, pp. 1527-1531, ISBN: 978-1-5090-0077-7.

[5] S.A.A.M Faudzi and N.Yahya, “Evaluation of LBP – Based Face Recognition Techniques”, International Conference on Intelligent and Advanced Systems (ICIAS), 2014, pp. 1-6, ISBN: 978-1-4799-4653-2.

[6] S.Lee, S.Jang, J.Kim and B.Choi, “A Hardware Architecture of Face Detection for Human-robot Interaction and its Implementation”, IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), 2016, pp. 1 -2 ISBN: 978-1-5090-2743-9.

[7] S. Naveen, R.S. Fathima and R.S. Moni, “Face Detection and Authentication using LBP and BSIF mask detection and Elimination”, International Conference on Communication Systems and Networks (ComNet), International Conference, Thiruvananthapuram, India, 2016, pp.99-102, ISBN: 978-1-5090-3349-2.

[8] Z.Jun, H. Jizhao, T.Zhenglan and W. Feng, “Face detection based on LBP”, 2017 IEEE 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), 2017, pp. 421-425, ISBN: 978-1-5090-5035-2.

[9] Z.Liu, L.Lv and Y Wu, “Development of face Recognition System Based on PCA and LBP for Intelligent Anti-Theft Doors”, 2nd IEEE International Conference Computer and Communications (ICCC), 2016, pp. 341-346, ISBN: 978-1-4673-9026-2.

[10] Y.P. Chen, C.H. Liu , K. Y. Chou and S.Y. Wang, “Real-time and Low-memory Multi-face Detection System Design based on Naive Bayes Classifier using FPGA”, International Automatic Control Conference (CACS), 2016, pp. 7-12, ISBN: 978-1-5090-4109-1.

[11] I.S.Z.Ugli and B.M.M. Ugli, “ Optimization detection on smiling and opening eyes in faces with algorithm LBP”, International Conference on Information Science and Communications Technologies (ICISCT),2016, pp. 1-4, 978-1-5090-3546-5.