*International Journal of Research and Reviews in Applicable Mathematics & Computer Science,*
*Vol. 1, No.4, 2011*

# An Optimized Approach for Face Recognition with Fast Discrete Cosine Transform and Robust RBF Neural Networks

Kazi Md. Rokibul Alam[1], Bikash Chandra Karmokar[2] and Md. Kibria Siddiquee[3]

Department of Computer Science and Engineering
Khulna University of Engineering and Technology
Khulna–9203, Bangladesh
Email: [1]rokibcse@yahoo.com, [2]bikash_kuet@yahoo.com, [3]shakil_2k7@yahoo.com

## *Abstract*

***This paper presents a modified approach for high speed face recognition with robust radial basis function (RBF) neural network and fast discrete cosine transform (FDCT). This approach is computationally inexpensive so it is fast enough and its important feature is that it works correctly with small training samples. In this approach we have proposed the system in which every step is specified. In the presentation we at first do preprocess the input image with FDCT which is computationally inexpensive but high data packing capability. Then the FDCT coefficient vectors are clustered with our proposed clustering algorithm. Then we have applied the Fisher's Linear Discriminant [4] Analysis to make the system tolerant for illumination variation. At the end of preprocessing, a booster is used to work better in case of small training samples. At the last step we feed the data to an exact robust RBF neural network to perform the recognition task. The network has the ability to train with small samples with a high speed learning rate. A high speed learning algorithm is also specified for this task. In every step we try to minimize the computation as a result our proposed model is seen quite speedy, noise tolerant and accurate.***

***Keywords:*** *Fast discrete cosine transform (FDCT), UMIST database, Fisher's linear discriminant (FLD), illumination invariance, robust RBF neural networks.*

## I. INTRODUCTION

Due to increasing security demand human face recognition has become a very active research area in recent years and its potential commercial and law enforcement applications. Numerous approaches have been proposed for face recognition and considerable successes have been reported. However, it is still a difficult task for a machine to recognize human faces accurately in real-time, especially under variable circumstances such as variations in illumination, pose, facial expression, makeup, etc.

Generally speaking, research on face recognition can be grouped into two categories. Namely, feature-based approach and holistic approach. Feature- based approaches are based on the shapes and geometrical relationships of individual facial features including eyes, mouth, nose, and chin. On the other hand, holistic approaches handle the input face images globally and extract important facial features based on the high-dimensional intensity values of face images automatically. However, the computational requirements of old approaches are greatly related to the dimensionality of the original data and the number of training samples. When the face database becomes larger, the time for training and the memory requirement will significantly increase. Moreover, the system based on the PCA should be retrained when new classes are added. As a consequence, it is impractical to apply the PCA in systems with a large database. The discrete cosine transform (DCT) has been employed in face recognition [7], [8].

The outlined is as follows. Section II describes current high speed face recognition techniques. Section III mentions our proposed modified and optimized approach for high speed face recognition. This section also contains all the needed algorithms for the scheme Experimental results are presented and discussed in Section IV. Finally, conclusions are explained in Section V.

## II. EXISTING WORKS

High information redundancy and correlation in face images result in inefficiencies when such images are used directly for recognition. Discrete cosine transforms are used to reduce image information redundancy because only a subset of the transform coefficients are necessary to preserve the most important facial features such as hair outline, eyes and mouth. When DCT coefficients are fed into a radial basis function neural network for classification, a high recognition rate can be achieved by using a very small proportion of transform coefficients. This makes DCT-based face recognition much faster than other approaches. This approach is first introduced by Zhengjun Pan and Hamid Bolouri in the year 1999 [1]. Then in the year 2002 face recognition with RBF neural network is done successfully by Meng Joo Er. , Shiqian Wu, Jewei Lu and Hock Lye Toh [2]. In the year 2005 discrete cosine transform is added with the previous idea by Meng Joo Er., Weilong Chen and Shiqian Wu [3]. But these ideas lack for specified algorithms and optimized performance.

The DCT has several advantages over the PCA. First, the DCT is data independent. Second, the DCT can be implemented using a fast and computationally inexpensive version of DCT called Fast Discrete Cosine Transform (FDCT). The FDCT is applied for dimensionality reduction and then the selected low-frequency DCT coefficient vectors are fed into a multilayer perceptron (MLP) classifier. It is well-known that the problems arising from the curse of dimensionality should be considered in pattern recognition. It has been suggested that as the dimensionality increases, the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities. In face recognition applications, the original input data are usually of high dimension, whereas, only limited training samples are available. Therefore, dimensionality reduction is a very important step which will greatly improve the performance of the face recognition system. However, if only the FDCT is employed for dimensionality reduction, we cannot keep enough frequency components for important facial features in order to compromise the problem of curse of dimensionality. Besides, some variable features exist in the low-dimensional features extracted by the FDCT. So, the FLD is also employed in the FDCT domain to extract the most discriminating features of face images. With the combination of DCT and FLD, more DCT coefficients can be kept and the most discriminating features can be extracted at high speed. In [2], the clustering process is implemented after the FLD is employed. The FLD is a linear projection method and the results are globally optimal only for linearly separable data. There are lots of nonlinear variations in human faces such as pose and expression. Once the faces with different poses are put into the same class, they will actually smear the optimal projection such that we cannot get the most discriminating feature and good generalization. For this reason, we should ensure that the face images in each class should not have large nonlinear variations.

## II. PROPOSED OPTIMIZED APPROACH

In this paper, instead of regarding each individual as one class, we used a sub clustering method to split one class into several subclasses before implementing the FLD. Moreover, the sub clustering process is very crucial to structure determination of the following radial basis function (RBF) neural networks. In fact, the number of clusters is just the number of hidden neurons in the RBF neural networks. Neural networks have been widely applied in pattern recognition for the reason that neural-networks-based classifiers can incorporate both statistical and structural information and achieve better performance than the simple minimum distance classifier. Multilayered networks (MLNs), usually employing the back propagation (BP) algorithm, are widely used in face recognition. Recently, RBF neural networks have been applied in many engineering and scientific applications including face recognition. The RBF neural networks possess the following salient features:

1) They are universal approximators,
2) They have a simple topological structure,
3) They can implement fast learning algorithms because of locally tuned neurons.

Based on the advantages of RBF neural networks and the efficient feature extraction method, a high-speed RBF neural networks classifier whereby near-optimal parameters can be estimated according to the properties of the feature space instead of using the gradient descent training algorithm is proposed in this thesis. As a consequence, our system is able to achieve high training and recognition speed which facilitates real-time applications of the proposed face recognition system. The block diagram of our proposed face recognition system is shown in Figure 1.

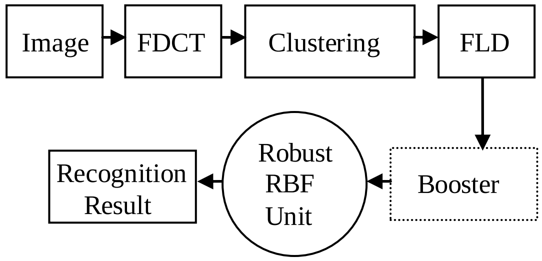


**Figure 1:** Block diagram of proposed optimized face recognition system with FDCT and robust RBF neural network.

### A. Fast Discrete Cosine Transform (FDCT) Algorithm

The N-point DCT is

$$X(n) = \frac{2}{N} e(n) \sum_{k=0}^{N-1} x(k) \cos\left[\frac{(2k+1)n\pi}{2N}\right] \quad \text{for } n = 0,1,\ldots,N-1$$

Its inverse (IDCT) is given by

$$x(k) = \sum_{n=0}^{N-1} e(n) X(n) \cos\left[\frac{(2k+1)n\pi}{2N}\right] \text{ for k=1,2,\ldots N-1}$$

Where $e(0) = 1/\sqrt{2}$ and $e(n) = 1$ otherwise, and define C such that

$$C_{2N}^{(2k+1)n} = \cos\left[\frac{(2k+1)n\pi}{2N}\right]$$

Then the N-point IDCT becomes

$$x(k) = \sum_{n=0}^{N-1} \hat{X}(n) C_{2N}^{(2k+1)n} \qquad \text{where } \hat{X}(n) = e(n) X(n)$$

Decomposing x(k) into even and odd indexes of n (assuming that N is even)

$$x(k) = g(k) + h^{'}(k)$$

$$x(N-1-k) = g(k) - h^{'}(k), \qquad \text{for } k = 0,1,\ldots,\frac{N}{2}-1$$

where

$$g(k) = \sum_{n=0}^{\frac{N}{2}-1} \hat{X}(2n) C_{2N}^{(2k+1)2n}$$

$$h^{'}(k) = \sum_{n=0}^{\frac{N}{2}-1} \hat{X}(2n+1) C_{2N}^{(2k+1)(2n+1)}$$

since

$$C_{2N}^{(2k+1)2n} = C_{N}^{(2k+1)n} = C_{2(N/2)}^{(2k+1)n}$$

We rewrite g(k) in the form

$$g(k)=\sum_{n=0}^{\frac{N}{2}-1}\hat{X}(2n)C_{2(N/2)}^{(2k+1)n}$$

since

$$2C_{2N}^{(2k+1)}C_{2N}^{(2k+1)(2n+1)}=C_{2N}^{(2k+1)2n}+C_{2N}^{(2k+1)2(n+1)}$$

we have

$$2C_{2N}^{(2k+1)}h^{'}(k)=\sum_{n=0}^{\frac{N}{2}-1}\hat{X}(2n+1)C_{2N}^{(2k+1)2n}+\sum_{n=0}^{\frac{N}{2}-1}\hat{X}(2n+1)C_{2N}^{(2k+1)2(n+1)}$$

so, if we define

$$\hat{X}(2n-1)|_{n=0}=0$$

then

$$\sum_{n=0}^{\frac{N}{2}-1}\hat{X}(2n+1)C_{2N}^{(2k+1)2(n+1)}=\sum_{n=0}^{\frac{N}{2}-1}\hat{X}(2n-1)C_{2N}^{(2k+1)2n}$$

because

$$C_{2N}^{(2k+1)2(N/2)}=C_{2}^{(2k+1)}=0$$

So

$$2C_{2N}^{(2k+1)}h^{'}(k)=\sum_{n=0}^{\frac{N}{2}-1}\left[\hat{X}(2n+1)+\hat{X}(2n-1)\right]C_{2N}^{(2k+1)2n}$$

Now we define

$$G(n)=\hat{X}(2n)$$

$$H(n)=\hat{X}(2n+1)+\hat{X}(2n-1)\qquad \text{for } n=0,1,\ldots,\frac{N}{2}-1$$

and

$$g(k)=\sum_{n=0}^{\frac{N}{2}-1}G(n)C_{2(N/2)}^{(2k+1)n}\qquad \text{for } k=0,1,\ldots,\frac{N}{2}-1$$

Finally we get

$$x(k)=g(k)+\left[\frac{1}{2C_{2N}^{(2k+1)}}\right]h(k)\quad x(N-1-k)=g(k)-\left[\frac{1}{2C_{2N}^{(2k+1)}}\right]h(k)\qquad \text{for } \mathrm{k}=0,1,\ldots,\frac{\mathrm{N}}{2}-1$$

so the final equation becomes

$$h^{'}(k)=\frac{1}{2C_{2N}^{(2k+1)}}\sum_{n=0}^{\frac{N}{2}-1}\left[\hat{X}(2n+1)+\hat{X}(2n-1)\right]C_{2N}^{(2k+1)2n}$$

**B. Clustering Algorithm**

Step 1: Set each class to be one cluster. Let u and s be number of total clusters and classes respectively. So, u=s.

Step 2: Find two training samples $x^k(f), x^k(g)$ with the largest Euclidean distance $d^k(f.g)$ in class k, k =1, 2, ……, s. These two samples are called clustering reference sample (CRS).

Step 3: Compute the Euclidean distance from the two samples to the samples $x^j(i)$ in other classes j, j=1, 2 ,…..,s, $j \neq k$, denoted as $d^{kj}(f,i)$ and $d^{kj}(g,i)$, i =1,2,……, $n^{\bar{k}}$. Where $n^{\bar{k}}$ is the number of samples not belonging to class k, respectively as follows:

$$d^{kj}(f,i) = \left\| x^j(i) - x^k(f) \right\|, \qquad j \neq k$$

$$d^{kj}(g,i) = \left\| x^j(i) - x^k(g) \right\|, \qquad j \neq k$$

where ||.|| is the Euclidean norm.

Step 4: Compute the mean value and standard deviation of $d^{kj}(f,i)\, d^{kj}(g,i)$ as follows :

$$\bar{d}^k(f) = \frac{1}{n^{\bar{k}}} \sum_{i=1}^{n^{\bar{k}}} d^{kj}(f,i) \qquad v^k(f) = \left[ \frac{1}{n^{\bar{k}}} \sum_{i=1}^{n^{\bar{k}}} \left( d^{kj}(f,i) - \bar{d^k}(f) \right)^2 \right]^{\frac{1}{2}} \qquad v^k(f) = \left[ \frac{1}{n^{\bar{k}}} \sum_{i=1}^{n^{\bar{k}}} \left( d^{kj}(f,i) - \bar{d^k}(f) \right)^2 \right]^{\frac{1}{2}}$$

$$\bar{d}^k(g) = \frac{1}{n^{\bar{k}}} \sum_{i=1}^{n^{\bar{k}}} d^{kj}(g,i)$$

$$v^k(g) = \left[ \frac{1}{n^{\bar{k}}} \sum_{i=1}^{n^{\bar{k}}} \left( d^{kj}(g,i) - \bar{d^k}(g) \right)^2 \right]^{\frac{1}{2}}$$

Step 5: Define the scope radius of the two CRSes $r^k(f), r^k(g)$ as follows:

$$r^k(f) = \bar{d^k(f)} - \alpha v^k(f)$$

$$r^k(g) = \bar{d^k(g)} - \alpha v^k(g)$$

where α is a positive constant clustering factor. If $d^k(f.g)$>max($r^k(g), r^k(f)$), then split the cluster into two clusters with CRSes $x^k(f)$ and .$x^k(g)$ respectively, and set u=u+1.

Step 6: There are three scenarios for any sample $x^k(h)$ in class k ($x^k(h)$ is not CRS). These scenarios as depicted in figure 3.3 will be handled as follows:

i) If only one CRS's scope comprises $x^k(h)$, then $x^k(h)$ will be merged with the cluster to which this CRS belongs.

ii) If more than one CRS's scope comprise $x^k(h)$, then $x^k(h)$ will be merged into the cluster to which the CRS with the shortest distance to $x^k(h)$ belongs.

iii) If no CRS's scope comprise $x^k(h)$, then $x^k(h)$ is regarded as another CRS belonging to a new cluster. Set u=u+1 and compute the radius $r^k(h)$ according to steps (4)–(5). Repeat step (6), until u dose not change.

Step 7: Apply steps (2)–(6) to all classes.

**C. The Boosting Algorithm**

Boosting, proposed by Freund and Schapire [6], is a technique to combine weak classifiers, having a poor performance, in a strong classification rule with a better performance. In boosting, classifiers and training sets are obtained sequentially, in a strictly deterministic way. At each step, training data are reweighted in such way that incorrectly classified objects get larger weights in a new modified training set. By that, one actually maximizes margins between training objects. It suggests the connection between boosting and Vapnik's Support Vector Classifier (SVC), as objects obtaining large weights may be the same as the support objects. Boosting is organized by us in the following way.

Step 1: Repeat for b=1,2,...,B.

a)Construct classifier $c^b(x^*)$ on the weighted version $x^* = (w_1^b x_1, w_2^b x_2, ...., w_n^b x_n)$ of training data set $x = (x_1, x_2, ...., x_n)$, using weights $w_i^b$, i=1,2,….n ( $w_i^b$ =1 for b=1).

b) Compute probability estimates of the error $err_b = \frac{1}{n}\sum_{i=1}^{n} w_i^b \xi_i^b$

$$\xi_i^b = \begin{cases} o, & \text{If } x_i \text{ is classified correctly} \\ 1, & \text{Otherwise} \end{cases}$$

and $c_b = \frac{1}{2}\log(\frac{1-err_b}{err_b})$

c)If $0 < err_b < .5$ set $w_i^{b+1} = w_i^b \exp(-c_b \xi_i^b)$ $i$=1,...,$n$, and renormalize so that $\sum_{i=1}^{n} w_i^{b+1} = n$ .Otherwise, set all weights $w_i^b = 1$, $i$=1,...,$n$, and restart.

Step 2: Combine classifiers $c^b(x^*)$ by the weighted majority vote with weights $c_b$ to a final decision rule.

**D. Algorithm for Implementing Robust RBF Network**

In this section, the algorithm of implementing the SRBF network with a robust objective function and proposed adaptive growing technique is presented. First of all, we introduce the following useful notations which will be used in the description of the algorithm.

1) Check period T: if the number of iterations between consecutive updates of the objective function is a multiple of the period, we should check the state of the network.

2) Objective function $E_R(i)$: the value of the objective functions in the ith iteration.

3) Threshold $\partial_{en\_upper}$: if the difference between the current and previous values of the objective function is larger than the threshold, it implies that the SRBF network moves more closer to underlying function; hence the confidence interval of the residual is reduced.

4) Threshold $\partial_{en\_lower}$: if the difference between the current and previous values of objective function is smaller than the threshold, it means that the number of nodes of network is insufficient in approximating the underlying function; hence a new node needs to be added to the network

5) Threshold $\partial_{weight}$:Nodes with weights smaller than this threshold will be deleted.

6) Threshold $\partial_{stop}$:criterion of terminating the learning process. With all necessary notations defined, we are ready to describe the algorithm as follows.

Step 1: Set up the network initial conditions.

a) Select an initial number of nodes for the network.

b) Set initial parameter values of each node.

c) Set $E_s = \in$ to a small value, where $E_s$ is used for recording the value of the objective function.

d) Set i=1 and initialize the check period T.

Step 2**:** Construct the robust objective function.

a) Select proper s(r) and t(r).

b) Compute extreme of y(r)=s(r)t(r) by solving the equation of $y'(r)=0$ The extreme positions represent the two cutoff points. Let this two cutoff points be $\pm s$

c) Let [-s, s] be the initial confidence interval of the residual.

d) Compute the corresponding robust objective function $E_R(r_p)$ by integrating y(r)

Step 3: Compute output of the network for all training patterns.

Step 4: Compute the value of robust objective function of the network.

Step 5: If the iteration number of i is a multiple of T adjust the size of the network and the confidence interval of the residuals based on the following procedure,

a) If $|E_R(i)-E_s| > \delta_{en\_upper}$(objective function changes rapidly), then reduce the confidence interval of the residual by finding new cutoff points

$$\sigma_{new} = \sigma_{old} * r_d \quad \text{if } \sigma > \sigma_L$$
$$= \sigma_{old} \quad \text{otherwise}$$

where is the decreasing rate and $\sigma_L$ is the lower bound of the confidence interval of residual.

b) If $|E_R(i)-E_s| < \delta_{en\_lower}$ (objective function does not change rapidly), then add a new node to the network by the adaptive growing technique with a memory queue. c) $\sigma_{weight}$ Check the outgoing weight of each node and remove those nodes with outgoing weights smaller than prescribed threshold

d). Store the current objective function $E_R(i)$ to $E_s$

Step 6: If $E_r(i) < \delta_{stop}$stop then terminate the learning procedure; otherwise, go to step no 7.

Step 7: Update parameters of the network and set i=i+1 go to step 3.

## IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

In order to evaluate the proposed face recognition system, our experiments are performed on one benchmark face database: the UMIST face database consists of 564 images of 20 people. Each covering a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. We have implemented our proposed technique in MATLAB 7.

### A. Speed Comparison between DCT and FDCT

We use uniform distribution and normal distribution for generating artificial pixel data (grey level intensity ranging form 0-255) for image of different size starting from 1x1, 2x2, 3x3,……1000x1000. These entire artificial images were fed to the DCT and FDCT functions and we compute time for processing each image. From the figure 2 and figure 3 we can see that for every case the time curve of FDCT lies behind the DCT line which means the FDCT is much faster than the traditional DCT. From those figures we can also see that there are some peaks which are caused by the other running programs in computers. The expected line is linear. Actually the DCT and FDCT calculation time are very small. So to make the comparison clearly visible we multiplied the times with 100.

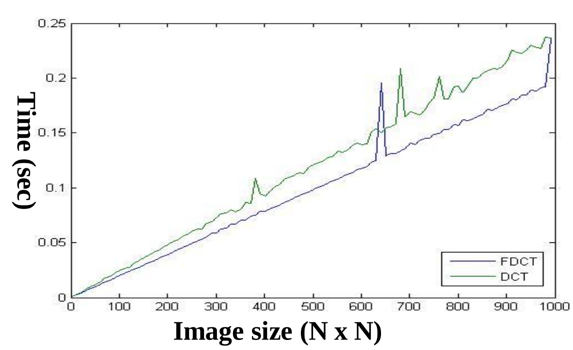

**Figure 2:** DCT vs FDCT (Gaussian distribution)

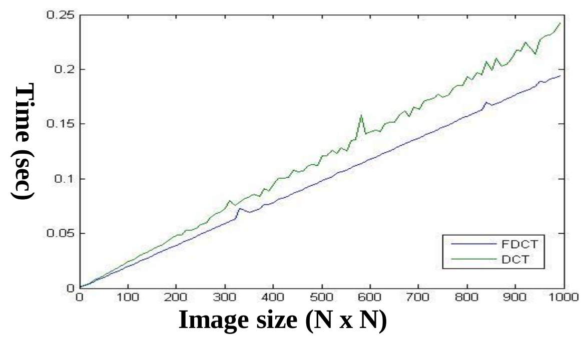


**Figure 3:** DCT vs FDCT (Normal distribution)

**B. Consistency of the Proposed System**

We use UMIST face database to test the consistency of our proposed system. The UMIST Face Database We randomly chose different sample image for each person and 8 simulation runs and averaging the times we found the following graph (Figure 4). In this graph our proposed system is trained with very small sample (1,2,3…5) per person and as booster is used there were no errors to identify the person with the test images. We can see that if the system is provided with only one image per person the recognition rate is quite higher but if the system is trained with two images per person the rate goes to quite constant. One remarkable point is that if the system is trained with only one image per person we have quite a tolerable and speedy recognition rate.

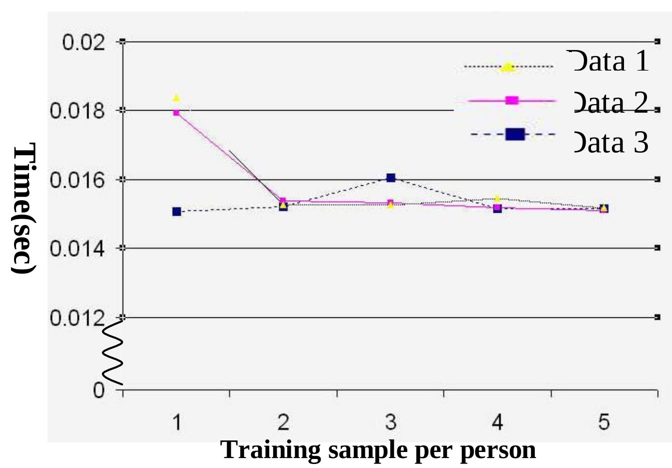


**Figure 4:** Consistency graph for dataset 1,2 and 3

### C. Comparison with Other approaches

We worked with the latest face recognition system based on DCT and traditional three layered RBF neural network. From our simulation and we summarize the Table I to have
the recognition rate comparisons with different face recognition approaches and our proposed optimized face recognition approach.

**Table I:** Recognition performance comparison of different approaches

| Approach | Error rate (%) (best) | Recognition time | Platform |
|---|---|---|---|
| DCT + HMM [20] | 0.5 | 3.5 sec | Pentium 200M |
| Pseudo-2D HMM+DCT [21] | 0.0 | 1.5 sec | Pentium 400M |
| DCT + MLP [7] | 3.0 | 0.0002 sec | IBM 450M |
| FND [23] | n/a | 3.23 sec | Pentium II 400M |
| NFA-II+NFL [24] | n/a | 0.16 sec | IBM 1700M |
| RBF+DCT+FLD [2] | 0.0 | 0.055 sec | Pentium II 350M |
| **Our proposed system** | **0.0** | **0.015676 sec** | **Pentium III 933M** |

From this table we can see that our approach is quite faster than some recent face recognition approaches. But its most important feature is that it works fast and accurate for small sample size which is absent all other techniques.

## V. CONCLUSION

Today we live in the age where security is a high demand. But the security system must provide consistency, accuracy, memory efficiency and time efficiency. Our proposed model is quite fast and can deals with both small and large number of training samples so it can meet the current security demand. In this paper we try to give specification for all the steps for our proposed method for face recognition which may be helpful for implementing a high speed face recognition system with higher accuracy level.

**REFERENCES**


[1] [Z. Pan, R. Adams, and H. Bolouri, "*Image redundancy reduction for neural network classification using discrete cosine transforms*," in Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Networks, vol. 3, Como, Italy,2000, pp. 149–154.

[2] Z. M. Hafed and M. D. Levine, "*Face recognition using the discrete cosine transform,*" Int. J. Computer Vision, vol. 43, no. 3, pp. 167–188, 2001.

[3] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "*Face recognition with radial basis function (RBF) neural networks,*" IEEE Trans. Neural Networks, vol. 13, no. 3, pp. 697–710, May 2002.

[4] Meng. Joo Er, Weilong Chen and Shiqian Wu, "*High speed Face recognition based on discrete cosine and RBF neural networks,*" IEEE Trans. Neural Networks, vol. 16, no. 3, pp. 697–710, May 2005.

[5] Freund, Y., Schapire, R.E.: "*Experiments with a New Boosting Algorithm*". In: Machine Learning: Proceedings of the Thirteenth International Conference, pp:148-156, 1996.

[6] Schapire, R.E., Freund, Y., Bartlett, P., Lee, "*Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods*". The Annals of Statistics, pp: 1651-1686, 1998.

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

[8] http://images.ee.umist.ac.uk/danny/face.tar.gz

[9] V. V. Kohir and U. B. Desai, "*Face recognition using a DCT-HMM approach,*" in Proc. IEEE Workshop on Applications of Computer Vision (WACV'98), Princeton, NJ, 1998, pp. 226–231.

[10] S. Eickeler, S. Müller, and G. Rigoll, "*High quality face recognition in JPEG compressed images,*" in Proc. IEEE Int. Conf. Image Processing (ICIP), Kobe, Japan, 1999, pp. 672–676.

[11] T. Tan and H. Yan, "*Object recognition based on fractal neighbor distance,*" Signal Process., vol. 81, pp. 2105–2129, 2001.

[12] Y.-S. Ryu and S.-Y. Oh, "*Simple hybrid classifier for face recognition with adaptively generated virtual data,*" Pattern Recognit. Lett., vol. 23, pp. 833–841, 2002.

## Authors Profile

**Dr. Kazi Md. Rokibul Alam** received his Ph.D degree from University of Fukui, Japan in 2010. Currently he is a faculty member in the department of Computer Science and Engineering (CSE) of Khulna University of Engineering and Technology (KUET), Bangladesh.

**Bikash Chandra Karmokar** is a final year student of Computer Science and Engineering (CSE) of Khulna University of Engineering and Technology (KUET), Bangladesh. He completed his H.S.C from Notre Dame College, Dhaka. His major research Interests are in Artificial Intelligence, Neural Network, Natural Language Processing, Genetic Algorithm, Mobile Computing and Web-Engineering.

**Md. Kibria Siddiquee** is a final year student of Computer Science and Engineering (CSE) of Khulna University of Engineering and Technology (KUET), Bangladesh. He completed his H.S.C from Government Science College, Dhaka. His major research Interests are in Artificial Intelligence, Neural Network, Natural Language Processing, Genetic Algorithm, Mobile Computing and Web-Engineering.