Normalized Feature descriptor for Face recognition using RBF-Neural network

V. Vinay Kumar
Anurag Group of Instititution, Ghateskar, RR-Dist, Telangana, India,
.vemulavinay383@rediffmail.com

Abstract: *Face recognition is a major application in biometric based security system. The accuracuy of such learnig system are however dependednt on the descritpive feature and the classification system used. It is required to have a faster and accurate retreival so as to acheive hgihest rate of retreivation, improvizing robustness of a face recogntion system. The facial feature hence paly an important role in retreival accuarcy. In this paper a nromilzed feature descritpor is defiend to overcome the effect of surrounding environemnt on face recogntion. Towards the retreival accruacy these normized features are trainied with RBF based neural netork to acheive faster and accurate retrival. This approach improvizes the retreival accuracy hence increasing the robutness of the face recogntion system.*

Keyword: Face recognition, normalized feature, RBF, Neural Network.

I. INTRODUCTION

The most remarkable abilities of human vision is that of face recognition. It develops over several years of childhood, is important for several aspects of our social life, and together with related abilities, such as estimating the expression of people with which we interact, has played an important role in the course of evolution. The problem of face recognition was considered in the early stages of computer vision and is now undergoing a revival after nearly 20 years. Different specific techniques were proposed or reproposed recently This interest is motivated by wide applications ranging from static matching of controlled format photographs such as passports, credit cards, driving licenses, access control systems, model-based video coding, criminal identification and authentication in secure system like computer or bank teller machines and mug shots to real-time matching of surveillance video images presenting different constraints in terms of processing requirements. Although many face recognition by human beings and machines are developed, it is still difficult to design an automatic system for the task because in real world, illumination, complex background, visual angle and facial expression for face images are highly variable. Several methods have been proposed for face detection, including graph matching, neural networks, and also geometric feature based. Although researchers in psychology, neural sciences and engineering, image processing and computer vision have investigated a number of issues related to face recognition by human beings and machines, it is still difficult to design an automatic system for this task, especially when real-time identification is required. Face images are highly variable and Sources of variability include individual appearance, three-three dimensional (3-D) facial expression, facial hair, makeup, and so on and these factors change from time to time. Furthermore, the lighting, background, scale, and parameters of the acquisition are all variables in facial images acquired under real-world scenarios. The variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to changes in the face identity. This makes face recognition a great challenging problem. When designing a complex system, it is important to begin with strong foundations and reliable modules before optimizing the design to account for variations. Provided a perfectly aligned standardized database is available, the face recognition module is the most reliable stage in the system. Biggest challenge in face recognition still lies in the normalization and preprocessing of the face images so that they are suitable as input into the recognition module. Hence, the face recognition module was designed and implemented first.

II. FACE RECOGNITION SYSTEM

When designing a complex system, it is important to begin with strong foundations and reliable modules before optimizing the design to account for variations. Provided a perfectly aligned standardized database is available, the face recognition module is the most reliable stage in the system. Biggest challenge in face recognition still lies in the normalization and preprocessing of the face images so that they are suitable as input into the recognition module. Hence, the face recognition module was designed and implemented first. Extensive researches have already been conducted and completed in face recognition by many research groups, producing many papers and available literature that have addressed the issue of recognizing faces based on a preprocessed face database. Similar to other research groups, the task of implementing a face recognition module is therefore accomplished first by considering a set of predefined face inputs rather than using variable images. It is important to begin with as little variable parameters as possible, and a pre-processed face database omits any possible uncertainty from the detection and normalization modules. Given a perfect set of faces such that the scale, rotation, background and illuminance is controlled, the recognition module can be designed to work with the optimal ideal inputs, since it is crucial that the performance of this foundation module be as optimized as possible. Its ability to recognize an ideal database will determine the best possible performance attainable by the overall complete system. Any subsequent development and implementation of the face detection and normalization module will therefore be aimed at providing this ideal set of database. The eigenface approach of describing the features of a face in which the key idea is to calculate the best coordinate system for image compression, in which each coordinate is actually an image that is called an eigenpicture. However, the eigenfaceparadigm, which uses principal component analysis (PCA), yields projection directions that maximize the total scatter across all classes, i.e., across all face images. In choosing the projection, which maximizes the total scatter, the PCA retains unwanted variations caused by lighting, facial expression, and other factors. Accordingly, the features produced are not necessarily good for discrimination among classes. In the face features are acquired by using the fisher face or discriminant Eigen feature paradigm. This paradigm aims at overcoming the drawback of the Eigen face paradigm by integrating Fisher's linear discriminant (FLD) criteria, while retaining the idea of the eigenface paradigm in projecting faces from a high-dimension image space to a significantly lower-dimensional feature space. Instead of using statistical theory, neural-networks-based feature extraction has been reported recently. The goal of face processing using neural networks is to develop a compact internal representation of faces, which is equivalent to feature extraction. Therefore, the number of hidden neurons is less than that in either input or output layers, which results in the network is used to design a classifier. Although this paradigm has been successfully applied to solve various problems in pattern classification, it has difficulty in encoding inputs in a smaller dimension that retains most of the important information. Then, the hidden units of the neural network can serve as the input layer of another neural network to classify face images. In many pattern recognition systems, the methodology frequently used is the statistical approach, whereby decision theory derived from statistics of input patterns expressing structural information unless an appropriate choice of features is made possible. Furthermore, this approach requires much heuristic information to design a classifier. Neural-networks-based paradigms, as new means of implementing various classifiers based on statistical and structural approach, have been proven to possess many advantages for classification because of their learning ability and good generalization. Generally speaking, Multilayered networks (MLNs), usually coupled with the back propagation (BP) algorithm, are most widely used in face recognition. Yet, two major criticisms are commonly raised against the BP algorithm:

1) It is computationally intensive because of its slow convergence speed and

2) There is no guarantee at all that the absolute minima can be achieved.

On the other hand, RBF neural networks have recently attracted extensive interests in the community of neural networks for a wide range of applications. The invariance properties of moments of images have received considerable attention in recent years. The term invariant denotes an image feature remains unchanged if that image undergoes one or a combination of the changes such as: change of size (scale), change of position (translation), change of orientation (rotation), and reflection. Above properties of moments, occurred that moments have been proposed as pattern sensitive features in classification and recognition applications.

III. PCA FACE REOGNITION

Until G. Bors and M. Gabbouj [4] applied the Karhunen-Loeve Transform to faces, face recognition systems utilized either feature-based techniques, template matching or neural networks to perform the recognition. The groundbreaking work of Kirby and Sirovich not only resulted in a technique that efficiently represents pictures of faces using Principal Component Analysis (PCA), but also laid the foundation for the development of the "eigenface" technique of Turk and Pentland [1], which has now become a de facto standard and a common performance benchmark in face recognition. Starting with a collection of original face images, PCA aims to determine a set of orthogonal vectors that optimally represent the distribution of the data. Any face images can then be theoretically reconstructed by projections onto the new coordinate system. In search of a technique that extracts the most relevant information in a face image to form the basis vectors, Turk and Pentland proposed the eigenface approach, which effectively captures the variations within an ensemble of face images. Mathematically, the eigenface approach uses PCA to calculate the principal components and vectors that best account for the distribution of a set of faces within the entire image space. Considering an image as being a point in a very high dimensional space, these principal components are essentially the eigenvectors of the covariance matrix of this set of face images, which Turk and Pentland termed the eigenface. Each individual face can then be represented exactly by a linear combination of eigenfaces, or approximately, by a subset of "best" eigenfaces – those that account for the most variance within the face database characterized by its eigenvalues,

Consider an N-by-N face image I(x, y) as a vector of dimension N^2, so that the image can be thought of as a point in N^2-dimensional space. A database of M images can therefore map to a collection of points in this high dimensional "face space" as $\Gamma_1,\Gamma_2,\Gamma_3,\ldots\Gamma_M$. With the average face of the image set defined as

$$\Psi=1/M\sum^{M}_{n-1}\Gamma_n \quad (1)$$

each face can be mean normalized and be represented as deviations from the average face by $\Phi_t=\Gamma_t-\Psi$. The covariance matrix, defined as the expected value of $\Phi\Phi^T$ can be calculated by the equation;

$$C=1/M\sum^{M}_{n-1}\Phi_n\Phi^T \quad (2)$$

It is reasonable to assume that each face image is independent, that is, each image is taken differently. If we further assume that all the input images into the face recognition module have been perfectly normalized, we can conclude that the variances of each Φ_t, lie correspondingly, thus resulting in a set of time-aligned images. The significance of these assumptions is because together with the consideration that covariance is a zero-mean process, we can conclude that each Φ_t is independent and that cross multiplication between different Φ_t is possible due to their time-aligned characteristics, with the multiplication resulting in an expected value of zero. The above conclusion allows us to express the covariance matrix in an alternative form. Letting matrix $A = [\Phi_1,\Phi_2,\ldots\Phi_M]$, based on our conclusions, Eq. (2) can be rewritten as

$$C= 1/M\,(AA^T) \quad (3)$$

Since the factor 1/M will only affect the scaling of the output during eigenvector analysis, we can omit this scaling factor in our calculation resulting in

$$C=(AA^T) \quad (4)$$

Given the covariance matrix C, we can now proceed with determining the eigenvectors u and eigenvalues λ of C in order to obtain the optimal set of principal components, a set of eigenfaces that characterize the variations between face images. Consider an eigenvector $u_i$of C satisfying the equation

$$Cu_i=\lambda_i u_i \quad (5)$$

$$u_i^TCu_i=\lambda_i u_i^T u_i \quad (6)$$

The eigenvectors are orthogonal and normalised hence

$$u_i^Tu_j=\{1 \; i=j \quad i\neq j \quad (7)$$

Combining Eq. (2) and (7), Eq. (6) thus become

$$\lambda_i=1/M\sum^{M}_{n-1}\mathrm{var}\,(u_i\Gamma_n^T) \quad (8)$$

Eq. (8) shows that the eigenvalue corresponding to the *i*th eigenvector represents the variance of the representative face image. By selecting the eigenvectors with the largest corresponding eigenvalues as the basis vector, the set of dominant vectors that express the greatest variance are being selected. Recall however, that an N-by-N face image treated as a vector of dimension N^2 is under consideration. Therefore, if we use the approximated equation derived in Eq.(4), the resultant covariance matrix C will be of dimensions N^2 by N^2. A typical image of size 256 by 256 would consequently yield a vector of dimension 65,536, which makes the task of determining N^2 eigenvectors and eigenvalues intractable and computationally unfeasible.

IV. NORMALIZED FACIAL FEATURE DESCRIPTOR

An alternative to computing N^2 eigenvectors was proposed by Turk and Pentland [1]. Although the algorithm and methodology have been widely accepted and applied since its introduction, we believe that a complete prove of the mathematics behind the more efficient eigenface algorithm has not been seen and published in any literature. Similar to the calculations of the eigenvectors, u_i, of the covariance matrix, $C = AA^T$in Eq. (5), consider the eigenvectors, v_i, of A^TA such that,

$$A^TAv_i=\mu_i v_i \quad (9)$$

Pre-multiplying both sides by A and using Eq. (4), we obtain,

$$C\,A\,v_i= \mu_i\,A\,v_i \quad (10)$$

Comparing to the standard eigenvectors definition in Eq. (5), Eq. (10) implies that the product, A v_i, are the eigenvectors of C, with $\mu_i$being its corresponding eigenvalue. Thus rather than calculating the N2 eigenvectors of AA^T, we can instead compute the eigenvectors of A^TA, and multiply the results with A in order to obtain the eigenvectors of the covariance matrix, $C = AA^T$. Recalling that $A = [\Phi_1,\Phi_2,\ldots\Phi_M]$, the matrix multiplication of A^TA results in an M-by-M matrix. Since M is the number of faces in the database, the eigenvectors analysis is reduced from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practise, the training set is relatively small ($M \ll N^2$), making the computations mathematically manageable. The simplified method calculates only M eigenvectors while previously it was proven that there are mathematically N^2 possible eigenvectors. As demonstrated in Eq. (8), only the eigenvectors with the largest corresponding eigenvalues from the N^2 set are selected as the principal components. Thus, the eigenvectors calculated by the alternative algorithm will only be valid, if the resulting eigenvectors correspond to the dominant eigenvectors selected from the N^2 set. Consider the singular value decomposition (SVD) of matrix A, which would result in a diagonal matrix S of the same dimension as A with nonnegative diagonal elements in decreasing order, and unitary matrices U and V such that $A = U*S*V^T$ and $A^T= V*S*U^T$. The covariance matrix, $C = AA^T$, would therefore be equal to

$$AA^T=USV^TVSU^T \qquad (11)$$

We can split this into a combination of rank one matrices such that,

$$AA^T=\sum_{n=1}^{N2} S_n^2u_nu_n^T \qquad (12)$$

If the total N^2 eigenvectors of the covariance matrix were found, it would follow from Eq. (5) that a spectral decomposition (decomposing a matrix into orthogonal sub-basis) of the covariance matrix would give, $C=(AA^T)=\sum_{n=1}^{N2}\lambda_n e_n e_n^T$ (13)

if λ_n represents the eigenvalues and en the basis vectors. As expected, Eq. (12) and (13) shows that the s^2 values are the eigenvalues, while the u_n values correspond to the eigenvectors of C. By convention, it is acceptable to assume that an SVD analysis decomposes the basis vectors in descending order. With each eigenvector corresponding to a specified eigenvalue, it follows that the eigenvectors are also ordered according to its dominance with u_1 being the most dominant eigenvector since λ_1 is the largest eigenvalue. This conclusion is significant when considering the singular value decomposition of A^TA, from Eq. (11),

$$A^TA=VSU^TUSV^T=VS^2V^T=\sum^{M}_{n-1}S_n^2v_nv_n^T \qquad (14)$$

$$=S_1^2v_1v_1^T+S_2^2v_2v_2^T+S_3^2v_3v_3^T \qquad (15)$$

From Eq. (14) that the SVD expansion of A^TA results in only M terms.This is expected since the multiplication A^TA gives an M-by-M matrix. Using the same assumption derived in Eq. (13) and comparing the outcome of Eq.(15) to Eq. (12), we observe that the ordered s^2 values are the same in both equations. This gives strong evidence that each derived eigenvector v_n corresponds to a respective eigenvector u_n. Since we have concluded that the eigenvectors u_n are ordered according to their dominance, we have demonstrated that the M derived eigenvectors v_n are the first M eigenvectors of u_n, such that the set of v_1 to v_m forms our desired dominant basis vectors. This conclusion flows not only mathematically but also logically if we consider the number of meaningful data points. If the number of data points in face space is less than the dimension of the space itself, which in our case is true since $M \ll N^2$, it follows logically that there will only be M – 1, rather than N^2, meaningful eigenvectors. The remaining eigenvectors will therefore have associated eigenvalues of zero. This coincides with the outcome of the SVD analysis. If the first M eigenvectors of Eq. (12) are essentially the vectors derived in Eq. (15), the remaining eigenvalues corresponding to the non-dominant eigenvectors of Eq. (15) must therefore be zero in order for the equations to be mathematically correct. Based on the work of Cendrillon [11], Turk and Pentland [1], a face recognition module based on the eigenface approach was implemented. There are two separate phases included in the recognition module depending on the task to be performed,

a) The training stage
b) The testing stage. The training stage takes in a database of normalized faces and returns the weights of each face after projecting each onto the selected dominant basis vectors. After a set of weights have been determined and the database initialized, the testing stage can accept a new face image, subject it to eigenface decomposition, and compare the resultant weights with the closest matched weights in the database to identify the identity of the input. Without discussing the mathematics of the algorithm, the eigenface training stage can be broken down into the following steps:

1. Collect and construct a matrix T, of M fixed sized face images of known individuals. It is assumed that the input images into the recognition module have been pre-processed by the face detection and normalization modules. Each face should also be organized as a one-dimensional vector, and the corresponding pixels of each face should be aligned in the matrix.
2. Compute the average face of the database by summing the intensity values of each M

corresponding pixels and dividing it by the total number of faces, M.

3. Subtract the average face from T and obtain matrix A, a zero mean matrix where each element represents the variance of the pixel's intensity values.
4. Matrix C′ should be constructed by considering the optimized eigenface algorithm presented in Chapter 3, such that C′ is an M-by-M matrix computed by the product of AT and A.
5. The eigenvectors of C′ is calculated and sorted in descending order according to each eigenvector's associated eigenvalues.
6. Obtain the eigenfaces – the eigenvectors of the covariance matrix C – by multiplying the eigenvectors of C′ calculated in step 5 by matrix A.
7. Obtain the dominant basis vectors that describe and characterise the face database by normalising the eigenfaces in step 6, by dividing each by its vector norm.
8. Determine the weights of the M input faces of the database by projecting each face image into face space, transforming each face into its eigenface components. This can be accomplished by multiplying each individual zero mean face from A by the eigenfaces obtained in step 7.
9. Each face is now represented by a set of weights, which can uniquely describe the M input faces, and be used to reconstruct any face in the database given the eigenfaces components.

For the testing stage, it is necessary to save the computed average face, eigenfaces, known weights and the name of the faces. The nine steps described above will transform a database of face images into a set of projections into the constructed face space. If a large database is present for training, for example if M is large, such that a representative set of eigenfaces has been obtained, it is possible to use less than M eigenfaces to describe the database [1], since eigenfaces that have small corresponding eigenvalues tends to overfit and begin to describe peculiarities of individual faces. Under this circumstance, when a new face is presented for addition into the database, rather than recalculating all the eigenfaces, only the weights need to be determined by projecting the new face into the existing face space.Each eigenface is dedicated to describing a set of facial features that captures the greatest variances between different faces. Along with the average face, due to the perfect alignment of the database, linear combinations of these eigenfaces can reconstruct a large ensemble of faces. Thus after the above initialization stage is completed, any subsequent faces can also be projected into face space described by the set of eigenfaces, and be tested to see if the weights matches any of the M faces of the database, performing the recognition stage of the module. The following steps can summarize the recognition stage:

1. A test image T that is pre-processed by the same face detection and normalization modules as the face database is inputted into the recognition stage of the face recognition module as a one-dimensional vector.
2. The saved average face, eigenfaces, known weights and the name associated with each face from the initialization are loaded into memory.
3. The input image is translated to zero mean by subtracting the average face from it, resulting in vector N.
4. Since the face space is represented by the eigenfaces dominant vectors, the zero mean input image can also be projected onto the space by multiplying the eigenfaces with the normalized input image, N, in order to determine its weights.
5. The calculated weights can be compared with the set of known weights to find the minimum distance between the calculated weights and each face's set of weights. The minimum distance symbolizes the closest match between the input test image and the faces in the database.
6. If the minimum distance found is less than a certain sensitivity value, the input test image can be identified as the identity of the matched face. Whereas, if the minimum distance found is larger than the sensitivity value, the input test image can be claimed to be an unknown identity and can prompt the system to add this new face into the database repeating the training stage.

If the input into the face recognition module is normalized and adjusted for scale, rotation and lighting, the training and the testing stage of the module described above will function flawlessly. However, although the eigenface approach works theoretically, the criteria for perfectly aligned faces make a perfect face recognition system difficult to accomplish. As noted throughout the literature on this topic, the biggest challenge remains in designing normalization modules that can provide such ideal databases of faces for recognition.

V. RBF-NN MODELING

Neural networks are important for their ability to adapt. Neural nets represent entirely different models from those related to other symbolic systems. The difference occurs in the way the nets store and retrieve information .The information in a neural net is found to be distributed throughout the network and not localized. The nets are capable of making memory associations. They can handle a large amount of data, fast and efficiently. They are also fault tolerant, i.e. even if few neurons fail; it will not disable the entire system. The architecture of RBF network is a multiplayer feed forward network in which 'n' number of input neurons and 'm' number of output neurons with the hidden layer existing between input and output layer .The interconnection between the input layer and the hidden layer forms hypothetical connection and between hidden layer and output layer forms the waited connections. The training algorithm is used for updation of weigths in all interconnections.

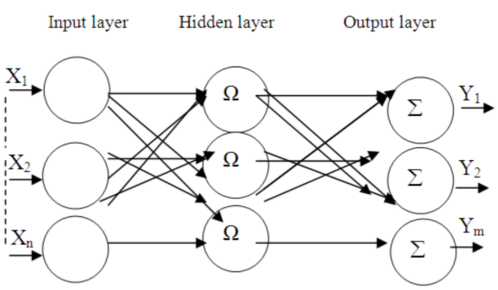


Figure 1: Radial Connected neural network

An RBF neural network, shown in Figure 1, can be considered as a mapping:

$\Re^T \Re^S \longrightarrow$

Let $P \in \Re^T$ be the input vector and $C_i \in \Re^T$ $(1 \leq i \leq u)$ be the prototype of the input vector. The output of each RBF unit is as follows:

$R_i(P)=Ri(\|P- C_i\|)$ $\quad i=1\ldots\ldots u$ (18)

Where($\|\text{-}\|$) indicates the Euclidean norm on the input space. Usually, the Gaussian function is preferred among all possible radial basis functions due to the fact that it is factorizable. Hence

$R_i(P)=\exp[-(\|P- C_i\|)^2 s \ \sigma_i^2]$ (19)

Where σ_i^2 is the width of the ith RBF unit. The jth output $y_i(P)$of an RBF neural network is

$y_i(P)=\sum^u_{i=1} R_i(P) X\ w(j,i)$ (20)

where $R_o=1$, w(j,i) is the weight or strength of the i th receptive field to the j th output and w(j,0) is the bias of the j th output. In order to reduce the network complexity, the bias is not considered in the following analysis. We can see from (19) and (20) that the outputs of an RBF neural classifier are characterized by a linear discriminant function. They generate linear decision boundaries (hyperplanes) in the output space. Consequently, the performance of an RBF neural classifier strongly depends on the separability of classes in the k-dimensional space generated by the nonlinear transformation carried out by the' u' RBF units.

The proposed methodology comprises the following parts:

1. The number of input variables is reduced through feature selection, i.e., a set of the most expressive features is first generated by the PCA and the FLD is then implemented to generate a set of the most discriminant features so that different classes of training data can be separated as far as possible and the same classes of patterns are compacted as close as possible;
2. A new clustering algorithm concerning category information of training samples is proposed so that homogeneous data could be clustered and a compact structure of an RBF neural classifier with limited mixed data could be achieved.
3. Two important criteria are proposed to estimate the initial widths of RBF units which control the generalization of RBF neural classifier.
4. A hybrid-learning algorithm is presented to train the RBF neural networks so that the dimension of the search space is significantly reduced in the gradient paradigm.

Actually, the PCA paradigm does not provide any information for class discrimination but dimension reduction. Accordingly, the FLD is applied to the projection of the set of training samples in the eigenface space $X = (X_1,X_2,\ldots X_n) \subset \Re^{rxn}$. The paradigm finds an optimal subspace for classification in which the ratio of the between-class scatter and the within-class scatter is maximized. Let the between class scatter matrix be defined as

$S_B=\sum^c_{i=1} n^i(\bar{X}^i- \bar{X}) (\bar{X}^i- \bar{X})^T$ (21)

and the within-class scatter matrix be defined as

$S_w=\sum^c_{i=1}\sum_{Xi\in n^i}(X_{k-} \bar{X}^i) (X_{k-} \bar{X}^i)^T$ (22)

Where $\bar{X}=(1/n)\sum^n_{j=1}X_j$ is the mean image of the ensemble, and $\bar{X}^i =(1/n^i)\sum^{ni}_{j=1}X_j^i$ is the mean image of the ith class and c is the number of classes. The optimal subspace $E_{optimal}$ by the FLD is $(e_1,e_2,e_3,\ldots e_{c-1})$ is the set of generalized eigenvectors of S_B and Sw corresponding to the c-1 largest generalized eigen values λ_i=1,2,3,….c-1 i.e.

$S_B E_i= \lambda_i S w E_i$ (23)

Thus the feature vectors P for any query face image Z in the most discriminant sense can be calculated as follows:

$P=E^T_{Optimal}.U^T.Z$ (24)

VI. EXPERIMENTAL RESULTS

To evaluate the proposed approach, a face recognition system is defined as shown below.

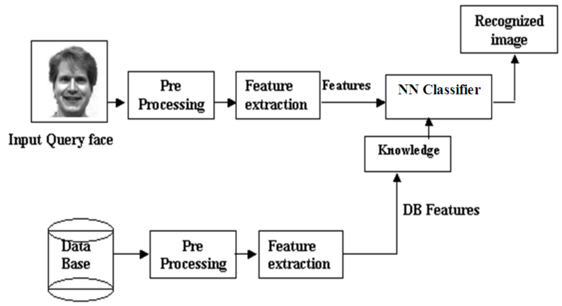


Figure 2: Block diagram of RBF-NN used for face Recognition

For the implementation of face retrieval a well known face database called YALE face database is used. YALE face database contains 165 gray level face images of 15 persons. There are 11 images per subject, and these 11 images are, respectively, under the following different facial expression or configuration: center-light, wearing glasses, happy, left-light, wearing no glasses, normal, right-light, sad, sleepy, surprised, and wink. In this implementation, all images are sized to a size of 128 x 128.

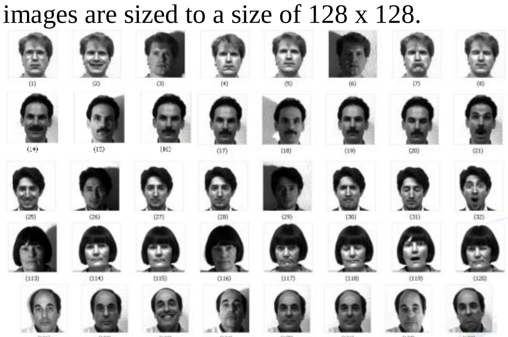


Figure 3: samples of Yale database used for training the face retrieval system.

For the evaluation of the proposed approach different samples were tested randomly selected out of the data base information. The recognition obtained are as presented;

For a Test with same samples present in database the retrieval is found to be 100%. This illustrates the processing algorithm accuracy for the developed system.

.

(a) (b)

Figure 4: Obtained recognition for test set from Database image using PCA
(a) Original Image, (b) Recognized image

(a) (b)

Figure 5: Obtained recognition for test set from Database image using Normalized FLD-RBF
(a) Original Image, (b) Recognized image

For the samples out of the dataset images the recognition is tested, the obtained observations are as illustrated below,

(a) (b)

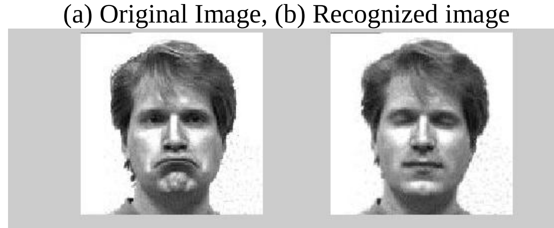
Figure 6: Obtained recognition for test set from Database image using PCA
(a) Original Image, (b) Recognized image

(a) (b)

Figure 7: Obtained recognition for test set from Database image using Normalized FLD-RBF
(a) Original Image, (b) Recognized image

(a) (b)

Figure 8: Obtained recognition for test set from Database image using PCA
(a) Original Image, (b) Recognized image

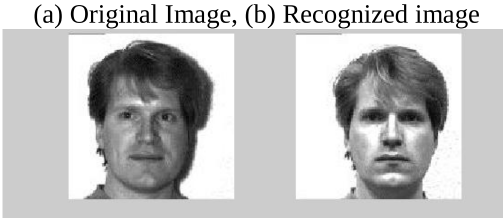


(a) (b)

Figure 9: Obtained recognition for test set from Database image using Normalized FLD-RBF
(a) Original Image, (b) Recognized image

(a) (b)

Figure 10: Obtained recognition for test set from Database image using PCA
(a) Original Image, (b) Recognized image



(a) (b)

Figure 11: Obtained recognition for test set from Database image using Normalized FLD-RBF
(a) Original Image, (b) Recognized image

The retrieval accuracy for the proposed normalized feature descriptor with FLD RBF NN model is observed to have higher retrieval accuracy in comparisons to the conventional PCA based face recognition.

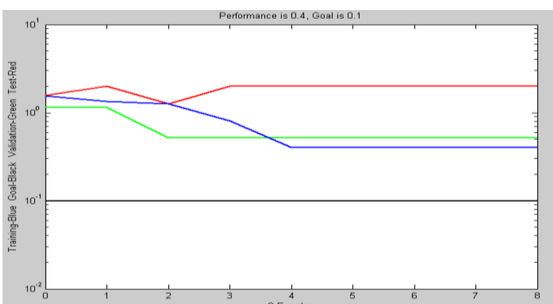


Figure 12: The learning, training and testing characteristic of the proposed NN-Model.

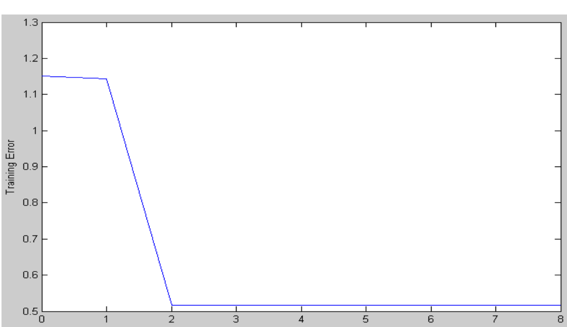


Figure 13: The Mean learning error for the proposed system

The performance for the NN model under variant parametric variation is observed and outlined in table I.

Images	No. of RBF units	β	η	Epochs	MSE	NOM
Q_1	40	0.5	0.1	300-500	25.58	2/5
Q_2	40	0.6	0.01	400-500	27.35	1/5
Q_3	40	0.7	0.2	200-500	32.54	3/5
Q_4	40	0.8	0.5	500	33.014	3/5

Table I. Classification performance for the developed NN-Model

*MSE – Mean squared error
*NOM – Number of misclassifications

The feature vectors used for the training is as outlined in table II.

0.003198	0.002752	0.001391	0	0	0	0	0	0	0	1
-0.00152	0.002202	0.001193	0	0	0	0	0	0	0	1
0.001698	-0.01135	-0.00588	0	0.00E+00	0	0	0	0	0	1
-0.02365	0.000429	0.020011	0	0.00E+00	0	0	0	0	1	0
-0.00649	0.000498	-0.00053	0	0	0	0	0	0	1	0
-0.00132	-2.27E-05	-7.14E-05	0	0	0	0	0	0	1	0
-0.02572	0.000762	0.00045	0	0	0	0	0	0	1	1
-0.00116	0.001425	-0.00523	0	0	0	0	0	0	1	1
0.005151	-0.0007	0.003498	0	0	0	0	0	0	1	1

Table II. Feature Vectors for training Neural Network

VII. CONCLUSION

The paper outlines a RBF based NN modeling for face recognition. The facial features having high variational components are normalized using the approach of modifiedEigen features. With the dimensional reduction using FLD approach the performance of the prosed system is furtherimproved. In the classification of the face sample it is observed to have higher retrieval in proposed system as comparison to the PCA based face recognition system.

VIII. REFERENCE

[1]. R. Chellappa, C. L.Wilson, and S. Sirohey, “Human and machine recognition of faces: A survey,” Proc. IEEE, vol. 83, pp. 705–740, 1995.

[2] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces versus fisherfaces: Recognition using class specific linear projection,” IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 711–720, 1997.

[3] R. Brunelli and T. Poggio, “Face recognition: Features versus templates,” IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 1042–1053, 1993.

[4] G. Bors and M. Gabbouj, “Minimal topology for a radial basis functions neural networks for pattern classification,” Digital Processing, vol. 4, pp. 173–188, 1994.

[5] M. T. Musavi,W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels, “On the training of radial

basis function classifiers," Neural Networks, vol. 5, pp. 595–603, 1992.
[6] R. Lotlikar and R. Kothari, "Fractional-step dimensionality reduction," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, pp. 623–627, 2000.
[7] J. L. Yuan and T. L. Fine, "Neural-Network design for small training sets of high dimension," IEEE Trans. Neural Networks, vol. 9, pp. 266–280, Jan. 1998.
[8] J. Moody and C. J. Darken, "Fast learning in network of locally-tuned processing units," Neural Comput., vol. 1, pp. 281–294, 1989.
[9] S.Wu and M. J. Er, "Dynamic fuzzy neural networks: A novel approach to function approximation," IEEE Trans. Syst, Man, Cybern, pt. B: Cybern, vol. 30, pp. 358–364, 2000.
[10] S. J. Raudys and A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners," IEEE Trans. Pattern Anal. Machine Intell., vol. 13, pp. 252–264, 1991.
[11] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, "Classifying facial actions," IEEE Trans. Pattern Anal. Machine Intell., vol. 21, pp. 974–989, 1999.