INTERNATIONAL JOURNAL OF RESEARCH – GRANTHAALAYAH
A knowledge Repository

Science

RECOGNITION OF MULTI-VIEW HUMAN FACES BASED ON MACHINE INTELLIGENCE USING KLT ALGORITHM

Haripriya.K[*1], Ramya Lakshmi.V.[2], Rajeswari.S[3], Rama.T[4], Vinothini.K.R[5]
[*1, 2, 3, 4]Department of Electronics and Communication Engineering, A.V.C College of Engineering (Anna University), Mayiladuthurai, Tamilnadu, India
[5]Assistant Professor, Department of Electronics and Communication Engineering, A.V.C College of Engineering (Anna University), Mayiladuthurai, Tamilnadu, India

Abstract

Nowadays Image Processing has become a proficient domain due to the prolific techniques like face detection and face recognition. They play an important role in our society due to their use in wide range of applications such as surveillance, security, banking, and multimedia. One of major challenges faced in this technique of face recognition is difficulty in handling arbitrary pose variations in three dimensional representations. In video retrieval system, many approaches have been developed for recognition across pose variations and to assume the face poses to be known. These constraints made it semi-automatic. In this paper we propose a fully automatic method for multi-view face recognition of improving the accuracy or efficiency using local binary patterns. It uses tree-based data structure to create sub-grids. In this system we use KLT algorithm to detect and extract features automatically by using Eigen vectors and estimation of hessian value.

Cite This Article: Haripriya.K, Ramya Lakshmi.V., Rajeswari.S, Rama.T, and Vinothini.K.R. (2017). "RECOGNITION OF MULTI-VIEW HUMAN FACES BASED ON MACHINE INTELLIGENCE USING KLT ALGORITHM." *International Journal of Research - Granthaalayah*, 5(3), 123-134. https://doi.org/10.5281/zenodo.439558.

1. Introduction

The human face processing techniques for video, including face detection, tracking, and recognition, has attracted a lot of research interest because of its value in various applications, such as criminal identification, security system, image and film processing as the human face provides rich information for spotting the appearance. Automated face recognition system is a big challenging problem and undergoing several researches. There are many approaches and proposed algorithms to identify and recognize human being face. The recent development in this field has improved the processing capacity and high accuracy. There are many existing systems

to identify faces and recognized them. But the systems are not so efficient to have automated face detection, identification and recognition. And there are difficulties in the development of efficient feature extracting and high processing power for retrieval from a huge image database. A video image is a complex high dimension (3D) matrix. The processing of matrix operation is not so fast and perfect, hence, this directs us to handle with a huge image database and focus on the new algorithms which are more real-time and more efficient with maximum percentage of accuracy. And another problem in face recognition is machine learning to find the relevant characteristics of face and non-face samples and multi –view faces .The present system have good performance in detection of frontal faces, but locating multi-view faces remains a difficult problem due to the much more complication within the multi-view face class. Statistics show that most of the photos and videos faces are non-frontal, and therefore, the ability to deal with multi-view faces is important for many face related applications. New efficient and robust multi-view face recognition in video is presented in this paper.

2. Methodology

The general block diagram for face recognition in video system have two main blocks, the first is the face tracking and second is the face recognition. The basic face recognition model is shown in figure1.
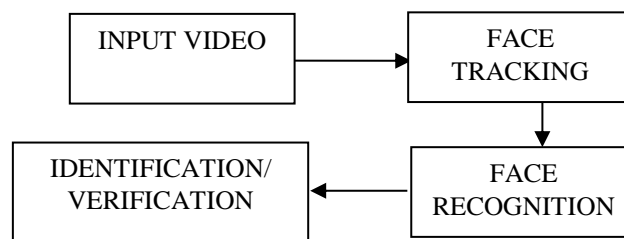

Figure 1: The basic block diagram of face recognition

2.1.Face Tracking

The process of locating a moving faces over a period of time is known as face tracking. First, the face detector initializes the given face. Then the subsequent video frames are analyzed by face tracker and the motion parameter of face is estimated to determine the outputs from the initialized face within these frames. Face tracking acquires the information on many consecutive faces within subsequent frames in which the faces are identical. At the run time, the faces in the video frame are outlined and tracks are ranked according to the similarity. Since tracking is corresponds to one face identity, the workload of face matching is reduced efficiently. In addition to this, face tracking is also used for the association of face is its objective is to label television or CCTV footage with the identity of the person present in each frame of the video.The face tracking consists of two blocks such as face detector and feature extraction.

2.1.1.   Working of Face Tracker

Face tracking is a kind of algorithm that analyzes video frames and the location of moving faces within each frame are output. For each tracked face, three steps are involved, i.e., initialization, tracking, and stopping.
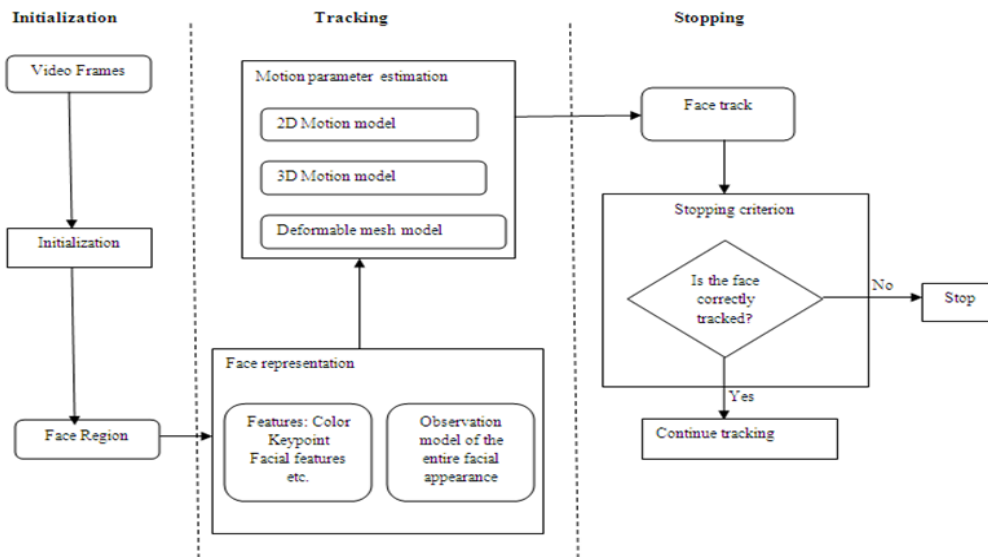

Figure 2: Working of face tracker

2.1.1.1.Detector

The first step of tracking is initialization. There are many methods in initialization. In most methods, the initialization of the tracking process is done by face detector. But the two problem arises while detecting. They are: non-facial problem and non-frontal face. Although there are many studies in profile or intermediate pose face detectors, they all suffer from the major problem called false detection problem. To alleviate these problems, instead of using a fixed threshold, we uses two face probability maps, one for frontal views and one other for profile view. All local maximum values in these maps are chosen as face candidates whose probabilities goes to zero or remain low over time are determined t be non-face and the face probabilities of which are propagated throughout the temporal sequence are estimated. The intermediate head pose is represented by combining the information from the two face probabilities maps. This type of proposed probabilistic detector is more accurate than a traditional face detector. And by this method, we could handle head movement covering ±90 degree out-of -plane rotation called yaw.

2.1.2.   Feature Extraction

After initialization, one should choose the features to track before tracking a face. The exploitation of color is one of the more common choices. However, color-based face trackers do not work on unknown videos with varying illumination conditions or on faces of people of different races. The other choices of facial features that are more robust to varying illuminations are eyes, nose, mouth, etc.

2.1.3. Stopping

After tracking, there is a need for checking weather the face tracked correctly or not. The process is continued until it is tracked correctly otherwise process is stopped and starts tracking from beginning.

2.2.Face Recognition

The process of identifying or verifying one or more persons appearing in a scene by using a stored database of faces is called Face Recognition.
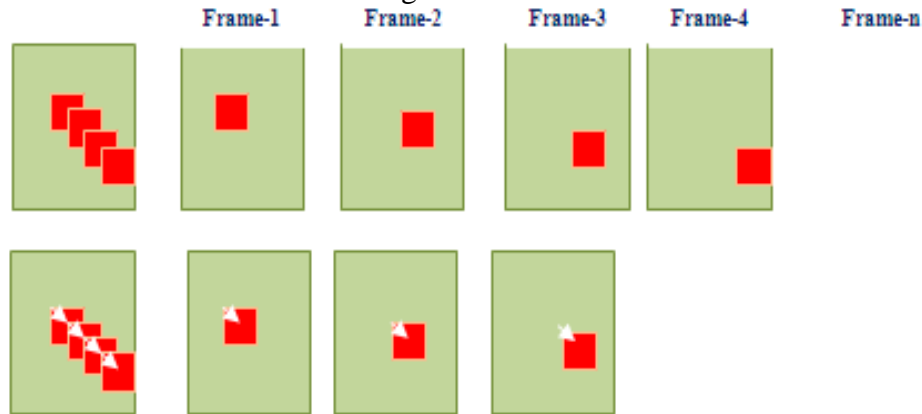

Figure 3: Frames

2.2.1. Recognition Techniques

In general, a single image can be viewed as a point in a Euclidean image space. In dimensions, D is equal to the number of pixels of the input face image.

Usually D is large, leading to the curse of dimensionality problem. However, the surfaces of faces are mostly smooth and have regular texture, making their appearance quite constrained. As a result, it can be expected that face images can be confined to a face space, a manifold of lower dimension d << D embedded in the image space. The eigen-face method is a popular technique for dimensionality reduction when a set of training face images is given. A face is represented as a point in a high dimensional eigen-face space, which is computed from a set of training faces using principle component analysis (PCA). Figure shows an example of a set of eigen-faces computed from 3,816 faces.
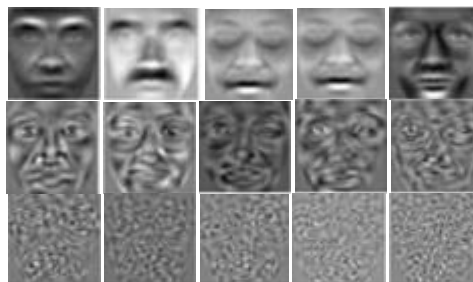

Figure 4: Eigen Faces

3. KLT Algorithm

Human tracking is one of the most important for various applications including surveillance, activity monitoring and gate analysis. Many efficient object tracking algorithms have been proposed in literature, however part of those algorithms are semi-automatic requiring human interference. As for the fully automated algorithms, most of them are not applicable to real-time applications. This project presents a low cost automatic object tracking algorithm suitable for video based systems. The novelty of the proposed system is that it uses a simplified version of the Kanade-Lucas-Tomasi (KLT) technique to detect features in pose variation. Lucas-Kanade proposed the method for image alignment(tracking) and Tomasi-Kanade proposed the method for best features extraction(image patch).

The KLT algorithm can be divided into two main parts. One is Feature point detection and second is Feature point tracking.During the detection process, salient feature points are found and added to the already existing ones. Afterwards, in the tracking process for each feature point its corresponding motion vector is calculated.

3.1.Feature Point Detection

The task performed here is to detect the feature points in a given image *I* and add them to the already existing feature points. The structure matrix L can be defined as

$$\mathbf{L} = \sum_{x \in W(p)} \Delta\, I\,(x)\,.\,\Delta\, I(\,x\,)^{T} \qquad\qquad\qquad \ldots\ldots\ldots\ldots\ldots\text{equation (3.1)}$$

The information about the neighborhood region *W*is given by the its eigen values $\lambda 1, \lambda 2$. If neighbourhood region*W* is
   a. completely homogeneous, then $\lambda 1 = \lambda 2 = 0$.
   b. Contain edge, then $\lambda 1 > 0, \lambda 2 = 0$.
   c. a corner, then$\lambda 1 > 0, \lambda 2 > 0$.

The feature point detection undergoes the following steps:
   1) Calculate structure matrix *L*and λ for each pixel in the given image *I*.
   2) Calculate the maximum λ_{max} occurring in the image.
   3) 3. Eliminate all the pixels which have λ equal to or smaller than λ_{max} .Keep all pixels that have a λ larger than λ_{max}.
   4) Perform a non-maxima suppression within the 3 x 3 pixel neighborhood of the remaining points to keep only the local maxima.
   5) From the remaining points, add new points to the already existing points, starting with the points with the highest λ values.

3.2.Feature Point Tracking

Here, we want to calculate for each feature point qin image *I* its corresponding motion vector iso that its tracked position in image *T* is q+ i. The following are the steps for face tracking:
   1) Set initial motion vector   i (0,0)
   2) Calculate image gradient $\nabla I = \partial I / \partial(x, y)$

3) Calculate Structure matrix

    i. $\mathbf{G} = \sum\limits_{x \in W(q)} \Delta I(x) \cdot \Delta I(x)^T$ ………………… equation (3.2)

4) for k = 1 to *maxIter*
a. Image difference h (x) = I (x) - J (x + ik)
b. Calculate mismatch vector

    a. $b = \sum_{x \in W(p)} h(x) \cdot \Delta I(x)$ ………………… equation (3.3)

c. Calculate updated motion

    a. $i_{k+1} = i_k + G^{-1}b$ ………………… equation (3.4)

d. if $\| i_k +1 - i_k \| <$ eps then stop.
e. Report final motion vector i.

3.3. Overall Algorithm Process

The steps involved in KLT algorithm are as Follows:
Let *I* as the current image and
   T as the immediately next image in the sequence.

STEP 1: Wrap the image to get   I(w(x ; p))

STEP 2: Compute the error image
       I(W(x; p)) - T(x))

STEP 3: Minimize the SSD with respect to p
    $\sum\limits_{x} [I(W(x; p)) - T(x)]^2$

STEP 4: Solve for increments to current estimate
    $\sum\limits_{x} [I(W(x; p + \Delta p)) -- T(x)]^2$ ………………… equation (3.5)

where Δp is calculated from equation (3.10).

STEP 5: On Linearing
$\sum_{x}[I(W(x;p)) + \nabla I \frac{dW}{dp} \Delta p - T(x)]^2$ …………………… equation (3.6)
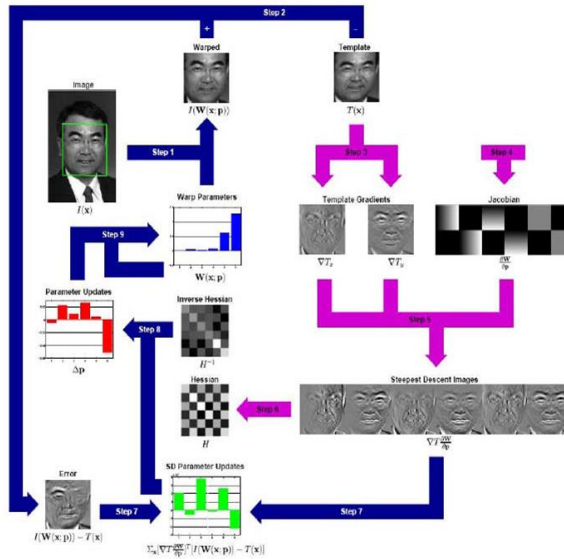
Figure 5: KLT Algorithm

STEP 6: solve for $\Delta\mathbf{p}$

$$\Delta p = \sum_x H^{-1} \left[\left[\nabla I \frac{dW}{dp} \right]^T \right] [T(x) - I(W(x;p))] \qquad \text{......………………… equation (3.7)}$$

$$H = \sum_x \left[\nabla I \frac{dW}{dp} \right]^T \left[\nabla I \frac{dW}{dp} \right] \qquad \text{…..………………… equation (3.8)}$$

STEP 7: Compute image gradient $\Delta\mathbf{I}$
$$\nabla I = \partial I / \partial(x, y)$$

STEP 8: Compute Jacobian $\frac{\mathbf{dW}}{\mathbf{dp}}$

Image coordinates $\quad \mathbf{x} = (x, y)^T$
Warp parameters, $\quad \mathbf{p} = (p_1, p_2, \ldots, p_n)^T = (dx_1, dy_1, \ldots, dx_n, dy_n)^T$

$$\frac{dW}{dp} = \begin{pmatrix} \dfrac{dW_x}{dp_1} & \dfrac{dW_x}{dp_2} & \cdots & \dfrac{dW_x}{dp_3} \\ \dfrac{dW_y}{dp_1} & \dfrac{dW_y}{dp_2} & \cdots & \dfrac{dW_y}{dp_3} \end{pmatrix}$$

STEP 9: compute Hessian **H**

$$H = \sum_x \left[\nabla I \frac{dW}{dp} \right]^T \left[\nabla I \frac{dW}{dp} \right] \qquad \text{………………… equation (3.9)}$$

STEP 10: compute $\Delta\mathbf{p}$

STEP 11: Update the parameter

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$$ ………………… equation (3.10)

4. Local Binary Pattern

LBPs is very effective for representation of image which have been applied to visual inspection, motion detection and outdoor scene analysis. The most important properties of LBP features is the monotonic illumination changes tolerance and it is computational simplicity and they specifically indicate the edges, spots, and flat areas of the image.

The operator labels the pixels f_P (P = 0,. . . ,7) of an image by threshold a 3 X 3 neighborhood of each pixel with the value of the center pixel f_c and considering the result as a binary number $S(f_p - f_c)$.
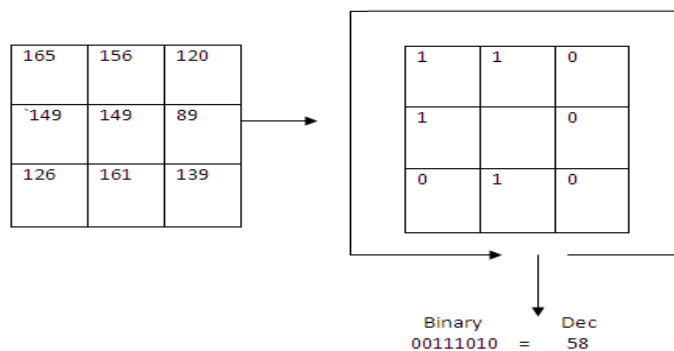

Figure 6: LBP

An example of the LBP operator is shown in Figure.6

$$S(f_p - f_c) = \begin{cases} 1 & if \ f_p \geq f_c \\ 0 & otherwise \end{cases}$$ ………………… equation (4.1)

Then, by assigning a binomial factor 2^P for each S(fp - fc) the LBP is computed as follows:
$$LBP = \sum_{p=0}^{7} S(f_p - f_c)2^p$$ ………………… equation (4.2)

This operator was extended to use neighborhoods of different sizes, to capture dominant features at different scales. The Notation LBP(P,R) denotes the neighborhood of P which is equally spaced sampling points on a circle of radius R. Figure 6 is a basic LBP where P = 8 and R = 1. A small subset of the 2^P patterns accounted for the majority of the texture of images are called uniform patterns which contain at most two bitwise transitions from 0 to 1 or vice versa for a circular binary string. Non-uniform LBPs are less useful for texture recognition because they the noise occur in the image makes them ineffective for discriminating between patterns.

For example, Let us consider two uniform patterns such as 01100000 and 11011111. These binary patterns are used to represent texture primitives such as spot, flat area, edge and corner. The uniform patterns contain (P - 1)P + 2 total binary patterns, Where (P - 1)P are rotational patterns which includes edges and two non-rotational patterns such as spot and flat area. Patterns where U(x) > 2, are defined as non-uniform patterns:

$$LBP_{P,R}^{u2} = \begin{cases} z \ if \ U(LBP_{P,R}) \le 2, LBP_{P,R} = I_z \\ \quad where \ |I| = (P-1)P \\ (P-1)P + 2 \ \ otherwise \end{cases}$$ ....

equation (4.3)

Where ,

$$U(LBP_{P,R}) = ||S(f_{p-1} - f_c) - S(f_0 - f_c)|| \ + \Sigma^p{}_{p=0} ||S(f_p - f_c) - S(f_{p-1} - f_c)||$$ ....

equation (4.4)

LBPs are most commonly used for facial recognition. The face is divided into smaller regions, typically a 'uniform grid' of an odd number of equal divisions in both the horizontal and vertical directions. The LBP operator is applied to each of these regions, and the resulting image is then used to create a histogram. The histograms from the regions are appended to create a 'spatially enhanced histogram'.

5. Result

This section contains results of our experiments. This involves databases, performances, pose estimation and comparison.

5.1. Database

This section consists of template images of input video that is extracted frames of video. This comprises of set of poses in multiple views. The train images to be compared and recognized are also stored in databases such as CalTech data set and the Database of Faces dataset. This chooses one random image from each class in dataset to be the training sample and uses other remaining images for testing.

### 5.2. Performance Evaluation

The faces were labeled on the basis of poses. For the angles like 22.5, 45, 67.5 faces, eyes and chin were labeled. For the 90 faces, eyes, nose and chin were labeled. On the basis of ground table, faces were extracted and scaled to 19×19 pixel window according to non-frontal face model depending on pose. In frontal faces, distances between eyes are constant whereas non-frontal is not constant. Frontal faces can be used to compute a ratio between original and window image. The ratio is calculated using the distance between eye and chin in the original image that is EC, and the distance between the eye and chin in window (i.e.,) EYE_CHIN. EYE_CHIN is computed as:

$$EYE\_CHIN = \frac{en\_gn \cdot model\_width}{zy\_zy}$$

where, zy_zy is width of the face, model_width is width of the window and en_gn is the lower half of the craniofacial height. We can compute ratio as:

$$ratio = \frac{EC}{EYE\_CHIN}$$

For feature extraction is done using points, we have taken two eyes, nose and chin.
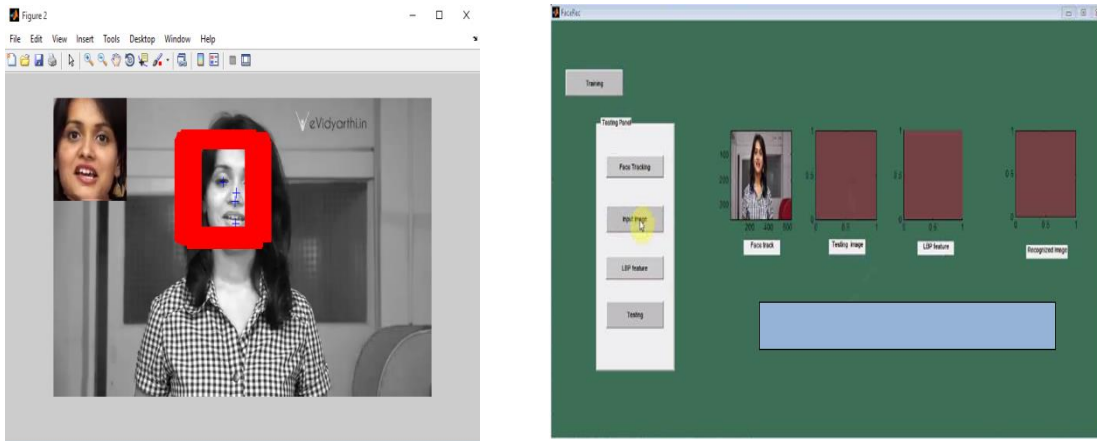
(a)   (b)

Figure 9: (a) Tracked face; (b) face tracking

5.3.Face Recognition

Face recognition process involves the matching of the template with the train image. This took place on the basis of the comparison of train images with template. In verification, an image is matched to only one image in the database (1:1). For example, an image taken of a subject may be matched to an image in the Department of Motor Vehicles database to verify the subject is who he says he is. If identification is the goal, then the image is compared to all images in the database resulting in a score for each potential match (1:N).The image is warped using the jacobian and steepest descendent values. These parameters are updated using the Hessian values. It is then inverted and Eigen images are detected. So the train images are compared with the Eigen faces stored in the dataset. The output will interpret the recognition rate as follows:
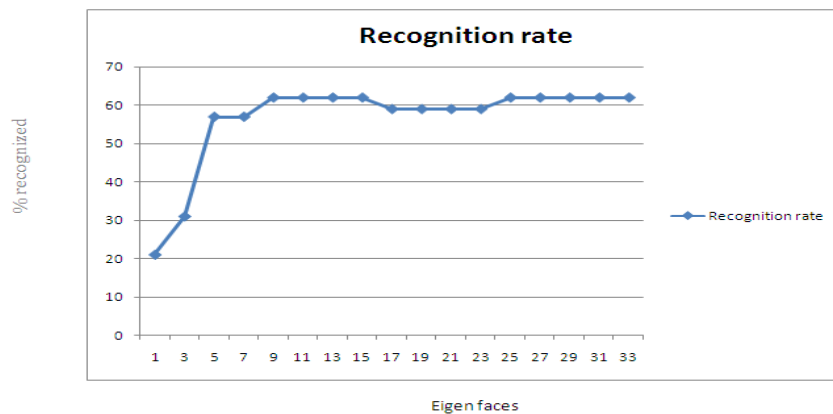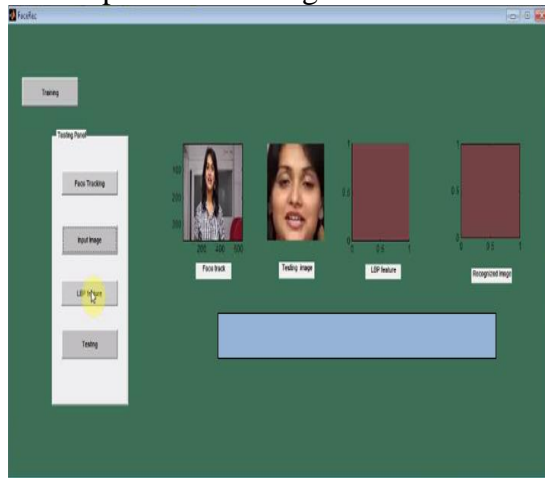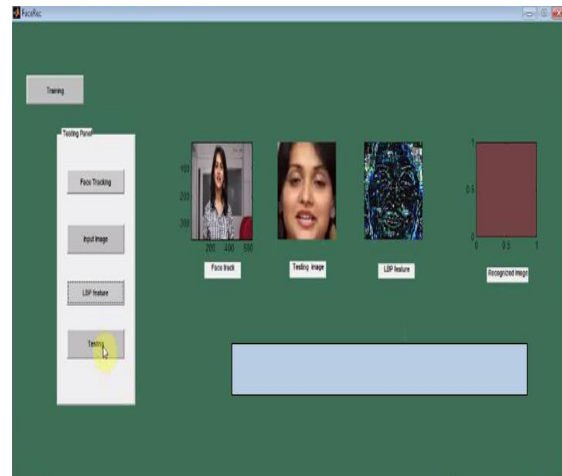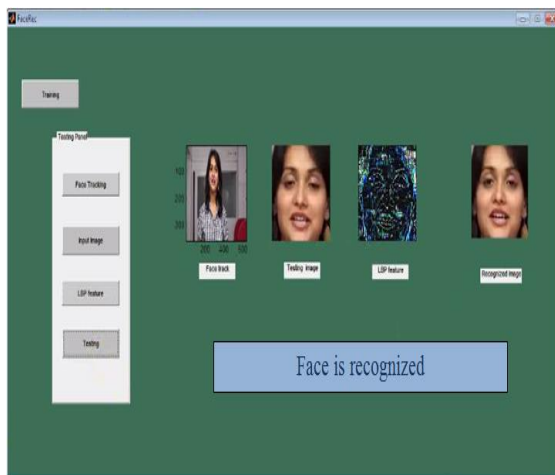

Figure 10: Recognition Rate

The output of face recognition as follows:

(a)

(b)

(c)

Figure 11: (a) Testing image; (b) LBP extraction;   (c) Recognizes image

6. Conclusion

In this paper, a novel framework for rapid and pose-invariant face detection has been presented. In this framework, multi-view recognition of faces involves two steps: face detection and face recognition. Face detection process involves feature extraction by using point trackers. These point trackers are assigned on the basis of local binary patterns. Face recognition involves the matching or verification of train and template images by using Kanade-Lucas-Tomasi algorithm. This technique has demonstrated the effectiveness of real time detection of multi-view faces and poses estimation by hierarchical combination of face pose classifiers.

References

[1]   R. Jafri and H. R. Arabnia, A Survey of Face Recognition Techniques Journal of Information Processing Systems, vol. 5, no. 2, pp. 41–68, Jun. 2009.

[2]   L. Ding, X. Ding, and C. Fang , Continuous pose normalization for pose-robust face recognition, IEEE Signal Processing Letters, 19(11), 2012 ,721–724.

[3]   M. Turk,A. Pentland , Eigen faces for recognition,   Cognitive Neuroscience, vol.3, no.1, pp.71–86, 1991.

[4]   T. Ojala, M. Pietikainen, and T. Maenpaa , Multi resolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. PAMI, 24(7): 971–987, 2002.

[5]   Rafel.c.Gonzalez, Richard E.Woods digital image processing(Pearson education inc.)

[6]   Rajkamal kishor Gupta, Object detection and tracking in video image, Department of Computer Science and Engineering, National Institute of Technology Rourkela, Rourkela, India.

[7]   Naotoshi Seo, simultaneous multi-view face tracking and recognition in video using particle filtering, the Graduate School of the University of Maryland, College Park in partial fulfillment of the requirements for the degree of Master of Science,2009.

[8]   Saeed U., Matta F. and Dugelay J.-L., Person recognition based on head and mouth dynamics, in IEEE Proceedings on Multimedia Signal Processing, pag. 29-32, October 2006.

[9]   Storring, M.—Andersen, H. J.—Granum, E. , Estimation of the Illuminant Color from Human Skin Color, Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

[10]   Schwerdt, K.—Crowley, J. L.: Robust Face Tracking Using Color. Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

*Corresponding author.
E-mail address: priyavaishu07@gmail.com