**British Journal of Mathematics & Computer Science**
**11(5): 1-27, 2015, Article no.BJMCS.19543**
***ISSN: 2231-0851***

**SCIENCEDOMAIN** *international*
www.sciencedomain.org

# A Method for Human Emotion Recognition System

**Sarbani Ghosh[1] and Samir K. Bandyopadhyay[1*]**

[1]*Department of Computer Science and Engineering, University of Calcutta, Kolkata, India.*

***Article Information***

DOI: 10.9734/BJMCS/2015/19543
*Editor(s):*
(1) Kai-Long Hsiao, Taiwan Shoufu University, Taiwan.
*Reviewers:*
(1) Anonymous, University of Science and Technology Houari Boumediene, Algeria.
(2) Anonymous, Dalian University of Technology, China.
(3) Anonymous, Firat University, Turkey.
Complete Peer review History: http://sciencedomain.org/review-history/11384

***Original Research Article***

*Received: 15 June 2015*
*Accepted: 07 August 2015*
*Published: 12 September 2015*

## Abstract

This paper studied computationally efficient algorithm for facial feature selection based on template matching method which further leads to identification of smiling face or neutral face. At first, minimal pre-processing including gray scale conversion is done on the image. After that, matching between original image and template image is done using normalized cross-correlation technique. Each matching area is bounded by box to identify that region of interest. Then the mid points between the eye regions are found and the distance between the mid points and the corners of the mouth region is calculated. On the basis of the distances between these features, emotions are recognized. After detecting neutral or smiling face, different types of facial expressions are classified using linear support vector machine used as multiclass classifier.

*Keywords: Support vector machine; multiclass classifier and human emotion.*

## 1 Introduction

As the number of social media applications and image based applications increase day by day, Facial Feature Detection system is fast becoming a familiar feature in 'apps' and on websites on different purpose. Human face localization and detection is often the first step in applications such as video surveillance, human computer interface, and face recognition and image database management. Furthermore facial feature characteristics are very much effective in both biometric identification which automatically identifies a person from a digital image or a video image. We use facial expressions not only to express our emotions, but also to provide important communicative cues during social interaction, such as our level of interest, our desire to take a speaking turn and a continuous feedback signaling about the understanding of the information conveyed. Among all the facial features, eye feature has more application domain. It is reported that facial expressions have considerable effects on listener, near about 55 percent effect of the spoken words depend on eye movements and facial expressions of the speaker.

**Corresponding author: E-mail: skb1@vsnl.com;*

This study represents a computationally efficient algorithm for facial feature selection based on template matching method which further leads to identification of smiling face or neutral face. In a smiley face distance between eye pair and mouth must be greater than normal face (without any notable emotion). Thus it can be identified whether a person is happy or not. Any face image database with various expressions can be taken for application. At first, minimal pre-processing including gray scale conversion is done on the image. After that, matching between original image and template image is done using normalized cross-correlation technique. Each matching area is bounded by box to identify that region of interest. Then the mid points between the eye regions are found and the distance between the mid points and the corners of the mouth region is calculated. On the basis of the distances between these features, emotions are recognized. After detecting neutral or smiling face, different types of facial expressions are classified using linear support vector machine used as multiclass classifier. Experiments on different human faces with four different facial expressions show that the proposed method is competitive. Finally a front-end interactive user interface is created to automate corresponding task or calculation done in the back-end programming methods. This interface provides the user to find input images stored in any folder in the computer's disk drives and perform corresponding operations on them.

# 2 Literature Review

Human-like robots and machines that are expected to enjoy truly intelligent and transparent communications with human can be created using automatic facial expression recognition with a set of specific desired accuracy and performance requirements. Facial expression recognition deals with the problem of classifying facial images into expression classes. It has been of interest to a growing number of researchers and much progress has been made during the last two decades. Expression recognition involves a variety of subjects such as perceptual recognition, machine learning, affective computing etc.

One case study uses skin color range of human face to localize face area. Then high frequency noise is removed by masking with a low pass filter from the preprocessed image and skin color blocks are detected. After face detection, various facial features are identified by calculating the ratio of width of multiple regions in human face. Finally the test image is partitioned into a set of sub-images and each of these sub-images is matched against a set of sub-pattern training set. Partitioning is done using Aw-SpPCA algorithm. Given as input any emotion of face, this pattern training set will classify the particular emotion [1].

Face component extraction by dividing the face region into eye pair and mouth region and measurement of Euclidean distance among various facial features is also adopted by a case study. Similar study is done by Neha Gupta to detect emotions. This research includes four steps: pre-processing, edge detection, feature extraction and distance measurement among the features to classify different emotions. This type of approach is classified as Geometric Approach [2].

Another research includes Face detection method using segmentation technique. First, the face area of the test image is detected using skin color detection. RGB color space is transformed into YCbCr color space in the image and then skin blocks quantization is done to detect skin color blocks. As next step, a face cropping algorithm is used to localize the face region. Then, different facial features are extracted using segmentation of each component region (eyes, nose, mouth). Finally, vertical & angular distances between various facial features are measured and based on this any unique facial expression is identified. This approach can be used in any biometric recognition system [3].

A template matching based facial feature detection technique is used in a different case study [4].

Different methods of face detection and their comparative study are done in another review work. Face detection methods are divided into two primary techniques: Feature based & View based methods [5].

Gabor filters are used to extract facial features in another study. This approach is called Appearance based approach. This classification based facial expression recognition method uses a bank of multilayer perceptron neural networks. Feature size reduction is done by Principal Component Analysis (PCA) [6].

Another study represents a robust face detection and gender classification strategy in color images under non-uniform background. This is done by localizing human face region in the given image and detecting facial features after converting the given RGB image to YCbCr color space for skin detection. Here, first mouth region is detected from the localized face region. From this, eye regions and nose regions are located and they are used as feature points. Gender classification is done using these features on images with different sizes. Linear support vector machine is used as the classifier which gives best classification rate [7].

Thus existing works primarily focused in detecting facial features and they are served as input to emotion recognition algorithm. In this study, a template based feature detection technique is used for facial feature selection and then distance between eye and mouth regions is measured.

# 3 Proposed Methodology

The facial features detection and expression classification methods are described in the flowchart given below:

I. *Facial feature & smiling face detection*

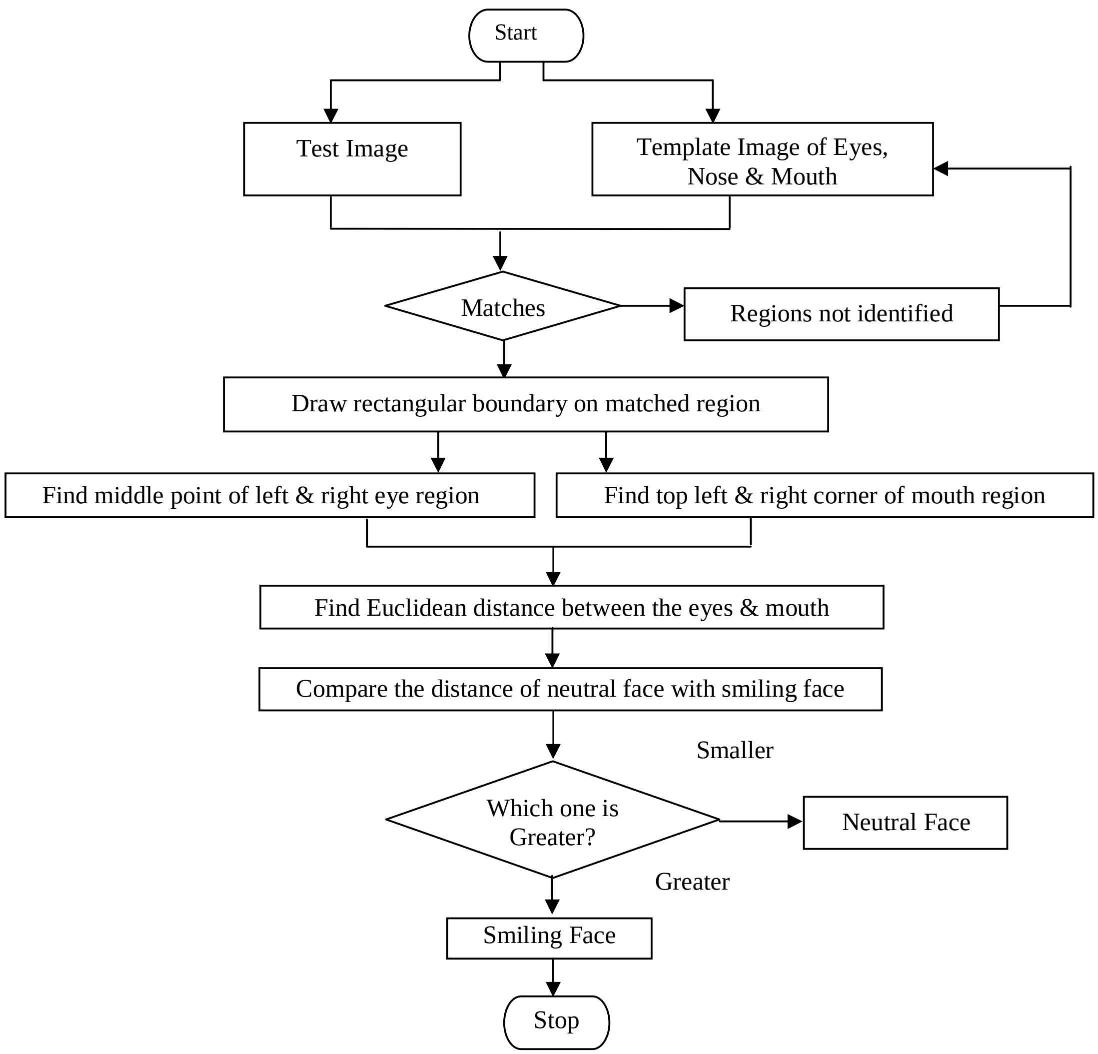



**Fig. 1. Overall processing of neutral or smiling face detection**

*II. Expressions classification:*

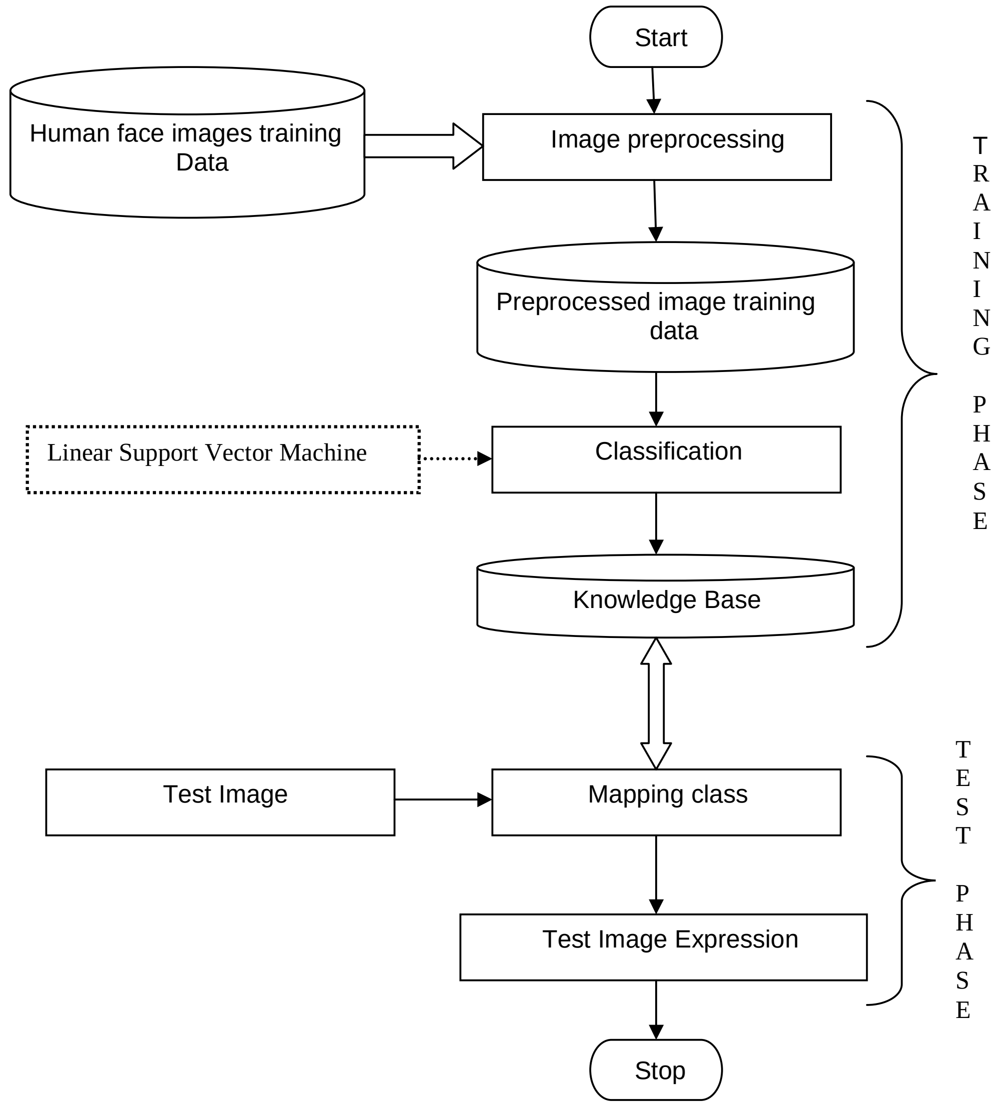



**Fig. 2. Overall processing of expression classification**

## 3.1 Template Matching Based Facial Feature Detection

The algorithm detects four basic facial feature regions of interests: Left & right eye, nose and mouth. To detect these regions of interests, the method takes two types of input images: one is the test image whose facial features need to be identified and the template images like eye templates, nose & mouth templates. Minimal preprocessing is done on the input image and the template images. The input images are taken in RGB color space and they are converted to gray scale images.

This method correlates the face image against all the feature template images. For example, to detect the eye we need to correlate the given image with the eye template and the best-correlated region will be marked as an eye.

Cross-correlation is a measure of similarity of two series as a function of the lag of one relative to the other. This is also known as a sliding dot product or sliding inner-product. For continuous functions *f* and *g*, the cross-correlation is defined as**:**

$$(f \star g)(\tau) \stackrel{\mathrm{def}}{=} \int_{-\infty}^{\infty} f^*(t)\ g(t+\tau)\,dt,$$

where $f^*$ denotes the complex conjugate of $f^*$ and $\tau$ is the lag.

As an example, consider two real valued functions $f^*$ and $g$ differing only by an unknown shift along the x-axis. One can use the cross-correlation to find how much $g$ must be shifted along the x-axis to make it identical to $f^*$. The formula essentially slides the $g$ function along the x-axis, calculating the integral of their product at each position. When the functions match, the value of $(f \star g)$ is maximized. This is because when peaks (positive areas) are aligned, they make a large contribution to the integral. Similarly, when troughs (negative areas) align, they also make a positive contribution to the integral because the product of two negative numbers is positive.

Following steps demonstrate the above method:

**Acquisition of input image**

Read the Input Human Face Image. If the Input Image is color (RGB), then convert it to gray scale image.

**Acquisition of template image**

Main features of a Human Face to be extracted are: Left Eye, Right Eye, Nose and Mouth. Read four different Template Images for the corresponding features. If the Template Images are color (RGB), convert them to corresponding gray scale images respectively.

**Normalized 2D - cross correlation between input image and template images**

Correlation mask *w* (x , y) of size m*n, with an image *f* (x, y) may be expressed in the form

$$C(x,y) = \sum_{s} \sum_{t} w(s,t)\, f(x+s,y+t)$$

where the limits of summation are taken over the region shared by *w* and *f* . This equation is evaluated for all values of the displacement variables x and y so that all elements of *w* visit every pixel of *f,* where *f* is assumed to be larger than *w.* When the input image *f* is larger than the template image, we say that it is *normalized.* Here *w* is referred to as a *template* and correlation is referred to as *template matching.* The values of template cannot all be the same. The resulting matrix C contains the correlation coefficients, which can range in value from -1.0 to 1.0. Perform 2D – Cross Correlation between Gray scale Input Image and 4 different Gray scale Template Images separately.

**Determine the maximum correlation value**

Following is performed for all four 2D – Cross Correlation operations mentioned in the previous step:

The Maximum Value of C occurs when the corresponding normalized region in f are identical i.e where the maximum correlation occurs. Find the corresponding rectangular region where maximum matching is found. Pixel coordinate of top left corner (let $(x_1,y_1)$), width (let w), height (let h) of the rectangle are found.

**Draw a boundary around the region**

Following is performed for all four 2D Cross Correlation operations mentioned in step 3:

Draw Rectangle with pixel coordinates $(x_1,y_1)$, $(x_2,y_1)$, $(x_1,y_2)$, $(x_2,y_2)$, where $x_2=x_1+w$ and $y_2=y_1+h$.

## 3.2 Neutral or Smiling Face Detection

The facial feature detection method based on template matching detects four important facial regions in human face, i.e. left & right eye, nose and mouth. These feature regions are very important to identify the changes in a smiling face than the neutral face of the same person. To detect accurately whether a person is smiling or not, some feature points from the regions of interest need to be identified.

Generally, the distance between the eyes and the mouth of a person's face is varied when he/she is smiling. So, the Euclidean distances between the middle point of the left eye & the top left corner of mouth and the middle point between the right eye & the top right corner of the mouth are calculated first. Experimentally, we can see that the distance between eye & mouth of a person's face become larger than his neutral face without any expression. The calculation is done on more than twenty images where the above result is seen. Probabilistically, we can say if a person is smiling then, the Euclidean distances between the left eye & mouth and right eye & mouth are larger than his/ her normal face. Following steps describe the above method:

**Determine the middle point of the rectangle around left eye**

If the pixel coordinates of the drawn rectangle around left eye are $(x_1,y_1)$, $(x_2,y_1)$, $(x_1,y_2)$, $(x_2,y_2)$, then middle point coordinate of the rectangle are $(x_{mid1},y_{mid1})$ where $x_{mid1} = (x_1+x_2)/2$ and $y_{mid1} = (y_1+y_2)/2$.

**Determine euclidian distance between middle point of rectangle around left eye and top left corner pixel coordinate of rectangle around mouth**

If the middle point pixel coordinate of rectangle around left eye is $(x_{mid1},y_{mid1})$ and top left corner pixel coordinate of rectangle around mouth is $(x_1',y_1')$ then Euclidian distance between these 2 points is = $\sqrt{( x_{mid1} - x_1')^2 + ( y_{mid1} - y_1')^2}$ unit.

**Determine the middle point of the rectangle around right eye**

If the pixel coordinates of the drawn rectangle around left eye are $(x_1,y_1)$, $(x_2,y_1)$, $(x_1,y_2)$, $(x_2,y_2)$, then middle point coordinate of the rectangle are $(x_{mid2},y_{mid2})$ where $x_{mid2} = (x_1+x_2)/2$ and $y_{mid2} = (y_1+y_2)/2$.

**Determine euclidian distance between middle point of rectangle around right eye and top right corner pixel coordinate of rectangle around mouth**

If the middle point pixel coordinate of rectangle around right eye is $(x_{mid2},y_{mid2})$ and top right corner pixel coordinate of rectangle around mouth is $(x_1'',y_1'')$ then Euclidian distance between these 2 points is = $\sqrt{( x_{mid2} - x_1'')^2 + ( y_{mid2} - y_1'')^2}$ unit.

## 3.3 Expressions Classification

Classification of available data sets can be viewed as the task of separating classes in features space. In machine learning, **Support Vector Machines (SVMs) [7]** are supervised learning models with associated learning algorithms that analyze data and recognize patterns used for classification. Given a set of training examples, each marked for belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic **binary linear classifier.**

Given some training data $\mathcal{D}$, a set of $n$ points of the form

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\, y_i \in \{-1, 1\}\}_{i=1}^{n}$$

where the $y_i$ is either 1 or −1, indicating the class to which the point $\mathbf{x}_i$ belongs. Each $\mathbf{x}_i$ is a $p$-dimensional real vector. We want to find the maximum-margin hyper plane that divides the points having $y_i = 1$ from those having $y_i = -1$. Any hyper plane can be written as the set of points $\mathbf{x}$ satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

where **.** denotes the dot product and $\mathbf{w}$ the (not necessarily normalized) normal vector to the hyper plane. The parameter b/ |w| determines the offset of the hyper plane from the origin along the normal vector $\mathbf{w}$.

If the training data are linearly separable, we can select two hyper planes in a way that they separate the data and there are no points between them, and then try to maximize their distance. The region bounded by them is called "the margin". These hyper planes can be described by the equations

$$\mathbf{w} \cdot \mathbf{x} - b = 1 \text{ and } \mathbf{w} \cdot \mathbf{x} - b = -1.$$

An SVM model is based on the concept of decision planes that defines decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. A schematic example is shown in the figure illustrated below:

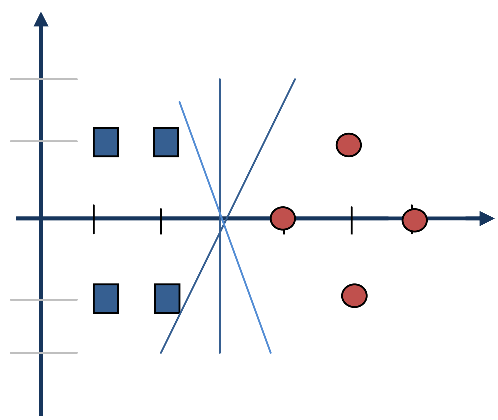


**Fig. 3. Decision plane in SVM**

So, in the above figure there are two classes: Blue class and Red class. The task here to find out the boundary that separates the two classes effectively. There may be more than one decision boundary that separates these two classes – light blue lines in Fig. 3. defines the possible boundaries that can separate the classes. SVM defines the optimum boundary from the above boundaries which separate the two classes accurately. Now, we are going to explain how linear support vector machine does this:

Often, SVM choose 3 or 4 support vectors or decision points from the two different classes to classify. Suppose here three vectors named $S_1$, $S_2$ & $S_3$ are used. The idea is to get the support vectors from the available decision points which are closer to the space or the hyper planes. This is actually the area that may common to both the classes depending on how the decision boundary is defined.

For example, here we choose $S_1$ & $S_2$ from the blue class and $S_3$ from the red class. This is shown in Fig. 4. We utilize these three vectors to define the optimum decision boundary using support vector machine. The vectors are defined as follows:

$$S_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad S_2 = \begin{bmatrix} 2 \\ -1 \end{bmatrix} \qquad S_3 = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$$

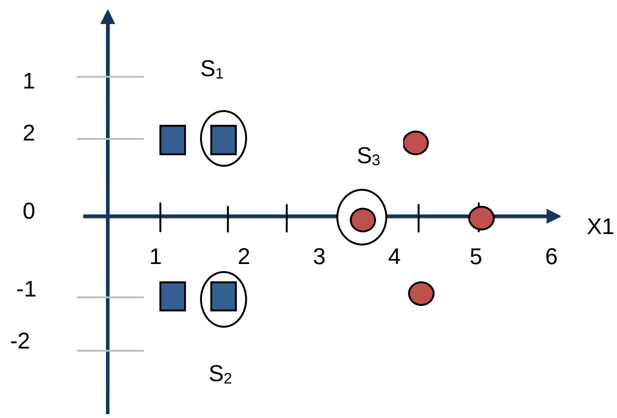



**Fig. 4. Support vectors used in SVM classification**

The values of the three vectors correspond to the two feature coordinates X1 and x2. The features may be different for different types of classification. Minimal modifications are done on these vectors for processing advantages. Basically, the vectors are augmented with a 1 as a bias input and for clarity, the names of the vectors are differentiated with an over tilde. So the support vectors become:

$$\sim S_1 = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} \qquad \sim S_2 = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix} \qquad \sim S_3 = \begin{bmatrix} 4 \\ 0 \\ 1 \end{bmatrix}$$

The extra 1 in these vectors are added as bias input and the vectors are defines as $\sim S_1$, $\sim S_2$ & $\sim S_3$.

According to SVM theory, we need to find three parameters a1, a2 & a3 based on the following three linear equations:

$$a1 \sim S_1 * \sim S_1 + a2 \sim S_2 * \sim S_1 + a3 \sim S_3 * \sim S_1 = -1 \text{ (-ve class)}$$
$$a1 \sim S_1 * \sim S_2 + a2 \sim S_2 * \sim S_2 + a3 \sim S_3 * \sim S_2 = -1 \text{ (-ve class)}$$
$$a1 \sim S_1 * \sim S_3 + a2 \sim S_2 * \sim S_3 + a3 \sim S_3 * \sim S_3 = -1 \text{ (-ve class)}$$

The blue classes are classified as –ve class & red classes are classified as +ve class.

Now we have to substitute the values of $\sim S_1$, $\sim S_2$ & $\sim S_3$ in the above equations chosen for this particular example. After substituting the vectors in the equations and calculating we get three simplified equations:

$$6a1 + 4a2 + 9a3 = -1$$
$$4a1 + 6a2 + 9a3 = -1$$
$$9a1 + 9a2 + 17a3 = +1$$

Simplifying the above three equations we get: a1= a2=-3.25 and a3=3.5.

Now we are going to get the boundary between the two classes. We obtain the values of three parameters a1, a2 & a3 and the three vectors, $\sim S_1$ & $\sim S_2$ from blue class and $\sim S_3$. from red class. The hyper plane that discriminates the positive class from the negative class is given by the equation:

$$\sim w = \sum_{i} ai \sim S_i$$

ai is the constant whose value need to be calculated and Si is the support vector. In this example, i=3 as we use three support vectors and get values of three constants.

Substituting the values of three vectors $\sim S_1$, $\sim S_2$ & $\sim S_3$. and the constants a1, a2 & a3, we get:

$$\sim w = (-3.25)\begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25)\begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5)\begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

Thus, the feature coordinates of the hyper plane that may discriminate the blue and red classes is obtained. Following describes how this hyper plane will distinguish the classes.

First, the support vectors are augmented with a bias previously for processing purpose. This bias will now even be represented as the bias of the hyper plane i.e. 3 and the two coordinates of x1 & x2 will be 1 & 0. Hence we can equate the entry in the ~w as the hyper plane with an offset b.

Therefore, the separating hyper plane equation will be:

$$y = wx + b \text{ with } w = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and offset } b = -3$$

Basically, the coordinates of w denote that the hyper plane is a vertical line with an offset 3.

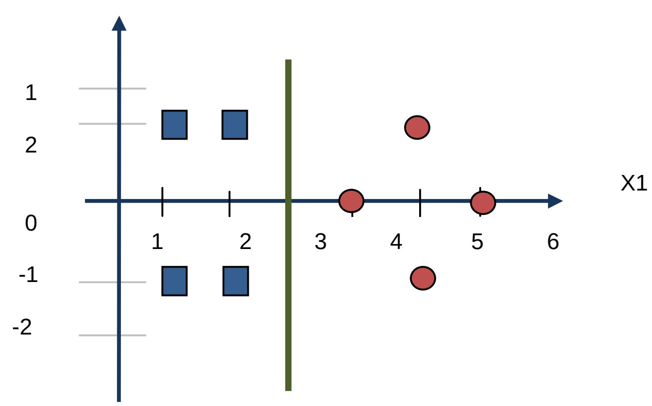



**Fig. 5. Decision plane with optimum decision boundary**

The green line in Fig. 5 is the expected decision boundary in linear SVM for the corresponding classes. The two classes now either belong to –ve class (blue class) or +ve class (red class). Any test image can be classified in any of the two classes after performing SVM training. The classification is linear because there is very clear separation between two classes which is distinguished by the hyper plane w. Sometimes, there may not be clear separation between the classes so that support vectors from the two different classes cannot be chosen directly. This type of SVM classification is Non-linear classification.

In the proposed algorithm, linear binary support vector machine model is used as multiclass classifier.

Four different types of facial expressions of human face need to be classified according to the algorithm. As SVM is a binary classifier, the algorithm uses “one against all” approach to use SVM as multiclass classifier.

The algorithm uses four facial expressions:

- Neutral
- Smiling
- Angry
- Sad

The proposed work uses Database for face images for training set [8]. Each type of expression is presented by four images. The 16 images are arranged into three train database. First database contains all the 16 images of four types of expression. Second train database contains 12 images of Smiling, Angry & Sad expressions. Third train database contains 8 images of Angry & Sad expressions. These three training databases are used jointly to train the whole set of images.

First, preprocessing is done on the training set of images. Preprocessing of the training images and test images is needed because the used images are of different sizes and in RGB color space and they need to be of same features so that classification can be done accurately.

Preprocessing includes:

- The face images in the training set are converted to “double” type.
- The images are converted to gray scale from RGB color space.
- The images are zoomed or shrank to 200X200 pixels size.
- The images are reshaped so that the 2D matrices of the gray scale images are converted to 1D array of 40000 columns.

After image preprocessing, the three train databases are assigned different classes. The steps include:

- In the first training set, there are 12 images of which first four are of neutral expression. This expression is assigned class 1 and left 9 images are assigned class -1.
- In the second training set, there are 9 images of which first four are of smiling expression. This expression is assigned class 2 and left 6 images are assigned class -2.
- In the third set, there are 6 images of which first four are of angry expression assigned class 3 and left four are of sad expression assigned class -3.

Finally, the test image is mapped to either neutral class or smiling class or angry class or sad class and the same is displayed in the user interface.

Following steps describe the above method:

## 3.4 Acquisition of Input Training Images

Read the input images used for training set.

## Image pre-processing

### Change image data type

Convert the input images from type 'unit8' to 'double' precision. If the input image is in double type, the output image is identical.

### Convert to gray scale

Convert the input images from RGB color space to gray scale image.

### Image resizing:

The input images are zoomed or shrank to 200X200 resolution of pixels depending on the original resolution of the input images.

### Image reshaping

The input images' dimensions are reduced and reset to 1 to 40000 matrix. Therefore, after reshaping, all the train images become 1D array 40000 indexes.

### Create array of all zeroes

Create three 2D array with number of rows equal to the total number of training set of images and the number of columns are 40000. The cell values of these 3 arrays are initialized to zero.

### Assign group membership to each expression class

In the proposed method, only the columns vectors of the train image matrices are used as features implicitly to classify the membership of the images. No other facial feature points are used explicitly to measure the percentage of membership of a particular image to a particular class.

Four types of expressions are assigned to different class membership in one against all approach:

- For the first train set, neutral expression is assigned class 1 and all other expressions are assigned class -1.
- For the second train set, smiling expression is assigned class 2 and all other expressions are assigned class -2.
- For the third train set, angry expression is assigned class 3 and sad expression is assigned class -3.

### Store the images from the training set to the arrays:

The images from the three training subsets are stored in the three 2D arrays and make the three traindata arrays for SVM training.

### Training:

Load the three train subsets of images and map the train images to the corresponding classes according to the feature space (column vectors of the traindata arrays). Each image in the training sets is mapped to any of the four expression class.

### Classify the test image into a train class:

Any test image is first go through the same pre – processing steps. Then classification is done and the result classes are stored in result, result1 & result2 for the three train sets.

- If the test image has result class of 1, then the image is of Neutral expression.
- If the test image has result class of -1 & the result1 class of 2, then the image is of smiling expression.
- If the test image has result class of -1 & the result1 class of -2 & result2 class of 3, then the image is of angry expression.
- If the test image has result class of -1 & the result1 class of -2 & the result2 class of -3, then the image is of sad expression.

## 3.5 User Interface Design

A simple interactive User Interface is created to represent the processing done in the proposed work. Any processing step is done after clicking the corresponding button on the interface. The output of the processing will be displayed on the interface and every change in the output will be notified by interactive messages.

- Input image for facial feature detection can be browsed from any disk storage of the machine.
- Conversion from RGB to gray scale is done on the input image and the same is showed in the interface.
- After clicking the "Facial feature extraction" button, four primary facial features are detected and the same is showed in the interface.
- The distances between the eyes and mouth are calculated for normal & smiling face and the same are showed in the interface so that we can compare them.
- For classification of expressions, training images are done after clicking the "Train svm" button. Then any test image can be browsed for test purpose and the corresponding expression of the test image is shown in the interface.
- Some interactive messages in between these steps are shown to notify the internal processing done in the programme code.

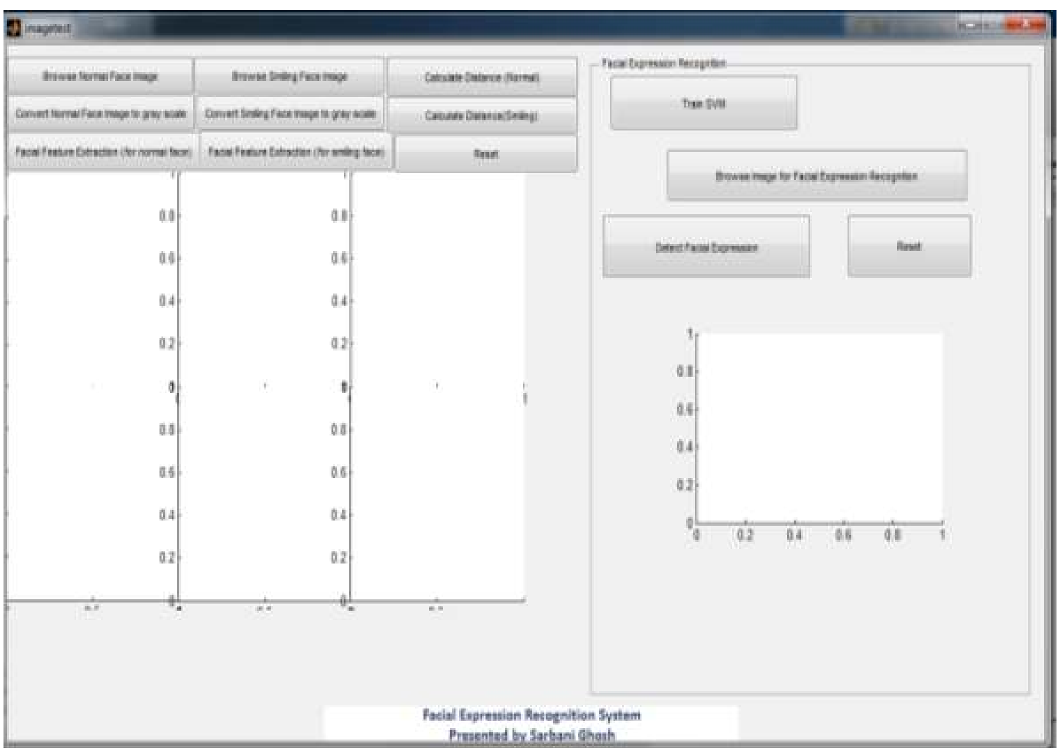



**Fig. 6. User interface for the expression recognition system**

# 4 Algorithm

## 4.1 Facial_Feature_Detection (Input Image, Template Images)

**Step1.** Start

**Step2.** Read Input Human Face Image.
If the Input Image is color (RGB), then
convert it to Gray scale Image and save the pixel values to a 2D array let gface.
Else
save the pixel values of the input image to a 2D array let gface.

**Step3.** Read Left eye template image.
If the template image is color (RGB), then
convert it to Gray scale Image and save the pixel values to a 2D array let gleft.
Else
save the pixel values of the input image to a 2D array let gleft.

**Step4.** Read Right eye template image.
If the template image is color (RGB), then
convert it to Gray scale Image and save the pixel values to a 2D array let gright.
Else
save the pixel values of the input image to a 2D array let gright.

**Step5.** Read Nose template image.
If the template image is color (RGB), then
convert it to Gray scale Image and save the pixel values to a 2D array let gnose.
Else
save the pixel values of the input image to a 2D array let gnose.

**Step6.** Read Mouth template image.
If the template image is color (RGB), then
convert it to Gray scale Image and save the pixel values to a 2D array let gmouth.
Else
save the pixel values of the input image to a 2D array let gmouth.

**Step7.** Declare 4 2D Array C1, C2, C3 & C4 of size m*n where m*n is the size of gface.

**Step8.** Calcualte C1[][] = 2D_norm_crosscorr(gleft,gface)
C2[][]= 2D_norm_crosscorr(gright,gface)
C3[][] = 2D_norm_crosscorr(gnose,gface)
C4[][] = 2D_norm_crosscorr(gmouth,gface)

**Step9.** Call $(x_{11},y_{11},w_1,h_1)$ = Find_max(C1)
$(x_{21},y_{21},w_2,h_2)$ = Find_max(C2)
$(x_{31},y_{31},w_3,h_3)$ = Find_max(C3)
$(x_{41},y_{41},w_4,h_4)$ = Find_max(C4)
where $(x_{11},y_{11},w_1,h_1)$, $(x_{21},y_{21},w_2,h_2)$, $(x_{31},y_{31},w_3,h_3)$, $(x_{41},y_{41},w_4,h_4)$ are top – left pixel coordinate, width, height of the matched rectangular area around left eye, right eye, nose and mouth respectively.

**Step10.** Calculate $x_{12} = x_{11} + w_1$ & $y_{12} = y_{11} + h_1$
$x_{22} = x_{21} + w_2$ & $y_{22} = y_{21} + h_2$
$x_{32} = x_{31} + w_3$ & $y_{32} = y_{31} + h_3$
$x_{42} = x_{41} + w_4$ & $y_{42} = y_{41} + h_4$
where $(x_{12},y_{12})$, $(x_{22},y_{22})$, $(x_{32},y_{32})$, $(x_{42},y_{42})$ are bottom right pixel coordinate of the matched rectangular area around left eye, right eye, nose and mouth respectively.

**Step11.** Draw Boundary Rectangle around left eye in gface with top left, top right,bottom left and bottom right pixel coordinates as $(x_{11},y_{11})$, $(x_{12},y_{11})$, $(x_{11},y_{12})$ & $(x_{12},y_{12})$ respectively.
Draw Boundary Rectangle around right eye in gface with top left, top right, bottom left and bottom right pixel coordinates as $(x_{21},y_{21})$, $(x_{22},y_{21})$, $(x_{21},y_{22})$ & $(x_{22},y_{22})$ respectively.

Draw Boundary Rectangle around nose in gface with top left, top right, bottom left and bottom right pixel coordinates as $(x_{31},y_{31})$, $(x_{32},y_{31})$, $(x_{31},y_{32})$ & $(x_{32},y_{32})$ respectively.
Draw Boundary Rectangle around mouth in gface with top left, top right, bottom left and bottom right pixel coordinates as $(x_{41},y_{41})$, $(x_{42},y_{41})$, $(x_{41},y_{42})$ & $(x_{42},y_{42})$ respectively.
Calculate middle point pixel coordinate $(x_{1mid},y_{1mid})$ of the boundary rectangle around Left eye as $x_{1mid} = (x_{11}+x_{12})/2$ and $y_{1mid} = (y_{11}+y_{12})/2$.

**Step12.** Calculate Euclidian Distance between middle point pixel coordinate $(x_{1mid},y_{1mid})$ of the boundary rectangle around left eye and top – left pixel coordinate $(x_{41},y_{41})$ of the boundary rectangle around mouth as:

$$\mathbf{Dist1 = \sqrt{( x_{1mid} - x_{41})^2 + ( y_{1mid} - y_{41})^2}} \text{ unit.}$$

**Step13.** Calculate middle point pixel coordinate $(x_{2mid},y_{2mid})$ of the boundary rectangle around right eye as $x_{2mid} = (x_{21}+x_{22})/2$ and $y_{2mid} = (y_{21}+y_{22})/2$.

**Step14.** Calculate Euclidian Distance between middle point pixel coordinate $(x_{2mid},y_{2mid})$ of the boundary rectangle around right eye and top – right pixel coordinate $(x_{42},y_{41})$ of the boundary rectangle around mouth as:

$$\mathbf{Dist2 = \sqrt{( x_{2mid} - x_{42})^2 + ( y_{2mid} - y_{41})^2}} \text{ unit.}$$

**Step15.** Write the value of Dist1 and Dist2 in a output text file for comparison.
**Step16.** Repeat step 1 to 15 for another same human face but with smiling facial expression.
**Step17.** Compare both input face images according the distances measured between eyes & mouth. The image with larger distance is considered as Happy face or smiling face, in general.
**Step18.** Exit

## 4.2 2D_Norm_Crosscorr (Template Gray scale Image, Input Gray scale Image)

**Step1.** Start
**Step2.** Perform 2D Cross Correlation between Template Image and Input Image pixel values and return 2D array C of size m*n with values of the corresponding Cross correlation where m*n is the size of the Input Image.
**Step3.** End

## 4.3 Find_Max(C[ ][ ])

**Step1.** Start
**Step2.** Find Maximum Value of 2D Array C[ ][ ] and determine the corresponding rectangular region where the maximum value is found.
**Step3.** Find top – left position coordinate (x,y), width (w) and height (h) of the rectangular region and return the values.
**Step4.** End

## 4.4 Facial_expression_recognition (Input Image, 3 Training Image Databases)

**Step19.** Start
**Step20.** Read Input Human Face Image and Store the pixel values to an array let face.
**Step21.** Call Processed_Face = imPreprocess(face)
**Step22.** Set i=1
**Step23.** Repeat Step 6 to 9 for every Image of Train_Neutral_Other Image Database

**Step24.** Read the Image from the Database and Store the pixel values to an array let t.
**Step25.** Call t1= imPreprocess(t)
**Step26.** Store t1 into image cell Train_Neutral_Other_Cell as
Train_Normal_Other_Cell(1,i)=t1
**Step27.** Set i=i+1
**Step28.** Set i=1
**Step29.** Repeat Step 12 to 15 for every Image of Train_Smiling_Other Image Database
**Step30.** Read the Image from the Database and Store the pixel values to an array let t.
**Step31.** Call t1= imPreprocess(t)
**Step32.** Store t1 into image cell Train_Smiling_Other_Cell as
Train_Smiling_Other_Cell(1,i)=t1
**Step33.** Set i=i+1
**Step34.** Set i=1
**Step35.** Repeat Step 18 to 21 for every Image of Train_Angry_Sad Image Database
**Step36.** Read the Image from the Database and Store the pixel values to an array let t.
**Step37.** Call t1= imPreprocess(t)
**Step38.** Store t1 into image cell Train_Angry_Sad_Cell as
Train_Angry_Sad_Cell(1,i)=t1
**Step39.** Set i=i+1
**Step40.** Create 3 2D Array of size (no_of_images * mn) for 3 Training Databases where no_of_images refers to the total number of images in the corresponding training databases respectively and m,n refers to the predefined size mentioned in Impreprocess function. Let traindata1 (n1 *mn), traindata2 (n2*mn) and traindata3 (n3*mn) are 3 arrays for Train_Neutral_Other, Train_Smiling_Other and Train_Angry_Sad Training Image Databases respectively, with n1, n2 and n3 are number of images in the corresponding databases.
**Step41.** Initialize all elements of traindata1, traindata2 and traindata3 array to 0.
**Step42.** Set i=1
**Step43.** Repeat step 26 to27 for n1 times
**Step44.** Set traindata1(i,:)= Train_Normal_Other_Cell(1,i)
**Step45.** Set i=i+1
**Step46.** Set i=1
**Step47.** Repeat step 30 to31 for n2 times
**Step48.** Set traindata2(i,:)= Train_Smiling_Other_Cell(1,i)
**Step49.** Set i=i+1
**Step50.** Set i=1
**Step51.** Repeat step 34 to35 for n3 times
**Step52.** Set traindata3(i,:)= Train_Angry_Sad_Cell(1,i)
**Step53.** Set i=i+1
**Step54.** Create 3 1D Arrays, namely class1, class2 and class3 of size n1,n2 and n3 respectively corresponding to Train_Neutral_Other, Train_Smiling_Other and Train_Angry_Sad Training Image Databases respectively.
**Step55.** Set i=1
**Step56.** Repeat Step 39 to 40 for all images of Train_Neutral_Other Image Database
**Step57.** If $i^{th}$ image of Train_Neutral_Other is of Neutral expression
then set class1(i)= 1
else set class1(i)= -1
**Step58.** Set i=i+1
**Step59.** Set i=1
**Step60.** Repeat Step 43 to 44 for all images of Train_Smiling_Other Image Database
**Step61.** If $i^{th}$ image of Train_Smiling_Other is of Smiling expression
then set class2(i)= 2
else set class2(i)= -2
**Step62.** Set i=i+1

**Step63.** Set i=1
**Step64.** Repeat Step 47 to 48 for all images of Train_Angry_Sad Image Database
**Step65.** If $i^{th}$ image of Train_Angry_Sad is of Angry expression
then set class3(i)= 3
else set class3(i)= -3
**Step66.** Set i=i+1
**Step67.** Call SVMTrained1=SVM_Training(traindata1,class1)
SVMTrained2=SVM_Training(traindata2,class2)
SVMTrained3=SVM_Training(traindata3,class3)
**Step68.** Call result1=SVM_Classify(SVMTrained1, Processed_Face)
result2=SVM_Classify(SVMTrained2, Processed_Face)
result3=SVM_Classify(SVMTrained3, Processed_Face)
**Step69.** Call FinalExpression=Recognize_Expression(result1,result2,result3)
**Step70.** Display FinalExpression as output
**Step71.** Exit

### 4.5 Impreprocess (Image_Pixel_Array)

**Step1.** Start
**Step2.** Convert Image_Pixel_Array to its corresponding double format let Image_Pixel_Array_Double.
**Step3.** If Image_Pixel_Array_Double is of format a*b*3, then
convert it to Corresponding Gray scale and save the pixel values to a 2D array let gImage.
Else
save the pixel values of the input image to a 2D array let gImage.
**Step4.** Resize gImage to a Predefined size say m*n & save the pixel values to a 2D array let gImage_Resized.
**Step5.** Reshape gImage_Resized Array to a 2D array of size 1*(mn) & save the pixel values to a 2D array let gImage_Reshaped.
**Step6.** Return the Array gImage_Reshaped.
**Step7.** End

### 4.6 SVM_Training (Training_Data, Group_Membership_Class)

**Step1.** Start
**Step2.** Train Linear Support Vector Machine with Training_Data and Group_Membership_Class and store the value in an array let SVM1.
**Step3.** Return the array SVM1
**Step4.** End

### 4.7 SVM_Classify (SVM_Trained, Img_Array)

**Step1.** Start
**Step2.** Classify Img_Array in one of the classes with SVM_Trained and SVM Binary Classifier and store the value in a variable let Classifier1
**Step3.** Return the value Classifier1
**Step4.** End

### 4.8 Recognize_Expression (Val1, Val2, Val3)

**Step1.** Start
**Step2.** If Val1=1

Set Expression=Neutral
Else if Val1= -1 and Val2=2
Set Expression=Smiling
Else if Val1= -1 and Val2= -2 and Val3=3
Set Expression=Angry
Else if Val1= -1 and Val2= -2 and Val3= -3
Set Expression=Sad

**Step3.** Return the value of Expression

**Step4.** End

# 5. Results and Discussion

## 5.1 Test Results for Neutral or Smiling face Detection Using Facial Features:

- Testing includes sets of images with relatively different lighting condition. Each set of image is the images of same human face with two different emotions (Neutral & with smiling). We perform template matching on both faces of a set of image using different templates. Templates, i.e. pair of eyes, nose & mouth area used, may be the regions of the same test image or different image other than the test image.
- After template matching, Euclidian distances between midpoint of the rectangle region of eye areas & the top two corner points (left & right) of the rectangle region of mouth area are calculated on both images.
- The distances are compared and the distance with larger value primarily yields that the person is smiling. A neutral face has smaller distance than the smiling expression of the same face, generally.

Test result: Template Matching

Templates used:

- Left eye template
- Right eye template
- Nose template
- Mouth template

**Case 1:** Matched with templates of test image itself

**Fig. 7. Test image**

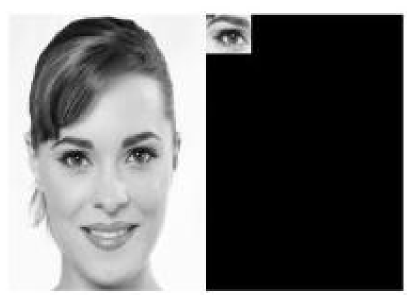


**Fig. 8. Left eye template**

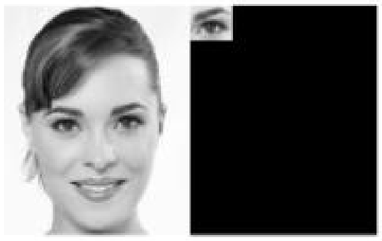


**Fig. 9. Right eye template**

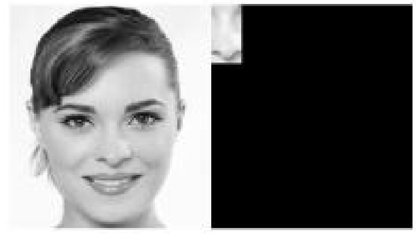


**Fig. 10. Nose template**

**Fig. 11. Mouth template**

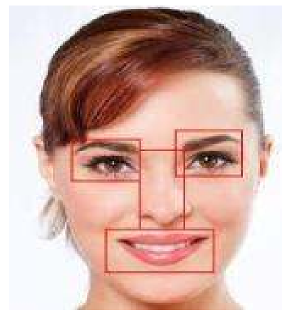
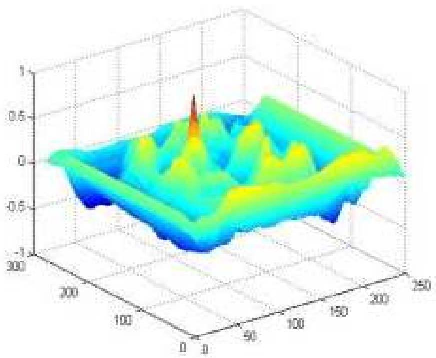
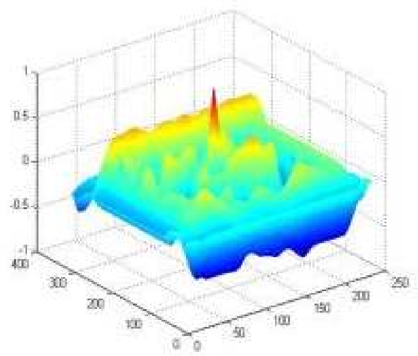


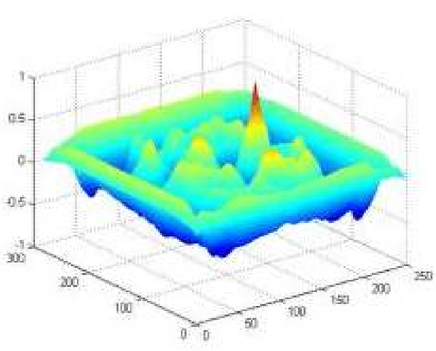
**Fig. 12. Output image**



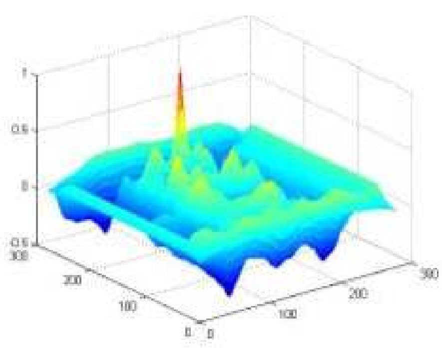
**(i)**



**(iii)**



**(ii)**



**(iv)**

**Fig. 13. Peak normalized cross-correlation value when matched with (i) Left eye (ii) Right eye (iii) Nose (iv) Mouth template**

**Case 2:** Matched with different templates

**Fig. 14. Test image**

**Fig. 15. Templates of this image**

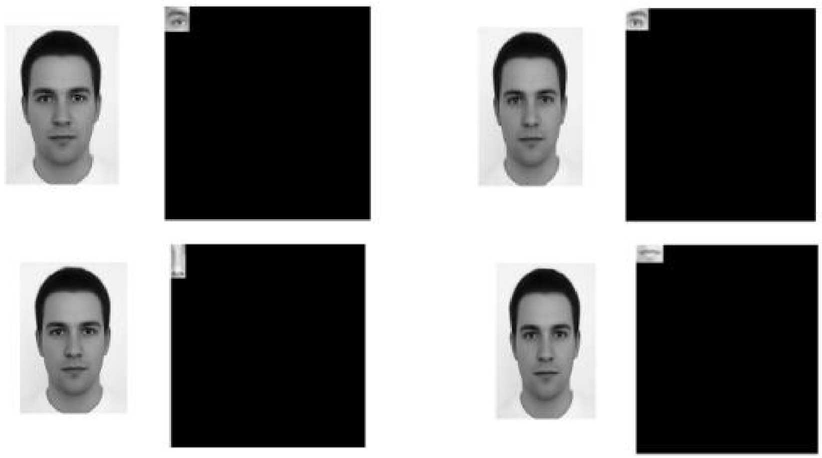


**Fig. 16. Templates of different image than the test image**

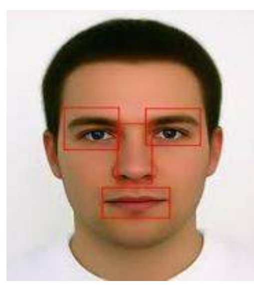


**Fig. 17. Output image**

**Distance calculations and neutral or smiling expression detection:**

**Fig. 18. Neutral face**

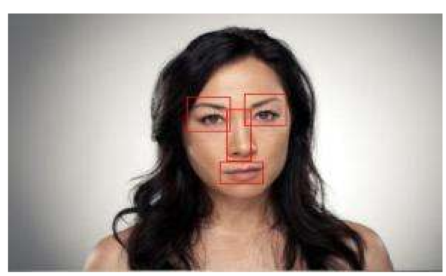


**Fig. 19. Neutral face template matching**

Fig. 20. Smiling face

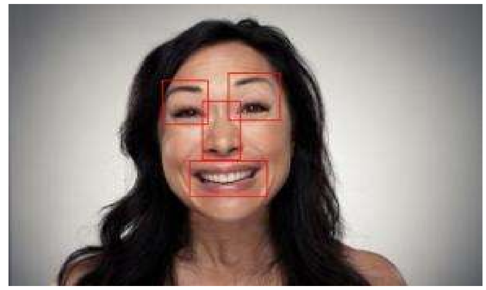


Fig. 21. Smiling face template matching

**<u>Euclidean distance calculation</u>:**

Neutral face:

Distance between Left eye & Mouth: 50.01 unit
Distance between Right eye & Mouth: 53.04 unit

Smiling face:

Distance between Left eye & Mouth: 54.08 unit
Distance between Right eye & Mouth: 60.02 unit

The distances between the eyes and the mouth are larger for smiling face.

**Displaying the processing in user interface:**


Fig. 22. Processing done for facial feature detection


**Fig. 23. Distance between eyes and mouth for neutral face**

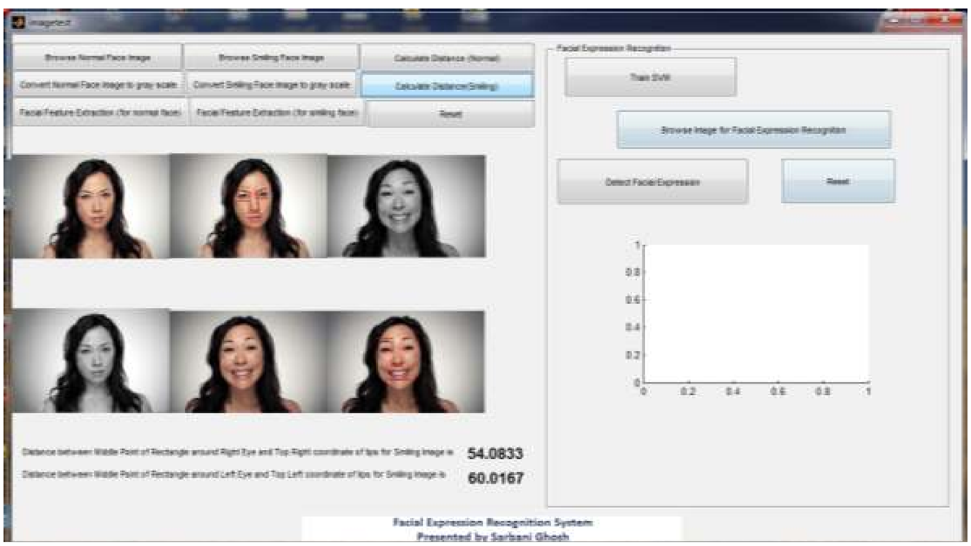



**Fig. 24. Distance between eyes and mouth for smiling face**

## Test Results for Expression Recognition:

- In the proposed method, four categories of expressions of human face using database [8] are considered for classification. They are:
  - Neutral
  - Smiling
  - Angry
  - Sad

- Training set includes 16 face images of which each type of expression include four images. The training set is divided into three subsets of images. First set contains all the 16 images with four different expressions. Second set contains 12 images of three different expressions excluding Neutral expression. Third set contains 8 images of two different expressions excluding smiling expression.
- Test images may be of any size and varying lighting condition. The test images will undergo some image pre-processing steps and become identical size and shape like train set of images.
- The column vectors of the training image matrices are used as features to train the training set of images. Test images are also classified to any expression based on these features.

**SVM Training:**

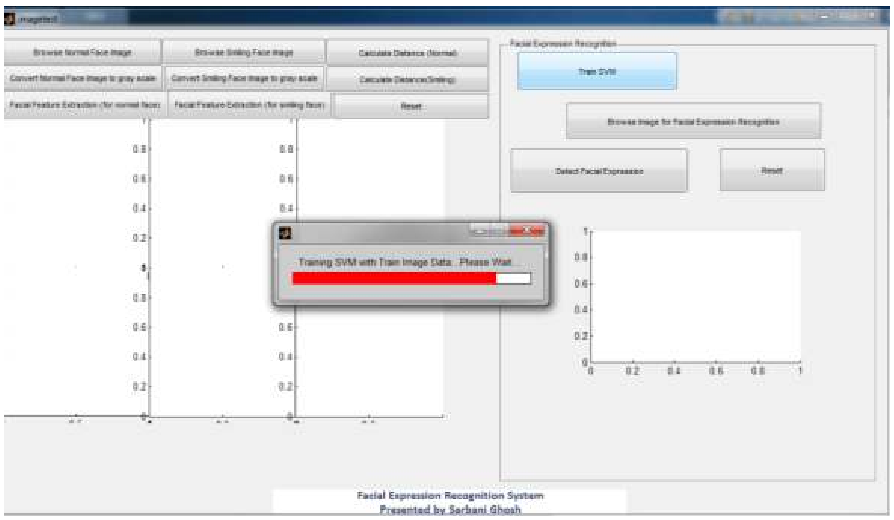



**Fig. 25. Training image database**

**Neutral expression recognition:**

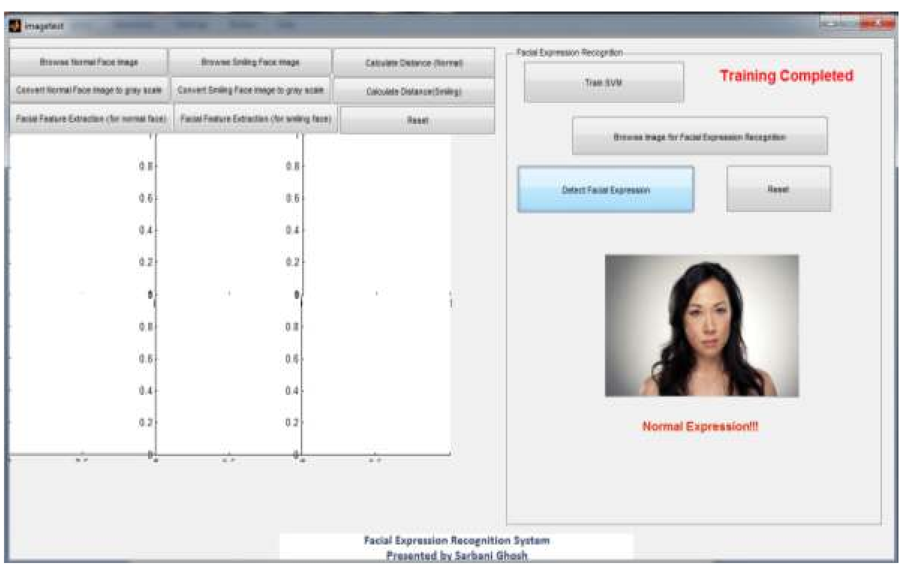



**Fig. 26. Normal expression**

**Smiling expression recognition:**

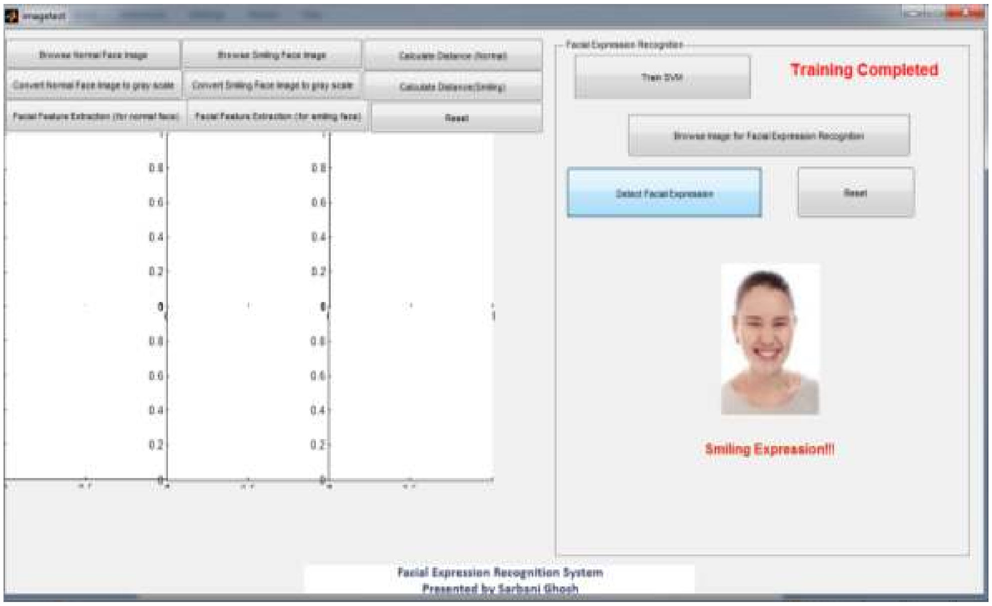



**Fig. 27. Smiling expression**

**Angry expression recognition:**

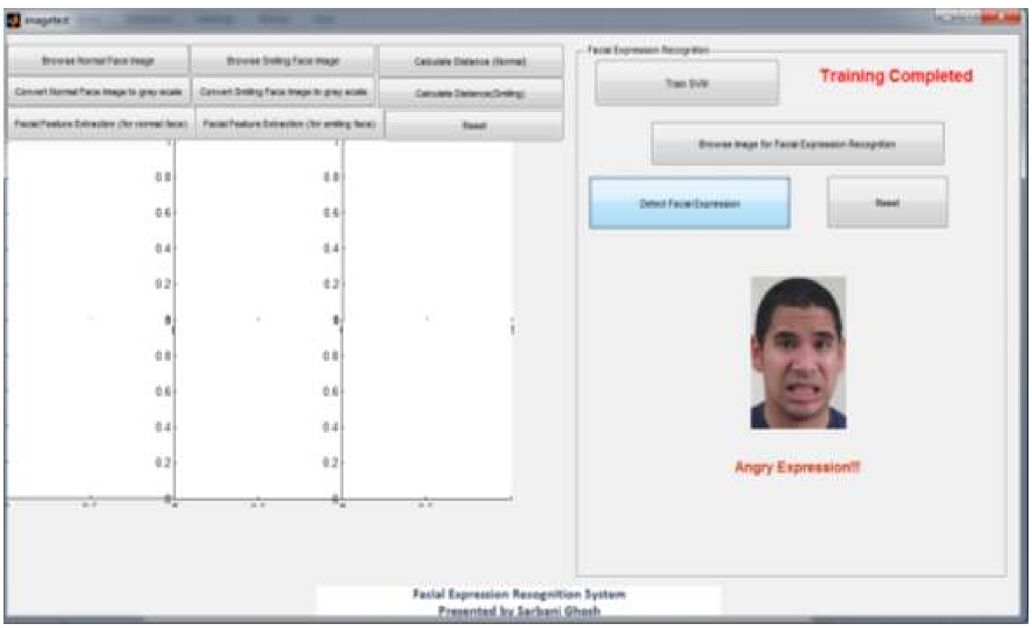



**Fig. 28. Angry expression**

**Sad expression recognition:**

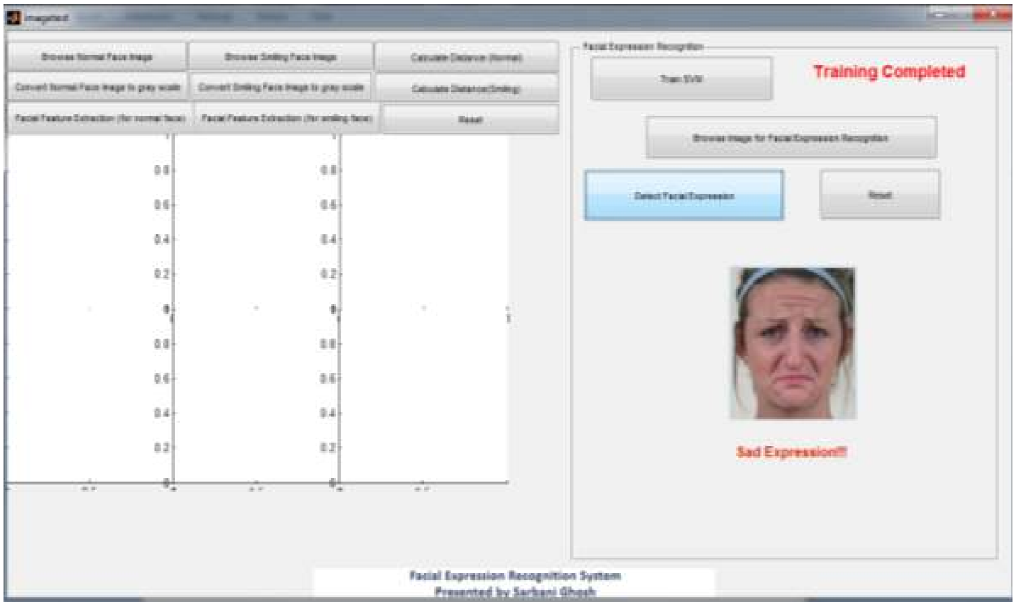



**Fig. 29. Sad expression**

# 6 Performance Analysis

## 6.1 Smiling or Neutral Face Detection Using Facial Features

Since the main purpose of work is smile recognition, therefore sample pictures must be taken with two different emotions, i.e. with smile or without smile. This research is based on some common assumption:

- Although templates of eyes, nose or mouth from different images can be used to match with a test image, best possible matching & region of interest detection can occur in the case where templates images are cropped from the same test image.
- Distance between eyes & mouth depends on the size of the template image & finally on the size of the rectangle region of interest. There may be some cases where, distance between eyes & mouth will be smaller than the distance measured with the same neutral face, due to very small size of template images, so as to size of region of interest. Generally, the template images should cover the facial features broadly so that the whole eye region or nose region or mouth region is covered.
- Templates of neutral image cannot be matched on a test image of smiling face & vice versa. Both set of templates should be different.

A comparative study on various approaches are made to identify the person's face and thereafter recognize their emotion. Among the various techniques implemented, Neural Networks, Hidden Markov Model and Dimensionality reduction techniques have been made in comparison with our proposed method. It is shown in Tables 1 and 2.

**Table 1. Comparison of the results**

| Method | Recognition rate |
|---|---|
| K-Means | 86.75 |
| Fuzzy Ant with Fuzzy C – means | 94.82 |
| Proposed | 97.018 |

**Table 2. Matching percentage of three methods**

| **Rank** | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Eigen-face | 26 | 33.8 | 47.1 | 52.4 | 55 |
| Sketch transform method | 71 | 78 | 81 | 84 | 88 |
| Proposed method | 80 | 82.1 | 84 | 90.1 | 92.4 |

**Performance of the Template matching algorithm:**

**Case 1. Matched with templates of the test image itself**

| **No.** | **Type of Emotion** | **No. of Input Images** | **Recognized** | **Result (%)** |
|---|---|---|---|---|
| 1 | Neutral | 20 | 20 | 100 |
| 2 | Happy | 10 | 10 | 100 |

**Case 2. Matched with templates of different images other than the test image**

| **No.** | **Type of emotion** | **No. of input images** | **Recognized** | **Result (%)** |
|---|---|---|---|---|
| 1 | Neutral | 20 | 15 | 75 |
| 2 | Happy | 10 | 7 | 70 |

**Performance of Smiling face identification algorithm after Template matching:**

| **No. of sets of images** | **No. of neutral faces** | **No. of happy faces** | **No. of sets of images where distance between eyes & mouth is larger for one face than the other in the particular set** | **No. of sets of images where smiling face is identified.** | **Result (%)** |
|---|---|---|---|---|---|
| 10 | 10 | 10 | 8 | 8 | 80 |

**Smiling or neutral face detection using facial features:**

The main purpose of the work is to recognize four basic expressions of human face. Linear support vector machine is used as the classifier as the performance measurement and accuracy of SVM is high.

- The train set of images is able to classify the predefined four types of expressions on various human faces. Due to lack of different image sets with various expressions, the work is confined into four basic facial expressions although the method can be further extended to classify many more different expressions in human faces. Only small changes including class definition and train set arrangement need to be done to extend the method.
- For some face images the method cannot properly identify the exact expression in the image as human facial expression is too much variable. For any particular expression, human face may vary to a great extent. So there is a need of many different face images to load the train set so that the set can identify any test image's expression exactly. But we have lack of this.
- If some explicit feature points can be identified and used as features based on which classification is done, then more accurate result can be expected.

**Performance of Expression Recognition System:**

| **No. of training sets** | **Total number of images in training set** | **No. of Neutral/ Smiling/Angry/Sad faces in training set** | **Total no. of tested images with neutral & smiling & angry & sad expression** | **No. of images whose expression are identified accurately** | **Result (%)** |
|---|---|---|---|---|---|
| 3 | 16 | 4 | 20 | 17 | 85 |

# 7 Conclusion

Facial expression recognition or Expression detection system has numerous applications in image processing domains, security applications domain or any type of biometric system. This research work is a primary step of an emotion detection system using which one specific emotion can be identified after extracting different facial features. Future work will be done to classify more different types of expressions in human face and identify each emotion properly. The proposed approach will help to define uniqueness of human face component accurately to some extent which can be used greatly in biometric recognition system. Also the algorithm can be used further in face recognition systems, machine learning system and other types of image processing applications.

## 7.1 Advantages

- The proposed method takes the advantage of geometric features of a human face to detect facial expressions leads to accurate detection of facial features and expression of a human face.
- Support Vector Machine is used as multiclass classifier to classify more than two expressions in the proposed algorithm which overcome the limitations of binary SVM which is able to do only two-class classification. This method can be further extended to more than four class classification to classify various human facial expressions easily.
- The expression recognition method can detect up to twenty images of any of the four predefined expression. If the training database can be loaded with more different images then the method will work more accurately.

## 7.2 Limitations

- Although neutral or smiling face using geometric facial features can be identified for 80% of input images, for some smiling face images, distance between eyes & mouth become smaller than the normal expression of the same person. This is because human face shape and expression is too much variable and may vary sometimes from the general probability.
- Due to lack of a huge set of training set of images in expression classification, some facial expression cannot be identified accurately by the proposed method. If the training database can be loaded with more different images then the method will work more accurately.
- No other geometric facial feature or measurement is used for SVM classification of facial expressions. Some specific feature points like, distance between eyes & mouth, distance between two eyes, width of left & right eye , distance between eyes & nose-peak, total skin area- all these can be used as feature space for classification and the proposed method will work more efficiently.

# 8 Future Scope

A primary work is done to develop an Expression Detection system which will be able to identify many different types of Emotion or facial expression in human face. Here, in the specified algorithm, we first tried to extract all the regions of interest i.e., left eye, right eye, nose & mouth region using template matching technique. Then, we measured the vertical distance between two eyes & mouth to detect one expression that whether a person is smiling or not.

## 8.1 Next Steps

**Measurement of distances among other facial features :** To identify many more different types of expression in human face, further we have to measure distances between multiple face components like:

- Right eye-left eye
- Left eye-nose peak
- Right eye-nose peak
- Nose peak-mouth
- Nose height & width

All of these features will define the uniqueness of a whole facial component. These features can be used in SVM train data and the set of images can be classified based on these properties of human faces. For test images, the same feature points can be used to map the images to particular classes. The accuracy of the classification will be competitive.

**Large set of images for training database:** If more than 200 images with different facial expressions can be used for train database and they arranged properly in one against all approach in more than one train subsets, then the SVM classifier can be used as multiclass classification which will be able to detect facial expressions more than four. The corresponding classes for each train set need to be defined properly.

The facial expression recognition system using facial features can be further used to uniquely identify any human face which can be used in biometric system design or any other real time face recognition system.

## Competing Interests

Authors have declared that no competing interests exist.

## References

[1] Alom Md. Zahangir, Piao Mei-Lan, Alam Md. Ashraful, Kim Nam, Park Jae-Hyeung. Facial expressions recognition from complex background using face context and adaptively weighted sub-pattern PCA" on World Academy of Science. Engineering and Technology. 2012;6.

[2] Dewi Agushinta R, Adang Suhendra, Sarifuddin Madenda, Suryadi HS. Face component extraction using segmentation method on face recognition system. On Journal of engineering Trends in Computing & Information Science. 2011;2:2.

[3] Gupta Neha, Kaur Prof. Navneet. Design and implementation of emotion recognition system by using matlab. On International Journal of Engineering Research and Applications (IJERA). 2013;3(4) : 2002-2006.

[4] Roy Sunita, Prof. Bandyopadhyay Samir Kumar. Extraction of facial features using a simple template based method. On International Journal of Engineering Science and Technology (IJEST).

[5] Roy Sunita, Roy Sudipta, Prof. Bandyopadhyay Samir K. A tutorial review on face detection. On International Journal of Engineering Science and Technology (IJEST). 2012;1(8).

[6] Lajevardi SM, Lech M. Facial expression recognition using neural networks and log-gabor filters. Digital Image Computing: Techniques and Applications IEEE; 2008.

[7] Ravi S, Wilson S. Face Detection with facial features and gender classification based on support vector machine. Special Issue - International Journal of Imaging Science and Engineering2010.

[8] Gross R, Face Databases, Handbook of Face Recognition, Stan Z. Li, Anil K. Jain, ed. Springer-Verlag. 2005;22.



***Peer-review history:***
*The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)*
*http://sciencedomain.org/review-history/11384*