3. Chapter 3

CHAPTER 3

RECOGNISING A POSE VARIANT FACE IMAGE

3. RECOGNISING A POSE VARIANT FACE IMAGE

3.1Introduction

One of the challenges in face recognition that deteriorates the recognition rate of this biometric recognition system is with respect to pose variant faces that are captured in the frame of a video. When the input to the system is a video sequence, it can carry faces with different poses and variations. It is a difficult task in recognising these faces of the varied pose. Various methods have been proposed to solve this issue. Figure 3.1 shows the framework of pose invariant face recognition (Ding and Tao 2015).

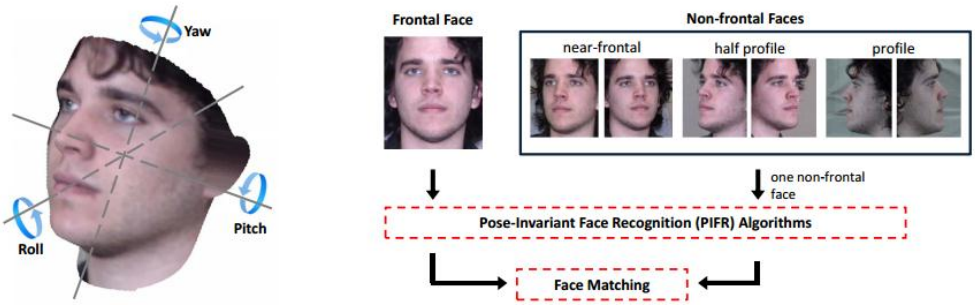

Figure 3.1: Faces with different poses (Ding and Tao 2015)

The figure describes a frontal face image that is to be matched with a non-frontal face. If the frontal face is to be matched with a non-frontal face, it turns out to be a difficult task. Hence to identify faces with different poses at the same time is a very important task in face recognition system. Methods have developed with deploying different cameras for this purpose. Recognising face from different poses is gaining attention as face biometric is a passive biometric and users need not cooperate with the system. This chapter discusses a method to recognise face with varying poses.

3.2 Proposed Model

In this work, a model is proposed that recognises faces that are inclined at a different angle. Figure 3.2 gives a diagrammatical representation of the proposed model. The input is a sequence of video. From this video sequence, a face is to be detected and matched with the stored face image in the gallery. If a matched face is found, access should be provided. As a first step, the face needs to be detected from a frame of the video sequence. From the detected face image, features are extracted. As discussed in Chapter 2.2.3, Curvelets are always considered superior in recognising smooth curves and for this reason; Curvelet Transform is used for extracting features in the process of face recognition. The extracted features are matched against the feature extracted from gallery images and further matched for similarity. If a face is found matching it is accepted and access is allowed else it is rejected.

In order to check the efficiency of this framework, (i) Youtube dataset and (ii) Honda/UCSD data sets are used. These datasets contain videos with varied pose variation. Recognition rate and time required to recognise the face are calculated. YouTube dataset contain noisy real-world videos. This dataset includes video clips of 35 celebrities. The videos in the dataset are of low resolution and recorded at high compression rates. YouTube dataset are used for studying the problem of unconstrained face recognition in videos. This dataset contains 3,425 videos of 1,595 different people. In a video clip, there are 181.3 frames of faces in an average. The second dataset, the Honda/UCSD dataset contains videos of faces with large variation in head movement. It includes movement of the head in all the four directions (left/right and

up/down). The set contains many subjects each appearing in at least two sessions. The experimentation is done using MatLab.

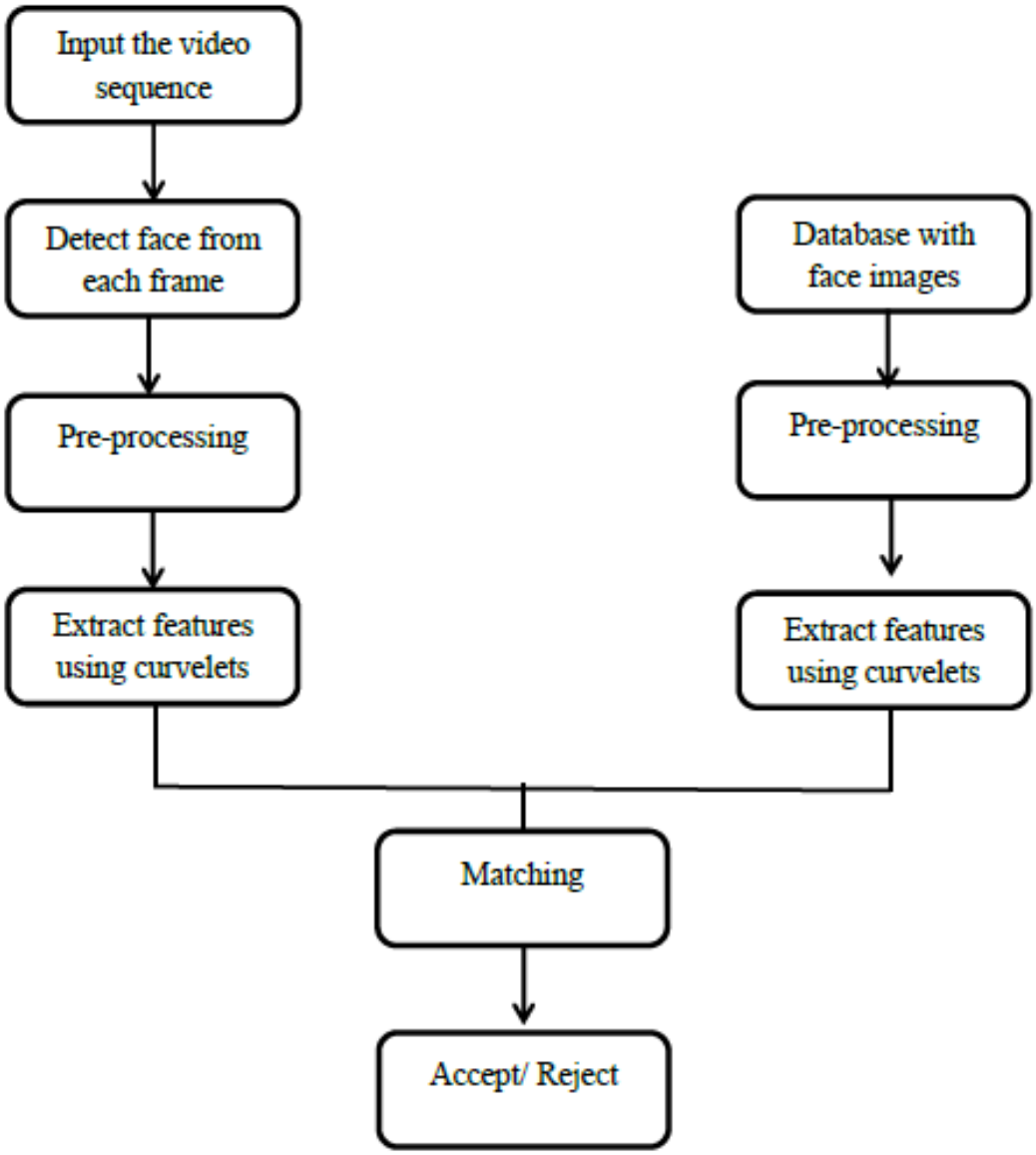

Figure 3.2: Proposed model for recognising faces with pose

3.3 Preprocessing Method

For constructive recognition of face image, the image is cropped into a size of 120×120

3.4 Experimental Data

For experimental analysis of this study two publicly available databases are considered. These databases namely, You-Tube Face database and HONDA/UCSD are used in evaluating the proposed approach. The first database, You-Tube face database is one of the databases used for face recognition from video sequences. The major purpose of using this database is the input to the study is unconstrained faces. You-Tube face database consist of 3,425 videos of 1595 different subjects. The dataset is created by downloading videos from You-tube. There is an average of 2.15 videos for each subject. There are videos with varying duration. The duration of the shortest clip is 48 frames and the longest is 6070 frames. Hence the average can be computed as 181.3 frames. This database includes videos where faces vary with different poses and consisting of real world videos.

Figure 3.3 and 3.4 represents frames extracted from you tube dataset.

Figure 3.3: Frame1 of faces extracted from YouTube database

Figure 3.4: Frame2 of faces extracted from You Tube database

The second database considered is the HONDA/UCSD database. This is a standard database used for evaluating face tracking and recognition algorithms. In this database each of the faces in the video sequence is recorded in an indoor environment at 15 frames per second. The duration of each video sequence is a minimum of 15 seconds. The resolution of each video sequence is 640×480. In this database, all the video sequence contains a 2-D and 3-D head rotation. This is very significant while a problem on faces with varying poses is considered. While rotating the head by each individual in his/her speed, for 15 seconds each are able to provide wide range of different poses. This video database contains two datasets. One of them included three subsets for training, testing and occlusion testing and with respect to 20 different human faces there are

20, 42 and 13 videos respectively. The second dataset contains two subsets containing 30 videos from 15 different human faces for testing and training. Figure 3.5 represents frames extracted from Honda/UCSD dataset.

Figure 3.5: Frames of faces extracted from Honda/UCSD database

One of the other publicly available databases is MoBo (Motion of Body) database. This database was originally collected for identifying human motion from distance. This database consists of 25 individuals walking in a treadmill. This includes videos with different walk patterns like slow, fast and incline walk of the individuals. This database is in general used for human gait. Hence this is not used for the study.

3.5 Face Detection

Face detection is an important step in face recognition technique. (Rowley et al. 1998) and (Schneiderman and Kanade 2000) for their work, extended their previous work for rotated and profile faces, as well as new sets of rectangle features, are defined for rotated face detection. The authors in their work used classifiers based on the weak classifier set from the AdaBoost algorithm by (Freund and Schapire 1995). For real-time face detection, viola Jones algorithm gives a good detection rate. The work proposed by (Viola and Jones 2001) describes a machine learning approach for detecting objects which give high detection rate. According to the work carried out, it is a composition of three key features: (i) new image representation which authors represented as an Integral image. This helps in quickly computing the features used by the detector. (ii) Learning algorithm based on AdaBoost. This selects features from and yields efficient classifiers. (iii) Combines complex classifiers which discard the background region and does the computation of object-like regions. Based on this work, extended face detection for multi-view faces is proposed in (Jones and Viola 2003). In this work, authors used building different detectors for different views.

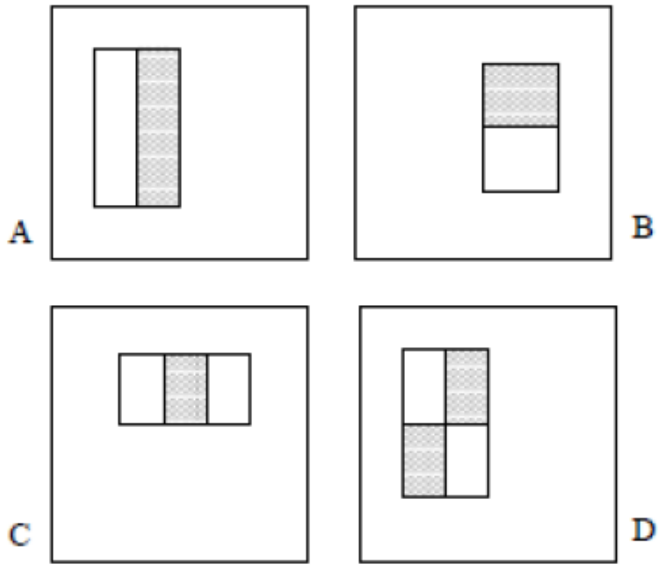

Figure 3.6: Haar features considered for Adaboost.

The features shown in Figure 3.6 are the important Haar features selected for detecting face region using AdaBoost.

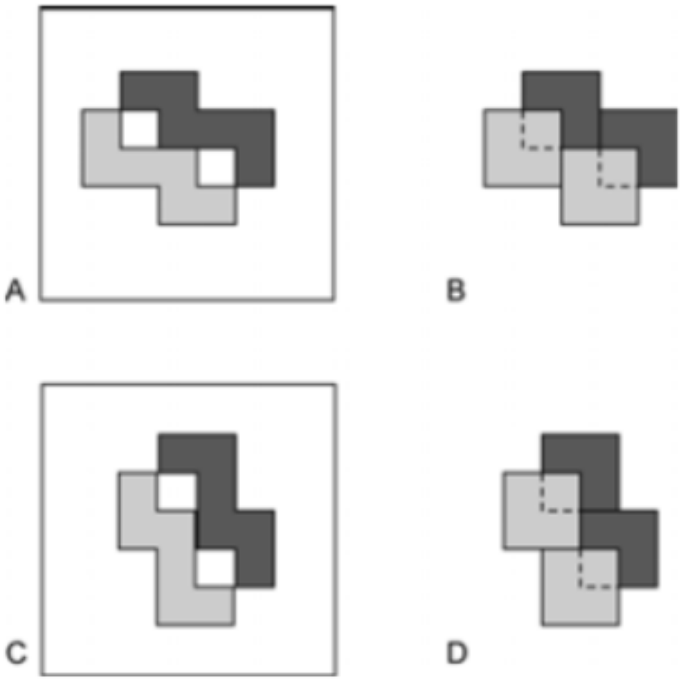

Figure 3.7: Diagonal Haar Features Designed for Face Detection

Figure 3.7 shows the diagonal filters used by the authors that are used to detect non-upright faces and non-frontal faces giving highly accurate results. Authors used a sequence of the complex classifier to improve the efficiency in computation. Input window is evaluated on the first classifier and if it returns false then computation ends on that window if it returns true, next classifier in the cascade evaluates the window. In a similar way, window passes through all the classifiers and if true, the detector will return a true value for that window. In order to detect the face with different poses, the poses are divided into various classes and different detectors are trained for each class. (Meynet et al. 2003) described the algorithm for AdaBoost in their work in detail and is summarised in Table 3.1. The working of AdaBoost algorithm includes one weak classifier that is selected at each step. A weight is assigned to the data at each step and a weak learner is constructed based on the weight (Schapire 2013). To estimate how the data is classified, the weak learner produces an empirical error. The weak classifier with the best classifier and with the lowest error is selected and weights are updated accordingly. The iteration loop stops if the empirical error calculated is$\mathbf{0}\ \boldsymbol{or}\ \geq \mathbf{1/2}$.

Table 3.1 AdaBoost Algorithm to Detect Face.

Input: Set $(x_1,y_1),\ldots.(x_m,y_m)$ where $y_i \epsilon \{-1,+1\}$

Set $D_1(i)=1/m$ for i=1,….,m.

For t=1,….,T:

- Normalize the weights to get the probability distribution D_t as $D_t(i) = \frac{w_{t,i}}{\sum w_{t,n}}$
- A weak classifier h_j is generated for each feature f_j.
- Determine the error ε_j of h_j with respect to D_t:

$$\varepsilon_j = \sum_{n=1}^{N} w_{t,n} |h_j(x_n) - y_n|$$

- Choose classifier h_j with lowest error ε_j and assign $(h_t, \varepsilon_t) = (hj, \varepsilon_j)$.
- Update the weights $w_{t+1,n} = w_{t,n}{\beta_t}^{1-e_n}$ where $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$ and e_n =0 if x_n is classified correctly or else 0.
- The final strong classifier:

$$h(x) = 1 \; if \sum_{t=1}^{T} \log\frac{1}{\beta_t} h_t(x) \geq \frac{1}{2}\sum_{t=1}^{T} \log\frac{1}{\beta_t}$$

$$h(x) = 0 \; otherwise$$

3.6 Feature Extraction using Curvelet Transform

Curvelets are delivered remembering the finished objective to vanquish the controls that are gone up against by wavelets. (Candes et al. 2006) proposed curvelets and said to have various orientations and positions at each length scale and at fine scales they possess needle formed parts. The basic advantage with respect to discrete curvelet transforms are (i) they provide optimally sparse representation that displays smoothness along

the curve. (ii) Curvelets are the very important tool for analyzing and computing Partial Differential Equations (PDE). Curvelets can behave both like waves and particles. (iii) In severely ill-posed problems, curvelets can be optimally used in image reconstruction. Figure 3.8 represents the Curvelet tiling. Curvelets are supported near a parabolic wedge in the Fourier space and the shaded area in the figure on the left represents a generic wedge. The figure on the right represents Cartesian grid associated with a given orientation (Candes et al. 2006, Demanet 2006).

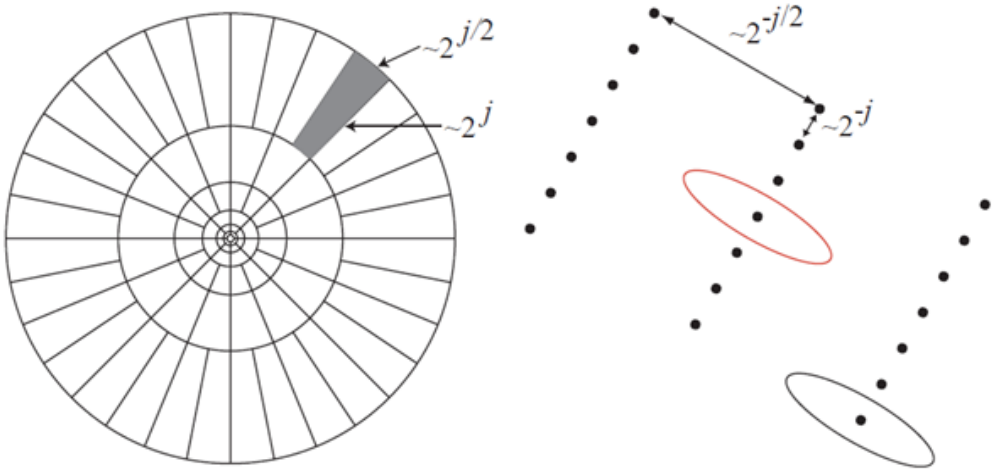

Figure 3.8: Curvelets tiling

The major disadvantage with respect to wavelet is that they will denoise the image. To overcome this disadvantage curvelet transform were developed.

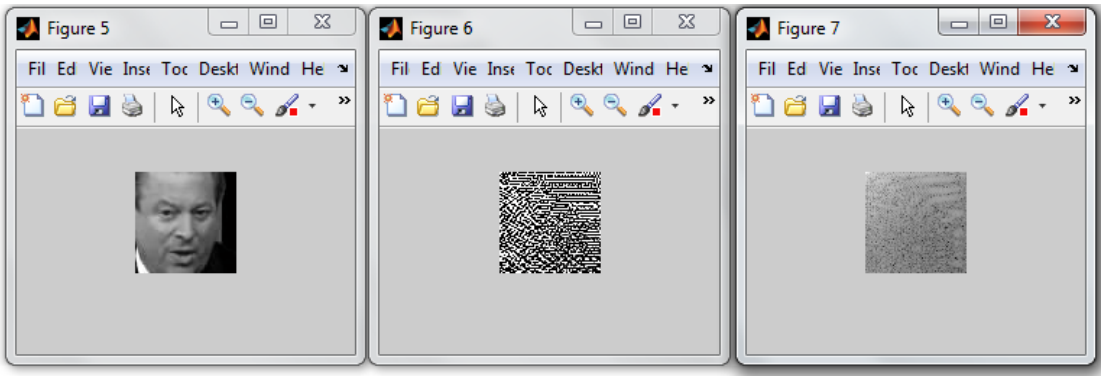

Figure 3.9: Feature extraction using DCT

Variables - I

I <70x70 uint8>

	1	2	3	4	5	6	7	8	9	10	11	12
1	51	58	68	78	88	100	113	122	138	154	174	184
2	55	64	76	88	99	110	121	129	134	150	169	180
3	48	59	76	91	103	113	122	128	131	145	162	173
4	37	51	71	90	104	114	122	127	132	144	158	167
5	37	52	75	95	110	120	128	132	137	146	157	165
6	45	59	80	99	113	122	130	134	142	149	158	164
7	58	70	87	102	113	121	128	133	145	152	160	165
8	71	82	96	108	117	123	130	135	146	152	161	166
9	56	64	78	94	108	121	132	137	143	148	155	162
10	64	73	87	103	117	129	138	142	143	148	155	161
11	71	80	95	112	127	138	144	147	143	147	153	159
12	73	81	96	114	131	143	149	151	145	148	153	157
13	74	81	94	112	131	145	153	156	151	153	156	159
14	79	83	92	108	128	145	155	159	157	158	159	161
15	85	85	89	102	122	141	153	158	158	158	157	157
16	88	85	86	97	115	135	148	154	155	154	153	151
17	74	70	75	94	115	130	138	143	154	152	151	152
18	65	64	72	92	113	128	137	143	154	153	152	154

Figure 3.10: RGB to Grayscale values

J <70x70 double>

	1	2	3	4	5	6	7	8	9	10	11	12
1	6.9419e+03	2.8999e+03	-1.5727e+03	399.9883	-276.6934	35.8537	-186.6848	-233.7864	66.9609	-153.1977	-117.4095	-100.6860
2	1.0682e+03	-254.9064	-879.7950	-66.5777	-414.9714	-247.8072	-300.0897	-199.1100	21.8965	-57.6681	-40.0100	16.5089
3	-214.6531	194.8822	-123.3157	-227.1187	208.3502	-144.7823	269.7909	57.2580	-258.4398	-30.2017	-73.8718	23.5191
4	374.6181	-52.8692	-353.5349	-137.5821	-257.8344	-77.3688	130.1977	10.3009	25.2506	133.2907	106.0066	93.1726
5	401.5608	178.3518	-42.7222	-9.3574	74.7238	163.8891	-157.7477	15.6075	133.7465	-3.2641	-32.7442	-126.5633
6	-414.0553	-114.9695	218.7291	20.0912	-54.3258	-37.7399	-9.4422	19.8858	148.8620	117.5133	84.0147	131.2885
7	-237.4121	-84.8012	97.8479	-56.1851	-10.6317	12.2550	-139.5937	-60.9113	-174.1236	-223.0343	-86.6320	-156.6265
8	183.7432	-17.8468	-198.2185	22.9556	-13.3967	-39.7041	129.5709	108.8187	65.8510	92.3101	40.0294	46.5972
9	261.9342	105.4080	-220.5215	-79.3295	-67.1501	-121.3313	15.9821	-120.1171	-101.0725	-39.0821	-30.3853	-4.9891
10	17.1722	7.6888	31.1110	40.1767	17.8732	28.7123	13.2055	51.1501	30.4083	59.1578	124.9713	62.9221
11	-154.3702	13.6039	121.3302	2.8914	70.2252	-6.1317	-76.7830	50.4934	86.7295	-3.3404	-73.1057	-20.0887
12	-79.5347	-4.0151	32.9850	-38.1562	-74.7606	-32.8717	97.4544	30.2880	-25.7123	23.1466	-79.0232	-64.4757
13	16.8090	16.2515	39.8593	-13.6646	-33.1876	49.7130	10.9594	-24.1844	-66.9469	-60.8621	47.2404	47.4706
14	-43.8823	-7.0862	75.2143	86.8281	4.2212	-75.1241	-63.0555	-9.4868	57.8373	31.5557	7.1103	-11.2487
15	73.5742	29.8288	-138.6560	-39.1535	126.0470	14.1167	-74.7858	-57.4483	3.5405	11.9111	-5.8134	16.6522
16	-161.0433	-20.7673	206.6468	44.0722	-120.1571	-14.5021	133.9182	80.3968	-72.8143	9.1688	91.7763	14.2225
17	5.7894	-18.0084	-77.4005	-39.1212	60.0379	21.5745	-89.1746	-66.4790	62.3943	23.5017	-89.9446	9.2091
18	13.9834	14.9550	6.7918	-34.0167	-66.4280	-2.9612	53.5415	31.2106	-5.1629	-23.1537	2.0779	9.3280

Figure 3.11: Features from DCT algorithm

Figure 3.9 shows the DCT feature vector when a face image is subjected to DCT. The features selected and compressed from the face input image are represented in Figure 3.9. In order to apply DCT on to a colour image, the colour image is converted to grey scale image. Figure 3.10 represents the screen shot of the grey scale values of the input colour image and Figure 3.11 shows the converted DCT values of each pixel.

3.7 Matching

Matching is performed using minimum distance classifier. This method is used to classify unknown image data to classes which minimize the distance between image data and the class in multi feature space. In this case, the distance is expressed as an index of similarity. Hence, the minimum distance is identical to maximum similarity.

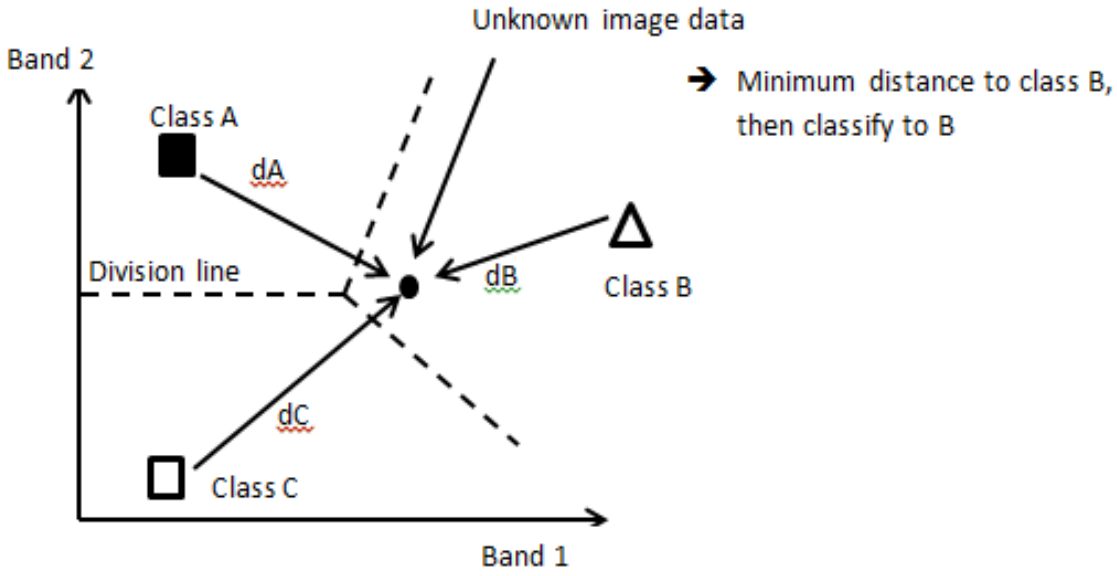

Figure 3.12: Concept of Minimum Distance Classifier

The distance used in this work is Eucleidian distance. This is given by the equation (3.1).

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \quad (3.1)$$

3.8 Proposed Algorithm

The major objective on this work is to recognise faces from a video sequence irrespective of the orientation of the face. It is undeniably true that when a face is captured using a camera, the video sequences may contain faces oriented in different angles. When database contains stored faces with fixed orientation, matching the dynamic input with the fixed content from the database need not give an accurate result for every frame. To increase the efficiency of recognition rate, the following methodology is implemented.

In the proposed work, a generic shape based approach is followed to recognise the angle of orientation of the face image. Face orientation from all the three axes are considered to conclude the result. To find the orientation, face features namely two eye points and tip of nose are considered. This method first detects the location of the eyes and the tip of the nose. By considering the distance between the eye points and the relationship of them with the nose tip, the angle of rotation is calculated. To calculate the angle of rotation of the face with respect to the camera, following considerations are carried out.

The face is captured from the three planes namely, X-Z plane, X-Y plane and Y-Z plane. Let us assume that the face is planar and parallel to the X-Z plane. Consider the eye points of a face as E1, E2 and is parallel to the Y-axis. The mean position formed by the eyes and nose of the face lies in the Y-Z plane.

Case i: Considering X-Z plane and rotation along Y axis clockwise and anticlockwise at 90°

In the first case the face that is to be captured by the camera is assumed to be in the X-Z plane and rotation is considered along the Y axis. With this assumption rotation along both clock wise and anti-clock wise directions are considered.

Figure 3.13 represents the three axes X, Y and Z and when the plane of rotation is along the Z axis. As a first step to finding the angle formed due to rotation, an assumption is made as there takes place a rotation along Z axis. Both clockwise and anti-clockwise directions of rotation are considered.

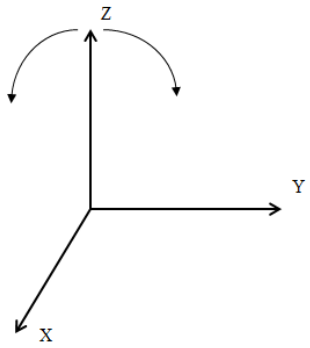

Figure 3.13: The Planes and the Rotation with respect to z-axis.

In this case, the two eye points and the nose tip are considered as the feature points. The term E_1 denotes the left eye point and E_2 denotes the right eye point. The term N is used to signify the nose tip of a face as depicted in Figure 3.14.

In this case, $\Delta E_1\ E_2 N\ ||\ Y - Z\ plane$ (3.2)

Equation (3.2) indicates that the points considered E_1, E_2 and N lie on the Y-Z plane.

The scenario is depicted in Figure 3.14. On rotation with respect to 90° the two eye point E_1 and E_2 overlaps and forms an angle with respect to

the plane of observation. From Figure 3.14 the distance between N and E_2 is depicted as $d'(= d)$.

As the eye points $E_1 \, and \, E_2$ overlap we can conclude that the two feature points E1 and E2 can be equal and can be represented by the equation (3.3)

$$E_1 = E_2 \tag{3.3}$$

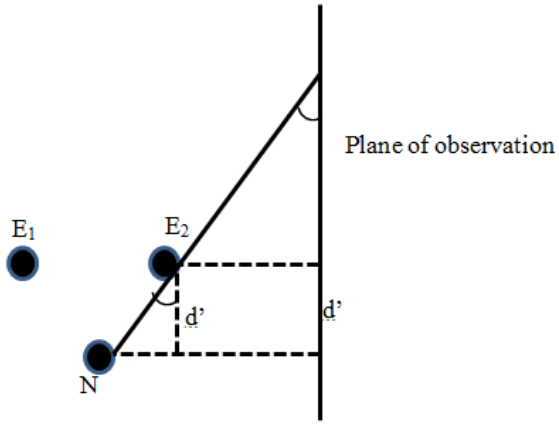

Figure 3.14: Plane of Rotation along Z-axis Clockwise.

From the above observation, we can state that the angle formed by the feature points as represented in equation (3.4)

$$\mathrm{E_1\hat{E}_2N} = 90° = \mathrm{E_2\hat{E}_1N} \tag{3.4}$$

$$\text{and } \hat{\mathrm{N}} = 0° \tag{3.5}$$

Hence, it can be concluded that due to a rotation of 90° the angles formed by $E_2 \, and \, \hat{E}_1$ increases to 90° and N decreases to 0°.

The separation between the eyes and nose is ascertained as derived in equation (3.9),

$$\frac{d\prime}{E_2N} = \cos\theta \tag{3.6}$$

$$\mathrm{d}' = \mathrm{E_2N}\cos\theta \tag{3.7}$$

$$\theta = 90° - \hat{E}_2 \tag{3.8}$$

$$d' = E_2N \sin E_2 \tag{3.9}$$

The second scenario when the eye points and nose tip is in the X-Y plane and there exist a rotation of 90° about Z axis in a clockwise direction is explained here.

In this case, the two eye points and the nose tip are considered. The term E_1 denotes the left eye point and E_2 denotes the right eye point. The term N is used to signify the nose tip of a face.

Due to rotation along Z-axis in a clockwise direction, $E_2\ and\ \hat{E}_1$ increases to 90° and N decreases to 0°. As point $E_1\ and\ E_2$ overlaps it can be stated as given in equation (3.10):

$$E_2 = E_1 \tag{3.10}$$

$$E_1\hat{E}_2N = 90° = E_2\hat{E}_1N \tag{3.11}$$

$$\text{and } \hat{N} = 0° \tag{3.12}$$

To ascertain the separation between the eyes and nose consider the Figure 3.15. From the Figure 3.15, we can get the distance between eye and nose tip as

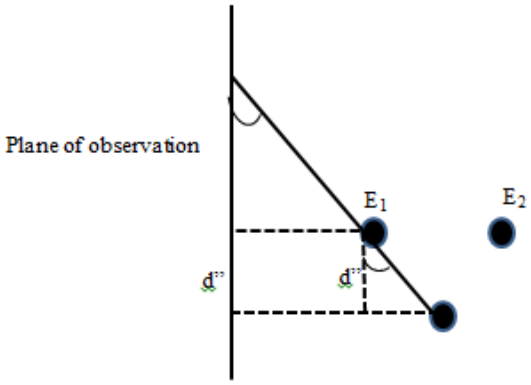

Figure 3.15: Plane of Rotation along Z-axis Anti clockwise.

$$d'' = E_1 N \cos \theta_2 \tag{3.13}$$

$$d'' = E_1 N \sin E_1 \tag{3.14}$$

Considering the eye points and the nose tip as given in Figure 3.16, KN denotes the altitude from the vertex N which is described as the distance using a variable d.

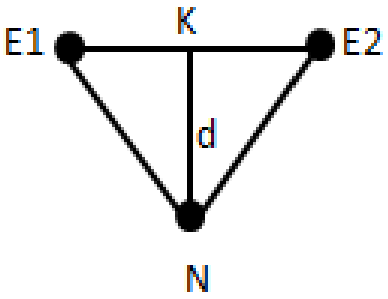

Figure 3.16: Distance between the nose point and eye points

The movement and the formation of the angle are shown in Figure 3.17 as the rotation takes place.

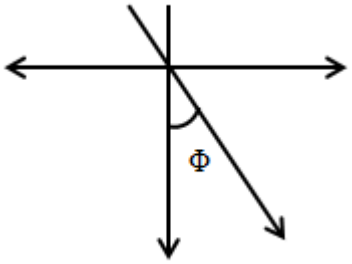

Figure 3.17: Formation of Angle as Rotation.

From the above observation it can be concluded that the angle at the eye increases with respect to rotation as described in equation (3.15).

$$E_1 = E_2 = \left(\frac{\pi}{2} - \varepsilon\right) sin^2\Phi + \varepsilon \qquad (3.15)$$

Φ is the angle formed due to rotation.

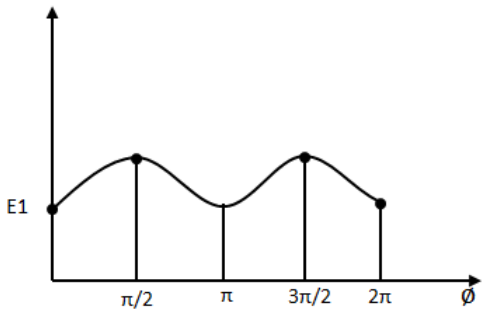

Figure 3.18: Graphical Representation of Angle Formation as Rotation.

$E_1 and\, E_2$ at any angle of rotation can be demonstrated as given in equation (3.16).

$$\hat{E} = (90° - \varepsilon) sin^2\Phi + \varepsilon \qquad (3.16)$$

Similarly,

$$\widehat{N} = f_2(\Phi) \tag{3.17}$$

$$\widehat{N} = N°cos^2\Phi \tag{3.18}$$

From equation 3.17 and 3.18 it can be concluded as

$$\Phi = \sin^{-1}\sqrt{\frac{\hat{E}-\varepsilon}{90°-\varepsilon}} \tag{3.19}$$

$$\Phi = \cos^{-1}\sqrt{\frac{\widehat{N}}{N°}} \tag{3.20}$$

The above observations were when the plane of observation is parallel to the Y-Z plane at a mean position at 90-degree rotation of the observing plane.

Considering the distance d as the angle formed due to rotation changes is illustrated in Table 3.2.

Table 3.2: Table Representing Distance between Eye Points as Rotation

Distance between E1 and E2	**Angle of rotation**
$\boldsymbol{l}$	0°
0	90°
$\boldsymbol{l}$	180°
0	270°

From Table 3.2, it can be concluded that at 0° of rotation the distance between the two eye points is l and as the rotation angle increases to 90° the distance between the eye points reduces to 0. At 180° rotation,

distance changes back to l and at 270° rotation changes back to 0. This observation can be graphically represented in Figure 3.19.

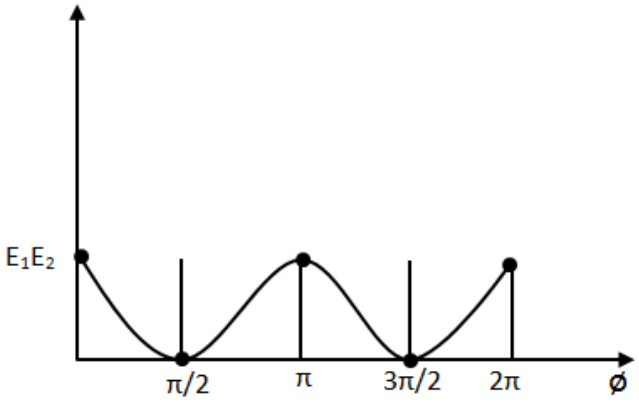

Figure 3.19: Graphical Representation of Angle formed as Rotation.

From the above observation,

$$E_1E_2 = l\,cos^2\Phi \tag{3.21}$$

$$NK = d \tag{3.22}$$

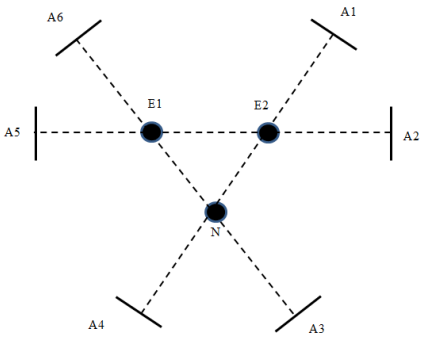

Figure 3.20: Lines Formed by Feature Points to the Plane

At the point when a face is rotated at 90 degrees concerning the observing plane, the line framed as for the eyes E_1 and E_2 gets to be perpendicular to the plane. Consequently, E_1 and E_2 overlap one another. Therefore at places A1, A2, A3, A4, A5, A6 the observing plane perpendicular lines of two out of three points of a triangle.

Table 3.3: Overlapping Features with respect to points on Face.

Point1	**Point2**	**Overlapping features**
A1	A4	E_2 and N overlap
A2	A5	E_1 and E_2 overlap
A3``	A6	E_1 and N overlap

According to the initial assumption, line E_1, E_2 is parallel to Y axis so that triangle E_1E_2N || Y−Z plane.

That is, E_1E_2 || Y axis and altitude NK perpendicular to Y-axis. Assume that a plane is passing through E_1E_2 || X−Y plane.

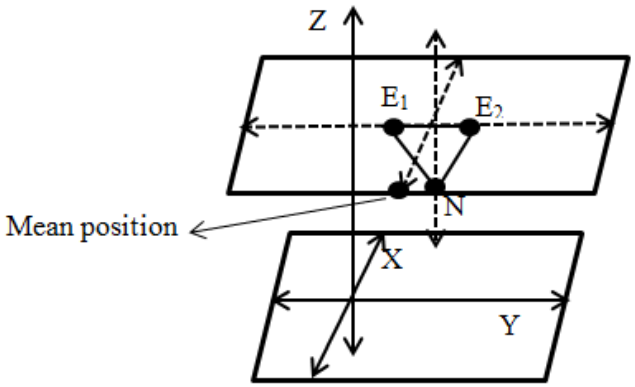

Figure 3.21: The Feature points on the Plane on Rotation

From equations (3.19) and (3.20), Φ the angle of rotation of point of observation with respect to mean is given as:

$$\Phi = \sin^{-1}\sqrt{\frac{\hat{E}-\varepsilon}{90°-\varepsilon}} \tag{3.23}$$

$$\Phi = \cos^{-1}\sqrt{\frac{\hat{N}}{N°}} \tag{3.24}$$

Case ii: Considering Y-Z plane and rotation along X-axis clock wise and anticlockwise at 90°

Assume that the plane of observation is perpendicular to the plane1. Let the plane of observation be at an angle ω with plane1. The first case is agreed if ω = 90°. Now at 90° rotation to the right,

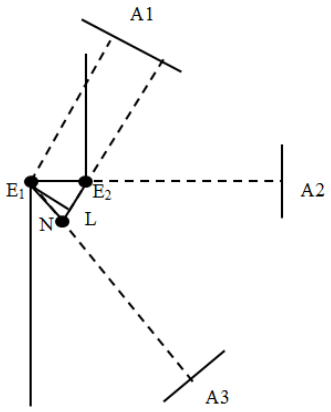

Figure 3.22: Feature Points with respect to Different Planes

From the above Figure 3.22, we observe that at A1, E2 and N overlap.

$$\hat{E}_2 and\ \hat{N} = 90° \tag{3.25}$$

$$\hat{E}_1 = 0° \tag{3.26}$$

Apparent Length = $E_1L = E_1N \sin N°$ (3.27)

From A1 to A2, as the plane revolves E2 overlaps N to E1.

$$E_1E_2 = l\,cos^2\omega \tag{3.28}$$

$$NK = dsin^2\omega \tag{3.29}$$

Where l is the distance between the eye points and ω is the angle formed as the face rotates to different orientation along X-axis. This observation can be depicted as in Figure 3.23.

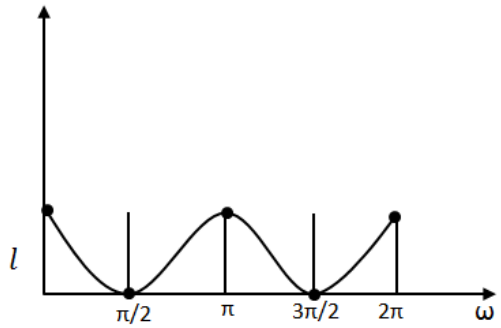

Figure 3.23: Angle formed on Rotation with respect to the distance ***l***

Analysing Figure 3.23, at 0° rotation, the distance between the eye points is l and as the rotation angle increases to 90°, the distance l becomes 0 as the eye points E_1 and E_2 overlap each other. At 270° rotation, distance tends to 0 as the points overlap and again to l at 360° rotation angle.

Case iii: Considering X-Z plane and rotation along Y axis clockwise and anti-clockwise at 90°.

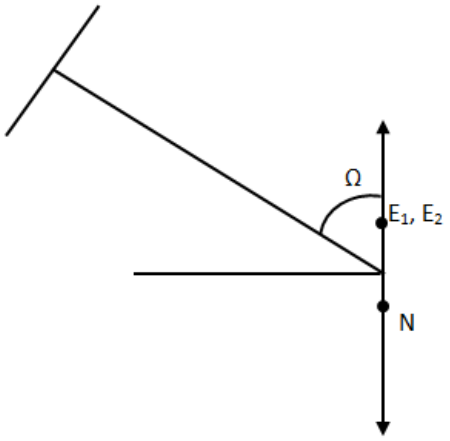

Figure 3.24: Angle formed on Rotation along Y axis.

Table 3.4: Distance between the Feature points on Rotation.

Angle of rotation	**Distance between the eyepoints E_1E_2**	**Distance NK**
0°	l	0
90°	l	d
180°	l	0

Observation from Table 3.3, it can be concluded that,

$$E_1E_2 = l \tag{3.30}$$

$$NK = dsin^2\Omega \tag{3.31}$$

The three rotation planes are Φ rotation, ω rotation and Ω rotation. At any point of observation at an angle θ with respect to X-axis, the angle θ depends on Φ and ω. Consider X-axis as the main axis for the initial reading.

Table 3.5: The angle formed with respect to different axes.

Axis	**Φ**	**Ω**	**Θ**
x-axis	0°	90°	0°
y-z plane	90°	0°	90°
x-axis and far from mean	180°	0°	180°

Consequently separating the eye points and the tip of the nose can give a clear thought of the edge at which the face is situated to.

The above finding can be summarised as in Figure 3.18. In the figure, the angle of rotation along the X-Y plane is depicted by Φ, an angle of rotation along the Y-Z plane by ω and angle of rotation of X-Z plane by Ω.

The final transformation is given in Table 3.5 from equations (3.21) and (3.22) where the rotation was considered along the Z axis, equation (3.25) and (3.26) with rotation along the x-axis and from equation (3.27) and (3.28) where the rotation is considered along Y axis respectively.

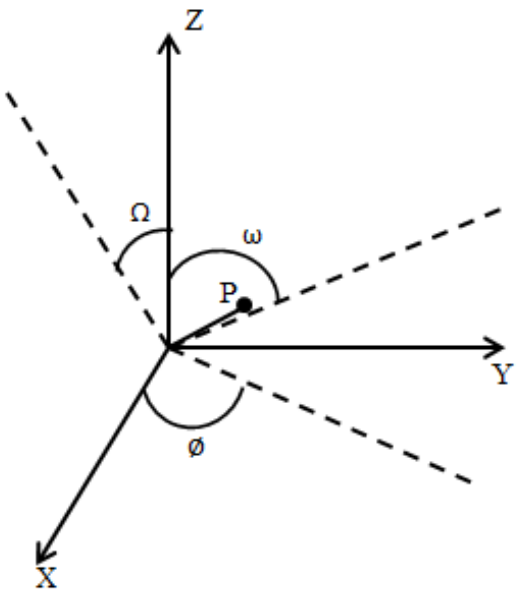

Figure 3.25: Representing the Formation of Rotation with respect to three axes.

Figure 3.25 shows the formation of three different angles of rotation about all the three axes. Observing this figure it is clear that when there is

a rotation along Y-Z plane of the object, it forms an angle ω. Upon rotation along X-Y plane, the angle formed is Φ and on X-Z plane, Ω.

Table 3.6: Distance between the Feature points on Rotation with respect to three axes.

	X-Y	**Y-Z**	**X-Z**
E_1E_2	$lcos^2\Phi$	$lcos^2\omega$	l
NK	d	$dsin^2\omega$	$dsin^2\Omega$

These aides into the procedure of breaking down the various poses of faces to match it with the images stored in the dataset. The above finding is condensed underneath to perceive a face from various orientations.

Table 3.7: Training Algorithm.

V: Defines the video sequence as the input.

I: Frame that is chosen from the video.

G: Set of gallery images.

N: Number of gallery subjects.

Creating a training set:

Step 1: For the Face image, we extract the skin tones using Gabor features.

Step 2: Also the texture feature prediction is done for the case of fixed poses and most common lighting variations. This is done by inserting noise patterns and generating the multi face image set.

Step 3: With all multi-face images generated, use Curvelet Transform to extract the features vector for each image and added to the training set.

Table 3.8: Testing Algorithm.

Step 1: In each frame of video, detect the face using face detection algorithm – Voila Jones. Step 2: Extract faces from the frames. Step 3: Pre-process the face image by resizing to standard size. Step 4: On the pre-processed face image, Gabor wavelet is executed to extract the skin texture. Also the Curvlet transform is executed to extract the feature vector for the image. Step 5: Matching is done against the training data set using minimum distance matching. Step 6: Once it is matched the training image, provide the result. Step 7: Between subsequent frames, the SIFT matching is done to locate the recognized face and mark in the video frames.

In the proposed work, the appearances are arranged into different logs based on the perception of points framed when the eye point and tip of the nose are considered from the input face image.

3.9 Performance Analysis

Video sequences are considered from YouTube datasets and Honda/UCSD dataset for this experimentation. There is a pose variation of subjects in the video sequence of each dataset. The variation in pose is considered for ten subjects of this dataset and is used as test data. Five face images per subjects are used as the training data for the experiment.

The result of the experiment is presented in Table 3.9.

i. Recognition rate on You tube datasets

The proposed method is tested on YouTube dataset. The dataset consists of video clips of mostly actors/actresses and politicians. The videos in this dataset are recorded at high compression and are of low resolution. This indicates that the videos are of low quality and with noise. Each video consists of hundreds of frames. This dataset has videos of about 1500 video clips each containing hundreds of frames. The videos in this dataset are considered for the experiment as they contain videos with large variation in face pose, illumination and expression.

In order test the recognition rate and time for recognition with respect to the proposed method, the video clips of 10 celebrities are collected from the dataset.

The proposed work is compared with patch based and S-LNMF as the literature survey carried out gives a better recognition rate using the above methods on occlusion and faces with different poses.

Table 3.9: Recognition Rate with respect to different methods.

Video sequences	**Proposed approach**	**Patch based**	**S-LNMF**
1	96	95	92
2	95	94	90
3	95	93	88
4	95	93	86
5	94	92	85
6	94	92	84
7	93	91	82
8	93	87	81
9	92	86	80

10	92	84	80

From the test performed on ten different videos from YouTube dataset, Table 3.9 shows an improvement in recognition rate compared to existing methods. From Table 3.9 it is clear that when there is a large variation in pose, the proposed method works well compared to existing methods.

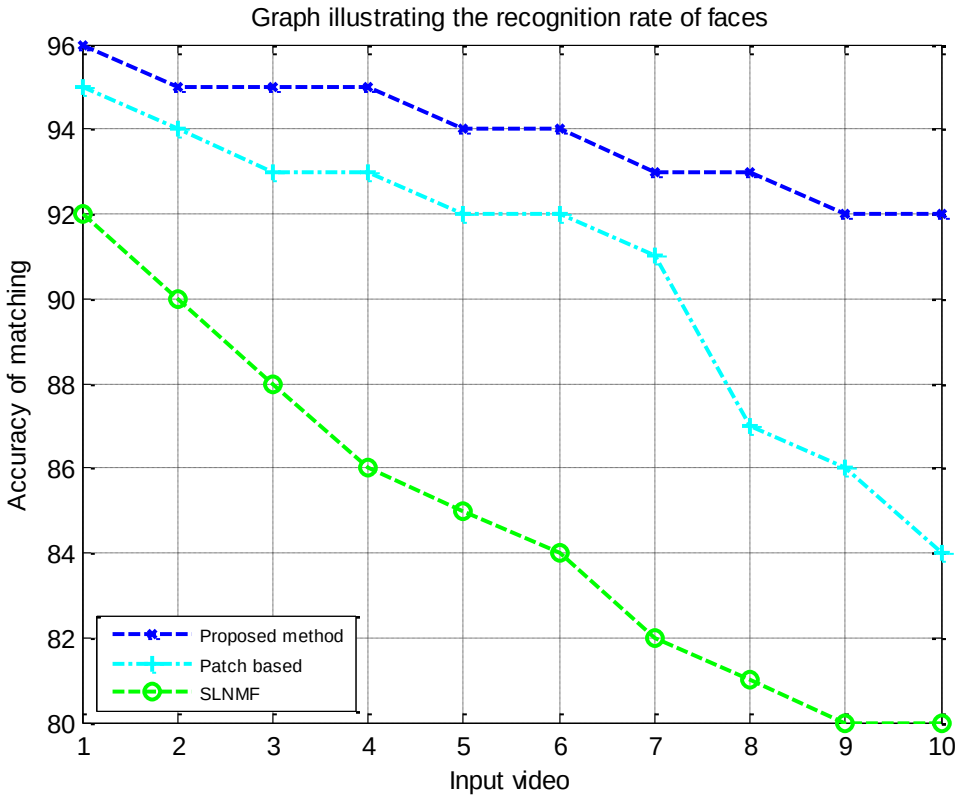

Figure 3.26: Graphical Representation of Recognition Rate

From Figure 3.26, it is significant that the recognition rate of the face from an input video sequence using the proposed method is comparatively better than existing methods. When compared with recognition rate on videos 8, 9 and 10 the recognition rate of existing

methods tend to come down whereas the proposed model shows a stable recognition rate.

ii. Recognition rate on Honda/UCSD datasets

Videos in this dataset include large variations in out-of-plane (left/right and up/down) head movement as well as in facial expression. The set contains several dozen subjects, each one appearing in at least two sessions. For this experimentation, 5 face images are stored in the training set of each subject.

Table 3.10: Recognition Rate with Honda/UCSD Dataset

Video sequences	**Proposed approach**	**Patch-based**	**SLNMF**
1	97	95	92
2	96	94	90
3	95	94	90
4	92	93	89
5	91	92	89
6	91	92	85
7	90	90	84
8	89	88	82
9	89	87	82
10	88	87	81

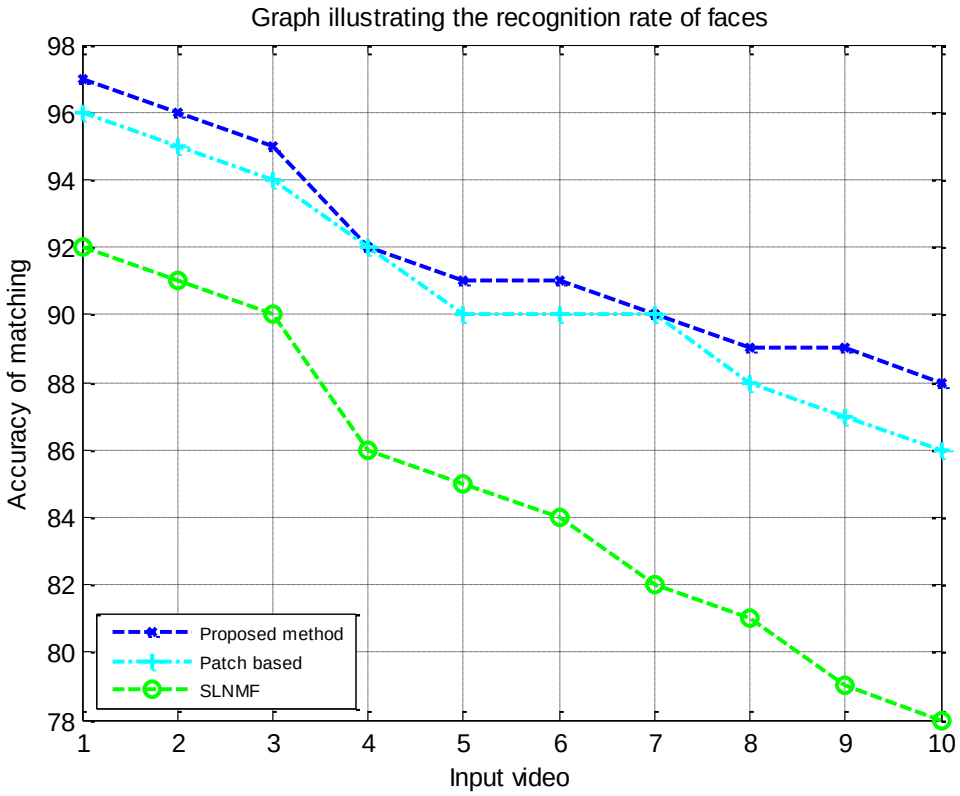

Figure 3.27: Comparative Analysis of Recognition Rate with Honda/UCSD Dataset

Figure 3.27 describes the recognition rate of videos with respect to Honda/UCSD dataset. This dataset has face with very large pose variation both left to right and top to bottom.

iii. Time required for recognition on You tube dataset

Table 3.11: Time required for Recognition using You Tube Dataset

Video sequences	**Proposed approach**	**Patch-based**	**SLNMF**
1	07	11	13
2	08	12	14
3	10	13	15
4	12	14	18
5	14	17	21
6	15	18	22
7	17	19	24
8	20	22	26
9	21	24	28
10	22	25	29

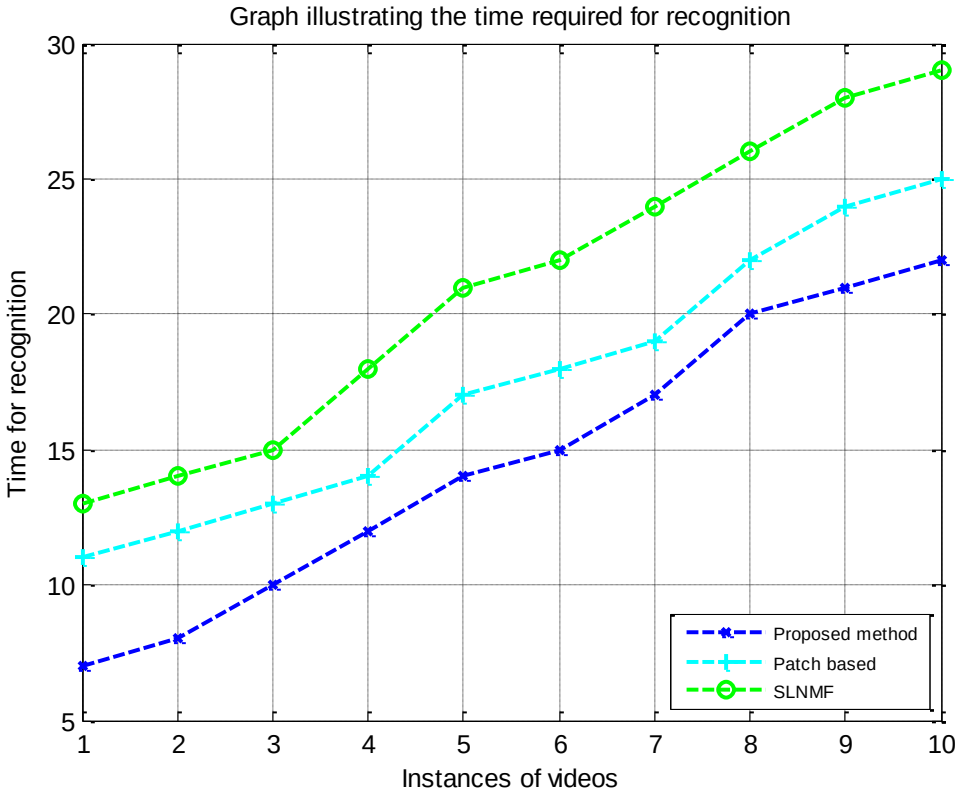

Figure 3.28: Comparative Analysis of the Time Required for Recognition.

Figure 3.28 describes the time required for recognising the faces from video sequences. To test the rate of time at which the recognition occurs, as an input, YouTube data set is considered. The time for the proposed method is comparatively less as with the existing methods.

iv. Time required for recognition of Honda/UCSD database

In this case Honda/UCSD dataset is used for analysis. This dataset includes faces with varying pose in all direction.

Table 3.12: Time Required to Recognise using Honda/UCSD Dataset.

Video sequences	**Proposed approach**	**Patch-based**	**SLNMF**
1	07	10	15
2	08	11	16
3	10	12	17
4	10	12	20
5	12	14	20
6	13	17	22
7	16	19	23
8	17	21	25
9	17	21	25
10	18	22	27

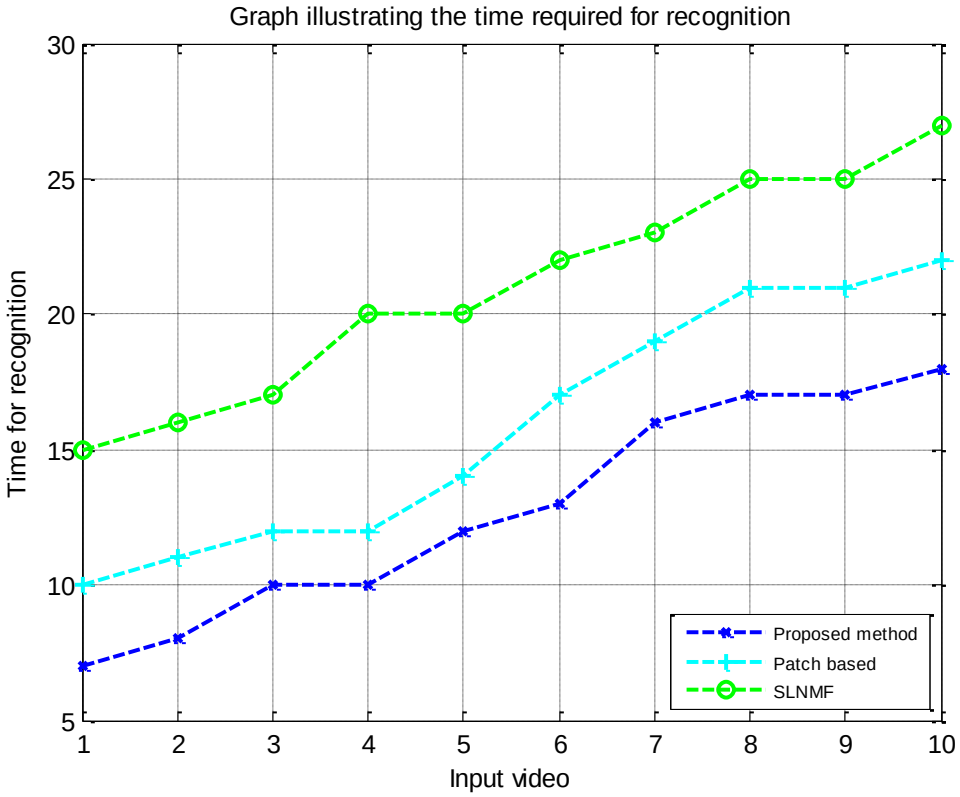

Figure 3.29: Comparative Analysis of the Time Required to Recognize faces.

Figure 3.29 describes the time required for recognising the faces from video sequences. To test the rate of time at which the recognition occurs, as an input Honda/UCSD data set is considered. This dataset has faces with varying poses inclined to all directions. The time for the proposed method is comparatively less as with the existing methods.

3.10 Summary

In this chapter, a method for video-based face recognition on pose is proposed. The angle formed with respect to a face is calculated taking into consideration the eye points and nose tip of a face. In this case three scenarios are considered namely, considering the X-Y plane with rotation along Z-axis in clockwise and anti-clockwise direction, considering the Y-Z plane with rotation along X-axis in clockwise and anti-clockwise

direction and considering X-Z plane with rotation along Y-axis in clockwise and anti-clockwise direction. Depending on the angle of orientation of the face with respect to the geometrical points considered, the input face image is matched with the gallery sets. To extract the feature points, Curvelet transform is used. The Curvelet Transform is coined because they are good at extracting the edge features. The extracted features are classified using minimum distance classifier. In order to test the efficiency of the system, YouTube dataset and Honda/UCSD data sets are used. Experiments show that this method reaches considerable recognition rate in recognising faces of varying pose from an input video. The comparative study on time required to recognise a face shows this approach accounts lesser time compared to state-of-arts methods.

The limitation with this system is that it cannot handle large changes in illumination and expression. These limitations will be addressed in future work.