Chapter 3

FACIAL FEATURE POINTS (FFP) LOCALIZATION

Objective

1. To prepare age-prototype-landmark-points template,
2. To fit the template landmark points and localize the facial landmark points in the given face image.
3. To extract the Facial Parts using the landmark points.

3.1 Introduction

Identification of facial feature points (or landmark points) is the first step in synthesizing age progressed facial images as well as many other facial image processing applications like, face detection and recognition, expression classification, age grouping, face modeling, video surveillance *etc.* Face landmarks are defined as certain characteristic points on the face, which are used to determine facial features. A facial landmark is a point of correspondence of different available facial parts like eye, nose mouth, chin, cheek, eyebrows *etc.* The shape of a face can be drawn easily if all the corresponding landmark points are given. In general, the landmarks aim to find the correct positions of the facial components in a face image. An example of facial landmark points are shown in Figure 3.1.

For a given a facial image, the respective landmark points can be localized using manual or automatic methods. The available methods mainly separated in two categories: (a) texture-based and (a) shape-based approaches. In texture-based approaches local texture e.g. pixel values of different facial parts have considered for the computation, whereas in shape-based approaches, location of the facial feature points are considered for the process.

Some major texture-based facial feature point extraction techniques can be found in [22, 28, 50, 55]. R Feris et al. [22] have proposed a technique for facial feature localization using Hierarchical Wavelet Networks, E. Holden et al. [28] have been proposed a technique using log Gabor wavelet networks by employing geometry cross-ratios relationships, S. K. Paul et al. [50] have proposed a technique using cumulative histogram for extraction of facial feature points and M. J. Reinders et al. [55] have used a technique of neural networks which can locate the facial

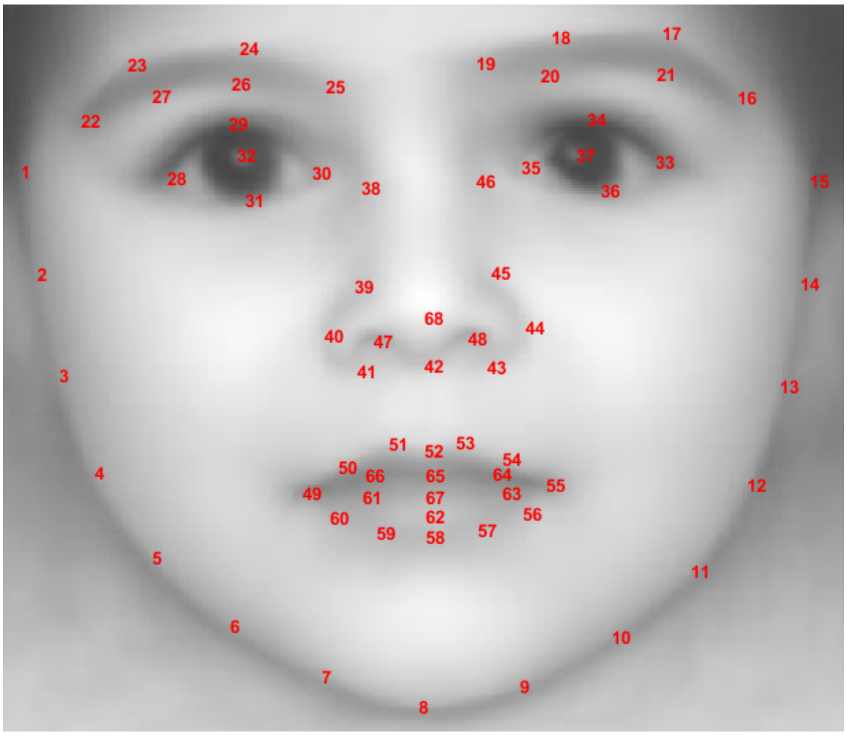


Figure 3.1: A set of 68 landmark points marked in prototype face image of 01-05 years old child age group.

features in sequences of images. The shape-based facial feature point extraction approaches are categorized in Active Appearance Model (AAM) and face detectors based techniques. Some of them are view-based direct appearance models [84], view-based active wavelet network [31].

Again some techniques have been proposed using combination of texture-based and shape-based algorithms, some of them are AdaBoost with Shape Constrains [15], elastic bunch graph matching [79], Probabilistic-like outputs for 3D shape constraint[13].

Recently, deep learning based facial landmark point detection techniques are very popular due to comparatively high accurate results. Some recent work of facial landmark point detection techniques based on deep learning have discussed in [5, 23, 54, 66, 74, 86]. Wissam J. Baddar et al. [5] have proposed a new facial landmarks detection method based on deep learning with facial contour and facial components constraints. Their proposed Deep Convolutional Neural Networks (DCNNs) for facial landmark detection consists of two deep networks: one DCNN detects landmarks constrained on the facial contour and the other one detects landmarks constrained on facial components. Their method branches the network at higher layers to capture the complex local facial components features. Frigieri et al. [23] have proposed a method which can find accurate and fast facial landmark points and can be applied in depth images inside a car. Ranjan et al. [54] have presented an algorithm based on deep multi-task learning framework, which can recognize gender, detect faces, localize the landmark points and also can estimate the diffrent poses of a given face images. Sun et al. [66] have proposed a facial point detection using cascaded deep convolutional network, which can estimate the positions of keypoints of facial parts. Basically it can find the five major facial of points of a given face image. Xin Tang et al. [74] have proposed method known as SEMI, which has been developed using convolutional neural network (CNN) for a semi-supervised landmark detection technique. This method can detect facial components and landmarks simultaneously. Using deep multi-task learning, Zhang et al. [86] have proposed a method which improve the results of facial landmark point detection.

There are many works on facial landmark detection including the recent deep-learning based methods. Advantages and limitations of Deep Learning based approaches can be found in [10, 34, 36, 68], which includes as follows:

Advantages of Deep Learning Approach

1. Deep learning has best performance on problem solving in the area of image, speech, language processing.

2. It reduces the need of feature engineering which is one of the most time-consuming fragments of machine learning technique.

3. The architecture of deep learning can be relatively easily altered as per the requirement of new problems.

Disadvantages of Deep Learning Approach

1. Very large amount of data is required for deep learning approached problems.

2. To train the system, it is very much expensive in terms of computation. To train a complex model it needs more than a week with expensive equipped GPU also.

3. The feature extraction does not have good theoretical bases.

4. The learned model can not be easily understand and it can not be easily explained.

Landmark Points	Facial Parts (ID)	Triangle IDs
1 to 18 and 22 to 24	Face Edges (1)	All Traingle IDs
28 to 32	Left Eye (2)	[10 21 28 45]
33 to 37	Right Eye (3)	[112 111 35 114]
22 to 27	Left Eye Brow (4)	[8 27 30 34]
16 to 21	Right Eye Brow (5)	[98 115 109 108]
38 to 46 & 68	Nose (6)	[41 100 103 43 47 94 75 16 53 50 73 102 101]
49 to 67	Mouth (7)	[56 57 58 15 60 69 52 42 70 62 19 81 61 72 71 74 83 64 2 82 78 20 87]

Table 3.1: Facial features points and corresponding facial parts & Triangle IDs (Table 3.3). The visual representation is given in Figure 3.10.

Keeping in mind the limitations of deep learning approach, and since facial land marking is not the main objective of our goal, therefore, we present a semi automatic method for localizing landmark points in a given facial image in this chapter. This method is based on statistical approach, which does not need any training images, can act directly on the input images, and as a result it can produce the required outputs very fast.

This method requires the locations of the left eye, right eye and the midpoint of mouth to be demarcated manually. Other landmark points are demarcated by using a pre-existing age-prototype-landmark-points template for the specific age group and gender, adjusted with help of the three manually marked points. The age-prototype-landmark-points templates are already prepared for different age groups and are readily available. The process is detailed in subsequent sections.

3.2 Preparation of Age Prototype Landmark Points Template

We consider a set of 68 landmark points (Figure 3.1) which is the same set of landmark points as used in FG-NET dataset. The significance of the points is shown in Table 3.1. By landmark points template we mean a table showing the relative position of the landmark points on a unit scale. To prepare such a table, landmark points are manually marked in a model image. A model image is a mean image which is the average of n number of images. Again this model image is know as age prototype if this one is a mean image for a particular age group. This age prototype image generally have the main characteristics of a particular age group. Such an age prototype (Figure 3.3) is constructed by getting the average of a set of facial images which are of the same age group.

3.2.1 Age-Prototype of Age Group $[a - b]$

To find the age prototype (described by Tiddeman et al. [75]) age group $[a - b]$ (where a and b are minimum and maximum age of the group respectively), we collected a set of images of age group $[a-b]$ of same gender (either male or female). The following steps are followed to prepare the age prototype.

1. Input: n number of images $(x_1, x_2, \ldots, x_n)$ of age group $[a - b]$
2. Normalize each x_i, where $i = 1, 2, \ldots n$ by aligning and cropping the face parts of original image x_i into a fixed sized image of size $M \times N$,
3. $Sum_p^q = \sum_{i=1}^{n} x_i^{pq}$, where x_i^{pq} is the pq^{th} pixel value of image x_i, $p = 1, 2, \ldots M$ and $q = 1, 2, \ldots, N$, finds the sum of pq^{th} pixel values of images,
4. $AP_p^q = \frac{Sum_p^q}{n}$
5. $AP = [AP_p^q]$, AP is the Age-Prototype of age group $[a - b]$ of size $M \times N$.

Some Age-Prototype of different age groups are shown in Figure 3.2

Landmark points on the age prototype is marked manually and then their coordinates are normalized to the scale 0–1.

We have 14 different age groups for both male and female face images for the dataset. The age groups are $1 - 5, 6 - 10, \ldots, 66 - 70$ years (last group is based on the maximum older image given in the database, which is 69).

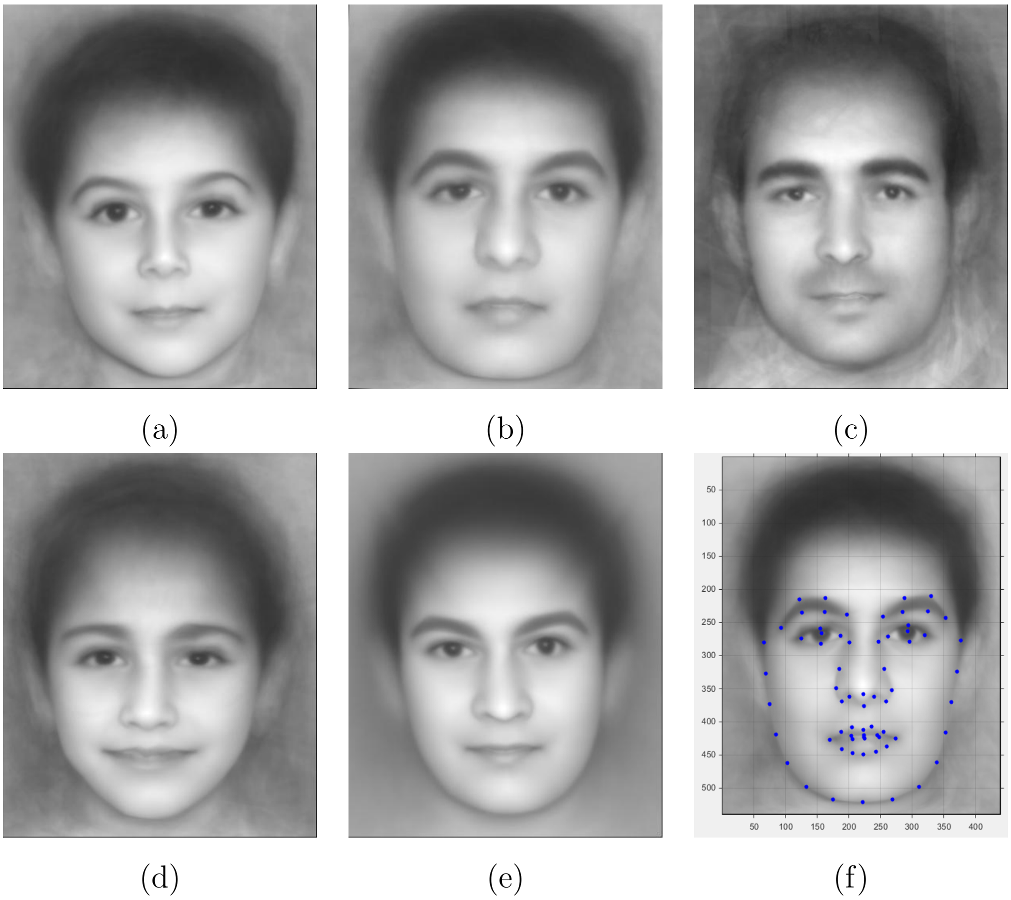


Figure 3.2: Age-Prototypes of different age groups. (a) Age group [6-10], gender: male, (b) Age group [11-15], gender: male, (c) Age group [26-30], gender: male, (d) Age group [6-10], gender: female, (e) Age group [11-15], gender: female, (f) Landmark plotted, Age group [11-15], gender: male

Figure 3.3: Prototype of face images or mean images of 01-05 years old child age group.

3.2.2 Perform Normalization of the Landmark Points to get Age-Prototype-Landmark-Points Template

Let L is a landmark point set of a prototype image. This set L is nothing but a set of 68 coordinate points of an image, and is given by,

$$L = \{(x_1, y_1), (x_2, y_2), \ldots (x_{68}, y_{68})\} \tag{3.1}$$

For this set L, any k^{th} coordinate point $(x_k, y_k), k = 1 \ldots 68$ of L is shown in Figure 3.1 and Table 3.1. Some special points of this set, like 32^{th} and 37^{th} are for two eye center positions, 68^{th} is for nose top position, 67^{th} is for mid of mouth position, *etc.*

In normalization, in the first step all the points are rotated *w.r.t.* the angle made by two eye locations 32 and 37 with the horizontal line (line connecting by points (x_{32}, y_{32}) and (x_{37}, y_{32}), Figure 3.1 and Table 3.2). This rotation is done with the Equation 3.6, described in the next section. After rotation, all the x and y coordinate values of this Ls are scaled to [0, 1], and it is done bye the following equation,

$$\begin{aligned} x'_k &= \tfrac{x_k - x_{min}}{x_{max} - x_{min}} \\ y'_k &= \tfrac{y_k - y_{min}}{y_{max} - y_{min}} \end{aligned} \tag{3.2}$$

where

$$\begin{gathered} x_{max} = max(x_k), x_{min} = min(x_k) \\ y_{max} = max(y_k), y_{min} = min(y_k) \\ k = 1 \ldots 68 \end{gathered} \tag{3.3}$$

Where (x'_k, y'_k) are normalised points of (x_k, y_k), and thus normalized point set $L' = \{(x'_k, y'_k) : k = 1 \ldots 68\}$ of L is obtained.

The normalized landmark point sets of face prototype shown in Figure 3.1 is given in Table 3.2. This normalized landmark point set is termed as Age-Prototype-Landmark-Points Template.

3.3 The Methodology for Facial Feature Points (FFP) Localization

The proposed methodology for facial feature points localization is described in the following steps (Figure 3.4):

SN	x	y	SN	x	y	SN	x	y	SN	x	y
1	0.00	0.20	18	0.67	0.01	35	0.63	0.20	52	0.51	0.62
2	0.02	0.36	19	0.58	0.04	36	0.73	0.23	53	0.55	0.61
3	0.05	0.51	20	0.66	0.06	37	0.70	0.18	54	0.61	0.63
4	0.09	0.65	21	0.81	0.06	38	0.43	0.23	55	0.66	0.67
5	0.17	0.78	22	0.07	0.13	39	0.42	0.38	56	0.64	0.71
6	0.26	0.88	23	0.13	0.05	40	0.39	0.45	57	0.58	0.74
7	0.38	0.95	24	0.28	0.02	41	0.43	0.50	58	0.51	0.75
8	0.50	1.00	25	0.39	0.08	42	0.51	0.49	59	0.45	0.74
9	0.63	0.97	26	0.27	0.08	43	0.59	0.49	60	0.39	0.72
10	0.75	0.90	27	0.17	0.09	44	0.64	0.44	61	0.43	0.69
11	0.85	0.80	28	0.19	0.22	45	0.60	0.36	62	0.51	0.72
12	0.92	0.67	29	0.26	0.13	46	0.58	0.22	63	0.61	0.68
13	0.96	0.52	30	0.37	0.21	47	0.45	0.46	64	0.60	0.66
14	0.99	0.37	31	0.28	0.25	48	0.57	0.45	65	0.51	0.66
15	1.00	0.22	32	0.27	0.18	49	0.36	0.68	66	0.44	0.66
16	0.91	0.10	33	0.80	0.19	50	0.40	0.64	67	0.51	0.69
17	0.81	0.00	34	0.72	0.13	51	0.47	0.61	68	0.51	0.42

Table 3.2: Normalized template landmark points of age group 01-05 years old child.

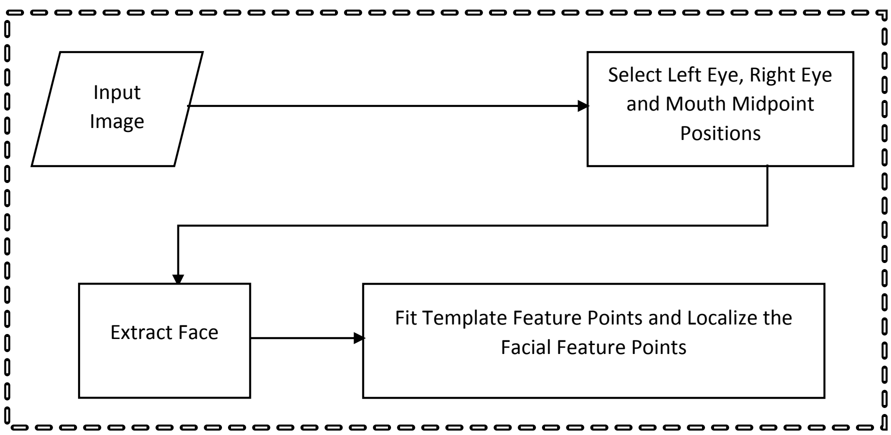


Figure 3.4: Block diagram of our proposed localization of facial feature points method.

1. Select eye center and mouth mid positions,
2. Extract Face,
3. Fit Template Feature Points, and Localize the Facial Feature Points.

3.3.1 Select Eye Center and Mouth Mid Positions

We manually select two eye centers positions, and middle point of the mouth position. We denote the co-ordinates of the left and right eye center as $e_1 = (x_1, y_1)$ and $e_2 = (x_2, y_2)$ respectively and the mouth mid position is $m_2 = (p, q)$.

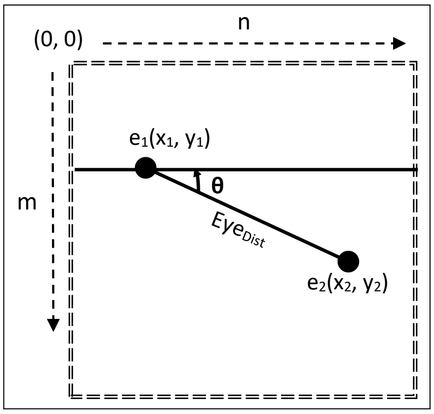


Figure 3.5: Geometric angle of eye line and horizontal line. Eye positions e_1 and e_2 and angle made by connecting eye line with horizontal line.

Based on statistical experiment for the measurement of face height, we found that distance between eye mid point and mouth mid point represents the face height. If this distance increases then the face looks like a long face and in reverse it looks like a short face.

Using two coordinates e_1, e_2, we calculate the eye distance and eye mid position m_1 (Figure 3.7) in the further steps. A distance fh (face height) which is the distance from eye mid point m_1 to mouth mid point m_2 can be computed by these three input points of eyes and mouth. Using these two distances we find the face width and face height.

To rotate and align the input image, we find the value θ (Figure 3.5), and which is the angle of two eyes made by the horizontal line using the equation,

$$\theta = \tan^{-1} \frac{y_2 - y_1}{x_2 - x_1} \tag{3.4}$$

After getting the rotation angle, we rotate the face with θ angle anticlockwise if $\theta > 0$, otherwise clockwise (Figure 3.5), *w.r.t.* the midpoint of the face image $c = (x_0, y_0)$, where $x_0 = m/2$ and $y_0 = n/2$ for the given image of size (m, n). This rotation of the image keeps the two eyes in horizontal line. After this rotation of angle θ, the new size (m', n') of the rotated image can found by some in-built function e.g. imsize in matlab, or computed by Equation 3.5 as shown in Figure 3.6.

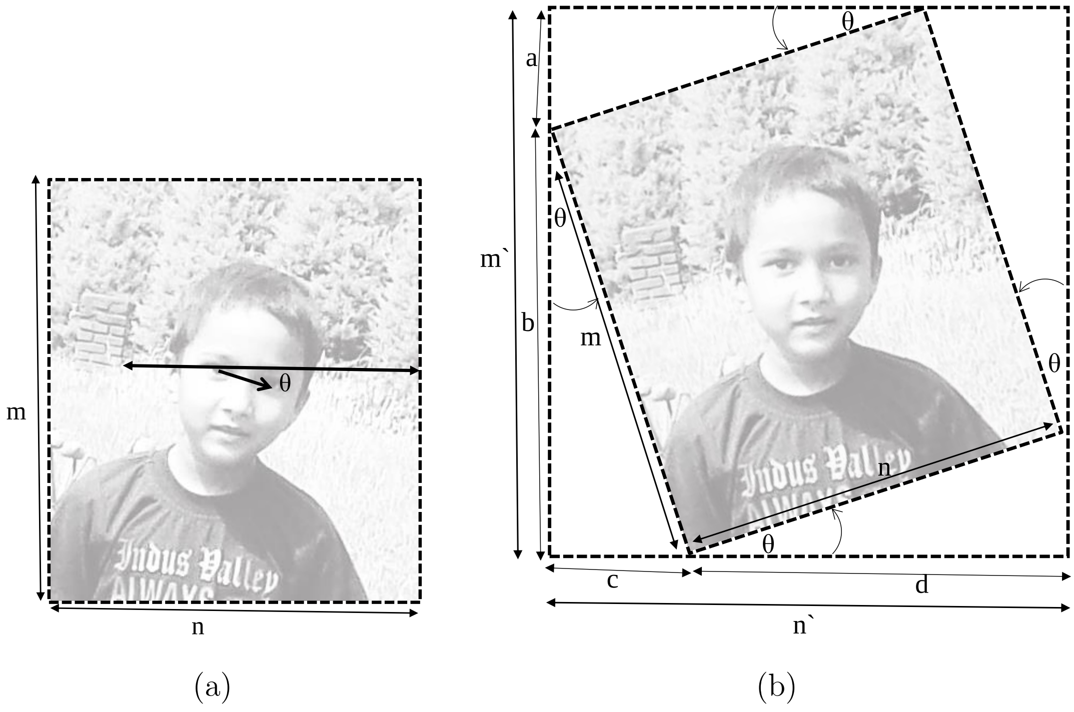


Figure 3.6: New image size (m', n') after the rotation with θ angle of an image of size (m, n), where $m' = a + b = n \times \sin\theta + m \times \cos\theta, n' = c + d = m \times \sin\theta + n \times \cos\theta$

$$
\begin{aligned}
m' &= n \times \sin\theta + m \times \cos\theta \\
n' &= m \times \sin\theta + n \times \cos\theta
\end{aligned}
\tag{3.5}
$$

The new center $c' = (x'_0, y'_0)$ of it, where $x'_0 = m'/2$ and $y'_0 = n'/2$. Accordingly we update the coordinates of the new eye center positions e_1 & e_2. We update these values of e_1 & e_2 with the following equation,

$$
\begin{aligned}
e_1 &= R \times (e_1 - c) + c', \\
e_2 &= R \times (e_2 - c) + c'
\end{aligned}
\tag{3.6}
$$

where the rotation matrix R is given by the following equation,

$$
R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}
\tag{3.7}
$$

Similarly we update the input mouth mid position also *w.r.t.* this rotation. Note that, in image axis system left top corner co-ordinate is (0,0), rather than $(0, m)$, where m is the maximum value of y-axis in the coordinate system as shown in Figure 3.5. In this step we have rotated the given image so that the new eye positions remain in the horizontal line of the resultant image. Till now the face part is not detected, therefore it has to be computed and in the next section we calculate the rectangular boundary region of the face part by the distance factor Eye_{Dist}, distance of two eyes given by the Equation 3.8, and cropped it out from the image, it is known as face detection and face extraction.

3.3.2 Extract the Face

In this step, face extraction is done with a semi-automatic method based on the selection of two eyes and mouth mid position as described in the previous step. In the previous step face is aligned properly based on two eye centers. The selected eyes make an angle θ with the horizontal line, then the given face is rotated with the angle θ, and new eye locations and mouth mid position have been updated accordingly.

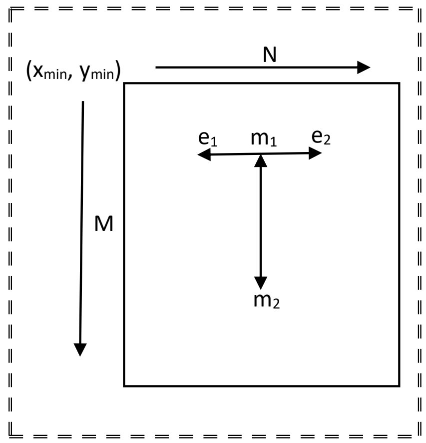


Figure 3.7: Face Model Preparation, (x_{min}, y_{min}) is the left top corner point, N, M are the *width* and *hight* respectively of a detected face image. The points e_1, e_2 are two eye positions, and m_1, m_2 are middle points of eyes mouth mid point respectively.

Now we detect the face part and compute the feature point locations of the preprocessed given image. The face part detection is done by selecting a rectangular region in the image whose two eye positions are in horizontal line. Region of the face part and facial feature point locations have been calculated by the distance factor Eye_{Dist} and fh which are the distances of two eyes and face height, distance from eye connecting line to mouth mid position ($\overline{e_1e_2}$ and $\overline{m_1m_2}$ in Figure 3.7). The distance factor Eye_{Dist} is the Euclidian distance of the two eyes $e_1 = (x_1, y_1)$, and $e_2 = (x_2, y_2)$ is given by the equation,

$$Eye_{dist} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.8}$$

and, the another distance factor fh is the face height, which is the distance from eye mid to mouth mid point is calculated by,

$$fh = \sqrt{(x_{midEye} - p)^2 + (y_{midEye} - q)^2} \tag{3.9}$$

where (p, q) is the mouth mid point and (x_{midEye}, y_{midEye}), is the middle point of two eye centers, which is obtained by the equation,

$$\begin{aligned} x_{midEye} &= (x_1 + x_2)/2, \\ y_{midEye} &= (y_1 + y_2)/2 \end{aligned} \tag{3.10}$$

To select the face part, identify the left top corner (x_{min}, y_{min}) and $(width, height)$ of a region in which the face part is located, refer to the Figure 3.7. These four parameters can be computed by the eye distance factor Eye_{Dist} and face height fh using the following equation,

$$\begin{aligned} x_{min} &= x_{midEye} - 1.2 \times Eye_{dist} \\ y_{min} &= y_{midEye} - .6 \times fh \\ width &= 2.5 \times Eye_{dist} \\ height &= 2.5 \times fh \end{aligned} \tag{3.11}$$

The constant value 1.2 indicates that left top corner position of the cropped image will be $1.2 \times Eye_{dist}$ far away in opposite of x-direction from the mid eye location, while value 0.6 indicates that the same position will be $.6 \times fh$ far away from the mid eye location, and the last one 2.5 directs to crop the face image of size of $2.5 \times Eye_{dist}$ and $2.5 \times fh$. These constant values (1.2, 0.6, 2.5) are selected based on several experiments, which shows that this amount is sufficient for selecting a face region from any given image where two eyes are located in the horizontal line.

Now we are ready to crop out the respective face part and will get the face region whose size will be $(M = width, N = height)$, and thus selection of the face part is done. The new location of the two eyes e_1, e_2, and mouth mid position m_2 can be found by equation,

$$\begin{aligned} e_1 &= e_1 - (x_{min}, y_{min}) \\ e_2 &= e_2 - (x_{min}, y_{min}) \\ m_2 &= m_2 - (x_{min}, y_{min}) \end{aligned} \tag{3.12}$$

end new eye middle point can be found by the Equation 3.10.

Thus the face part is extracted from a given image which is aligned horizontally with the connecting line of two eyes and accordingly we update the inputs of two eye point locations to keep them in horizontal line. In the next step we fit the 68 template facial feature points $P'_i, i = 1 \ldots 68$ as a part of our method of feature detection.

3.3.3 Fit Template Feature Points, and Localize the Facial Feature Points

Also, in our inputs, discussed in the earlier step, we recorded the three co-ordinate positions, namely, two eyes and mouth end points. After the preprocessing of our input, now we have to fit two eye positions of the template landmark points into the two selected eye positions, and mouth mid position into the selected mouth mid position, and with this aspect ratio we have to fit all other remaining template points also. To fit these template points, scaling and translation are required, which can be processed with the following few steps. Note that rotation of the template point is not required as the input face image is already rotated where two selected eyes are kept in horizontal line.

1. Input: $e_1 = (x_1, y_1)$, $e_2 = (x_2, y_2)$ and $m_2 = (p, q)$, eye coordinates of the given face image and mouth mid point respectively, which are in horizontal line.

2. Load $P'_i = (P'x_i, P'y_i), i = 1 \dots 68$, template landmark points,

3. Find the eye co-ordinates, $e'_1 = (x'_1, y'_1)$, $e'_2 = (x'_2, y'_2)$ and $m'_2 = (p', q')$ of the template landmark points, which are the 32^{th}, 37^{th} and 67^{th} points of P', where

$$\begin{aligned} (x'_1, y'_1) &= (P'x_{32}, P'y_{32}) \\ (x'_2, y'_2) &= (P'x_{37}, P'y_{37}) \\ (p', q') &= (P'x_{67}, P'y_{67}) \end{aligned} \tag{3.13}$$

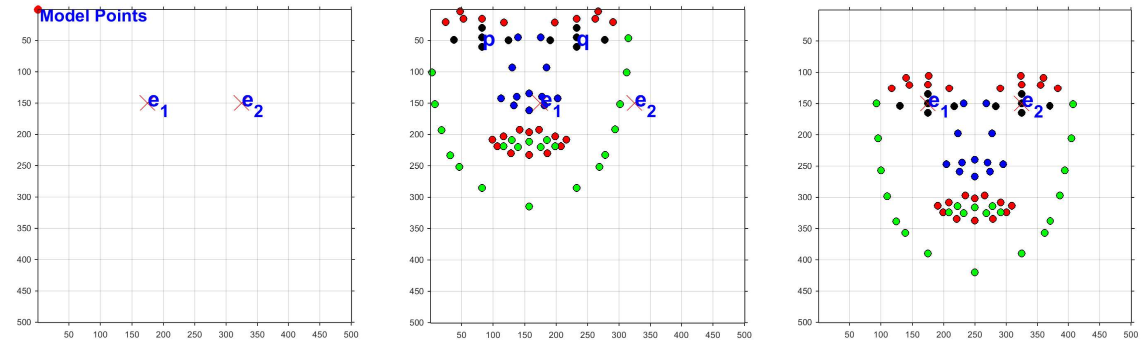
4. Find $(ModelEye_{Dist}, Eye_{Dist})$, and (fh', fh), the eye distances and face heights of template and given images

$$\begin{aligned} ModelEye_{Dist} &= x'_2 - x'_1 \\ Eye_{Dist} &= x_2 - x_1 \\ fh' &= q' - EyeMid'_x \\ fh &= q - EyeMid_x; \end{aligned} \tag{3.14}$$

   where $EyeMid'_x$ and $EyeMid_x$ are eye mid positions of template landmark point set and input face respectively.

5. Find scale factors S_x and S_y with

$$\begin{aligned} S_x &= Eye_{Dist}/ModelEye_{Dist} \\ S_y &= fh/fh' \end{aligned} \tag{3.15}$$



(a) (b) (c)

Figure 3.8: Template landmark points fitting. (a) Normalized template landmark points are shown in left top corner whose values are in [0, 1], and e_1, e_2 are target eye positions in where template landmark points have to be fitted, (b) template landmark points have been scaled $w.r.t$ to the value of eye distance e_1 to e_2 and face height fh, distance from eye mid to mouth mid point, (c) all template landmark points have been placed after translating them to e_1 and e_2.

6. Find $P_i, i = 1 \ldots 68$, the new points to be fitted in the given face, can be scaled and found by the equation

$$\begin{aligned} Px_i &= P'x_i \times S_x \\ Py_i &= P'y_i \times S_y \end{aligned} \tag{3.16}$$

7. Find translate factor T_x and T_y, to shift all the points such that the new eye points P_{32} and P_{37} are placed in the given eye points e_1 (or e_2) as shown in Figure 3.8, can be found by

$$\begin{aligned} T_x &= x_1 - Px_{32} \\ T_y &= y_1 - Py_{32} \end{aligned} \tag{3.17}$$

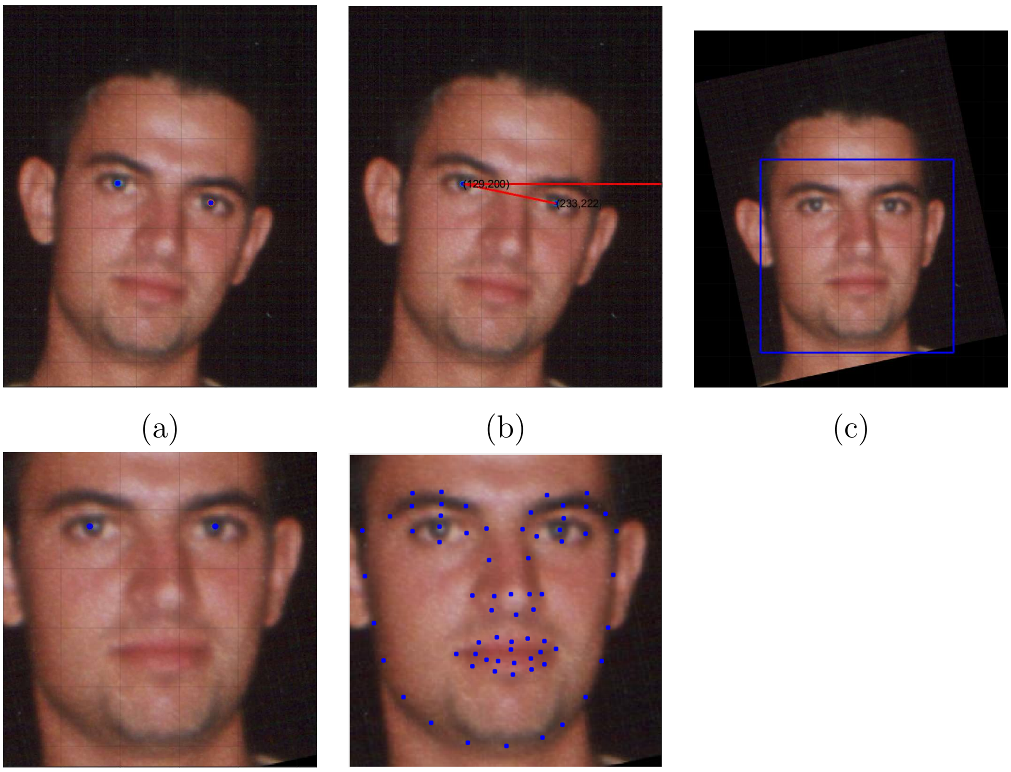
8. Translate all the points P_i to fit in the given image, shown in Figure 3.8(c),

$$\begin{aligned} Px_i &= Px_i + T_x \\ Py_i &= Py_i + T_y \end{aligned} \tag{3.18}$$

In the Figure 3.7, we have shown the details of important points in a face and the relationships among themselves.

In these way, we find the required face and the required facial landmark points $P_i, i = 1 \ldots 68$ from a given face image.

The template landmark points computation is completed in this step; and triangulations and identification of all important facial parts will be described in the next section.



(a) (b) (c)

(d) (e)

Figure 3.9: Steps in the process of finding the facial land mark points. In (a) two eye locations are selected, (b) angle θ is obtained from two eyes, (c) given image is rotated and estimated the facial region, (d) face extracted with updated eye locations, (e) extracted facial feature points.

3.4 Triangulation and Facial Parts Identification

In any face image processing, it is very important to identify the various facial parts of a given face image. For this necessity we have modeled a method which identifies the necessary facial parts and extract them from a face image. Also using this technique we process input face images for the proposed warp and morphing techniques, described in Chapters 4 and 5.

In the previous steps we get ready with the necessary 68 number of facial landmark points, by which we can locate and identify all the important facial parts of components like, eyes, eyebrows, nose, nostrils, mouth *etc.* The importance of 68 landmark points has been described earlier. We have applied and visualised these 68 number of points in a face prototype image (which is also known as mean or model image), and form fixed set of important triangles by which it is possible to identify the important facial parts as mentioned above. We get the 116 number of different non-overlapping triangles from these 68 number of points. We observed and tried to form maximum numbers of triangles in a specific region so that a set of some triangles can represent an unique object of the face image which is known as a particular facial part or a component. The structure of the triangles are shown in Figure 3.10.

After getting the 116 non overlapping triangles, a unique identification ID is given for each triangle, which are known as $TriID_k, k = 1 \ldots 116$, where every triangle is represented by a unique set of three landmark or feature points $[P_1^k, P_2^k, P_3^k]$ to represent k^{th} triangle $TriID_k$. The complete list of triangle IDs and respective feature point IDs are given in Table 3.3.

Again a set of unique triangle $TriID$s represent a unique facial part of a given face. For example facial part "left eye" is represented by the unique set of triangle $TriID$s $[TriID_{10}, TriID_{21}, TriID_{28}, TriID_{45}]$ whereas "right eyebrow" is represented by $[TriID_{98}, TriID_{115}, TriID_{109}, TriID_{108}]$. The complete list of such set of triangle $TriID$s and respective feature point P; and the respective set of boundary points Ps of the facial part is given in the Table 3.1, and visualised in Figure 3.10 to get the basic idea of this proposed method.

Thus we identified and mark every facial parts with some unique triangle IDs and a set of unique boundary points Ps. For example the facial part of the left eye is represented by triangle $TriID$s (10 21 28 45) and this triangle ID is represented by a set of points which includes point Ps (28, 29, 30, 31), shown in bottom image of

Figure 3.10 and Table 3.3. For the all other facial parts, the respective information has been given in Table 3.1. In this way we proposed the triangulation system which can identify all major facial parts of a given face. In the next section we present the method to extract all the important facial parts automatically using this facial feature or landmark points and triangle IDs.

3.5 Facial Parts Extraction

In this step we describe a method for extracting the facial parts from a given face image. In the previous section we have explained the method for localization of facial landmark points, and identifying all the facial parts from a given face. Some subsets of these facial landmark points represents some specific facial parts such as left eye, left eyebrow, right eye, right eyebrow, nose, mouth *etc.*, which technique has been described in the previous section.

Since every facial parts have been identified in the previous step, extraction of such parts is very simple. The method to extract such facial parts is described with the following steps:

1. Input: (k, F); k is the part number which represent the k^{th} facial part of a given face image F (Table 3.1),
2. Output: F_k; k^{th} facial part,
3. $B^k = \{(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)\}$, a set of n number of boundary points of F_i, which can be found from the given Table 3.1,
4. T is a blank dummy image of the same size (M_F, N_F) of C_F, whose pixel values are 0, is called as blank image.
5. Create a polygon in blank image T with the boundary points given in B^k, and fill the polygon with a specific color x, by default value of this x is 0 *i.e.* black color,
6. Update the F for every $(i, j)^{th}$ position with color x, *i.e.* replacement with blank if the corresponding $(i, j)^{th}$ value of T is blank *i.e.* x, $\forall i = 1 \ldots M, j = 1 \ldots N$.

   This process will remove all the unwanted information from the image F and will keep the necessary information (pixel values) inside the given boundary

Tri	Points			Tri	Points			Tri	Points			Tri	Points		
IDs	P_1	P_2	P_3	IDs	P_1	P_2	P_3	IDs	P_1	P_2	P_3	IDs	P_1	P_2	P_3
1	28	22	27	30	24	26	27	59	61	62	59	88	10	9	56
2	57	63	56	31	26	25	29	60	66	50	51	89	44	13	12
3	59	7	6	32	27	26	29	61	62	67	63	90	11	10	55
4	3	2	31	33	25	19	46	62	62	66	67	91	55	12	11
5	49	40	41	34	24	25	26	63	59	58	7	92	43	44	12
6	23	22	1	35	37	34	33	64	62	63	57	93	13	44	36
7	27	29	28	36	38	30	25	65	66	51	67	94	48	68	45
8	23	27	22	37	38	39	30	66	50	41	51	95	13	33	14
9	4	40	49	38	46	38	25	67	43	53	42	96	34	20	21
10	29	32	28	39	39	31	30	68	53	51	42	97	33	15	14
11	29	25	30	40	42	51	41	69	53	52	51	98	21	17	16
12	28	31	2	41	39	38	46	70	65	67	51	99	36	45	35
13	49	5	4	42	65	51	52	71	53	65	52	100	68	39	46
14	3	40	4	43	47	40	39	72	64	67	65	101	43	48	44
15	50	66	61	44	22	28	1	73	68	48	42	102	43	42	48
16	41	40	47	45	30	31	32	74	64	65	53	103	46	45	68
17	33	21	16	46	40	3	31	75	45	44	48	104	46	35	45
18	60	59	6	47	47	39	68	76	36	33	13	105	34	35	19
19	59	62	58	48	40	31	39	77	55	54	43	106	35	46	19
20	56	63	54	49	1	2	3	78	64	53	54	107	24	18	19
21	30	32	29	50	68	42	47	79	58	8	7	108	19	18	20
22	44	45	36	51	50	49	41	80	58	57	9	109	20	18	21
23	12	55	43	52	61	66	62	81	58	62	57	110	24	23	18
24	19	25	24	53	42	41	47	82	63	64	54	111	36	37	33
25	34	19	20	54	49	6	5	83	63	67	64	112	36	35	37
26	53	43	54	55	49	60	6	84	58	9	8	113	33	34	21
27	24	27	23	56	49	50	60	85	57	56	9	114	37	35	34
28	31	28	32	57	60	61	59	86	56	55	10	115	18	17	21
29	2	1	28	58	60	50	61	87	55	56	54	116	33	16	15

Table 3.3: 116 numbers of triangles formed with 68 numbers of Facial Feature Points (FFP), (P_1, P_2, P_3) are FFPs to form a particular $TriID_i, i = 1 \ldots 68$

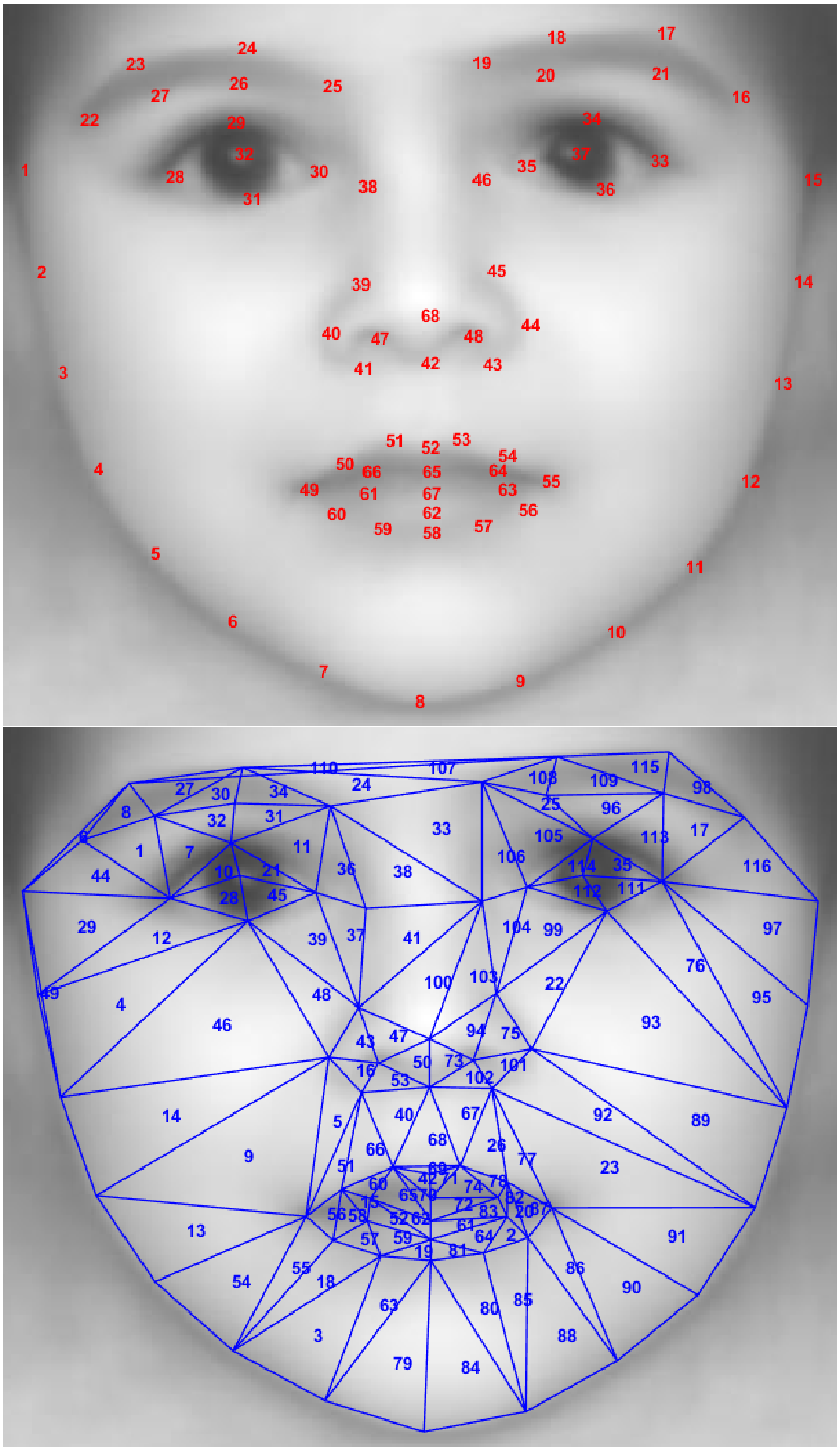
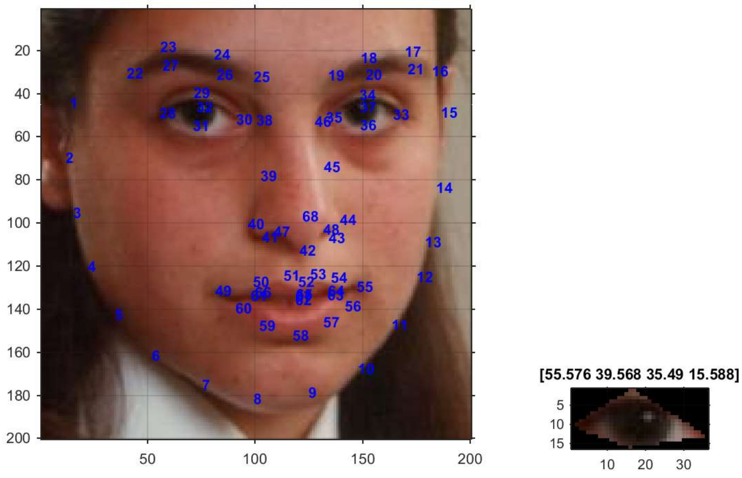


Figure 3.10: 116 Triangle IDs (bottom figure) and respective feature points (top figure), *e.g.* triangle $TriID_{14}$ is represented by points $[3, 4, 40]$, the complete list has been given in Table 3.3, the numbers in the top figure represents Point IDs whether in bottom figure represents point $TriIDs$.



(a) (b)

Figure 3.11: Extraction of left eye, a facial part, (a) Face image with feature positions, (b) Extracted left eye.

which are related to k^{th} facial part. Now crop or extract a region in which this polygon shaped image is available with the following steps.

7. $x_{min} = min(B_x^k), x_{max} = max(B_x^k)$, minimum and maximum values of x coordinates of B^k,

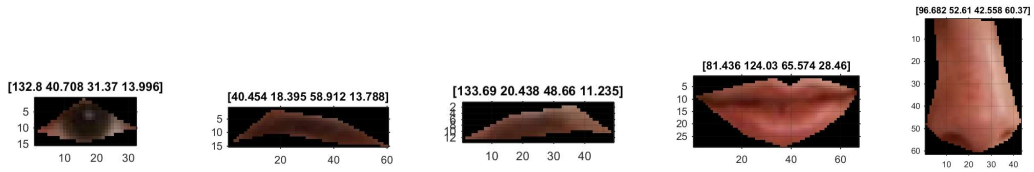
8. $y_{min} = min(B_y^k), y_{max} = max(B_y^k)$, minimum and maximum values of y coordinates of B^k,

9. $N = x_{max} - x_{min}$, which represents the *width* of the target feature image.

10. $M = y_{max} - y_{min}$, which represents the *height* of the target feature image.

11. Find cropped image F_k from given the image F by cropping F with rectangle $[x_{min}, y_{min}, N, M]$), which is the final extracted face part of F, whose left-top corner coordinate is (x_{min}, y_{min}) with size (M, N),

In this way, we get the F_k; k^{th} facial part of a given face F. One example is given in Figure 3.11 to extract the left eye from a given face image. The extracted some other facial parts are given in Figure 3.12.



(a) (b) (c) (d) (e)

Figure 3.12: The extracted facial parts of the face shown in Figure 3.11(a). (a) Right eye, (b) left eyebrow, (c) right eye brow, (d) mouth, (e) nose. In every facial part (a) to (e) the original left top corner position, width and height is mentioned in the top of figure displayed.

Figure 3.13: Normalized images of FG-Net dataset, in the first row, images are as given in the dataset, second row images are aligned *w.r.t* eyes, cropped *w.r.t.* the eye distance ratio, and resized to standard square size.

3.6 Comparisons of Localized and Manually Marked Feature Points in FG-NET Dataset Images

We have verified our localized feature points with the manually marked feature points which are available in FG-NET dataset images, sample images and their given corresponding land mark point sets are shown in Figure 3.13. We use two sets P and P' which represent the set of feature points localized by our method and manually selected given in the dataset respectively for each images. These P and P' are denoted by the point sets

$$\begin{aligned} P &= \{(x_i, y_i)\ i = 1 \ldots 68\} \\ P' &= \{(x'_i, y'_i)\ i = 1 \ldots 68\} \end{aligned} \tag{3.19}$$

Before applying any distance matric for two point sets, we bring them into unit scale, and therefore we have normalized both the point sets P and P' by

$$\begin{aligned} P &= \{(\tfrac{x_i - min_x}{max_x - min_x}, \tfrac{y_i - min_y}{max_y - min_y})\ i = 1 \ldots 68\} \\ P' &= \{(\tfrac{x'_i - min'_x}{max'_x - min'_x}, \tfrac{y'_i - min'_y}{max'_y - min'_y})\ i = 1 \ldots 68\} \end{aligned} \tag{3.20}$$

where (min_x, max_y) are minimum of x_i and y_i, similarly (min'_x, max'_y) are minimum of x'_i and y'_i. The Equation 3.20 will convert x and y coordinates values to the range of $[0, 1]$. Then we have calculated the differences of P and P' using Euclidean Distance.

The Euclidean distance between points $A = (x_1, y_1)$ and $B = (x_2, y_2)$ is the length of the line segment connecting them $\overline{AB}$

$$E_{Dist}(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.21}$$

For a normalized point sets, the maximum distance will occur when the values of two points will be (0, 0) and (1, 1), and minimum distance will occur when the values of two points will be same. Thus limits of E_{Dist} will be $[0, \sqrt{2}]$, and Normalized Euclidean Distance (limits [0, 1]) of a pair of normalized points (A, B) is computed by the Equation 3.22.

$$NE_{Dist}(A, B) = \frac{1}{\sqrt{2}}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{3.22}$$

The Normalized Euclidean Distance between two point sets P and P' are given by the following equation,

$$NE_{Dist}(P, P') = \tfrac{1}{n} \textstyle\sum_{i=1}^{n} NE_{Dist}(P_i, P'_i) \tag{3.23}$$

x	y	x'	y'	nx	ny	nx'	ny'	E_{Dist}
3	5	8	10	0.20	0.29	0.83	1.00	0.95
7	10	3	6	1.00	1.00	0.00	0.50	1.12
7	4	6	2	1.00	0.14	0.50	0.00	0.52
2	6	7	2	0.00	0.43	0.67	0.00	0.79
2	3	9	3	0.00	0.00	1.00	0.13	1.01
							Total	4.39
							Avg	0.88
x	y	x'	y'	nx	ny	nx'	ny'	E_{Dist}
3	5	2	5	0.20	0.29	0.20	0.43	0.14
7	10	6	9	1.00	1.00	1.00	1.00	0.00
7	4	6	4	1.00	0.14	1.00	0.29	0.14
2	6	2	6	0.00	0.43	0.20	0.57	0.25
2	3	1	2	0.00	0.00	0.00	0.00	0.00
							Total	0.53
							Avg	0.11

Table 3.4: Euclidian distance example of two different sets of sample points. Here $nx_i = \frac{x_i - min}{max - min}$, and same for ny_i, and (nx'_i, ny'_i)

.

where $P_i = (x_i, y_i)$, $P'_i = (x'_i, y'_i)$, $n = 68$ *i.e.* number of landmark points of the normalized point sets P and P'.

One example of this difference is given in Table 3.4 with a set of 5 sample points only. In the first set, (top portion) of this table, two sets of 5 sample points of P and P' are given, where points of P and P' are not so similar, because the differences of (x_i, y_i) and (x'_i, y'_i) are very high comparatively to the second pair of point sets, (lower portion), hence as a result average of Euclidian distance is $E_{Dist} = 0.88$. Again in the second sets of P and P', points are very similar, so as a result value of E_{Dist} is very less, which is 0.11.

3.6.1 Similarity between Two Point Sets

To find the similarity measures of two normalized point sets P and P' the following equation is used,

$$Sim_E(P, P') = 1 - NE_{Dist}(P, P') \tag{3.24}$$

Since the maximum Normalized Euclidean Distance $NE_{Dist}(P, P')$ of two normalized point sets P and P' is 1, therefore 0 will be the minimum similarity for this maximum distance, and hence this similarity measurement values are obtained by deducting the distances from 1. Again for minimum distance 0, the similarity will be maximum, for which maximum similarity will be 1. Therefore the similarity range of Sim_E will be $[0, 1]$. The similarity% of normalized point sets P and P' is nothing but $Sim_E(P, P') \times 100$.

The similarities of localized feature point locations of our method and manually marked point locations of FG-NET datasets are shown in Figure 3.14 by finding the similarities using Euclidean based similarity measure Sim_E by Equation 3.24. In the top figure, similarity% values are shown for point sets 1 to 1002 in sorted order of similarities, and in the bottom figure success rate (above 80% similarity) has been shown. The observed similarity information is as given below:

- Total number of experiments: 1002
- Points sets having similarity (≥80%): Approximately 888 out of 1002, i.e 88.62% experiments are successful whose similarity ≥80% (bottom Figure 3.14).

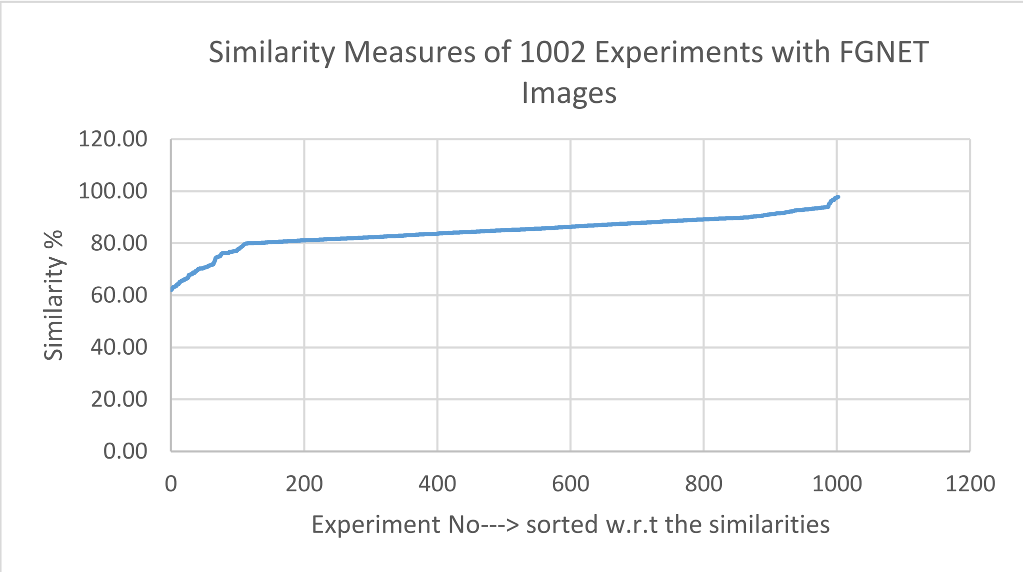
3.7 Databases Used in the Experiments

We have done our various experiments mainly with two different datasets (a) FG-NET Aging database and (b) The BioID Face Database.

3.7.1 FG-NET Aging Database

Ricanek, K. et al. [57] have developed the FG-NET (Face and Gesture Recognition Research Network) Aging database. The FG-NET Aging Database is an image database which contains 1002 number of face images, where images of a person at different ages are available.

The database has been divided into two parts. Part-A and Part-B. For our research purpose we get the access of the Part-A only, which has been distributed for open access. For each image in the database the corresponding pts file is available. Each pts files contain the coordinates of 68 landmark locations of the face. The



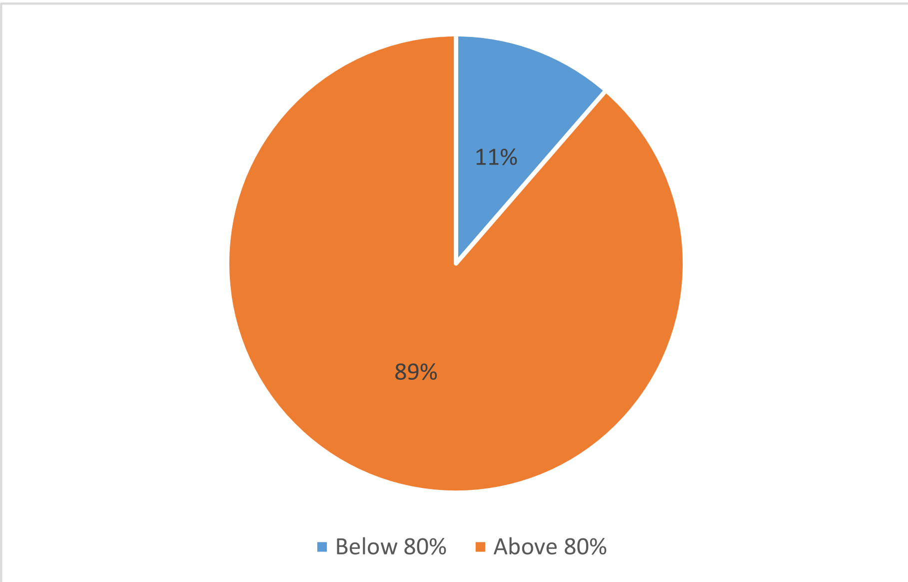


Figure 3.14: Comparisons of landmark points founded in FG-NET dataset by proposed method. In the top figure, similarity% values are shown for point sets 1 to 1002 in sorted order of similarities, and in the bottom figure success rate (above 80% similarity) has been shown.

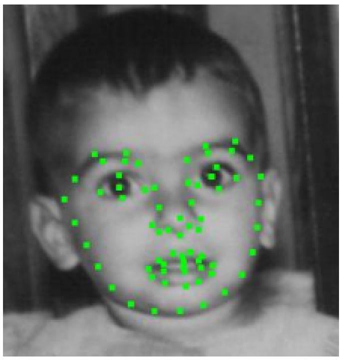


Figure 3.15: Landmark points manually plotted in FG-NET aging database [57].

Figure 3.16: Child to adult images of a subject in FG-NET aging database [57]

positions of the landmarks are shown in Figure 3.15. An example of a subject's images from child to adult is shown in Figure 3.16, and the details of this database is described in Tables 3.5 and 3.6.

3.7.2 BioID Face Database

O Jesorsky et al. [35] had created The BioID Face Database which has been given open access for the research in the area of face detection and image processing.

The dataset consists of 1521 gray level images with a resolution of 384×286 pixel of 23 different persons.

D Cristinacce and K Babalola, researchers from the University of Manchester have marked 20 manually placed points on each of 1521 images of the BioID Face Database. The manually marked feature points are shown in Figure 3.17.

3.8 Result Comparisons of Facial Landmark Localization

The FG-NET and BioID face datasets are used to compute the error of detected facial points on face images. We have used the normalized error detection technique from [35] for the accuracy measurement of the estimated landmark point locations.

Total number of face images	1002	
No. of person	82	
No. of face images per person	6 - 18 (avg 12 images/person)	
Min age	0	
Max age	69	
Image Type	JPG, color or grey scale	
Resolution of image	Variable- 400x500 (approx)	
Image Conditions	Illumination	Varying
	Pose	Varying
	Expression	Varying
	Beards	Yes
	Moustaches	Yes
	Spectacles	Yes
	Hats	Yes

Table 3.5: Specifications of the FG-NET database [57]

GroupID	Age Gp		Images#
	Min	Max	
1	0	5	234
2	6	10	178
3	11	15	163
4	16	20	155
5	21	25	81
6	26	30	62
7	31	35	38
8	36	40	31
9	41	45	26
10	46	50	13
11	51	55	12
12	56	60	2
13	61	65	5
14	66	70	2
Total			1002

SubID	Im#	SubID	Im#	SubID	Im#	SubID	Im#
1	15	22	13	43	11	64	6
2	16	23	12	44	10	65	15
3	12	24	11	45	13	66	12
4	12	25	12	46	13	67	10
5	11	26	11	47	14	68	10
6	12	27	11	48	16	69	11
7	9	28	11	49	10	70	11
8	15	29	13	50	8	71	13
9	13	30	11	51	11	72	14
10	12	31	13	52	11	73	16
11	14	32	12	53	13	74	16
12	15	33	11	54	13	75	10
13	12	34	13	55	8	76	18
14	10	35	14	56	8	77	16
15	13	36	13	57	10	78	16
16	13	37	12	58	11	79	14
17	13	38	14	59	9	80	14
18	11	39	14	60	12	81	12
19	10	40	14	61	13	82	11
20	13	41	10	62	12		
21	12	42	13	63	10		

Table 3.6: Age group and subject wise details of FG-NET aging database

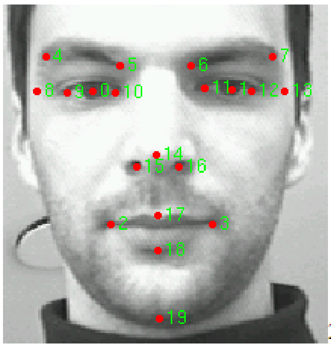


Figure 3.17: 20 manually placed points in BioID face image [35].

The point detection error is measured as

$$err = \sqrt{(x - x')^{+}(y - y')^2}/l \tag{3.25}$$

where (x, y) and (x', y') are the ground truth and the detected position, and l is the distance of two eyes. If an error is larger than 5%, it is counted as failure. The normalized error of two point sets (P, P') (where P and P' can be found by Equation 3.19) has been computed as follows:

$$e = \frac{1}{N}\sum_{i=1}^{N} err_i \tag{3.26}$$

where err_i is the point detection error of points (x_i, y_i) and (x_i', y_i'), $i = 1 \ldots N$, and N is the number of facial landmark positions.

We have randomly selected 100 images from BioID and 100 from FG-NET datasets to find the point detection error. Average error for detecting left eye LE (5 points), right eye RE (5 points) left eyebrow LEB (6 points), right eyebrow REB (6 points), Nose (10 points), Mouth (19 points) and face boundary FB (21 points) on BioID and FG-NET dataset are shown in Figure 3.18 (Figure 3.10 (top) and Table 3.1 for face parts and associated points).

Average error comparisons on BioID Face Dataset is shown in Figure 3.19. Comparisons were made with five[1] detected facial parts of BioID Face Dataset images, namely left eye (LE), right eye (RE), nose (N), left mouth (LM) and right mouth (RM). Some of the methods which detect above five major facial points (LE, RE, N, LM, RM) are Liang et al. [42], M. Valstar et al. [77], Yi Sun et al. [66]. We have compared our result with these methods, and comparative average errors of facial points show that error of our proposed method is comparatively lower.

[1]5, because each face is labeled with the five ground truth positions.

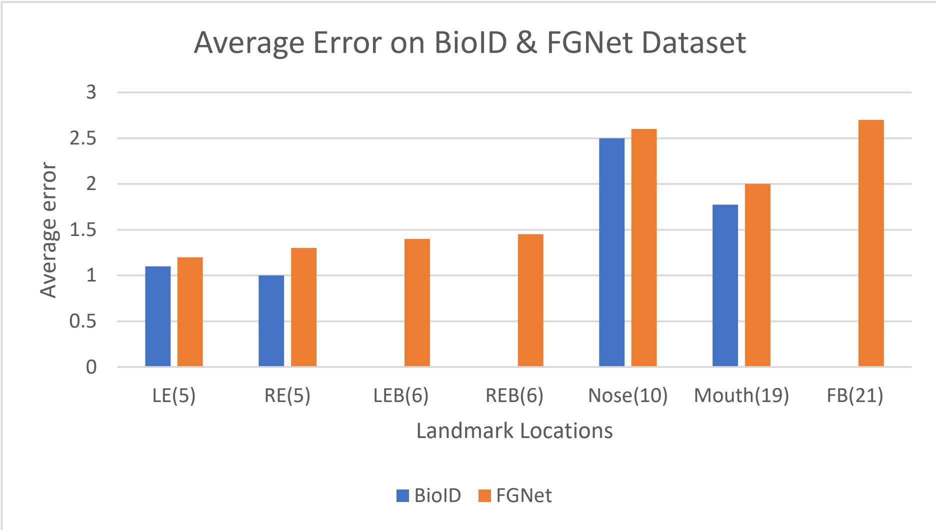


Figure 3.18: Average error for detecting left eye LE (5 points), right eye RE (5 points) left eyebrow LEB (6 points), right eyebrow REB (6 points), Nose (10 points), Mouth (19 points) and face boundary FB (21 points) on BioID and FG-NET dataset

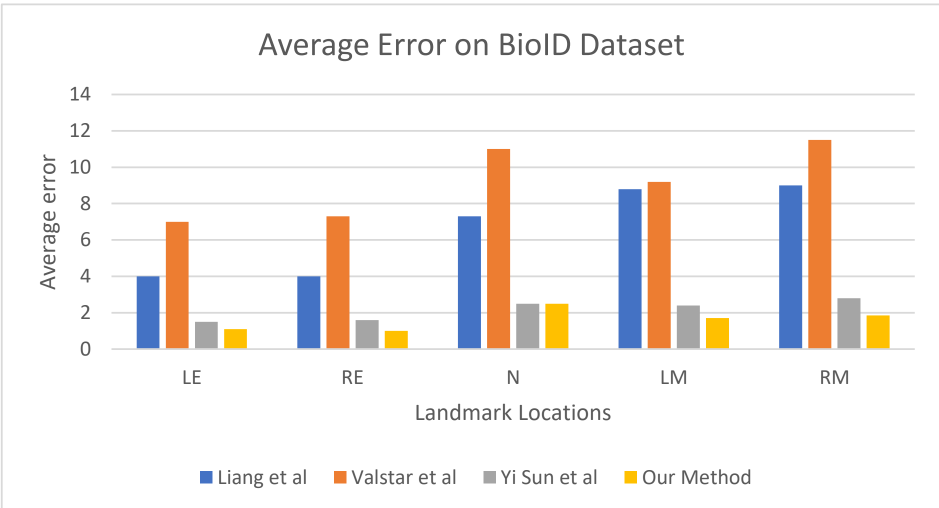


Figure 3.19: Average error comparisons on BioID Face Dataset. Comparisons were made with five detected facial parts only, namely left eye (LE), right eye (RE), nose (N), left mouth (LM) and right mouth (RM). The compared existing methods are Liang et al. [42], M. Valstar et al. [77], Yi Sun et al. [66] and our proposed method.

3.9 Conclusion

In this work we have developed a method by which a semi automatic face extraction and facial feature point extraction is done. In a face image 68 facial feature or landmark points are localized based on the selection of two eye position and mouth mid position. This specific number 68 has been chosen because majority of existing face image data-sets are having this number for various face image analysis methods. By these landmark points we create 116 triangles in a face image by which we represent different facial parts in the face.

The correct landmark positions create a situation where each triangle locates in a fixed location of a face, by which same facial parts can be located in different faces. The extraction of the facial parts like eyes, nose, mouth, eyebrows *etc.* from the face image is done with these feature point set. This feature points are very important initial information for any types of face image analysis.

In the next chapter a warping method have been developed by a triangle wise mapping technique by which a pair of face images can be mapped. This technique is based on these landmark points and associated triangulation. This technique is discussed in the next chapter.