Performance Evaluation Of Selected Principal Component Analysis-Based Techniques For Face Image Recognition

Aluko J. Olubunmi, Omidiora E. Olusayo, Adetunji A. Bola, Odeniyi O. Ayodeji

Abstract: Principal Component Analysis (PCA) is an eigen-based technique popularly employed in redundancy removal and feature extraction for face image recognition. In this study, performance evaluation of three selected PCA-based techniques was conducted for face recognition. Principal Component Analysis, Binary Principal Component Analysis (BPCA), and Principal Component Analysis – Artificial Neural Network (PCA-ANN) were selected for performance evaluation. A database of 400, 50x50 pixels images consisting of 100 different individuals, each individual having 4 images with different facial expressions was created. Three hundred images were used for training while 100 images were used for testing the three face recognition systems. The systems were subjected to three selected eigenvectors: 75, 150 and 300 to determine the effect of the size of eigenvectors on the recognition rate of the systems. The performances of the techniques were evaluated based on recognition rate and total recognition time.The performance evaluation of the three PCA-based systems showed that PCA – ANN technique gave the best recognition rate of 94% with a trade-off in recognition time. Also, the recognition rates of PCA and B-PCA increased with decreasing number of eigenvectors but PCA-ANN recognition rate was negligible.

Index terms: Principal Component Analysis, Binary Principal Component Analysis (BPCA), and Principal Component Analysis – Artificial Neural Network (PCA-ANN).

1 INTRODUCTION

The face is the primary focus of attention in the society, playing a major role in conveying identity and emotion The human ability to recognize faces is remarkable; a human can recognize thousands of faces learned throughout the lifetime and identify familiar faces at a glance even after years of separation. This skill is quite robust, inspite of large changes in the visual stimulus due to viewing conditions, expression, aging, and distractions such as glasses, beards or changes in hair style [1]. Using human face as a key to security, biometric face recognition technology has received significant attention due to its potential for a wide variety of both in law enforcement and non-law enforcement. In addition, face recognition serves the crime deterrent purpose because face images that have been recorded and archived can later help identify a person thereby reducing rate of criminal activities in the society like impersonation in schools. For face recognition system to be efficient and robust to serve it purpose of security, there is need to use the best technique out of the many techniques that have been proposed in literatures for face recognition.

- *Aluko J. Olubunmi, Directorate of Information Communication and Technology, Osun State College of Education, Ilesa, Osun State Nigeria. bnm_jnt@yahoo.com*
- *Omidiora E. Olusayo, Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria. eoomidiora@lautech.edu.ng*
- *Adetunji A. Bola, Department of Computer Science and Engineering, Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria. abadetunji@yahoo.com*
- *Odeniyi O. Olufemi, Department of Computer Science, Osun State College of Technology, Esa-Oke, Osun State, Nigeria. olufemiodeniyi@gmail.com*
- All Authors are Correspondent Authors.

1.1 Face Recognition Approaches

In general, face recognition techniques can be divided into two groups based on the face representation they use:

1. Appearance-based: which uses holistic texture features and is applied to either whole-face or specific regions in a face image.
2. Feature-based: which uses geometric facial features (mouth, eyes, brows, cheeks etc.) and geometric relationships between them.

Among many approaches to the problem of face recognition, appearance-based subspace analysis is one of the oldest and still gives the most promising results. Subspace analysis is done by projecting an image into a lower dimensional space (subspace) and after that recognition is performed by measuring the distances between known images and the image to be recognized. The most challenging part of such a system is finding an adequate subspace. When using appearance-based methods, an image of size n x m pixels is usually represented by a vector in an n x m dimensional space. These (n x m dimensional) spaces are too large to allow robust and fast object recognition. A common way to attempt to resolve this problem is to use dimensionality reduction techniques. PCA and LDA are the two most popular techniques usually used for dimensionality reduction [5].

2 Principal Component Analysis (PCA)

The Principal Component Analysis (PCA) is one of the most commonly classical statistic used techniques in prediction, redundancy removal, feature extraction, data compression, image recognition, etc. Because PCA is a classical technique which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc. 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 is needed to describe the data economically [3]. The basis vectors constructed by PCA have the same dimension as the input face images. The classical representation of a face image is obtained by projecting it to the coordinate system defined by the principal components. The projection of face images into the principal component subspace achieves information compression, de-correlation and dimensionality reduction to facilitate decision making [10]. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images (vectors).

2.1 Calculation of Eigenfaces with PCA

To generate a set of eigenfaces, a large set of digitized images of human faces, taken under the same lighting conditions, are normalized to line up the eyes and mouths. They are then all resample at the same pixel resolution. Eigenfaces can be extracted out of the image data by means of a mathematical tool called Principal Component Analysis (PCA) [3]. Identifying images through eigenspace projection takes four basic steps. First the normalized training images are stored in a vector of size N. Then, the eigenspace must be created using training images. Next, the training images are projected into the eigenspace. Finally, the test images are identified by projecting them into the eigenspace and comparing them to the projected training images [6]. The steps for the PCA algorithm are as follow:

Step 1: The normalized training image in the *N*-dimensional space is stored in a vector of size *N*. Let the normalized training face image set,

$$T = \{X_1, X_2, ..., X_N\} \text{where } T = \{x_1, x_2, x_3, ..., x_n\}^T \quad (1)$$

Step 2: Create Eigenspace

Each of the normalized training face images is mean centered. This is done by subtracting the mean face image from each of the normalized training images.

$$\overline{X}_i = \overline{X_i} - \overline{X} \quad (2)$$

where the average of the training face image set is defined as:

$$\overline{X} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad (3)$$

The training images are combined into a data matrix of size *N* by *P*, where *P* is the number of training images and each column is a single image.

$$\overline{X} = \{\overline{X}_1, \overline{X}_2, ..., \overline{X}_p\} \quad (4)$$

The column vectors are combined into a data matrix which is multiplied by its transpose to create a covariance matrix. The covariance is defined as:

$$\Omega = \overline{X}.\overline{X}^T \quad (5)$$

The eigenvalues and corresponding eigenvectors are computed for the covariance matrix using Jacobian transformation,

$$\Omega V = \Lambda V \quad (6)$$

Where *V* is the set of eigenvectors associated with the eigenvalues Λ. The eigenvectors ***V_i Є V***are orderaccording to their corresponding eigenvalues **λ Є Λ** from high to low with non-zero eigenvalues.

$$V = \{V_1, V_2, ..., V_p\} \quad (7)$$

Step 3: Project Training Images

Each of the centered training images $(\overline{X}_i)$ is projected into the eigenspace. The projected training images (T_p) are computed based on the dot product of the centered training images with each of the ordered eigenvectors denoted as,

$$T_p = V^T \overline{X}_i \quad (8)$$

The new vectors of the projected images are the feature vectors of the training face images. Let $T_p = \{X_{p1}, X_{p2}, ..., X_{pn}\}$ as feature vectors from the projection of the training set onto the principal components. The feature vector is defined as

$$X_p = \{P_1, P_2, P_3, ..., P_m\}^T.$$

Step 4: Project Testing Image

The vector of the testing face image (Y_p) is initially mean centered by subtracting the mean image.

$$\overline{Y} = Y - \overline{X} \quad (9)$$

The feature vector of the testing image (Y_p) is obtained by projecting the vector of the mean testing face image $(\overline{Y})$ into the eigenspace or the principal components,

$$Y_p = V^T \overline{Y} \quad (10)$$

3 Binary Principal Component Analysis (BPCA)

Binary PCA (B-PCA) is the combination of PCA and Haar-like binary box functions into a non-orthogonal subspace representation that can be computed very efficiently and at the same time capture the structure information of the data. B-PCA method can perform vector dot product very efficiently using integral image. Experiments on real image datasets show that B-PCA has comparable performance to PCA in image reconstruction and recognition tasks with significant speed improvement. A desirable property of these box functions is that their inner product operation with an image can be computed very efficiently [8]. Haar-like features are digital image features used in object recognition. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is then used to categorize subsections of an image. These features do not correspond to objects such as eyes and ears, but rather to classes such as relatively light areas next to relatively dark areas, dark areas surrounded by lighter areas, and lengths of lighter color bounded on two sides by dark areas [11]. One obvious property of binary box bases is that they are not necessarily orthogonal to each other and span as Non-Orthogonal Binary Subspace (NBS) [8]. PCA is an orthogonal subspace and has been proved to have good performance in image representation and recognition. Its capability to capture image structure information, orthogonal base vectors are complement to NBS. This makes it natural to combine NBS and PCA to the proposed binary PCA representation. The problem of searching for the best subspace representation in a set of predefined non-orthogonal base vector dictionary is known to be NP-hard. Two of the popular greedy solutions to this problem include the Matching Pursuit (MP) approach and the Optimized Orthogonal Matching Pursuit (OOMP) method [8].

3.1 Matching Pursuit (MP)

The matching pursuit (MP) method is a technique to compute adaptive signal representation by iterative selection of base vectors from *a* dictionary*.* Such a dictionary is usually non-orthogonal. MP method sequentially selects the base vector ϕ_k from D such that $|c_i| = \left|\left\langle x - R_{\Phi_{k-1}}(x), \phi_k \right\rangle\right|$ is maximized [9].

3.2 Optimized Orthogonal Matching Pursuit (OOMP)

Optimized Orthogonal Matching Pursuit (OOMP) is a technique to compute adaptive signal representation by iterative selection of base vectors from a dictionary. OOMP uses a similar criterion as MP to select the base vectors, but it maintains full backward orthogonality of the residual in each iteration. Thereby, the reconstruction of x using OOMP-selected base vectors Φ_k is orthogonal to the residual, hence, the name orthogonal MP. In addition, OOMP selects the base vector ϕ_i that minimizes the corresponding residual error [8].

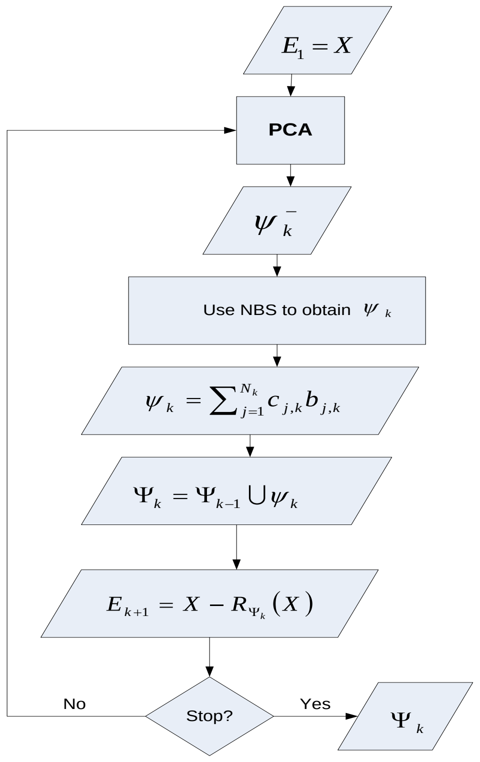
$E_1 = X$
PCA
ψ_k^-
Use NBS to obtain ψ_k
$\psi_k = \sum_{j=1}^{N_k} c_{j,k} b_{j,k}$
$\Psi_k = \Psi_{k-1} \cup \psi_k$
$E_{k+1} = X - R_{\Psi_k}(X)$
No
Stop?
Yes
Ψ_k

Figure 1.*Flow chart of the PCA-guided OOMP*

3.3 The Solution - PCA guided NBS

To overcome this problem, aPCA guided NBS method was performed to find a suboptimal solution efficiently shown in Figure 1. In the PCA guided NBS, the selected B-PCA base vectors was denoted up to iteration *k* as $\Psi_{k-1} = [\psi_1, \psi_2, ..., \psi_{k-1}]$. This set is empty at the beginning. The original PCA procedure was use to obtain the first principal component that captures the majority of the data variance. The first principal component is called the *Pre-BPCA* vector, denoted as ψ_1^-. NBS is then applied to approximate this vector as $\psi_1 = \sum_{j=0}^{N_1} c_{j,1} b_{j,1}$. Then, in iteration *k*, the data X is projected to the subspace spanned by the already selected B-PCA bases Ψ_{k-1}, and PCA is applied on the residual of the data $X - R_{\Psi_{k-1}}(X)$ to obtain the next *Pre-BPCA* ψ_k^- which is again approximated using NBS. The approximation of *Pre-BPCA* at iteration *k* is called the *k-th B-PCA base vector*. This procedure iterates until the desired number of B-PCA bases have been obtained or an error threshold is reached. Generally, it takes a large number of box functions to represent each *Pre-BPCA* perfectly. The computation cost term in the objective function prefers a solution with fewer box functions. To make the optimization simpler, computation cost constraint by finding the minimum number of box functions that satisfy was enforced:

$$\frac{1}{N}\sum_{i=1}^{N}\left|\frac{\left(\psi_1^- - \overline{\psi_1^-}\right)}{\left(\psi_1^-\right)_i}\right| \leq \zeta \qquad (11)$$

where $\overline{\psi_1^-}$ is the reconstruction of ψ_1^- using binary box functions. ζ Є[0,1] is the approximation error threshold that controls the precision. *N* is the dimension of the base vector. $(\cdot)_i$ means the *i−th*element of a vector. A smaller value of ζ tends to produce a more accurate approximation. This means the element-wise approximation error ratio should be smaller than a threshold. This constraint is stricter than the norm constraint which requires the difference in norm to be smaller than a threshold[8].

4 PCA- Artificial Neural Network (PCA-ANN)

In Principal Component Analysis - Artificial Neural Network technique, PCA is used during the feature extraction phase since it is found to be the simple and popular technique used for feature extraction. Meanwhile, the ANN based on feed-forward neural networks is used during classification phase because it is one of the machine learning algorithms which is widely used for classification and can perform both linear and non-linear operations [1]. An ANN is an adaptive nonlinear system that learns to perform a function from data. Adaptive means that the system parameters are changed during operation, normally called the training phase. After the training phase the ANN parameters are fixed and the system is deployed to solve the problem at hand (the testing phase). The ANN is built with a systematic step-by-step procedure to optimize a performance criterion or to follow some implicit internal constraint, which is commonly referred to as the learning rule [2].A wide range of models of ANN has been developed for varieties of purposes. They differ in structure implementation and principle of operation but share common features. ANNs are computing systems made up of a number of simple highly interconnected signal or information processing units (Artificial neurons) with the following features [7]. A successful face recognition methodology depends heavily on the particular choice of the features used by the pattern classifier. MLP and many other neural networks learn using an algorithm called back-propagation. Back propagation learning algorithm consists of forward pass and backward pass. Training a network by back-propagation involves three stages: the feed-forward of the input training pattern, the back-propagation of the associated error, and adjustment of the weights. With back-propagation, the input data is repeatedly presented to the neural network. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back (back-propagated) to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "training". The MLP refer to the network consisting of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction, from left to right and on a layer-by-layer basis [6].

4.2 Back Propagation Neural Networks Algorithm

A typical back propagation network with Multi-layer, feed-forward supervised learning is as shown in the Figure. 2. Here learning process in Back propagation requires pairs of input and target vectors. The output vector 'o' is compared with target vector ' t '. In case of difference of 'o' and 't' vectors, the weights are adjusted to minimize the difference. Initially random weights and thresholds are assigned to the network. These weights are updated every iteration in order to minimize the mean square error between the output vector and the target vector [4]. Input for hidden layer is given by

$$net_m = \sum_{z=1}^{n} x_z w_{mz} \qquad (12)$$

The units of output vector of hidden layer after passing through the activation function are given by

$$h_m = \frac{1}{1+\exp(-net_m)} \qquad (13)$$

In same manner, input for output layer is given by

$$net_k = \sum_{z=1}^{m} h_z w_{kz} \qquad (14)$$

and the units of output vector of output layer are given by

$$o_k = \frac{1}{1+\exp(-net_k)} \qquad (15)$$

For updating the weights, we need to calculate the error. This can be done by

$$E = \frac{1}{2}\sum_{i=1}^{k}(o_i - t_i)^2 \qquad (16)$$

$O_i$and t_i represents the real output and target output at neuron *i*in the output layer respectively. If the error is minimum than a predefined limit, training process will stop; otherwise weights need to be updated. For weights between hidden layer and output layer, the change in weights is given by

$$\Delta w_{ij} = \alpha\delta_i h_j \qquad (17)$$

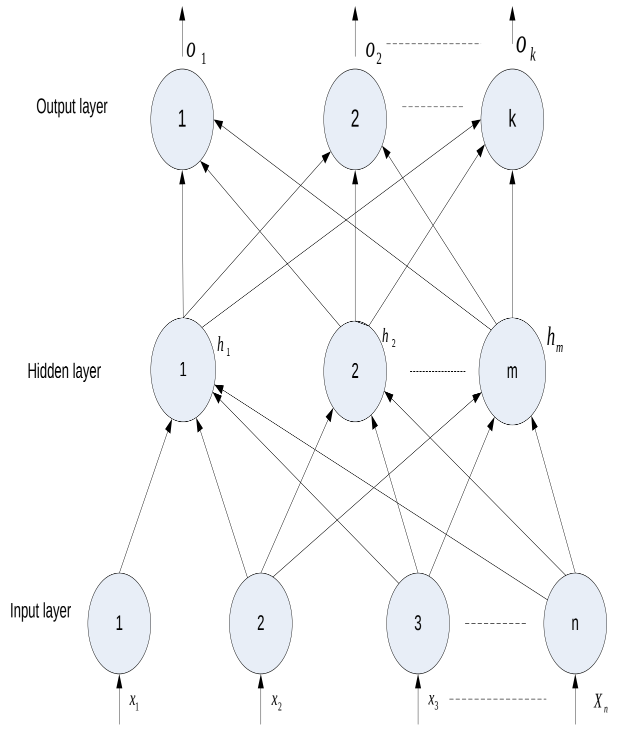

Figure 2. *Basic Block of Back propagation neural network*

where α is a training rate coefficient that is restricted to the range [0.01,1.0], h_{aj} is the output of neuron *j* in the hidden layer, and δ_i can be obtained by

$$\delta_i = (t_i - o_i)o_i(l - o_i) \tag{18}$$

Similarly, the change of the weights between hidden layer and output layer, is given by

$$\Delta w_{ij} = \beta \delta_{Hi} x_j \tag{19}$$

where β is a training rate coefficient that is restricted to the range [0.01,1.0], x_j is the output of neuron *j* in the input layer, and δ_{Hi} can be obtained by

$$\delta_{Hi} = x_i(l - x_i)\sum_{j=1}^{k} \delta_j w_{ij} \tag{20}$$

X_i is the output at neuron *i* in the input layer, and summation term represents the weighted sum of all δ_j values corresponding to neurons in output layer that obtained in equation. After calculating the weight change in all layers, the weights can simply updated by

$$w_{ij}(new) = w_{ij}(old) + \Delta w_{ij} \tag{21}$$

This process is repeated, until the error reaches a minimum value.

5 EXPERIMENTS

A database of 400, 50x50 pixels images consisting of 100 different individuals, each individual having 4 images with different facial expressions was created. The images were captured with a Coolpix s210 digital camera. The images were captured at different times, under different illumination, different facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). Three hundred images were used for training while 100 images were used for testing the three face recognition systems.

5.1 Varying Numbers of Eigenvectors

Experiments were performed on the created database using PCA, BPCA and PCA – ANN separately. The effect of the size of eigenvectors used for training was investigated by using 75, 150 and 300 eigenvectors. Figure 3.1 shows some samples of face images used.

6 Results of the Image Recognition Models

The result showed that PCA, BPCA and PCA-ANN had recognition rates of 91%, 86% and 94% when 75 eigenvectors were selected. When 150 eigenvectors were selected, the recognition rates of the systems were 75%, 70% and 93%. Selecting 300 eigenvectors, the recognition rates were 56%, 55% and 93%. The systems also had recognition time of 5.2 sec, 5.5 sec and 140.5 sec when 75 eigenvectors were selected. Selecting 150 eigenvectors, the recognition time of the systems were 5.5 sec, 5.6 sec and 143.5 sec. The systems had recognition time of 5.1 sec, 5.4 sec and 140.3 sec when 300 eigenvectors were selected.

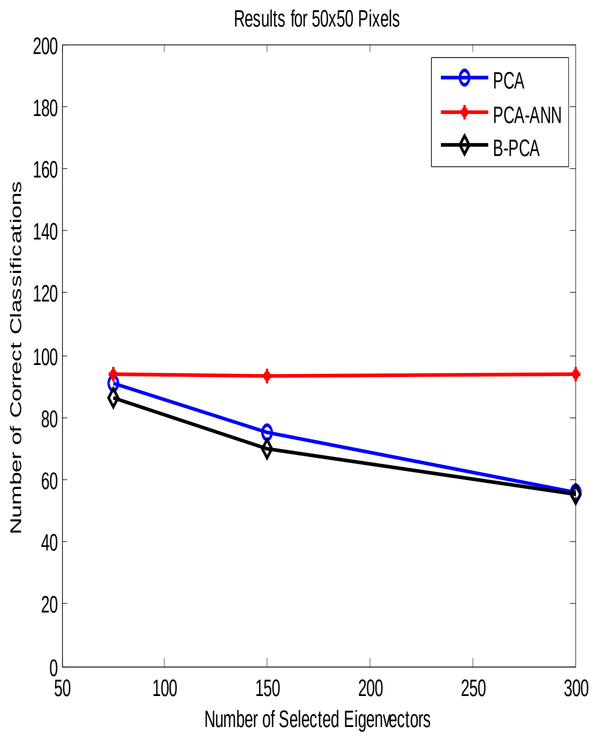

Figure 3. *Number of Correct Classification with selectedEigenvectors for 50x50 Image Resolution*

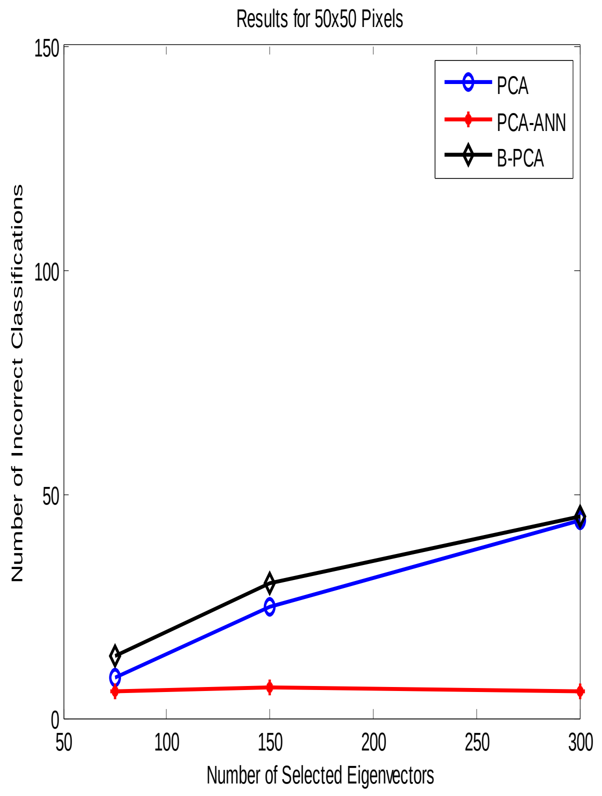

Figure 4. *Number of Incorrect Classification with selectedEigenvectors for 50x50 Image Resolution*

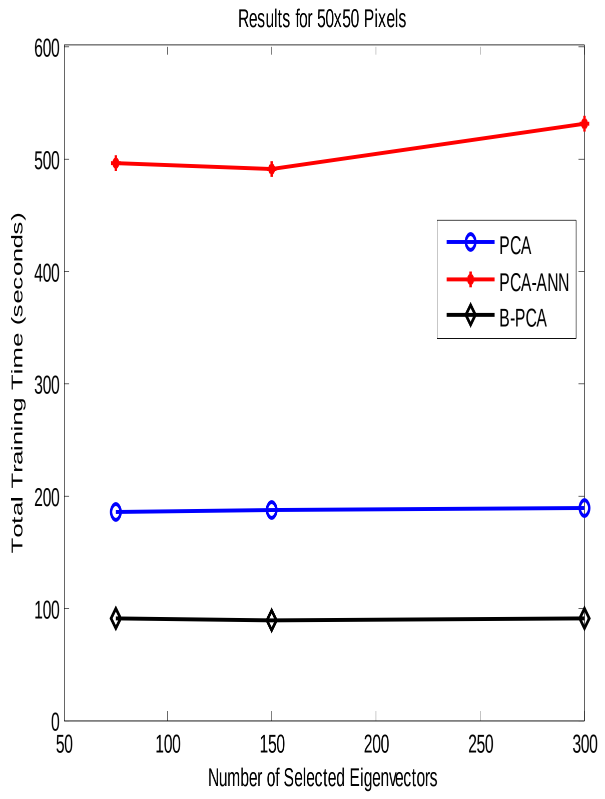

Figure 6.*Total Training Time with selected Eigenvectors for 50x50 Image Resolution*

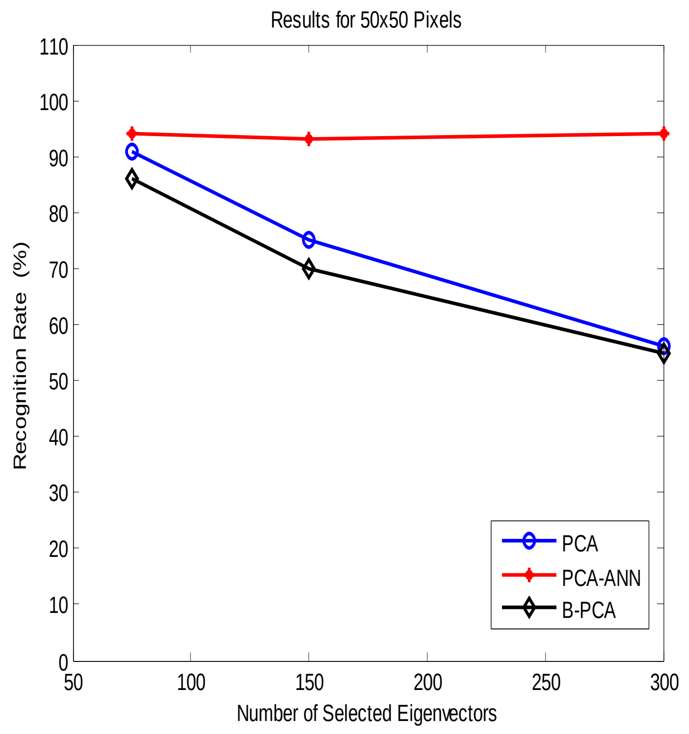

Figure 5. Recognition Rate with selected Eigenvectors for 50x50 Image Resolution

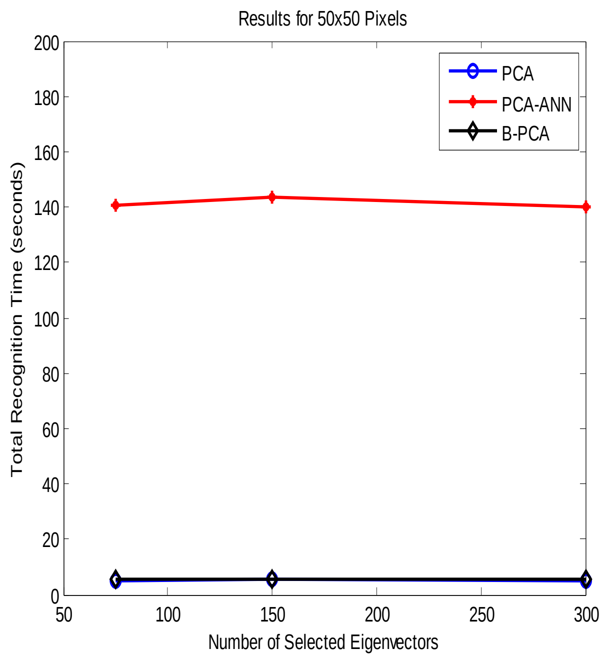

Figure 7.*Total Recognition Time with selected Eigenvectors for 50x50 Image Resolution*

6.1 Observation from the Result

PCA – ANN had the highest recognition rate and virtually constant recognition time when eigenvectors were varied. This could be attributed to the inherent non linearity nature of the ANN to learn sharply the reduced image dimension and correctly classify observed image signal. And also the robustness of the hybrid algorithm and its insensitivity to eigenvector's variation. The recognition rates of PCA and BPCA increased with decreasing size of eigenvectors, this was because there were more relevant and significant components in the reduced eigenvectors that well represented the image signals. Some of the irrelevant features (noise) of the face had been "cut-off" with the eigenvector reduction. The change in recognition rate was negligible in PCA-ANN because of the stable nature or robustness of the neural network that took part in the hybridization process. Also, PCA-ANN had highest accuracy to justify the non-linearity of the different shapes attained by the faces with different expressions which is virtually and highly captured by the neural network.It was observed from the results that, varying eigenvectors had significant effects on the system performance. All the parameters considered were actually affected, as the selected eigenvectors reduced, the recognition rates increased.

7 Conclusion

An evaluation of PCA-based techniques for the classification of images has been presented in this research work. PCA, BPCA and PCA-ANN techniques were evaluated to consider the quantitative effects of varying eigenvectors on recognition rate and recognition time with respect to a single image resolution. The experiment was performed on black facial images under different face views, expression and illumination. The performance evaluation of the three PCA-based systems showed that PCA – ANN technique gave the best recognition rate with a trade-off in recognition time for 50x50 pixels with different size of eigenvectors considered. Also, the recognition rates of PCA and BPCA increased with decreasing number of eigenvectors but PCA-ANN recognition rate was negligible.

REFERENCES

[1] Agarwal, M., Jain, N., Kumar M. and Agrawal H. (2010): "Face Recognition Using Eigen Faces and Artificial Neural Network". International Journal of Computer Theory and Engineering, 2(4): pp. 624-629.

[2] Bevilacqua, V., Mastronardi, G., Pedone, G., Romanazzi, G., and Paleno, D. (2006): "Hidden Markov Model for Recognition Using Artificial Neural networks". Springer-Verlag, Heidelberg, New York, 19(1): pp. 8-9.

[3] Jain K. and Singh S. (2011): "Performance Evaluation of Face Recognition Using PCA". International Journal of Information Technology and Knowledge Management, 4(2): pp. 427-430

[4] Latha, P., Dr. Ganesan L. and Dr. Annadurai S.: "Face Recognition Using Neural Networks". An International Journal (SPIJ) 3(5): pp. 155-157

[5] Martinez, A.M. and Kak, A.C. (2001): "PCA versus LDA". IEEE Trans. on Pattern Analysis and Machine Intelligence, 23(2): pp. 228-233.

[6] Nazeer S.A and Khalid M. (2009): "PCA-ANN Face Recognition System based on Photometric Normalization Techniques". ISBN-3-902613-42-4, pp. 250, I-Tech, Vienna, Austria.

[7] Omidiora E.O., Fakolujo O.A., Ayeni R.O., Olabiyisi S.O., and Arulogun O.T. (2008): "Quantitative Evaluation of Principal Component Analysis and Fisher Discriminant Analysis Techniques in Face images". Journal of Computer and its Applications, 15(1): pp. 22-37.

[8] Tang F. and Tao H. (2007): "Representing Images Using Non-orthogonal Haar-Like Bases". IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(12): pp.2120-2133.

[9] Tao H., Crabb R., and Tang F. (2005): "Non-Orthogonal Binary Subspace and its Applications in Computer Vision". In ICCV, pp. 864–870.

[10] Turk M.A. and Petland A.P. (1991): "Eigenfaces for Recognition". Journal of Cognitive Neuroscience. 3(1): pp.71-86.

[11] Wilson, P.I. and Fernandez, J. (2006): "Facial Feature Detection Using Haar Classifiers". J. Comput. Small Coll. 21(4): pp. 127-133.