Facial Expression Recognition using Principal Component Analysis Combined with Fisher Linear Discriminant Analysis

Nagaraj Bhat¹, Sumangala Bhavikatti²

¹,²Department of Electronics and Communication Engineering
SDM College of Engineering and Technology, Dharwad, Karnataka, India

Abstract—Human facial expressions play an important role in interpersonal relations. This is because humans demonstrate and convey a lot of evident information visually rather than verbally. Although humans recognize facial expressions virtually without effort or delay, reliable expression recognition by machine remains a challenge as of today. To automate recognition of emotional state, machines must be taught to understand facial gestures.

In this paper, we develop a facial expression recognition algorithm which is insensitive to variation in lighting direction and facial expression. Taking a pattern classification approach, we linearly project the image into a subspace in a manner which discounts those regions of the face with large deviation. Our projection method is based on Fisher’s Linear Discriminant (FLD) and produces well separated classes in a low-dimensional subspace, even under severe variation in lighting and facial expressions. The Eigenface technique (PCA), another method based on linearly projecting the image space to a low dimensional subspace, has similar computational requirements. Euclidean distances between the projected test image and the projection of all centered training images were calculated. Test image is supposed to have minimum distance with its corresponding image in the training database. The algorithm is tested in MATLAB. Experimental results were tested on databases like JAFEE, Cohn-Kanade and with our own images.

Keywords—FLD, PCA, Facial Expression recognition, LDA, Eigenface

I. INTRODUCTION

Human beings express their emotions in everyday interactions with others. Emotions are frequently reflected on the face, in hand and body gestures, in the voice, to express our feelings or liking. Recent Psychology research has shown that the most expressive way humans display emotions is through facial expressions. Mehrabian [1] indicated that the verbal part of a message contributes only for 7% to the effect of the message as a whole, the vocal part for 38%, while facial expressions for 55% to the effect of the speaker’s message. Emotions are feeling or response to particular situation or environment. Emotions are an integral part of our existence, as one smiles to show greeting, frowns when confused, or raises one’s voice when enraged. It is because we understand other emotions and react based on that expression only enriches the interactions. Computers are “emotionally challenged”. They neither recognize other emotions nor possess its own emotion [2]. To enrich human-computer interface from point-and-click to sense-and-feel, to develop non intrusive sensors, to develop lifelike software agents such as devices, this can express and understand emotion. Since computer systems with this capability have a wide range of applications in different research arrears, including security, law enforcement, clinic, education, psychiatry and Telecommunications [4]. There has been much research on recognizing emotion through facial expressions. Based on this we classify the emotion into positive and negative emotions. The six basic emotions are angry, happy, fear, disgust, sad, surprise. One more expression is neutral. Other emotions are Embarrassments, interest, pain, shame, shy, anticipation, smile, laugh, sorrow, hunger, curiosity.

A. Facial Expression Recognition

A survey on the research made regarding facial expression Recognition can be found in [4] and [5]. The approaches reported regarding facial expression recognition can be distinguished in two main directions, the feature-based ones and the template-based ones, according to the method they use for facial information extraction. The feature-based methods use texture or geometrical information as features for expression information extraction. The template-based methods use 3-D or 2-D head and facial models as templates for expression information extraction. To represent facial expression images effectively, several methods have been proposed such as PCA (Principal Component Analysis), ICA (Independent Component Analysis), Gabor representation, Optic flow, and geometrical tracking method. Among them, the Gabor representation has been favoured among many researchers because of its better performance and biological implication. The Enhanced Fisher discrimination Model (EFM) is very effective in a face recognition task. Moreover, when it
is combined with the Gabor representation, the face recognition performance is improved greatly. Given that there is evidence that Gabor representation is effective for the facial expression image, we thought that Gabor representation with the EFM could be used effectively in the facial expression task as well. This paper attempts to propose an Expression-Independent face recognition method which is invariant to variations of facial expression.

1) Feature-Based Approaches: Facial feature detection and Tracking is based on active Infrared illumination in [6], in order to provide visual information under variable lighting and Head motion. The classification is performed using a Dynamic Bayesian Network (DBN).

A method for static and dynamic segmentation and classification of facial expressions is proposed in [7]. For the static case, a DBN is used, organized in a tree structure. For the dynamic approach, multi level Hidden Markov Models (HMMs) classifiers are employed.

The system proposed in [8] automatically detects frontal faces in the video stream and classifies them in seven classes in real time: neutral, anger, disgust, fear, joy, sadness, and surprise. An expression recognizer receives image regions Produced by a face detector and then a Gabor representation Of the facial image region is formed to be later processed by a bank of SVMs classifiers.

Gabor filters are also used in [9] for facial expression recognition. Facial expression images are coded using a multiorientation, multiscale set of Gabor filters which are topographically ordered and aligned approximately with the face. The similarity space derived from this facial image representation is compared with one derived from semantic ratings of the images by human observers. The classification is performed by comparing the produced similarity spaces.

The images are first transformed using a multiscale, multiorientation set of Gabor filters in [10]. The grid is then registered with the facial image region either automatically, using elastic graph matching [11] or by manual clicking on fiducial face points. The amplitude of the complex valued Gabor transform coefficients are sampled on the grid and combined into a single vector, called a Labeled Graph Vector (LGV). The classification is performed using the distance of the LGV from each facial expression cluster center. Gabor features are used for facial feature extraction given a set of fiducial points in [12]. The classification is performed using Bayes, SVMs, Adaboost, and linear programming classifiers.

A Neural Network (NN) is employed to perform facial expression recognition in [13]. The features used can be either the geometric positions of a set of fiducial points on a face or a set of multiscale and multiorientation Gabor wavelet coefficients extracted from the facial image at the fiducial points. The recognition is performed by a two layer perceptron NN. A convolutional NN was used in [14]. The system developed is robust to face location changes and scale variations. Feature extraction and facial expression classification were performed using neuron groups, having as input a feature map and properly adjusting the weights of the neurons for correct classification. A method that performs facial expression recognition is presented in [15]. Face detection is performed using a Convolutional NN, while the classification is performed using a rule-based algorithm. Optical flow is used for facial region tracking and facial feature extraction in [16]. The facial features are inserted in a Radial Basis Function (RBF) NN architecture that performs classification. The Discrete Cosine Transform (DCT) is used in [17], over the entire face image as a feature detector. The classification is performed using a one-hidden layer feed forward NN.

A feature selection process that is based on principal component analysis (PCA) is proposed in [18]. A decision tree-based classifier that uses successive projections onto more precise representation subspaces, is employed. The image pixels are used in [19] as input to PCA and Linear Discriminant Analysis (LDA) to reduce the original feature space dimensionality. The resulted features are lexicographically ordered and concatenated to a feature vector, which is used for classification according to the nearest neighbor rule.

2) Model Template-Based Approaches: Two methods for facial expression recognition are proposed in [22], based on a 3-D model enriched with muscles and skin. The first method estimates facial muscle actuations from optical flow data. The classification is performed according to its similarity to the classical patterns of muscle actuation. The second method uses the classical patterns of muscle actuation to generate the classical pattern of motion energy associated with each facial expression, thus resulting in a set of simple facial expression “detectors,” each of which looks for the particular space-time pattern of motion energy associated with each facial expression.

A face model, defined as a point-based model composed of two 2-D facial views (frontal and profile views) is used in [3]. The deformation of facial features is extracted from both the Frontal and profile views and its correspondence with the FAUs is established. The facial expression recognition is performed based on a set of decision rules.

A 3-D facial model is proposed in [23]. Anatomically-based muscles are added to it. A Kalman filter in correspondence with optical flow computation are used to extract muscle action in order to form a new model of facial action, the so-called FACS.

A 3-D facial model used for facial expression recognition is also proposed in [24]. First, the head pose is estimated in a facial video sequences. Subsequently, face images are warped onto a face model with canonical face geometry, and then they are rotated to frontal ones, and are projected back onto the image plane. Pixels brightness is linearly rescaled and
resulting images are convolved with a bank of Gabor kernels. The Gabor representations are then channeled to a bank of SVMs to perform facial expression recognition.

II. PRINCIPAL COMPONENT ANALYSIS

A common problem in data processing is that large amounts of data are expensive to transmit, store or process. For transmitting we need high bandwidth, for storing large storage space and for processing we need complex computer systems to reduce the long processing time. To reduce the amount of data would mean a reduction in expenses. But simply throwing away part of the data would result in a loss of information, which could be important. In so called random data, like for example data from images, sounds or other samples, there is however a difference in how important each part of data is to the information which is stored in the data. By leaving out the part of data which is the least valuable to the information, we reach a reduction of the amount of data.

Principal Components Analysis (PCA) is used to compress data in such a way that the least information is lost. It does so by truncating data and thereby leaving out the data which is of the least importance to the information stored in the data. This PCA process is called dimensionality reduction, because a vector $\vec{x}$ which contains the original data and is N-dimensional is reduced to a compressed vector $\vec{c}$ which is M-dimensional, where M<N.

Principal Component Analysis is a standard technique used in statistical pattern recognition and signal processing for data reduction and Feature extraction. Principal Component Analysis (PCA) is a dimensionality reduction technique based on extracting the desired number of principal components of the multi-dimensional data. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables. The first principal component is the linear combination of the original dimensions that has the maximum variance; the n-th principal component is the linear combination with the highest variance, subject to being orthogonal to the n-1 first principal components. An ensemble of images maps to a collection of points in this huge space. Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principle component is to find the vectors that best account for the distribution of face images within the entire image space. These vectors define the subspace of face images, which we call “face space”.

PCA is an information theory approach of coding and decoding face images that give an insight into the information content of face images, emphasizing the significant local and global "features". Such features may or may not be directly related to face features such as eyes, nose, lips, and hair.

In the language of information theory, we want to extract the relevant information in a face image, encode it as efficiently as possible, and compare one face encoding with a database of models encoded similarly. A simple approach to extracting the information contained in an image of face is to somehow capture the variation in a collection of images, independent of any judgment of features, and use this information to encode and compare individual face images. And this can be done with the help of Eigen vectors.

These Eigen vectors can be thought of as a set of features that together characterize the variation between face images. Each image location contributes more or less of each Eigen vector; so that we can display the eigenvector as a sort of ghostly face which we call an Eigenface. Each individual face can be represented exactly in terms of a linear combination of the Eigenfaces. Each face can also be approximated using only the "best" Eigenfaces-those that have the largest Eigen values and which therefore account for the most variance within the set of face images. The best M eigenfaces span an M-Dimensional subspace- "face space" of all possible images.

A. Computation of PCA

Step 1: Get some data. The data can be of any dimensions.

Step 2: Subtract the mean.

For PCA to work properly, we need to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension. This produces a data set whose mean is zero.

Step 3: Calculate the covariance matrix

Recall that covariance is always measured between 2 dimensions. If we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated. For example, from a 3 dimensional data set dimensions(x, y, z) you could calculate cov(x, y) cov(y, z) and cov (z, x). In fact, for an n-dimensional data set, you can calculate $\frac{n!}{(n-2)!+2}$ different covariance values.

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are rather important, as they tell us useful information about our data. So, by this process of taking the eigenvectors of the covariance matrix, we have been able to extract lines that characterize the data.

Step 5: Choosing components and forming a feature vector
Here is where the notion of data compression and reduced dimensionality comes into it. If we look at the eigenvectors and eigenvalues from the previous section, it can be seen that the eigenvalues are of quite different values. In fact, it turns out that the eigenvector with the highest eigenvalue is the principle component of the data set. The eigenvector with the largest eigenvalue give the significant relationship between the data dimensions. In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance. Now, if we like, we can decide to ignore the components of lesser significance. We do lose some information, but if the eigenvalues are small, we don’t lose much. If we leave out some components, the final data set will have fewer dimensions than the original. What needs to be done now is we need to form a feature vector, which is just a fancy name for a matrix of vectors. This is constructed by taking the eigenvectors that we want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns.

Feature Vectors = (eig1 eig2 eig3 ……….eign)

Step 6: Deriving the new data set
This is the final step in PCA. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed.

Final Data = Row Feature Vector*Row Data Adjust
Where row feature vector is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and row data adjust is the mean adjusted data transposed, i.e. the data items are in each column, with each row holding a separate dimension.

III. LINEAR DISCRIMINANT ANALYSIS

Linear discriminant analysis (LDA) and the related Fisher’s linear discriminant (FLD) are methods used in statistics, pattern recognition and machine learning to find a linear combination of features which characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification.

LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made. LDA works when the measurements made on independent variables for each observation are continuous quantities. When dealing with categorical independent variables, the equivalent technique is discriminant correspondence analysis.

A. LDA for Two Classes

Consider a set of observations \( \hat{x} \) (also called features, attributes, variables or measurements) for each sample of an object or event with known class \( y \). This set of samples is called the training set. The classification problem is then to find a good predictor for the class \( y \) of any sample of the same distribution (not necessarily from the training set) given only an observation \( \hat{x} \). LDA approaches the problem by assuming that the conditional probability density functions \( P(\hat{x} | y = 0) \) and \( P(\hat{x} | y = 1) \) are both normally distributed with mean and covariance parameters \( (\mu', \Sigma_{y=0}) \) and \( (\mu', \Sigma_{y=1}) \), respectively. Under this assumption, the Bayes optimal solution is to predict points as being from the second class if the ratio of the log-likelihoods is below some threshold \( T \), so that

\[
\frac{(\hat{x} - \mu_0)\Sigma_{y=0}^{-1}(\hat{x} - \mu_0) + \ln|\Sigma_{y=0}|}{(\hat{x} - \mu_1)\Sigma_{y=1}^{-1}(\hat{x} - \mu_1) + \ln|\Sigma_{y=1}|} < T
\]

Without any further assumptions, the resulting classifier is referred to as QDA (quadratic discriminant analysis). LDA also makes the simplifying homocedastic assumption (i.e. that the class covariance’s are identical, so \( \Sigma_{y=0} = \Sigma_{y=1} = \Sigma \)) and that the covariance’s have full rank. In this case, several terms cancel and the above decision criterion becomes a threshold on the dot product

\[
\hat{w}.\hat{x} < T
\]

For some threshold constant \( c \), where

\[
\hat{w} = \Sigma^{-1}(\hat{\mu}_1 - \hat{\mu}_0)
\]

This means that the criterion of an input \( \hat{x} \) being in a class \( y \) is purely a function of this linear combination of the known observations. It is often useful to see this conclusion in geometrical terms: the criterion of an input \( \hat{x} \) being in a class \( y \) is purely a function of projection of multidimensional-space point \( \hat{x} \) onto direction \( \hat{w} \). In other words, the observation belongs to \( y \) if corresponding \( \hat{x} \)'s is located on a certain side of a hyper plane perpendicular to \( \hat{w} \). The location of the plane is defined by the threshold \( c \).

B. Fisher’s Linear Discriminant Analysis

The terms Fisher’s linear discriminant and LDA are often used interchangeably, although Fisher’s original article
actually describes a slightly different discriminant, which does not make some of the assumptions of LDA such as normally distributed classes or equal class covariance’s. Suppose two classes of observations have means \( \mu_{y=0}, \mu_{y=1} \) and covariance’s \( \Sigma_y = 0, \Sigma_y = 1 \). Then the linear combination of features \( \mathbf{w}^T\mathbf{x} \) will have means \( \mathbf{w}^T\mu_{y=i} \) and variances \( \mathbf{w}^T\Sigma_y\mathbf{w} \) for \( i = 0, 1 \). Fisher defined the separation between these two distributions to be the ratio of the variance between the classes to the variance within the classes. This measure is, in some sense, a measure of the signal-to-noise ratio for the class labeling. It can be shown that the maximum separation occurs when:

\[
\mathbf{w} = (\Sigma_{y=0} + \Sigma_{y=1})^{-1}(\mu_{y=1} - \mu_{y=0})
\]

When the assumptions of FLD are satisfied, the above equation is equivalent to LDA. Be sure to note that the vector \( \mathbf{w} \) is the normal to the discriminant hyper plane. As an example, in a two dimensional problem, the line that best divides the two groups is perpendicular to \( \mathbf{w} \). Generally, the data points to be discriminated are projected onto \( \mathbf{w} \); then the threshold that best separates the data is chosen from analysis of the one-dimensional distribution. There is no general rule for the threshold. However, if projections of points from both classes exhibit approximately the same distributions, the good choice would be hyper plane in the middle between projections of the two means, \( \mathbf{w}^T\mu_{y=0} \) and \( \mathbf{w}^T\mu_{y=1} \). In this case the parameter \( c \) in threshold condition \( \mathbf{w}^T\mathbf{x} < C \) can be found explicitly

\[
C = \mathbf{w}^T(\mu_{y=0} + \mu_{y=1})/2
\]

C. Computation of FLD

Step1: Compute the transformation matrix of K–L transformation

\[
U_{kl} = \arg \max |U^T S_T U| = [u_1, u_2, \ldots, u_n]
\]

Where \{u_i\} = 1; 2; \ldots; n \] is the set of eigenvector of \( S_T \) corresponding to the nonzero eigenvalue and \( S_T \) is the total scatter matrix defined as

\[
S_T = \sum_{k=1}^{N}(x_k - \mu)(x_k - \mu)^T
\]

Step2: Compute the transformed within-class scatter matrix \( S_w' \), which is a full rank matrix

\[
S_w' = U_{kl}^T S_b U_{kl}
\]

Step3: Compute the transformed between-class scatter matrix \( S_b' \)

\[
S_b' = U_{kl}^T S_b U_{kl}
\]

Step4: The standard FLD defined is used to the transformed samples to obtain \( W_{fld} \)

\[
W_{fld} = \arg \max \frac{|W^T S_b' W|}{|W^T S_w W|}
\]

Step5: Compute \( W_{opt}^T \)

\[
w_{opt}^T = W_{fld}^T U_{kl}^T
\]

IV. METHODOLOGY

The design and implementation of facial expression recognition System can be subdivided into three main parts: Image Detection, Recognition technique which Includes Training of the images, Testing and then result of classification of images.

A. Image Detection

In this paper we used the images from the JAFFE, Cohn-Kanade and with our own databases. Once the image is detected the image region containing the face is extracted and geometrically normalized. References to detection methods using neural networks and statistical approaches can be found in [15]. We are using Radial basis function network which is capable of handling noisy images also. Its gives better result than back propagation neural network.

B. Recognition Technique

It aims at modelling the face using some mathematical representation in such a way the feature vector can be fed into a classifier. The overall performance of the system mainly
depends on the correct identification of face or certain facial features such as eyes, eyebrows and mouth. After the face is detected, there are two ways to extract the features: Holistic Approach, Analytic Approach. In Holistic, raw facial image is subjected for feature extraction. While in analytic, some important facial features are detected. Here we used Holistic Approach, so that it means we send a raw image as an input without any feature selection. We fix the block size as fifty. The images are fed into fifty rows and fifty columns.

C. Statistical Method

After reading the image, the image must be analysed for duplication. So that correlation of the matrix will be found. Since the correlation matrix for each image is square, we can Calculate Eigen vector and Eigen value for each matrix. These are very important so that it gives useful information about the data.

1). Eigen Values and Eigen Vectors

In Eigen vectors any vector change in magnitude but not in direction is called as Eigen vector. In Eigen values, the magnitude that the vector is changed is called an Eigen value.

\[ \mathbf{A}\mathbf{x} = \lambda \mathbf{x} \]

Where \( \mathbf{A} \) is \( n \times n \) matrix. \( \mathbf{x} \) is the length of \( n \) column vector. \( \lambda \) is a scalar. It's an Eigen value and \( x \) is the Eigen vector.

The Eigen values for angry image1: \( 0.1369, 0.1371, 0.1372, 0.1373, 0.1375, 0.1368, 0.1366, 0.1375, 0.1382, 0.1285, 0.1394, 0.1402, 0.1406, 0.1408, 0.1412, 0.1417, 0.143 \). The Eigen values for angry image3: \( 0.1367, 0.1371, 0.1375 \) and 0.1371. The above property can improve the network classifier’s performance and generalization. By minimizing within class variance and maximizing between class variance. The most famous example of dimensionality reduction is principal component analysis. This technique searches for direction in the data that have largest variance and subsequently project the data onto it. That removes some of the noisy directions. There are many issues with how many directions one needs to choose. It is a unsupervised technique [13]. When compare with Principle Component Analysis, Independent Component Analysis, and FLD gives maximum percentage of the output. It is best for classification, improves the performance and reduction technique. Fisher Linear Discriminant Analysis considers maximizing the following objective.

\[ J(w) = W^T S_B W / W^T S_W W \]

Where \( S_B \) is the “between class scatter matrix” and \( S_W \) is the “within class scatter matrix”. Then we find two matrixes which contain specific information of the data.

D. Training Phase

We present the network with training examples, which consist of a pattern of activities for the input units together with the desired pattern of activities for the output units. We determine how closely the actual output of the network matches the desired output. We change the weight of each connection so that the network produces a better approximation of the desired output. The input to the training phase is a collection of images showing human faces. These images are also called as Face images. These face images are then passed through a feature extraction step. In the feature extraction step key attributes of the images are computed and stored as a vector called feature vector. These feature vectors define or represent the most important properties observed in the face image. Highest Eigen values are chosen.

There are two advantages of this step. First, the size of the data is reduced from the entire image to only a few selected important features. Second, the selection of features gives more structured information than just basic pixel values of the images. Thus the feature vectors can be considered as the minimal set that is adequate to represent the face image. The training could be done on face images showing a selected class of emotion or the entire set of emotions. If the training is done for selected class of emotions then the model is build for each class of emotion. Hence in this case three basic emotions mentioned earlier a separate will be build for each of three emotions. The input images for a particular model will be only the images that show the corresponding emotion. If the training is desired for building a single model for the entire set of emotions then the entire set of face images is used for the training space. In our case after the training the radial basis
functional network will get the separate values for each emotion.

E. Testing Phase

This phase can be performed to measure the classification rate. The inputs to this phase are the models that were built during training phase and the test images for which the emotions are to be recognized. Here again only the face region is used as rest of the image do not contribute information about the emotion. In a typical real time scenario the input image would be detected face image from an earlier face detection phase.

The first step here again would be a feature extraction phase where the key features from the face image are extracted. The extraction method must be same as the one used in the training phase. The output of this step is the feature vector of the face image that would then be subjected to a testing step. In the testing step the feature vector is tested against the models built during the training phase. The output of the testing step is a score that indicated the emotion that is detected by the model. This score is usually in the form of distance or probability and it defines which model was best suited for the feature vector extracted in the previous step.

In the testing step there are two possible ways that can be employed. The first possibility is in the case when one model was built per class of emotion. Here the feature vector is tested against all the models and their scores then define which model was the most suited one. The second possibility is the case when only one model was built for the entire set and a single score defines the possible emotion detected. During testing a simple image the approach can correctly classify it as the correct emotion expressed. In another case the approach can also wrongly classifies a sample image as the correct emotion expressed. These cases constitute false positives. It could also be the case that the approach wrongly classifies a sample images as incorrect emotion expressed. These cases constitute false negatives.

V. EXPERIMENTAL RESULTS

The key objective of Facial Expression Recognition system is to be able to achieve Expression recognition irrespective context, culture and gender. In the proposed method, facial expressions of the human face are identified from the input image using Eigen spaces method. To illustrate the feasibility of using Eigen space for facial expression recognition, the modified PCA reconstruction method is used. If the input image is similar to some expression training set, the reconstructed image will have less distortion than the image reconstructed from other eigenvectors of training expressions. Based on this idea, we divided the training set into six classes according to universal expressions and computed the Eigen spaces of each class. For a test face image, we first project it onto the Eigen space of each class independently and then derive reconstructed image from each Eigen space. By measuring the similarity (mean square error) between input image and the reconstructed image of each class, we can identify the class of input image whose reconstructed image is most similar to the input one.

Experiment is carried out for 7 basic expressions angry, happy, sad, disgust, fear, surprise and neutral. The experiment was conducted with JAFEE, cohn-kanade and with our own databases. Euclidean distances between the projected test image and the projection of all centered training images were calculated. Test image is supposed to have minimum distance with its corresponding image in the training database.

The following table shows results with each individual expressions and its associated MED (minimum Euclidian distance) and RI (recognized index) values. MED is close to zero when the input image best matches with the test image and is maximum when there is no match RI indicates the image number for which the expression is identified.

Table 1 shows the Expression Recognition values for JAFEE database. In this table we have used 21 images of different facial expression with different variations of illuminations. There are totally 7 expressions and each expression is recorded 3 times. Similarly Table 2 shows the expression recognition values for Cohn-Kanade database. In this we have used 7 images of 7 expressions. Similarly Table 3 shows the expression recognition values for our own database which has the images of different expression from my own images. In the table1,2,3 MED=0.0000 indicates the best match of the expression and RI indicates the image number for which the best match is there. This MED value increases as the best match corresponding to images decreases.

<table>
<thead>
<tr>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happy</th>
<th>Neutral</th>
<th>Sad</th>
<th>Surprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>MED</td>
<td>RI</td>
<td>MED</td>
<td>RI</td>
<td>MED</td>
<td>RI</td>
<td>MED</td>
</tr>
<tr>
<td>0.00</td>
<td>1</td>
<td>0.0</td>
<td>4</td>
<td>0.0</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>0.00</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.33</td>
<td>5</td>
<td>0.0</td>
<td>3</td>
<td>0.0</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>25</td>
<td>02</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.37</td>
<td>1</td>
<td>0.1</td>
<td>1</td>
<td>0.0</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>41</td>
<td>9</td>
<td>39</td>
<td>50</td>
<td>1</td>
<td>03</td>
<td>0.0</td>
</tr>
<tr>
<td>1.98</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
<td>0.0</td>
<td>1</td>
<td>0.0</td>
</tr>
<tr>
<td>27</td>
<td>40</td>
<td>6</td>
<td>06</td>
<td>7</td>
<td>07</td>
<td>0.0</td>
</tr>
</tbody>
</table>

TABLE I
THE TABLE SHOWING MINIMUM EUCLIDIAN DISTANCES AND CORRESPONDING RECOGNIZED INDEX VALUES OF DIFFERENT EXPRESSIONS FOR JAFEE DATABASE
<table>
<thead>
<tr>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happy</th>
<th>Neutral</th>
<th>Sad</th>
<th>Surpris e</th>
</tr>
</thead>
<tbody>
<tr>
<td>M ED</td>
<td>R M 1</td>
<td>M ED</td>
<td>R M 1</td>
<td>M ED</td>
<td>R M 1</td>
<td>M ED</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.19</td>
<td>0.05</td>
<td>1.26</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.04</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**TABLE II**

The table showing minimum Euclidean distances and corresponding recognized index values of different expressions for own database.

<table>
<thead>
<tr>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happy</th>
<th>Neutral</th>
<th>Sad</th>
<th>Surpris e</th>
</tr>
</thead>
<tbody>
<tr>
<td>M ED</td>
<td>M ED 1</td>
<td>M ED</td>
<td>M ED 1</td>
<td>M ED</td>
<td>M ED 1</td>
<td>M ED</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
<td>1.21</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**TABLE III**

The table showing minimum Euclidian distances and corresponding recognized index values of different expressions for own database.

<table>
<thead>
<tr>
<th>Anger</th>
<th>Disgust</th>
<th>Fear</th>
<th>Happy</th>
<th>Neutral</th>
<th>Sad</th>
<th>Surpris e</th>
</tr>
</thead>
<tbody>
<tr>
<td>M ED</td>
<td>R M 1</td>
<td>M ED</td>
<td>R M 1</td>
<td>M ED</td>
<td>R M 1</td>
<td>M ED</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>1.19</td>
<td>0.05</td>
<td>1.26</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.04</td>
<td>0.10</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
VI. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an approach to expression recognition in the static images. This emotion analysis system implemented using FLD, PCA for feature selection and RBF network for classification. This paper is designed to recognize emotional expression in human faces using the average values calculated from the training samples. We evaluated that the system was able to identify the images and evaluate the expressions accurately from the images. In future we may include other expression also. We will try to proceed with Video images also.

ACKNOWLEDGMENT

Many people have contributed to the success of this paper. Although a single sentence hardly suffices, we would like to thank Almighty God for blessing us with His grace.

I express my sincere thanks to Prof. Dr. Vijaya C., Head of the Department of E&CE, SDM College of Engineering and Technology, Dharwad and to Prof. Sumangala Bhavikatti, Department of E&CE, SDM College of Engineering and Technology, Dharwad and to Prof. S.B.Kulkarni, Department of CSE, SDM College of Engineering, Dharwad for providing their valuable assistance to carry out this paper.

I take this opportunity to extend my gratitude to our beloved Principal Prof. Dr. S. Mohan Kumar for allowing us to carry out this paper. I express immense pleasure and thankfulness to all the teaching and non-teaching staff of the Department of Electronics and Communication Engineering, SDMCET for their cooperation and support.

REFERENCES


Nagaraj Bhat, Department of Electronics & Communication Engineering, SDM College of Engineering and Technology, Dharwad, Karnataka, India. Mobile No 07353699907.Nagaraj bhat received BE in Electronics and Communication Engineering from SKYMCAET,Laxmeshwar in the year 2009.and he is currently pursuing his M.Tech in Digital Electronics from SDMCET,Dharwad.His area of interest includes Image Processing, Data Compression.

Sumangala N Bhavikatti, Department of Electronics and Communication Engineering, SDM College of Engineering and Technology, Dharwad, Karnataka, India. Prof.Sumangala Bhavikatti received her BE in Electronics and Communication from VTU Belgaum in the year 2008 and received M.Tech in Digital Electronics from BVBCET in the year 2008.She is currently working as a Assistant Professor in SDMCET Dharwad in the Department of Electronics and Communication Engineering. Her area of interest is Image Processing.