

Face Detection and Recognition from Real Time Video or Recoded Video using Haar Features with Viola Jones Algorithm and Eigenface Approach with PCA.

Md. Mohibullah^{a*}, Chowdhury Shahriar Muzammel^b, Khalil Ahammad^c, MD. Sakhawat Hosen^d, MD. Rakibul Islam^e

^{a*}Corresponding author's e-mail: mohib.cse.bd@gmail.com

^aAssistant Professor, Department of Computer Science & Engineering, Comilla University, Cumilla-3506, Bangladesh

^bAssistant Professor, Department of Computer Science & Engineering, Comilla University, Cumilla-3506, Bangladesh

^cAssistant Professor, Department of Computer Science & Engineering, Comilla University, Cumilla-3506, Bangladesh

^dStudent, Department of Computer Science & Engineering, Comilla University, Cumilla-3506, Bangladesh

^eStudent, Department of Computer Science & Engineering, Comilla University, Cumilla-3506, Bangladesh

Abstract

Looking at somebody's face is one of the easiest methods to figure out who they are. Face detection, which occurs incredibly rapidly in humans except when the object is positioned at a short distance away, and introduction, which recognizes a face as an individual by using a person's characteristics to identify who they are, are the two main phases of the human face recognition procedure. In our proposed research, we have used Haar Features with Viola-Jones algorithm for face detection and Eigenface approach with Principal Component Analysis (PCA) for face recognition. The major objective of this proposed research method is to analyze faces to detect and also recognize the unknown faces from real-time webcam or recorded video. If our method successfully detects and recognizes unknown faces with competent accuracy, we will be able to prevent future crimes through precautionary measures.

Keywords: Face Detection, Face Recognition, Eigen Face, PCA, Haar Feature, Viola Jones.

1. Introduction

Computers are becoming smarter as processing capacity expands at a rapid rate. Using webcams, many research papers and projects have demonstrated a computer's ability to engage with humans naturally. Face detection is one of the methods for identifying people in photos. Strange expressions, facial occlusion, illuminations, intricate backdrops, too many faces in the image, skin tone, and other aspects make face detection challenging. A computer's ability to overcome these obstacles to detect a face with increased accuracy is difficult. Face recognition is usually approached in one of two ways: geometric (feature-based) or photometric (view-based). The Eigenface approach and the Fisherface method are two types of methodologies that are currently prominent in developed face recognition patterns. The Eigenface process is based on utilizing Principal Component Analysis to minimize the face dimensional space for facial characteristics (PCA). The major goal of utilizing PCA on face recognition with Eigen's faces was to locate the eigenvector corresponding to the face image's biggest eigenvalue (face space). Many alternative algorithms were developed as researcher interest in face recognition grew, three of which have been thoroughly investigated in the face recognition literature. Principal Component Analysis (PCA) using Eigenfaces, Linear Discriminate Analysis, with Elastic Bunch Graph Matching using the Fisherface approach are all popular identification algorithms. Recognition algorithms can be divided into two main approaches:

1. Geometric: Based on the spatial structure of facial characteristics or the geometrical link between facial landmarks. That is, the key geometrical characteristics of the face, such as the eyes, nose, and mouth, are located first, and then faces are categorized by using geometrical distances and angles between features.
2. Photometric Stereo: Used to reconstruct an object's shape from a series of photos collected under various lighting conditions. The shape of the recovered item is defined by a gradient map composed of an array of surface normals.

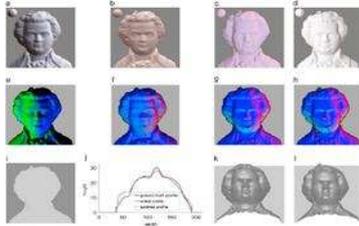


Figure 1: Photometric stereo image.

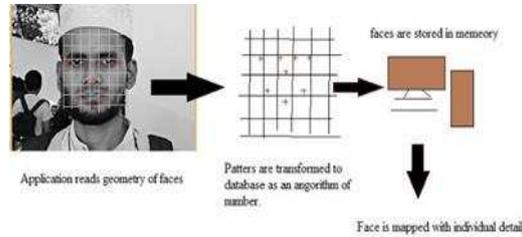


Figure 2: Geometric facial recognition.

Face Detection: Face detection involves separating image windows into two classes; one containing faces (turning the background (clutter)). It's challenging because, while there are some similarities across faces, they might differ significantly in age, skin color, and facial expression. Various lighting situations, image quality, and geometrics, as well as the potential for partial blockage and concealment, all add to the complexity of the problem. As a result, detecting the presence of any face would be possible with a face detector against any background, in any lighting condition. There are two major steps for the face detection process. The first phase is a classification task, which take any images as inputs then it will generate a binary value of yes or no, which will indicate if the image contains any face of human. The face localization process is the second step, which aim is to take images as input and it will produce output the location of any face within that image as some bounding box with some bound (x, y, width's co-ordinate, height's co-ordinate). The method of detecting faces can be broken down into the following steps:

1. Classification: Neural networks are trained on these samples to distinguish the image either faces or nonfaces. We employ both our neural network implementation process and Python OpenCV module in this task. Different network settings are tested to improve the results.
2. Pre-Processing: To reduce the diversity in the faces, the photos will be processed before being sent into the system. All good examples, which are face images, are obtained by cropping frontal facial images to just reveal the front perspective. After that, all of the cropped photographs are adjusted for lighting using normal procedures.
3. Localization: After that, the trained neural network is utilized to look for faces in an image and locate them in a bounding box if they are there. The work on Position Scale Orientation Illumination has been done on several features of the face.

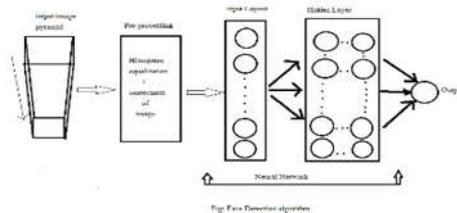


Figure 3: General Face Detection Model

2. Literature Review

There's been research for face detection and recognition system using surveillance data. But there are very few studies about criminal detection and recognition using surveillance data. Using Python and the OpenCV module, Chhoriya, P. [1] suggested a face detection and recognition system for criminal identification. There are four steps in the proposed system: (1) Image training (2) Haar cascade classifier face detection (3) comparison of taught images with those taken from the surveillance camera (4) comparison result.

Face detection is a computer system that allocates human faces in digital photos using learning algorithms. The Viola-Jones algorithm, the first-ever real-time face identification system, has been decrypted by Wang, Y. Q. [2]. The integral image for feature computation, Adaboost for feature selection, and an attentional cascade for effective computational resource allocation are three ingredients that work together to enable a quick and accurate detection. They present a detailed algorithmic explanation, a learning code, and a learnt face detector that can be used on any color image. Due to the fact that the Viola-Jones algorithm frequently produces many detections, a post-processing step is proposed to reduce detection redundancy using a robustness argument.

Face detection, feature extraction, and face recognition are all processes in a general face recognition system. Face detection and recognition consist of several interconnected components, each of which complements the others. Each part can work independently depending on the regular system. Padaruth, S., Indiawarsingh, F. & Bhugun, N. [3] creates a web-based application for face detection against a real-time background, which is then utilized to track staff attendance. Face pattern recognition is accomplished by PCA, while face detection is accomplished through the Haar Cascade approach. Face recognition accuracy is 68 percent with this method, according to tests. It's also employed in security systems. An access control system is built automatically. For detection, it employs the Viola-Jones method, and for face recognition, it employs PCA.

Identification of human faces is extremely useful [4] [5]. An intruder warning system for home security is being developed by Tharanga, J.G.R., Samarakoon, S.M.C. & Karunarathne, T.A.P. [6] and Vaidya, B., Patel, A., Panchal, A., Mehta, R., Mehta, K., & Vaghasiya, P. [7]. The system will recognize faces using the Eigenface algorithm if excessive movement is detected. It will send a warning and photos to the house owner's mailbox if it is unable to recognize faces. The accuracy of this facial recognition test system is 62%. Student attendance can also be taken using Face Recognition [8]. Facial ageing, attractiveness, and distinctiveness is useful to detect and recognize faces [9]. Face detection and recognition can also be done R-CNN [10] and using Raspberry Pi in recent years [11] [12].

3. Aim and Overview of the System

This project aims to develop such a system by which face will be detected with image details from a database with more accuracy. As better output even from a noisy image, the face will be detected with more accuracy. It will be able to recognize a face from a live video stream. It will help to recognize criminals to prevent crime. Adding this system in CCTV so that it can help our daily life. The proposed system will work as follows:

The system will capture video frames from the webcam. Those video frames are taken as still images. These images are converted from BGR to HSV image format. Different color objects on hand are detected by

masking. After that morphological operation is performed to remove noise and detected faces are figured out by a rectangle. The system will perform perfect recognition even in the movement of the head. The system will compare captured images details with the database's images details and will find out details of images.

4. Materials and Methods:

We are proposing a method of detecting a face from a real-time webcam or recorded video by adding a name with image details from the database. Our proposed system will work as the following flowchart:

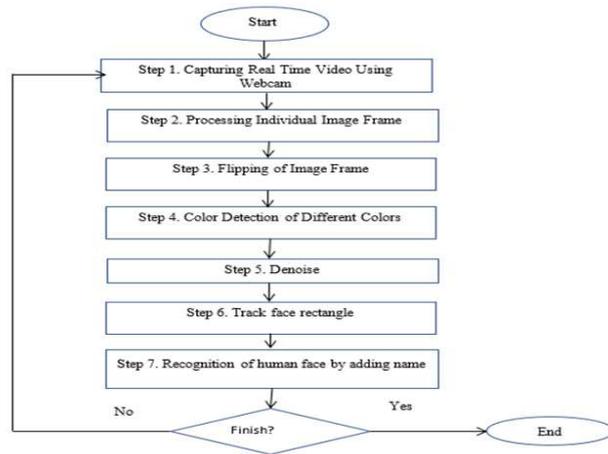


Figure 4: Flowchart of the system

4.1 Capturing Real-Time Video Using Webcam

Real-time video will be captured by using a webcam or from recorded video. By using a smartphone or a camera, we can simply capture or record live streaming video. Here we have used the HP laptop's camera for capturing the image.

4.2 Processing Individual Image Frame

After capturing real-time video using a webcam or from the recorded video we need to process individual image frames. If we need to flip the images we will see it in the next step (Flipping of Image).

4.3 Flipping of Images

When we use a camera to collect video frames, they are inverted, which means that if we move the color pointer to the left, the image of the pointer moves to the right, and vice versa. When we stand in front of a mirror, it's akin to an image (The left side is detected as right and the right side is detected as left). To avoid this issue, we must vertically flip the video frame's images. To accomplish the flipping operation, OpenCV has a built-in function. To perform the flipping operations, we used the `cv2.flip` function.

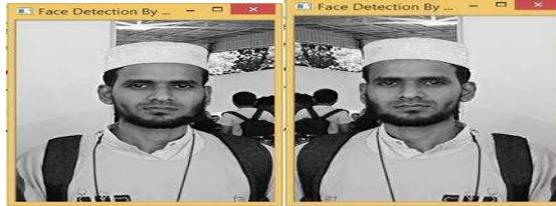


Figure 5: Flipping of image.

4.4 Detect an object of different colors

There are many ways of detecting colors from an image. We used a simple method to detect color pointers made with paper on a fingertip. A detailed explanation is given in the following sections.

HSV Color Space: Color can be represented in a variety of ways on a computer. We will just be concerned with BGR (Blue, Green, and Red) and HSV in our project (Hue Saturation Value). Throughout BGR, a pixel is represented by three characteristics: (i) blue, (ii) green, and (iii) red. A pixel in HSV is also represented by three parameters, but these are Hue, Saturation, and Value. The primary color is not used to describe a pixel in HSV, contrary BGR. Instead, it employs a hue, which is the pixel's color or shade. A pixel in the HSV color system is also represented by three parameters, but these are Hue, Saturation, and Value. The primary color is not used to describe a pixel in HSV, contrary BGR. Instead, it employs a hue, which is the pixel's shade or color. The saturation of a color determined the value of its intensity. A saturation of 0 represents white, whereas a saturation of 255 represents maximum intensity. Other thing to think about this is to consider saturation as a pixel's color intensity. A color's value refers to how bright or dark it is. As seen in the diagram below, HSV can be visualized as a three-dimensional cylinder.

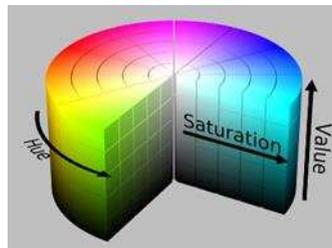


Figure 6: HSV color cylinder

RGB to HSV conversion formula: RGB color space can be converted to in HSV color space by following calculation. The R,G,B values are divided by 255 to change the range from 0..255 to 0..1:

$$R' = R/255$$

$$G' = G/255$$

$$B' = B/255$$

$$C_{\max} = \max(R', G', B')$$

$$C_{\min} = \min(R', G', B')$$

$$\Delta = C_{\max} - C_{\min}$$

$$H = \begin{cases} 0^\circ, & \Delta = 0 \\ 60^\circ \times \left(\frac{G' - B'}{\Delta} \bmod 6 \right), & C_{\max} = R' \\ 60^\circ \times \left(\frac{B' - R'}{\Delta} + 2 \right), & C_{\max} = G' \\ 60^\circ \times \left(\frac{R' - G'}{\Delta} + 4 \right), & C_{\max} = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0, & C_{\max} = 0 \\ \frac{\Delta}{C_{\max}}, & C_{\max} \neq 0 \end{cases}$$

Value calculation:

$$V = C_{\max}$$

In given Formula Hue values is 0 to 359, Saturation values are 0 to 100, and Value's range is 0 to 100, but in OpenCV, they are 0 to 179, 0 to 255, and 0 to 255 respectively. So they have to convert to map with OpenCV [14].

Segmentation: To detect face features first we have to convert the BGR color image frame to HSV format because HSV is more suitable than BGR in discriminating color objects and thus detecting them [13]. Once we get the HSV values of color then we can define an upper and lower range for that respective color. Using these range values, we can detect if color is presented in the image. If the respective color in the range is present in the image then the pixel is set to 0 and if not in the ranging pixel is set to 255. Thus, we get the binary image of the colored paper.

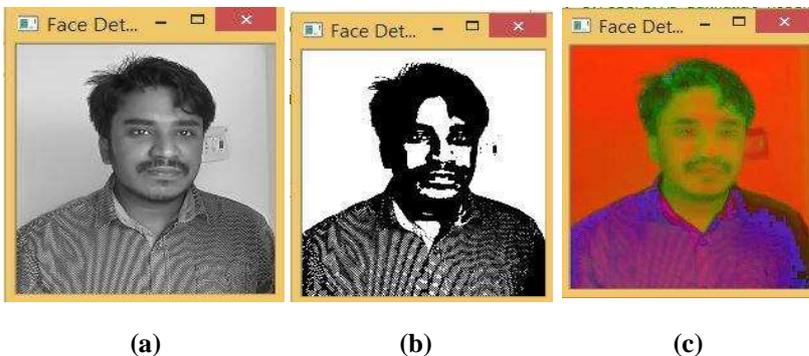


Figure 7: Binary image conversion. (a) (B/W) image. (b) HSV image. (c) Binary image.

4.5 Remove noise to have good colored pointers

We can't get a fair estimation of the color objects on the finger of the hand image after converting the original image to a binary image because of background noise. To acquire a better estimate of the hand, we

need to eliminate noisy pixels from the image. To remove noise, we employ an image morphology technique that performs picture erosion and dilation [14]. Dilation increases the amount of image pixels that aren't eroded, whereas Erosion reduces the amount of image pixels that aren't eroded.

Mathematically, Erosion is given by,

$$A \ominus B = \{x \mid (B)_x \cap A^c = \emptyset\}$$

The input image is denoted by A, and the structure elements are denoted by B. A Sliding window is used to operate the Structure element on the Image, and exact similarities are indicated. Figure 8 shows a graphical representation of the method. Dilation is defined by

$$A \oplus B = \{X \mid (B')_x \cap A \neq \emptyset\}$$

$$= \{X \mid [(B')_x \cap A] \subseteq \emptyset\}$$

The input image is denoted by A, while the structure element is denoted by B. On the image, the identical structural element is used, and if the center pixel is matched, the entire area surrounding that pixel is indicated. Erosion and Dilation are changed by the shape of B. Thus, B should be selected in advance.

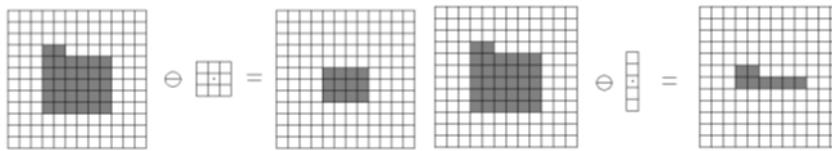


Figure 8: The Output of image erosion. The centre pixel is maintained when the structure element matches; when it does not, all pixels are destroyed.

Since erosion is destructive in that it degrades other parts foreground image with noise, we used opening operation which is simply erosion followed by dilation with the same structuring elements.

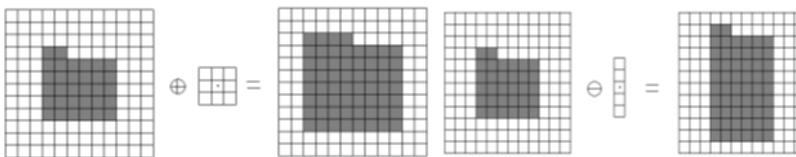


Figure 9: The output of image dilation. The bin of the structural element is filled when the structure's center matches any pixels.

We also perform closing operations followed by opening. The result of the noise removal operation is shown in figure 10.

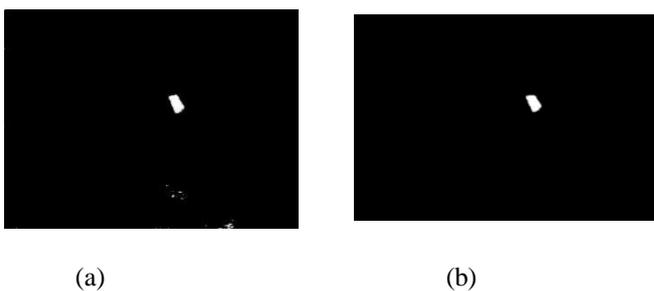


Figure 10: Output of image morphology. Carry out the opening and closing operations on the binary image. (a) The original photograph. (b) Image morphology's outcome. After noise removal, mouse pointer movement and different mouse operations based on detected colored objects can be done.

4.6 Track Face Rectangle

Face Detection Process: It is the process of distinguishing several features of a human face, such as the eyes, nose, and mouth, among others... This method may be accomplished with PyCharm Software. In this project, the authors will use image invariants to recognize faces in still photos. To accomplish so, you'll need to look at the greyscale intensity distribution of a typical human face. The 'average human face' below was created using a sample of 30 frontal view human faces, 12 of which were females and 18 of which were males. To highlight grey-scale intensity disparities, a properly scaled colormap was employed.

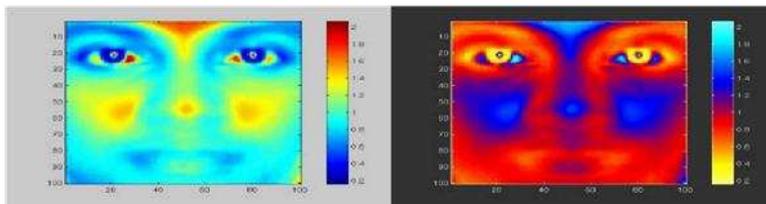


Figure 11: Grey-scale portrait of a typical human face

The grey-scale changes are dramatic, and they are consistent across all of the sample faces. Dark intensity (low) grey levels appear to always be present around the eyes and brows, whereas light intensity (high) grey levels appear to be present around the nose, forehead, and cheeks. The researcher discovered that the following areas of the human face were suitable for a face detection system based on picture invariants and a deformable template after a lot of testing.

Because of the distinct divisions of the bright intensity invariant area by the dark intensity invariant sections, the aforementioned facial area works well as a basis for a face template. After the face detection algorithm has discovered this pixel area, any desired area can be split using the proportions of an average human face. The author decided to utilize the following as a basis for dark intensity sensitive and brilliant intensity sensitive templates after reviewing the preceding photographs. A pixel region is 33.3 percent (of the width of the square window) below this once these are placed in a subject's face.

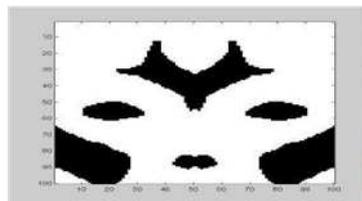


Figure 12: Basis for a bright intensity invariant sensitive template.

Note the minor changes to the bright intensity invariant sensitive template (see Figures 4), which were required due to the system's pre-processing to overcome uneven lighting. Now that suitable dark and bright

intensity invariant templates have been chosen, it's time to figure out how to use them to create two A-units for a perceptron, i.e., a computational model to assign neurons to the distributions shown.
Face Detection Algorithm:

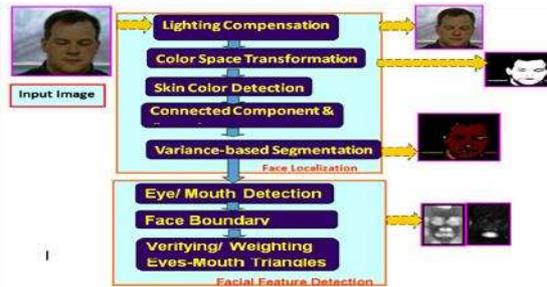


Figure 13: Face detection algorithm (Viola-Jones Algorithm).

Haar-like characteristics are digital image features used in object recognition, according to Wikipedia. They were utilized in the first real-time face detector and got their name from their intuitive resemblance to Haar wavelets. Haar-wavelet is a sequence of rescaled square-shaped “functions” which is very similar to Fourier-analysis. It was proposed by Alfred Haar in 1909.

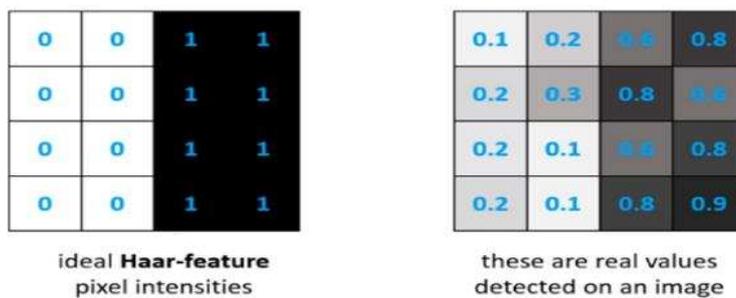


Figure 14: Haar-feature pixel intensities

Pixel intensities of 0 mean white and 1 mean black in the ideal Haar- feature. The Viola-Jones method will compare how near the actual situation is to the ideal condition.

- 1) Let's take a look at the white pixel intensities as a whole.
- 2) Add the black pixel intensities together.

Here, $\Delta = \text{Dark} - \text{White} = (1/n) (\sum I(\text{dark}) - \sum I(\text{white}))$

Generally Δ value for ideal Haar-features is 1.

Δ value for the real image: $0.74 - 0.18 = 0.56$.

The closer the value to 1, the more likely we have found a Haar-feature.



Figure 15: Applied algorithm on the image.

Using Haar features algorithm we have measured or detected images more than 94% with better accuracy. Face recognition process using template matching method: It is comparable to the template matching technique used during face detection, except we're trying to recognize a face instead of categorize this as a 'face' or a 'non-face.'

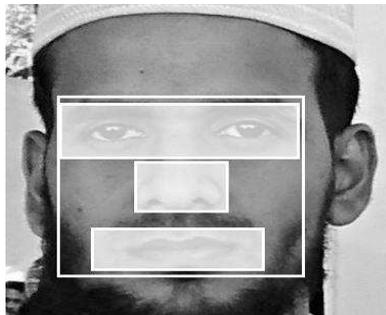


Figure 16: Face recognition using template matching

Recognition of human face by adding a name: A database is a collection of data that may be retrieved at a later time. They can be as small as a single table to as large as a group of tables with millions of tuples and columns. It is unlikely that the project will necessitate a database with more than a few tables. There was only one genuine consideration when it came to choosing a database, and that was MySQL. This is because of the following reasons: When compared to Oracle or Microsoft SQL Server, it is readily available and free to use, is extensively used by many enterprises, and provides good performance, dependability, portability, and stability. It is easily controlled using PHP, and there are numerous books and user guides available online.

5. Results and Discussions

After completing the whole process real-time face recognition on webcam and face recognition from a recorded video test was done.

5.1 Accuracy Calculation

True Positive (TP) is a genuine point of interest that's been appropriately identified. The correctly detected faces can be calculated as:

True Positives rate (TPR) = $TP / (TP + FP)$.

Where, TPR= True Positive Rate, TP= True Positive, FP= False Positive.

False Positive (FP) is a genuine object of curiosity that has been mislabelled as negative. False Negative Rate (FNR) can be calculated as:

$FNR = FN / (FN + TP)$.

Where FN= False Negative, FNR= False Negative Rate.

Now, Accuracy = $(TP + TN) / (TN + TP + FN + FP)$

Where TN= True Negative.

We measured or recognized images with more accuracy than 94 percent using the Haar features technique. We had taken around 1500 data and by analysing them using an accuracy equation we had measured the accuracy for our project. Where the detection rate was around 94 percent.

5.2 Real-time face recognition on webcam

We have used a webcam of HP laptop for recognizing both of our teammates and then we have become successful to recognize depending on our training data. After completing all the parts of our project, we got this can recognize image details with image names by capturing an image from the webcam. The output is given below:



Figure 17: Face recognition

5.3 Face recognition from a recorded video

At first, we have downloaded a movie part from the movie “Three Idiots”. Secondly, we have trained five faces of prominent characters of this movie. The detected names are listed below: Ranchor Dash, Engineering Sir, Raju Rastogi, Silencer, Farhan Quraishy, and the other persons are considered as “Student”.

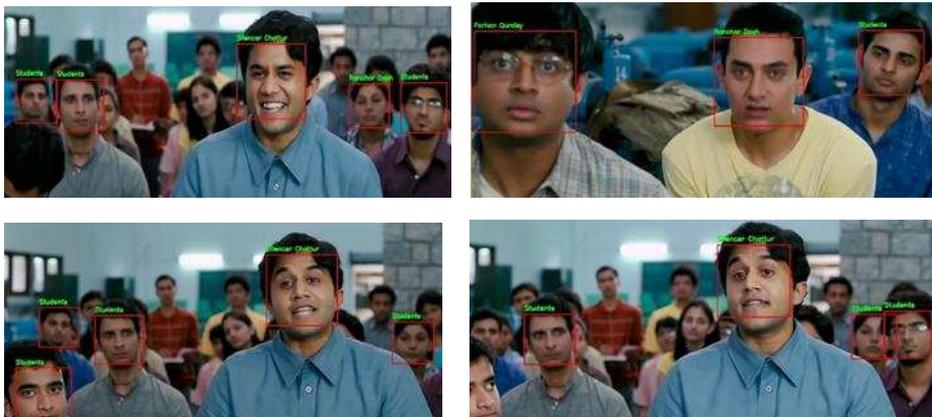


Figure 18: Recognized video

6. Conclusion

We expect our work to impact several fields of research and development in the future. The computational models used in this research project were picked after a thorough investigation, and the project's successful testing results demonstrate that our selections were sound. We also expect our work to be efficient and applicable to various fields. The limitation of our research work is the restricted number of eigenfaces employed in the PCA transform, the system, which used both manual and automatic face detection, failed to attain a recognition accuracy of more than 94 percent. In this study, the system was put through its paces under extremely severe conditions, and it is predicted that real-world performance will be far more exact. However, it is very easy to detect unknown persons in an office, home or institute etc. because this system is capable of taking faces from webcam or recorded video. We plan to extend our research work in future to prevent crime by adding eye detection technique. Adding an eye detection technique to the existing system would be a minor modification that would not necessitate a lot of further studies.

References

- Chhoriya, P. (2019). Automated criminal identification system using face detection and recognition. *International Research Journal of Engineering and Technology (IRJET)*, 6, 910-914.
- Wang, Y. Q. (2014). An analysis of the Viola-Jones face detection algorithm. *Image Processing on Line*, 4, 128-148.
- Padaruth, S., Indiwarsingh, F. & Bhugun, N., 2013, A Unified Intrusion Alert System using Motion Detection and Faces Recognition, 2nd International Conference on Machine Learning and Computer Science (IMLCS), Kuala Lumpur.
- Goldstein, A.J., Harmon, L.D., and Lesk, A.B. (1971). Identification of human faces. In *Proc. IEEE*, Vol. 59, page 748.
- Craw, I., Ellis, H., and Lishman, J.R. (1987). Automatic extraction of face features. *Pattern Recognition Letters*, 5:183-187, February.
- Lwin, H.H., Khaing, A.S. & Tun, H.M., 2015. Automatic Door Access System Using Face Recognition. *International Journal of Scientific & Technology Research*, 4(06), pp.294.
- Vaidya, B., Patel, A., Panchal, A., Mehta, R., Mehta, K., & Vaghasiya, P. (2017, June). Smart home automation with a unique door monitoring system for old age people using Python, OpenCV, Android and Raspberry pi. In *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 82-86). IEEE.
- Tharanga, J.G.R., Samarakoon, S.M.C. & Karunaratne, T.A.P., 2013, Smart Attendance using Real Time Face Recognition (Smart-FR), SAITM Research Symposium on Engineering Advancement, Sri Lanka.
- Deffenbacher K.A., Johanson J., and O'Toole A.J. (1998) Facial ageing, attractiveness, and distinctiveness. *Perception*. 27(10):1233-1243
- Jiang, H., & Learned-Miller, E. (2017, May). Face detection with the faster R-CNN. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)* (pp. 650-657). IEEE.
- Bansal, M. (2019). Face Recognition Implementation on Raspberrypi Using Opencv and Python. *International Journal of Computer Engineering and Technology*, 10(3).
- Gupta, I., Patil, V., Kadam, C., & Dumbre, S. (2016, December). Face detection and recognition using Raspberry Pi. In *2016 IEEE international WIE conference on electrical and computer engineering (WIECON-ECE)* (pp. 83-86). IEEE.
- Baron, R. J. (1981). Mechanisms of human facial recognition. *International Journal of Man Machine Studies*, 15:137-178.
- Adelson, E. H., and Bergen, J. R. (1986) The Extraction of Spatio-Temporal Energy in Human and Machine Vision, *Proceedings of Workshop on Motion: Representation and Analysis* (pp. 151-155) Charleston, SC; May 7-9.