Face Recognition Using Various Classifiers: Artificial Neural Network, Linear Discriminant and Principal Component Analysis

Technical Report UTDCS-05-06

Department of Computer Science
The University of Texas at Dallas

January 2006

Pallabi Parveen and Bhavani Thuraisingham

Face Recognition using Various Classifiers: Artificial Neural network, Linear Discriminant and Principal Component Analysis

Pallabi Parveen and Bhavani Thuraisingham
Richardson, TX 75083-0688
Email: pxp013300@utdallas.edu
bhavani.thuraisingham@utdallas.edu

Abstract

Automatic identification of users enables intelligent servicing in an interconnected home environment. In this paper, we propose a near real-time effective face recognition system for consumer applications. Since the nature of application domain requires real time result and better accuracy, it poses a serious challenge. To address this challenge, we propose a feature reduction technique using Principle component analysis (PCA) to facilitate real time face recognition along with better accuracy. In addition, to improve face recognition classifier accuracy, we propose a hybrid model that combines various classification techniques namely, Artificial Neural Networks (ANN), and Linear Fisher Discriminant Analysis (LDA) to resolve recognition using linear average weighting. Such fusion overcomes the drawbacks of each classifier. We have investigated the effectiveness of our hybrid model by comparing our results with ANN, and LDA based on a benchmark dataset.

1. Introduction

With the advancement of digital technology, we are required to consider, what, is the future of computing in our daily activities and how it will affect our lives. We would like to develop a face recognition system that can be embedded in a home environment to facilitate intelligent services. By automatic identification of home users, personalized services can be offered. For example, a face recognition-based smart TV program can offer a set of programs that are customized to that recognized user. After recognizing face of a user, corresponding user profile will be identified and matched with TV programs and presented to the user.

Current state of the art considers many face processing techniques, however, face recognition in an unconstrained environment, such as a home, is a challenging task. This happens due to the large variability of illumination and background conditions. In addition, the face recognition for consumer applications requires processing efficiency and robustness/ accuracy simultaneously. Note that it is difficult to achieve both at the same time. If we want to get result in real time, we may sacrifice accuracy. In that case, false positive and false negative may increase. On the other hand, we can improve accuracy by focusing on non real time detection. In this paper, we would like to sketch a solution that do real time face recognition and facilitate as perceptual interface for home devices. It is important to note that we will use current state of the art for face detection [4, 5, 6]. In other words, our focus is not concerned with the research of face detection.

Some researchers may argue that rather than face recognition why we do not use other biometric systems such as fingerprint and iris recognition for consumer applications. Face recognition has some advantages over finger and iris recognition such as: In traditional finger print and iris identification, users will pass through a 'pause and declare' procedure for authentication purpose. This may not be suitable for consumer applications. On the other hand, face recognition does not have that type of shortcoming. Furthermore, face recognition can make use of a wide range of inexpensive consumer camera such as DV cameras, and embedded cameras in mobile device.

In this paper, we use two classification techniques, namely, Artificial Neural Networks (ANN) [9] and Linear Discriminant Analysis (LDA) to increase the generalization accuracy in face recognition. ANN is a powerful technique, which can predict not only for the seen data, but also for the unseen data. It works well for both linear and non linearly separable datasets. However, when dealing with too many classes, ANN predictive power may decrease. The LDA is a powerful technique for predicting seen data; however, it cannot predict unseen data. Furthermore, LDA may not work for non linearly separable dataset. In particular when the dataset is high dimensional, its performance degrades. Hence, and Principal Component Analysis (PCA) is used for the reduction of features and select important features. Then these new features will be used for the training and testing of LDA classifier. By fusing all classifiers using average linear weighted method, namely ANN, LDA and PCA+LDA, we overcome major drawbacks in each technique and improve the predictive accuracy.

The organization of the paper is as follows. In Section 2, we discuss various steps of face recognition system. In Section 2.4.1, we present background of classifications for face recognition and how we can fuse various classifiers' outcome based on linear average weighted theory. In Section 3, we present what dataset we used and result of our work using a standard benchmark data set. In Section 4, we summarize the paper and outline some future research.

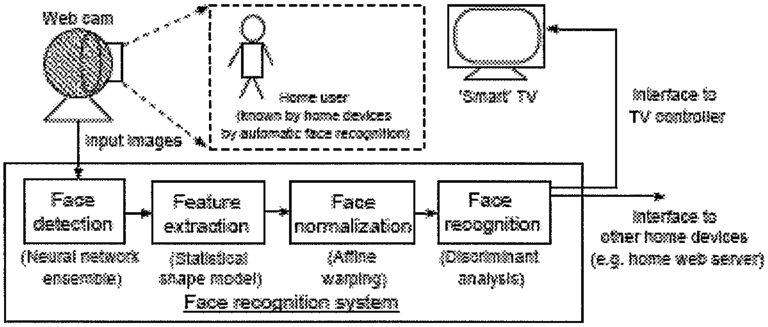


Figure1. Face Recognition Application in a Home Environment (taken from[17])

2. Face Recognition System

When embedded in a home environment, our face recognition system facilitates a 'smart' home, in which devices 'know' the identity of the user and adapt services accordingly (Fig. 1). The face recognition consists of four processing steps:

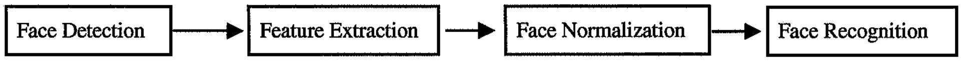


Figure2. Flow of Our Approach

I) Face detection from live video,
II) Facial feature extraction
III) Face normalization,
IV) Finally, Face recognition.

2.1. Face Detection using Cascaded Detectors

Here, our goal is to detect face area in an image. The recent approach of Zuo et al. [4, 17] uses successive detectors with incremental complexity and detection capability. This cascaded detector works in the following way. The detectors are cascaded in such a way that each detector progressively restricts the possible face candidates into fewer areas. The detector consists of a skin-color-based detector, a face structure detector by feature verification and a neural network- based detector. The initial pruning of large areas of non-face regions significantly reduces the number of candidate windows for the neural-network-based detection, thus significantly reducing the overall computation cost.

2.2. Model-based Facial Feature Extraction

Since varying face scale, position and poses can strongly interfere with the performance of the face recognition algorithm, we need an accurate extraction of salient facial features for normalization purposes. We would like to adopt the following technique: Zuo et al.[6, 17] adopt a two-step coarse-to-fine feature extraction technique. First, we use edge template matching for fast determination of approximate facial feature positions within the detected face region. Second, a deformable shape model is fit to the input image in an iterative two step procedure.

I. Local attribute matching: each local feature point is searched along a trace composed of 8 radial lines for a best fit, based on the Haar texture analysis.

II. Global shape regulation: the global shape is updated and regulated based on PCA shape statistics (similar to ASM). Note that in contrast with ASM [7], we use 2-D Haar wavelet based texture attributes to guide the local deformation process, which facilitates increased robustness and fast processing.

2.3. Face Normalization

Assuming all the features are coplanar and the face remains essentially rigid, Zuo et al. [4, 6, 17] use an affine transformation to warp an input face with varying scale, position and pose to a standard frame. By establishing correspondences between the extracted features of the input face and those of a standard front-view average face, we can quickly estimate the six affine parameters by least-squares fitting. Based on these parameters, the affine warping is performed to obtain a normalized face.

2.4. Face Recognition

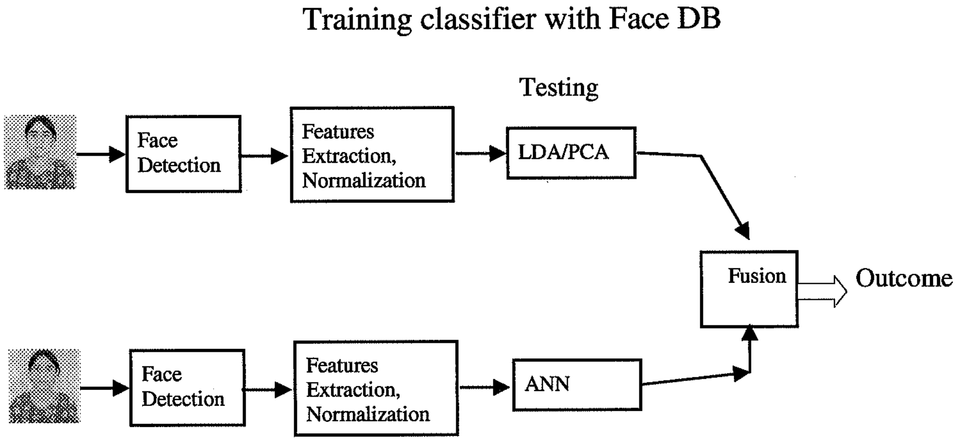

Training classifier with Face DB
Testing
Face Detection
Features Extraction, Normalization
LDA/PCA
Fusion
Outcome
Face Detection
Features Extraction, Normalization
ANN

Figure 3. Steps in Near Real Time Face Recognition System

In this Section, first, we present back ground for various classification mechanisms (i.e., ANN LDA) along with their advantages and disadvantages. Next we present feature reduction technique for LDA classifier. Finally, we present a new hybrid approach combining LDA, ANN models in face recognition using linear average combination.

2.4.1 Classification:

2.4.1.1 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) is a very well-know, powerful, and robust classification technique that has been used to approximate real-valued, discrete-valued, and vector-valued functions from examples [9]. ANN has been used in many areas such as interpreting visual scenes, speech recognition, learning robot control strategies, etc.

We employ an artificial neural network of two layers that uses the back propagation algorithm for learning. The back propagation algorithm attempts to minimize the squared error function.

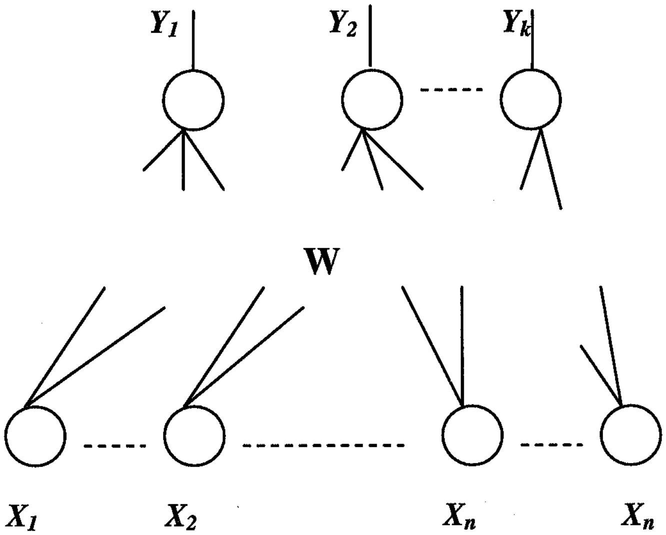


Figure 4. Design of Artificial Neural Network

A typical training example is a Face recognition, $<[x_1,\ldots,x_{n-1},x_n]^T, y_k>$. Where $[x_1,\ldots,x_{n-1},x_n]^T$ is the input (feature vector of a person's image) to the ANN and y_k is the output (face of a person). The output of the network is a boolean value, not a probability. We approximate the probability of the output by fitting a sigmoid function after ANN output. The approximated probabilistic output, $o^{'} = f(o(I))$, where I is an input session, $o(I)$ corresponds to y_k and $o^{'} = p(y_k \mid x_1,\ldots,x_n)$. We choose the sigmoid function, Equation 1, as a transfer function so that the ANN can handle non-linearly separable data set.

$$o = \sigma(W.X) = \sigma(y_k) = \frac{1}{1+e^{-2\times\beta\times y_k}}, \beta > 0 \qquad (1)$$

Where $\boldsymbol{X}$ is the input to the network, $\boldsymbol{O}$ is the output of the network, $\boldsymbol{W}$ is the matrix of weights, and σ is the sigmoid function.

We implement the back propagation algorithm for training the weights. The back propagation algorithm employs gradient descent to attempt to minimize the squared error between the network output values and the target values of these outputs. The sum of the error over all of the network output units is defined in Equation 2.

$$E(w) = \frac{1}{2}\sum_{k\in D}\ \sum_{i\in outputs}(t_{ik}-o_{ik})^2 \qquad (2)$$

Where the *outputs* is the set of output units in the network, *D* is the training set, and t_{ik} and o_{ik} are the target and the output values associated with the i^{th} output unit and training example k. For a specific weight w_{ji} in the network, it is updated for each training example as follows:

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}} \qquad (3)$$

$$w_{ji} = w_{ji} + \Delta w_{ji} \qquad (4)$$

Where, η is the learning rate and w_{ji} is the weight associated with the i^{th} input to the network unit j. As we can see from Equation 3 the search direction Δw is computed using the gradient descent which guarantees a convergent toward a local minimum. In order to mitigate that, we add a momentum to the weight update rule such that the weight update direction $\Delta w_{ji}(n)$ depends partially on the update direction in the previous iteration, $\Delta w_{ji}(n-1)$. The new weight update direction $\Delta w_{ji}(n)$ is shown in Equation 5.

$$\Delta w_{ji}(n) = -\eta \frac{\partial E_d}{\partial w_{ji}} + \alpha \Delta w_{ji}(n-1) \qquad (5)$$

Where n is the current iterations, α is the momentum constant. Notice that in Equation 5, the step size is slightly larger than in Equation 3. This contributes to a smooth convergence of the search in regions where the gradient is unchanging. In our implementation we set the step size η dynamically based on the distribution of the classes in the dataset. First, we set the step size to large values when updating the training examples that belong to low distribution class and vice versa. This is because when the distribution of the classes in the dataset varies widely (for example, positive examples 10% and negative examples 90%), the network weights converges towards the examples from the class of larger distribution, which causes a slow convergence. Second, we adjust the learning rates slightly by applying the momentum constant, Equation 5, to speed up the convergence of the network.

2.4.1.2 Face Recognition using LDA

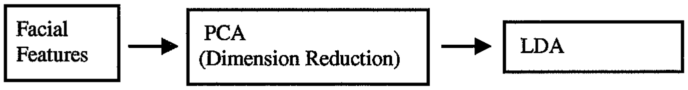
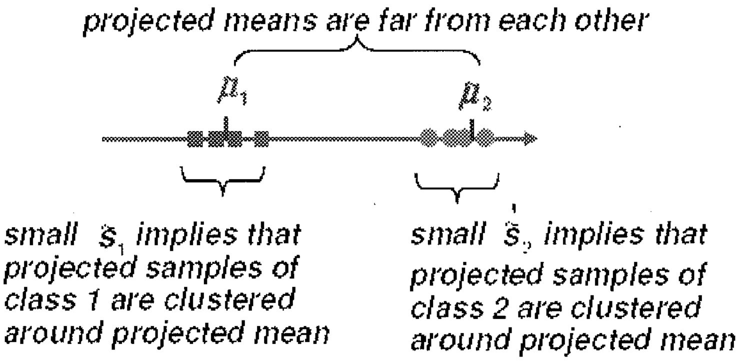


Figure 5. Variation of LDA with PCA

Fisher linear discriminant analysis [19, 20] tries to project input data on line in the direction v which maximizes $J(v)$, where, $\tilde{\mu}_1, \tilde{\mu}_2$ are means and $\tilde{s}_1, \tilde{s}_2$ are scatter values in projected space, S_B, S_W are between and within scatter value respectively in the original space.

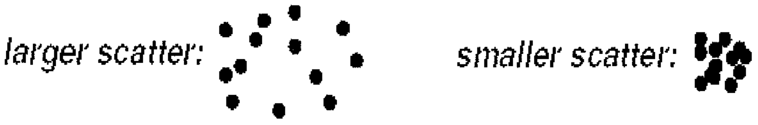
$$J(v) = \frac{(\tilde{\mu}_1 - \tilde{\mu}_2)^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \frac{v^t S_B v}{v^t S_W v}$$


projected means are far from each other
$\tilde{\mu}_1$
$\tilde{\mu}_2$
small $\tilde{s}_1$ implies that projected samples of class 1 are clustered around projected mean
small $\tilde{s}_2$ implies that projected samples of class 2 are clustered around projected mean

If z_1, z_2, z_3.... are samples and μ_z is mean of the input samples then scatter S

$$s = \sum_{i=1}^{n} (z_i - \mu_z)^2$$

Thus Scatter is just sample variance multiplied by n



So to maximizes $J(v)$

$$\frac{d}{dv} J(v) = 0$$

$$\Rightarrow S_B v - \frac{v^t S_B v (S_W v)}{v^t S_W v} = 0 \quad = \lambda$$

$$\Rightarrow \underbrace{S_B v = \lambda S_W v}_{\textit{generalized eigenvalue problem}}$$

Let
- n_i by the number of samples of class i
- and μ_i be the sample mean of class i
- μ be the total mean of all samples

$$\mu_i = \frac{1}{n_i} \sum_{x \in \textit{class } i} x \qquad \mu = \frac{1}{n} \sum_{x_i} x_i$$

within the class scatter matrix S_W is

$$S_W = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{x_k \in \textit{class } i} (x_k - \mu_i)(x_k - \mu_i)^t$$

between the class scatter matrix S_B is

$$S_B = \sum_{i=1}^{c} n_i (\mu_i - \mu)(\mu_i - \mu)^t$$

maximum rank is c -1

- At most ***c*-1** distinct solution eigenvalues
- Let v_1, v_2 ,..., v_{c-1} be the corresponding eigenvectors
- The optimal projection matrix ***V*** to a subspace of dimension ***k*** is given by the eigenvectors corresponding to the largest ***k*** eigenvalues
- Thus can project to a subspace of dimension at most ***c*-1**

By definition, an eigen vector of a transformation (S_B/S_W) represents a 1-D invariant subspace of the vector space in which the transformation is applied. For any C-class problem we would always have *C-1* non zero eigen values. After having the transformation metrics *V*, we transform the data sets using LDA transform.

$$Transformed\_set = transform\_spec^T \times data\_set^T$$

Here, *transform_spec* is *V*. Similarly the test vectors are transformed and are classified using the Euclidean distance of the test vectors from each projected or transformed class mean.

$$dist\_n = (transformed\_spec)^T \times \chi \ \text{-}\mu_{ntrans}$$

Here μ_{ntrans} is the mean of the transformed data set, *n* is the class index and χ is the test vector. Thus for n classes, n Euclidean distances are obtained for each test vector. The smallest Euclidean distance among the *n* distances classifies the test vector as belonging to class *n*.

2.4.1.3 Feature Reduction

Here, we deal with high dimensional data. For example, a person's face vector may be represented by 1024 dimensions which is high dimensional. We notice that LDA does not work well for high dimensional data. For this, we will investigate a techniques for the dimensionality reduction using PCA [18]. In addition, dimensionality reduction will save recognition time (i.e., real time).

For face recognition of a particular person, we consider a number of faces of that person. Data will be represented in a matrix form. Each row represents a vector of features of a particular person. To train classifier we need vector rather than matrix. Hence, we will obtain a feature vector for each matrix using the Principal component analysis (PCA) properties of the matrix. PCA projects data in the directions of maximum variance and optimally exposes the geometric structure of a matrix and this geometric structure can be exploited to obtain a set of feature vectors with reduced dimensions for a particular person. We will then classify the vectors for all persons.

Therefore, we will propose an efficient classification approach using first PCA and the classification, as shown in Figure 5. A feature vector is reduced from each matrix by PCA, and all the feature vectors are used as inputs to classifications. This training phase can be done offline (See Figure 3). Intuitively, if two matrices are similar, their feature vectors should be close to each other; otherwise their feature vectors would be different. Similarly, we generate testing datasets and these testing data can be classified that has already been trained offline with training datasets.

2.4.2 Fusion:

By fusing various classifiers, namely ANN and LDA model, we overcome major drawbacks in each technique and improve the predictive accuracy. For fusion, we use linear combination model.

$P_C = \lambda_1 P_{LDA} + \lambda_2 P_{(PCA+LDA)} + (1-\lambda_1 - \lambda_2) P_{ANN}$

Here Pc is the probability of a face belongs to a particular class. P_{LDA}, P_{ANN}, and $P_{(PCA+LDA)}$ are probabilities of this face given by LDA, ANN and PCA+LDA respectively. λ_1, λ_2 and $(1-\lambda_1 - \lambda_2)$ are weighting factors for LDA ,ANN and PCA+LDA.

3. Proposed Results

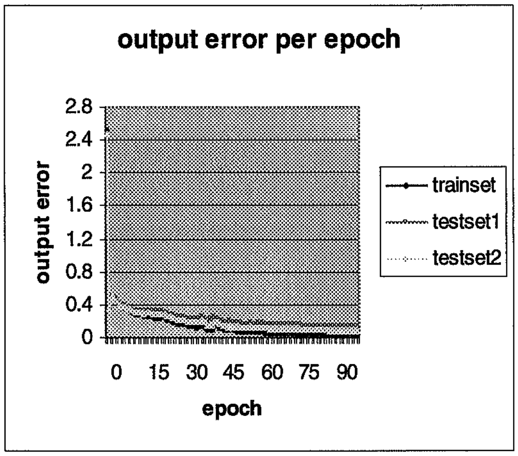
We use Carnegie Mellon University face recognition dataset (http://www.cs.cmu.edu/afs/cs.cmu.edu/user/avrim/www/ML94/face_homework.html).

It consists of 640 face images. 10 images of each of 16 students were taken with a variety of head positions and facial expressions. These images were then used to train and test classifier to recognize individual people, and to recognize different face poses. So, we have used a train set containing 160 images (16 different persons, each comprises of 10 images) and two test sets containing 40 and 41 images corresponding. The first test set have some images of persons, other than the 16 different persons that are included in the training data set. The second test set have different poses of those 16 different persons that are not used in the training set.

First testing set gives 75 percentage of accuracy, where the second test set gives 88 percentage of accuracy. Figure 6(a)-(b) and Table 1 show the total error and accuracy at each epoch for training and two testing data sets.

epoch	Train set			Test set1		Testset2	
	Total error	Accuracy	Output error	Accuracy	Output error	Accuracy	Output error
0	0	0	2.52967	0	2.43119	0	2.46391
10	26.5367	17.5	0.308014	12.1951	0.355595	14.2857	0.335306
20	21.054	42.5	0.243186	26.8293	0.316188	31.4286	0.28208
30	16.2648	70	0.152876	48.7805	0.248553	57.1429	0.204897
40	13.3732	75.625	0.0978488	58.5366	0.19373	68.5714	0.139836
50	13.2196	85.625	0.0892173	65.8537	0.179159	77.1429	0.12022
60	13.6633	91.875	0.0810445	73.1707	0.177783	85.7143	0.11931
70	8.48572	98.125	0.0382862	70.7317	0.159492	82.8571	0.096626
80	6.76837	100	0.0298756	75.6098	0.149207	88.5714	0.083415
90	6.06455	100	0.0274314	75.6098	0.147921	88.5714	0.080776
100	5.60823	100	0.0259482	75.6098	0.143939	88.5714	0.075755

Table 1. Output of neural network



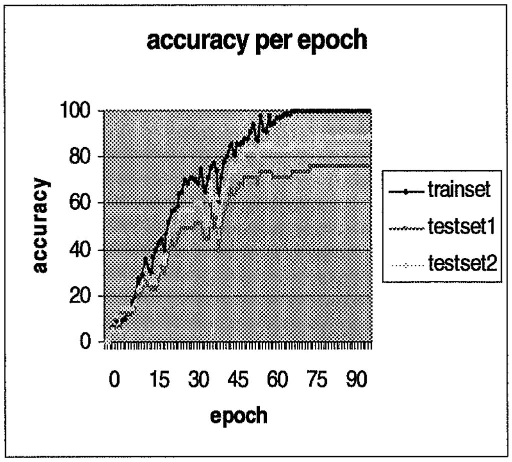


Figure 6 (a) Output error per epoch for training and two test 1 and 2 using ANN

Figure 6 (b) Accuracy per epoch for training and two test sets 1 and 2 using ANN

We take the same training set and the testing set (second one) for the classification technique LDA. But it gives a poor accuracy, 45.71%. Here, accuracy is percentage of truly classified images. So, we perform PCA on the images to extract the characteristics features from each image. Here we applied class dependant PCA. That means for each particular class we derived a transform metrics rather than a global metrics. In our training set, there were 16 classes and we have 16 such transformation metrics. We extract features by taking first 100, 200, …, 700 features (dimension) and so on. These features represent the important characteristics of each image for a particular class. The discarded portion are not significant characteristics of images, thus we can say that those are the noise parts that are eliminated.

But how much we should reduce the dimension that depends on image sets. In our case we found that when we take first 400 features it gives highest performance, after then it starts swinging. It is shown in Figure 7.

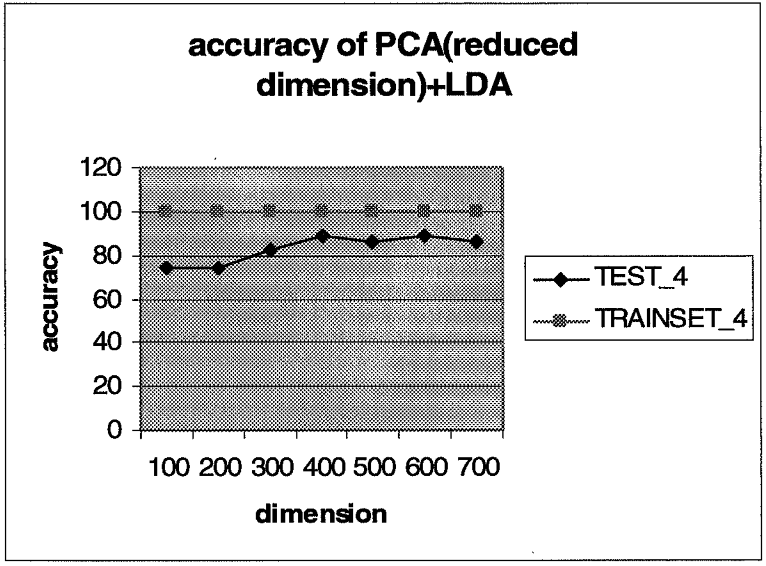


Figure 7. Applying PCA for dimension reduction and then LDA

After reducing dimension we apply LDA and obtain 88.571429 and 85.7124286 percentage of accuracy for 500 and 400 dimension respectively. The results using only single LDA and (PCA+LDA) are shown in Figure 8.

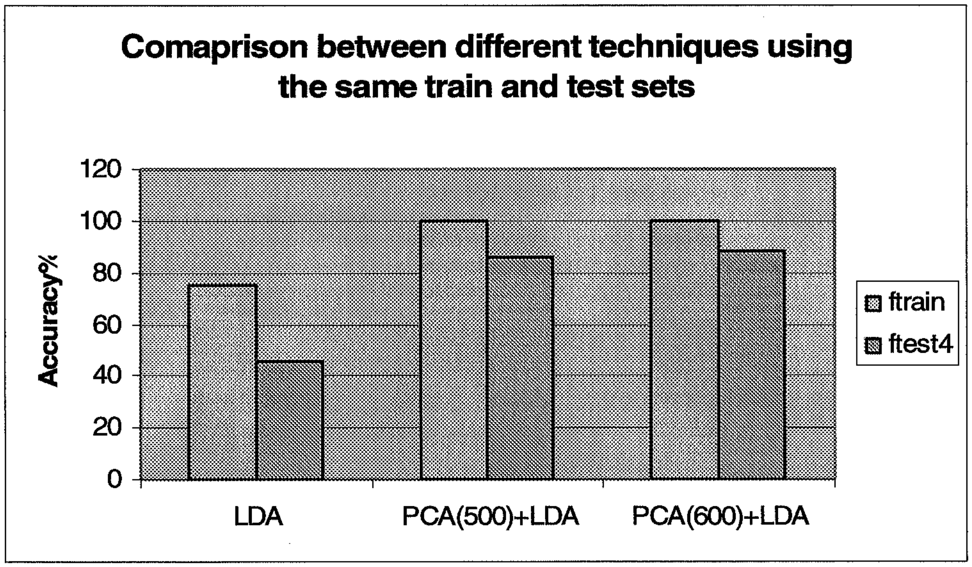


Figure 8. Accuracy using LDA and PCA+LDA

Then the outputs of all the classifiers (PCA+LDA), LDA and ANN for a particular class will be combined by using a linear combination model and see whether it is greater than a certain threshold. If it is, we declare that data will belong to this class. For example, in our case, we use the following equation:

```
If
    $0.1*LDA+0.2*ANN+0.1*\{PCA[100]+LDA)+(PCA[200]+LDA)+(PCA[300]+LDA)+(PCA[400]+LDA)+(PCA[500]+LDA)+(PCA[600]+LDA)+(PCA[700]+LDA)\}>=0.5$
then
    true classified.
else
    false classified.
```

Here PCA [300] means after dimensionality reduction, we use 300 features.

Finally, we get accuracy as 91.42857%.

4. Conclusion and Future Work

In this paper, we propose a near real time face recognition system for home application domain. For this, first, we advocate feature reduction technique to speed up retrieval. Second, we propose a weighted feature selection to improve face recognition accuracy. Finally, we use a hybrid method in recognition based on average linear weighting for evidence combination to improve recognition accuracy.

We would like to extend our research in the following directions. First, we would like to implement face detection technique to facilitate real time feature extraction. Second, we would like to extend this work as a full fledge prototype. Third we want to apply another classifier Support Vector Machine (SVM) together with the above methods. Finally, we would like to use boosting and bagging in the same context and compare it with our hybrid approach.

References

[1] HomeNet2Run project website: ***http://www.extra.research.philips.com/euprojects/hn2r/***

[2] J.Yang, W.Lu and A.Waibel, Skin-color modeling and adaptation, *Proc. ACCV*, pp.687–694, 1998.

[3] Y.Ma and X.Ding, Face detection based on hierarchical Support Vector Machines, *Proc. ICPR*, pp.222–225, 2002.

[4] F.Zuo and P.H.N. de With, Fast human face detection using successive face detectors with incremental detection capability, *Proc. SPIE*, 5022, 2003.

[5] H.Rowley, S.Balujua and T.Kanade, Neural network-based face detection, *IEEE Trans. PAMI. 20(1)*, pp. 23–28, 1998.

[6] F.Zuo and P.H.N.de With, Fast facial feature extraction using a deformable shape model with Haar-wavelet based local texture attributes, to be published in *Proc. ICIP*, 2004.

[7] T.Cootes, An introduction to active shape models, in Image Processing and Analysis, pp. 223–248, 2000.

[8] F.Zuo and P.H.N.de With, Two-stage face recognition incorporating individual-class discriminant criteria, *Proc. WIC 2004*, pp. 137–144, 2004.

[9] T. Mitchell, Machine Learning, *McGraw Hill*, 1997

[10] V. N. Vapnik, “The Nature of Statistical Learning Theory”. *Springer*, 1995.

[11] Glenn Shafer. A Mathematical Theory of Evidence, *Princeton University Press*, 1976.

[12] J. Platt. Probabilistic outputs for SVMs and comparisons to regularized likelihood methods. In Advances in Large Margin Classifiers. *MIT Press*, 1999.

[13] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179-188, 1936.

[14] R.L.Hsu, M.A.Mottalab and A.K.Jain, Face detection in color images, *IEEE transaction on pattern analysis and machine intelligence, Vol 24, No.5*, May 2002

[15] Y.H.Chan and S.A.R.Abu-Bakar, Face detection system based on feature-based chrominance color information, *Proceedings of the International Conference on Computer Graphics, Imaging and Visualization (CGIV’04*), 2004.

[16] Y.Gao and Maylor K.H. Leung, Face recognition using line edge map, *IEEE transaction on pattern analysis and machine intelligence, Vol 24, No. 6*, June 2002

[17] F. Zuo, Peter H.N de With, Real time embedded face recognition for smart home, *IEEE transactions on consumer Electronics, Vol. 51 , No.1*, February 2005

[18] C. Liu, H.Wechsler, Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition, *IEEE Transactions on image processing, vol. 11, No. 4, April, 2002.*

[19] O.Veksler, Pattern Recognition Lecture 8, http://www.csd.uwo.ca/faculty/olga/Courses/CS434a_541a/

[20] S. Balakrisnama, A. Ganapathiraju, *Linear Discriminant Analysis Tutorial,* http://lcv.stat.fsu.edu/research/geometrical_representations_of_faces/PAPERS/lda_theory.pdf