Face Recognition Using PCA and SVM

Md. Omar Faruqe
Dept. of Computer Science & Engineering,
BRAC University,
66-Mohakhali C/A, Dhaka-1212, Bangladesh
Email: faruqe@bracu.ac.bd

Md. Al Mehedi Hasan
Dept. of Computer Science & Engineering,
Rajshahi University of Engineering & Technology,
Rajshahi, Bangladesh
Email: mehedi_ru@yahoo.com

Abstract—**Automatic recognition of people has received much attention during the recent years due to its many applications in different fields such as law enforcement, security applications or video indexing. Face recognition is an important and very challenging technique to automatic people recognition. Up to date, there is no technique that provides a robust solution to all situations and different applications that face recognition may encounter. In general, we can make sure that performance of a face recognition system is determined by how to extract feature vector exactly and to classify them into a group accurately. It, therefore, is necessary for us to closely look at the feature extractor and classifier. In this paper, Principle Component Analysis (PCA) is used to play a key role in feature extractor and the SVMs are used to tackle the face recognition problem. Support Vector Machines (SVMs) have been recently proposed as a new classifier for pattern recognition. We illustrate the potential of SVMs on the Cambridge ORL Face database, which consists of 400 images of 40 individuals, containing quite a high degree of variability in expression, pose, and facial details. The SVMs that have been used included the Linear (LSVM), Polynomial (PSVM), and Radial Basis Function (RBFSVM) SVMs. We provide experimental evidence which show that Polynomial and Radial Basis Function (RBF) SVMs performs better than Linear SVM on the ORL Face Dataset when both are used with one against all classification. We also compared the SVMs based recognition with the standard eigenface approach using the Multi-Layer Perceptron (MLP) Classification criterion.**

Keywords- Face Recognition, Principal Component Analysis, Support Vector Machine, Multi-Layer Perceptron, Kernel Functions.

I. INTRODUCTION

Face recognition technology can be used in wide range of applications such as identity authentication, access control, and surveillance. Interests and research activities in face recognition have increased significantly over the past few years [1]. A face recognition system should be able to deal with various changes in face images. However, "the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity" [2]. In geometric feature-based methods [3], facial features such as eyes, nose, mouth, and chin are detected. Properties and relations such as areas, distances, and angles, between the features are used as the descriptors of faces. In contrast, template matching and neural methods [1] generally operate directly on an image-based representation of faces, *i.e.* pixel intensity array. Because the detection and measurement of geometric facial features are not required, these classes of methods have been more practical and easy to implement as compared to geometric feature based methods.

Support Vector Machines (SVMs) have been recently proposed by Vapnik and his co-workers [4] as a very effective method for general purpose pattern recognition. In this research, we focus on the face recognition problem, and show that the discrimination functions learned by SVMs can give much higher recognition accuracy than the popular standard eigenface approach using MLP classifier. Principal Component Analysis (PCA) is used to extract the feature from the face image. After the features are extracted, the SVMs are learned and the disjoint test set enters the system for recognition. We propose to construct a one versus all to recognize the testing samples. We perform experiment on the ORL face database of 400 images of 40 individuals.

The proposed system can be divided into two main modules: (1) the Feature extractor and (2) the Classifier. The Feature extractor module includes: (a) Data Acquisition, (b) Preprocessing, and (c) Feature Extraction. The Classifier module includes: (a) Learning Methods, and (c) Classification.

II. FEATURE EXTRACTOR: PCA

Let a face image *I(x, y)* be a two-dimensional N by N array of intensity values or a vector of dimension N^2. A typical image of size 256 by 256 describes a vector of dimension 65,536, or equivalently, a point in 65,536-dimensional space [5]. Consider our training set of images of 85 by 95 pixels. The main idea of principal component analysis is to find the vectors which best account for the distribution of the face images within the entire image space. Steps for Feature Extraction:

1. The first step is to obtain a set S with M face images. Each image is transformed into a vector of size N and placed into the set.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_4, \cdots\cdots, \Gamma_M\}$$

2. Second step is to obtain the mean image Ψ.

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

3. Then find the difference Φ between the input image and the mean image

978-1-4244-3884-6/09/$25.00 ©2009 IEEE

$$\Phi_i = \Gamma_i - \Psi$$

4. Next seek a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The k_{th} vector, u_k, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}\left(u_k^T \Phi_n\right)^2$$

is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{If } l=k \\ 0 & \text{Otherwise} \end{cases}$$

where u_k and λ_k are the eigenvectors and eigenvalues of the covariance matrix C

5. The covariance matrix C has been obtained in the following manner

$$C = \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^T$$
$$= AA^T$$
$$A = \{\Phi_1, \Phi_2, \Phi_3, \cdots\cdots\cdots, \Phi_n\}$$

6. To find eigenvectors from the covariance matrix is a huge computational task. Since M is far less than N^2 by N^2, we can construct the M by M matrix $L = A^T A$

7. Find the M eigenvector, v_l of L.

8. These vectors (v_l) determine linear combinations of the M training set face images to form the eigenfaces u_l

$$u_l = \sum_{k=1}^{M} v_{lk}\Phi_k \qquad l = 1,2,\cdots\cdots\cdots, M$$

9. Project each of the original images into eigenspace. This gives a vector of weights representing the contribution of each eigenfaces to the reconstruction of the given image.

$$\omega_k = u_k^T(\Gamma - \Psi) \qquad \Omega^T = [\omega_1, \omega_2, \cdots\cdots\cdots, \omega_M]$$

where u_k is the k^{th} eigenvector and ω_k is the k^{th} weight in the vector $\Omega^T = [\omega_1, \omega_2, \omega_3, \ldots \omega_M]$.

III. CLASSIFIER: SUPPORT VECTOR MACHINE

A. Binary Classification

SVMs perform pattern recognition between two classes by finding a decision surface that has maximum distance to the closest points in the training set which are termed support vectors [6] [11] [12]. Consider the examples in Fig. 1, where there are many possible linear classifiers that can separate the data, but there is only one that maximizes the margin shown in Fig. 2. This classifier is termed the optimal separating hyperplane (OSH).

Fig. 1. Arbitrary hyperplanes

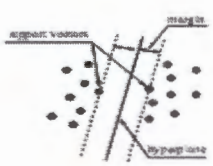

Fig. 2. Optimal hyperplane

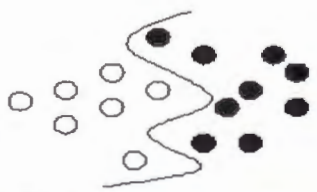

Fig. 3. Linearly Inseparable

Consider the problem of separating the set of training vectors belong to two separate classes, $(x_1, y_1), \ldots, (x_l, y_l)$, where $x_i \in R^n$, $y_i \in \{-1,+1\}$ with a hyperplane $w \cdot x = b = 0$. The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the margin is maximal. A canonical hyperplane [6] has the constraint for parameters w and b: $\min_{x_i} y_i(w \cdot x_i + b) = 1$.

A separating hyperplane in canonical form must satisfy the following constraints,

$$y_i[(w.x_i) + b] \geq 1 \ , i = 1, \ldots, l \qquad (1)$$

The distance of a point x from the hyperplane is,

$$d(w, b; x) = \frac{|w.x + b|}{\|w\|} \qquad (2)$$

The margin is $\frac{2}{\|w\|}$ according to its definition. Hence the hyperplane that optimally separates the data is the one that minimizes

$$\phi(w) = \frac{1}{2}\|w\|^2 \qquad (3)$$

The solution to the optimization problem of (3) under the constraints of (1) is given by the saddle point of the Lagrange functional,

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l}\alpha_i\{y_i[(w.x_i) + b] - 1\} \qquad (4)$$

where α_i are the Lagrange multipliers. The Lagrangian has to be minimized with respect to w, b and maximized with respect to w, b. Classical Lagrangian duality enables the primal problem (4) to be transformed to its dual problem,which is easier to solve. The dual problem is given by,

$$\max_{\alpha} W(\alpha) = \max_{\alpha}\left\{\min_{w,b} L(w, b, \alpha)\right\} \qquad (5)$$

The solution to the dual problem is given by,

$$\bar{\alpha} = \arg\min_{\alpha}\sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j x_i \cdot x_j \qquad (6)$$

with constraints,

$$\alpha_i \geq 0, \ i = 1, \ldots, l \qquad (7)$$

978-1-4244-3884-6/09/$25.00 ©2009 IEEE

$$\sum_{i=1}^{l} \alpha_i y_i = 0 \quad (8)$$

Solving Equation (6) with constraints (7) and (8) determines the Lagrange multipliers, and the OSH is given by,

$$\overline{w} = \sum_{i=1}^{l} \overline{\alpha}_i y_i x_i \quad (9)$$

$$\overline{b} = -\frac{1}{2}\overline{w} \cdot [x_r + x_s] \quad (10)$$

where x_r and x_s are support vectors, satisfying,

$$\overline{\alpha}_i, \overline{\alpha}_s > 0,\ y_r = 1,\ y_s = -1 \quad (11)$$

For a new data point x, the classification is then,

$$f(x) = sign(\overline{w} \cdot x + \overline{b}) \quad (12)$$

So far the discussion has been restricted to the case where the training data is linearly separable. The entire construction can be extended to the case of nonlinear separating surfaces shown in Fig 3. Each point x in the input space is mapped to a point $z = \phi(x)$ of a higher dimensional space, called the feature space, where the data are separated by a hyperplane. The key property in this construction is that the mapping $\phi(.)$ is subject to the condition that the dot product of two points in the feature space $\phi(x).\phi(y)$ can be rewritten as a kernel function $K(x, y)$. The decision surface has the equation:

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i K(x, x_i) + b$$

Again, the coefficients α_i and b are the solutions of a quadratic programming problem.

The kernel functions used in the experiments are linear, polynomial and radial basis functions (RBFs) defined as

Linear Kernel:

$$K(x_i, x_j) = x_i \cdot x_j$$

Polynomial Kernel:

$$K(x_i, x_j) = (x_i \cdot x_j + 1)^d$$

Radial Basis Kernel:

$$K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$$

where x_i and x_j denote two samples. The user-controlled parameters are the degree d in the case of the polynomial and the γ value in the case of the RBF kernel.

B. MULTI-CLASS CLASSIFICATION

Like many discriminative classifiers, SVM's are designed to solve binary classification problems. However, many real-world classification problems involve more than two classes. Attempts to solve q-class problems with SVM's have involved training q SVM's, each of which separates a single class from all remaining classes [7] [11], or training q^2 machines, each of which separates a pair of classes [8] [12]. The first types of classifiers are usually called one-vs-all, while classifiers of the second type are called pairwise classifiers. The pairwise classifier can be made in two ways: one is Bottom-UP decision tree and the other one is Top-Down decision graph. Let us see how each of these classifier are computed:

1) ONE VERSUS ALL

In the one-vs-all type scheme, there is one SVM classifier associated to each class. For each class $j \in \{1,.....,q\}$ the corresponding classifier is trained to separate the examples in this class (positive labeled) from the remaining ones (negative labeled). A new input vector is classified in one class for which associated classifier has the highest score among all classifiers.

2) BOTTOM- UP DECISION TREE

Initially, a binary tree is formed with q-1 nodes. This tree has no more than q-1 layers and exactly q leaves. Each node in the tree has two inputs and one output. A different class is assigned to each leaf. Classification of an input point starts at the bottom layer. For each node in this layer, the pairwise SVM's classifier is computed and the result of the classification (the winning class) assigned to the output node. The same procedure is repeated for the nest layer until the top node is reached. At the top node only two classes remain and the final classification is determined by the SVM corresponding to these two classes. We call this classification scheme bottom-up decision tree as the classification is carried out from the bottom of the tree to the top.

3) TOP-DOWN DECISION GRAPH

This scheme was recently proposed in [9]. The system architecture is based on the so called direct acyclic graph. Thus graph has the shape of a triangle, with q-1 layers. The j-th layer has j nodes, each with two edges. For each layer except the last one, the i-th node is connected to the i-th and (i +1)-th nodes of the next layer. The first layer contains only one node, which is the root node of the graph. The number of nodes is equal to q (q – 1)/2 and each node is associated to a pairwise SVM classifier as following : (i) An ordered pair of class at the root node is selected. (ii) If (a,b) is the ordered pair of classes assigned to the i-th node of the j-th layer, the i-th and (i +1)-the nodes in the (j+1)-th layer will have pairwise classifier (a, .) and (.,b) respectively.

There is no theoretical analysis of the two strategies with respect to classification performance. Regarding the training effort, the one-vs-all approach is preferable since only _ SVMs have to be trained compared to q (q - 1)/q SVMs in the pairwise approach. The run-time complexity of the two strategies is similar: The one-vs-all approach requires the evaluation of q, the pairwise approach the evaluation of (q – 1) SVMs. Recent experiments on person recognition show similar classification performances for the two strategies. Since the number of classes in face recognition can be rather large we opted for the one-vs-all strategy where the number of SVMs is linear with the number of classes.

IV. CLASSIFIER: MLP

Artificial Neural Networks were inspired by biological findings relating to the behavior of the brain as a network of units called neurons [10] [13]. The fundamental building block in an ANN is the mathematical model of a neuron the three basic components of the neuron are:

1. The synapses or connection links that provide weights, w_j, the input values, x_j for $j = 1, \ldots, m$

978-1-4244-3884-6/09/$25.00 ©2009 IEEE

2. An adder that sums the weighted input values to compute the input to the activation function

$$v = w_0 + \sum_{j=1}^{m} w_j x_j \text{, where } w_0 \text{ is called bias}$$

is a numerical value associated with the neuron. It is convenient to think of the bias as the weight for an input x_0 whose value is always equal to one, so that

$$v = \sum_{j=0}^{m} w_j x_j$$

3. An activation function g (also called a squashing function) that maps v to g(v) the output value of the neuron. This function is a monotone function.

A. BACKWARD PROPAGATION ALGORITHM

There is a minor adjustment for prediction problems where we are trying to predict a continuous numerical value. In that situation we change the activation function for output layer neurons to the identity function that has output value = input value.

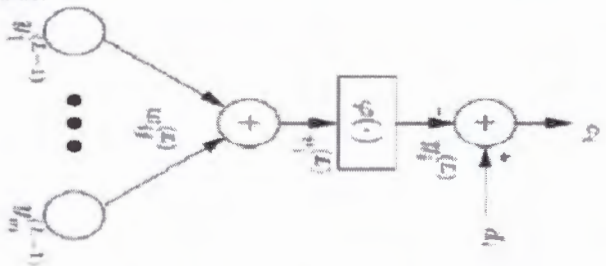

Fig. 4. Simple Multi-layer Perceptron

a) Forward pass: Computation of outputs of all the neurons in the network.

$$v_i^{\,L} = \sum_j w_{ij} y_j^{\,L-1}$$

$$y_i^{\,L} = \varphi(v_i^{\,L})$$

b) Backward pass: Propagation of error and adjustment of weights.

$$w_{ij}^{\,L}(t+1) = w_{ij}^{\,L} + \eta \zeta_i^{\,L}(t) y_j^{\,L-1}(t) \text{, where}$$

$$\zeta_i^{\,L} = \begin{cases} \varphi'\{v_i^{\,L}\} \sum \zeta_k^{\,L+1}(t) w_{ki}^{\,L+1} & \text{,hidden neuron} \\ e_i^{\,L}(t) \varphi'\{v_i^{\,L}\} & \text{,output neuron} \end{cases}$$

V. EXPERIMENTAL RESULT

In this work, the experiment is performed on the ORL face database, which contains 40 distinct persons. Each person has ten different images, taken at different times. We show five individuals with five images for each person of the ORL face images in Fig. 5. There are variations in facial expressions such as open/closed eyes, smiling/nonsmiling, and facial details such as glasses/no glasses. All the images were taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some side movements. There are also some variations in scale.

In our face recognition experiments on the ORL database, we select 200 samples (5 for each individual) randomly as the training set, from which we calculate the eigenfaces and train the support vector machines (SVMs). The remaining 200 samples are used as the test set. Such procedures are repeated for ten times, i.e., ten runs. In our experiment, the number of features will be changed up to 90. And then we take a closely look at the result corresponding to the number of features. In general, many feature vectors needs more computing time but gives more accuracy. When the number of feature vectors is chanced, input vectors of classifier are also changed, since output passed through PCA is directly connected to the input of the classifier. At first step we have performed experiment using SVMs based classifier. The SVMs that have been used included the Linear, Polynomial, and Radial Basis Function (RBF) SVMs. Fig. 6 shows the recognition accuracy among Linear, Polynomial and Radial Basis Function (RBF) SVMs when the number of feature vectors are chanced from 30 to 90. The degree d =5 in the case of the polynomial and the γ = 0.005 value in the case of the RBF kernel has been used in the Experiment. We provide experimental evidence which show that Polynomial and Radial Basis Function (RBF) SVMs performs better than Linear SVM on the ORL Face Dataset when both are used with one against all classification.

Fig 5. Five individuals with five images for each person

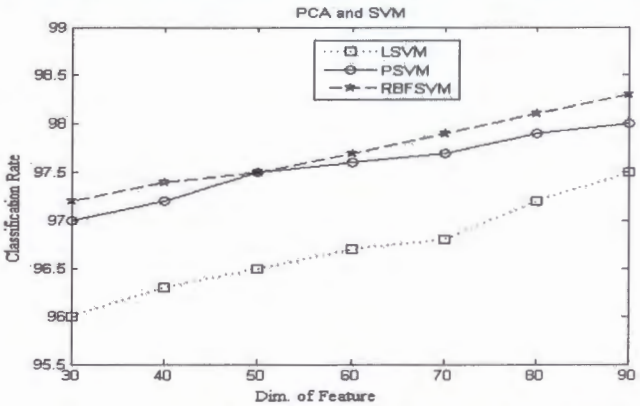

Fig. 6. Recognition Rates with Increasing Dimension of Features for Linear, Polynomial, and Radial Basis Functions SVMs

At the second step, we have performed experiment with the standard eigenface approach using Multi-Layer Perceptron (MLP) Classification criterion. The drawback of neural networks is as follows. There is no clearly proved rule so as to properly set network's parameters such as number of hidden neurons, number of epoches, learning rate, momentum and so on. But some guidelines give us a chance to reach a better performance. For instance, cross-validation is a well known technique to configure appropriate number of hidden neurons. Fig. 7 shows Classification rate corresponding to the number of hidden neurons. We can find out that when the number of neurons is 50, classification rate is increased up to 95%. Finally, Fig. 8 gives the result of the classification rate corresponding to the number of feature vectors with the

978-1-4244-3884-6/09/$25.00 ©2009 IEEE

standard eigenface approach using Multi-Layer Perceptron (MLP) Classification criterion.

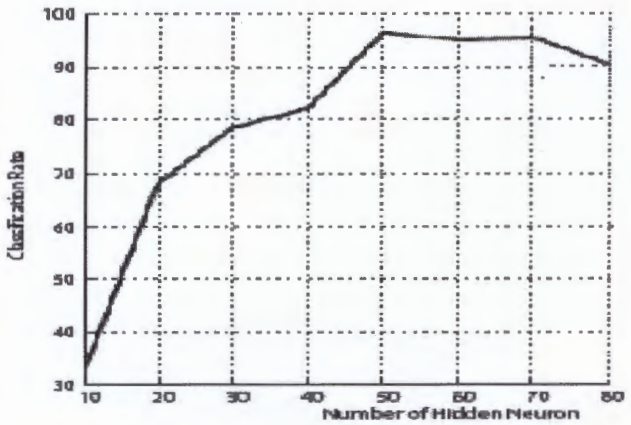

Fig. 7. Classification Rate VS Number of Hidden Neuron

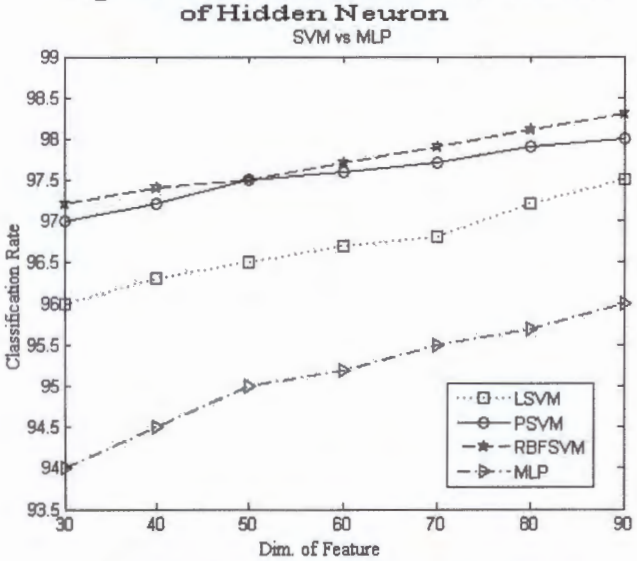

Fig. 8. Comparing with SVM and MLP w.r.t Classification Rate

For comparison, we show the results of SVMs and MLP in the same figure 6. It is obvious that the classification rates of SVMs are much higher than that of MLP. If we choose the best results among the results, the highest classification rate of the Radial Basis Function (RBF) SVM can achieve 98% whereas MLP gives 96%.

VI. CONCLUSION

We have presented the face recognition experiments using support vector machines with a one versus all classification strategy. Experimental evidence shows that Polynomial and Radial Basis Function (RBF) SVMs performs better than Linear SVM on the ORL Face Dataset. As shown in the comparison with other techniques, the experimental results show that the SVMs are a better learning algorithm than the standard eigenface approach using Multi-Layer Perceptron (MLP) Classification criterion for face recognition.

REFERENCES

[1] R. Chellappa, C. L.Wilson, and S. Sirohey. Human and machine recognition of faces: A survey. *Proc. IEEE*, 83:705– 741, May 1995.

[2] Y. Moses, Y. Adini, and S. Ullman. Face recognition: the problem of compensating for changes in illumination direction. *European Conf. Computer Vision*, pages 286–296, 1994.

[3] R. Brunelli and T. Poggio. Face recognition: Features versus templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15:1042–1052, 1993.

[4] V. N. Vapnik. Statistical learning theory. John Wiley & Sons, New York, 1998.

[5] J. Ruiz-del-Solar and P. Navarrete, Eigenspacebased face recognition: a comparative study of different approaches, *IEEE Transactions on Systems, Man and Cybernetics, Part C,* Vol. 35, Issue 3, Page(s):315 – 325, Aug. 2005.

[6] Guodong Guo, Li, S.Z., Kapluk Chan, Face recognition by support vector machines, Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on Volume , Issue , 2000 Page(s):196 - 201

[7] C. Cortes, V. Vapnik, Support Vector Networks, Machine Learning 20, pp. 273-297, 1995

[8] M.Pontil, A. Verri, Support Vector Machines for 3-d Object Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 35-39, 1998.

[9] J. Platt, N. Cristiani, J. Shawe-Taylor, Large Margin dags for multiclass classification, Advances in Neural Information Processing Systems, MIT press, pp. 547-553.

[10] http://home.postech.ac.kr/~taey16/index.files/ML2007/ML_FACE_KPCA.pdf

[11] Zhifeng Li, Xiaoou Tang, Using Support Vector Machines to Enhance the Performance of Bayesian Face Recognition, Information Forensics and Security, IEEE Transactions onVolume 2, Issue 2, Page(s):174 – 180, June 2007

[12] Gates, K.E. Fast and Accurate Face Recognition Using Support Vector Machines, Computer Vision and Pattern Recognition, 2005 IEEE Computer Society Conference on Volume 3, Issue , Page(s): 163 – 163, 20-26 June 2005

[13] Md. Al Mehedi Hasan, Mohammed Nasser, Face Recognition using PCA and MLP, International Conferenc on Electronics, Computer and Communication 2008, University of Rajshahi, Rajshahi, Bangladesh, pp-368-371, ISBN 984-300-002131-3.

978-1-4244-3884-6/09/$25.00 ©2009 IEEE