International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.12 (2015), pp.331-340
http://dx.doi.org/10.14257/ijmue.2015.10.12.32

Face Detection and Recognition Technology for HCI based on RBF Neural Network

Ning Zhang and Eung-Joo Lee

Department of Information & Communications Engineering,
Tongmyong University, Busan, Korea
ejlee@tu.ac.kr

Abstract

In this paper, feature extraction and facial recognition are studied in order to resolve problems like high-dimension problem, small size samples and no-linear separable problem that exist in facial recognition technology. In the part of feature extraction we use a Discrete Cosine Transform (DCT) algorithm, to extract the input features in building a face recognition system. The RBF neural network, which represents brilliant performance on small training sets, non-linear separable and high-dimension pattern recognition problems in the recognition stage, is used for pattern classification. The proposed approach is validated with the ORL database. Experimental results demonstrate the effectiveness of this method in the performance of face recognition.

Keywords: *Face Detection, DCT, Face Recognition, RBF*

1. Introduction

Neural network is an interdisciplinary, structure that works by simulating human brain networks to construct a model similar to the human brain. Among the hundreds of networks formed and algorithm models that had been proposed from its birth to the present, the BP neural network is the most mature and widely used algorithm. Face recognition has become a hot topic in digital image processing and pattern recognition. As one of the several most important external characteristics in recognition, human face recognition has important and wide theoretical and practical applications. The complexity of face recognition is that facial features continuously change depending on external and internal conditions. Despite these changes, humans are still able to recognize facial features in different environments, but machines do not have this capability. This inspired us to mimic the human brain structure and thinking model, to use artificial neural network to solve the face recognition problems.

During the past 30 years, many different face recognition method have been found. "Eigenspace" was one of those proposed under the PCA method [1, 2]. The PCA approach reduces the dimension of the data by means of basic data compression methods [3] and reveals the most effective low dimensional structure of facial patterns [4]. Recognition by Neural Network [5] and [6] are based on learning the faces in an "Example Set" by the machine in the “Training Phase” and carrying out recognition in the "Generalization Phase". The Support Vector Machines (SVM) technique, on the other hand one of the binary classification methods. The support vectors consist of a small subset of training data extracted by the algorithm given in [7]. Face recognition based on template matching represents a face in terms of a template consisting of several enclosing masks, and projecting features like the mouth, eyes, and nose [8]. In [9], a face detection method based on half face-template is discussed.

In this paper, we use a DCT method given in [10], to extract features. We first reduce the dimension using DCT. In the part of face recognition, we use the Radial Basis Function (RBF) neural network to take the place of BP neural network in order to play its

ISSN: 1975-0080 IJMUE

advantage of best global approximation, faster convergence speed, well classification capability, *etc*.

2. Feature Extraction in DCT Domain

DCT own Fast Fourier Transform algorithm (FFT), as great speed advantage than K-L transform. We implement the original face image with DCT based on the theory in [11], before the extraction of facial features. DCT is mainly used to compress data or images. It is possible to convert its signal to a spatial frequency domain, and it has good decorrelation performance. The basis of DCT is decomposed into an 8x8 or 16x16 image subinterval, and each subinterval separate DCT transforms, then quantizes and codes transform results. With the size of subinterval increasing, there comes a sharp rise in the complexity of the algorithm, therefore, the practical 8x8 subinterval is commonly used to transform. However, the use of larger subinterval can significantly reduce the effect of image block.

One-dimensional DCT transform is the basis of two-dimensional DCT transform.

$$F(u) = C(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi}{2N}(2x+1)u\right]$$

$$C(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{cases} \tag{1}$$

f(x) is the original signal, F(u) is the coefficients after DCT transformation, N is the point of original signal, C(u) can be considered as a compensation factor, it can make DCT transform matrix become orthogonal matrix.

Two-dimensional DCT transformation is based on the one-dimensional DCT transform to do a DCT transform.

$$F(u, v) = C(u)C(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x, y)\cos\left[\frac{\pi}{2N}(2x+1)u\right]\cos\left[\frac{\pi}{2N}(2y+1)v\right]$$

$$C(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \\ \sqrt{\frac{2}{N}}, & u \neq 0 \end{cases} \tag{2}$$

Two-dimensional image data should be square; if the data are not square, we usually transform after we make the data into a square. After rebuilding, the padded part can be removed to get the original image information. If the original information is the large correlation data such as images or other, large coefficient focus on the upper left corner of square, and almost all of the lower right corner is 0, wherein the upper left corner is a low-frequency component, then the lower right corner is a high-frequency component. Low-frequency coefficients reflect the characteristics of the image contours and intensity distribution in the target, while high-frequency coefficients reflect the details of the target shape, after the DCT transformation, energy is mainly concentrated in the low-frequency component; it is also reflecting the decorrelation of DCT transform.

3. RBF Neural Network Algorithm for Face Recognition

3.1. RBF Neural Network Structure

The RBF neural network structure showed in Figure 1 is similar to a 3-layer perceptron, which has a 3-layer feedforward structure: input layer, hidden layer, output layer. However, the RBF neural network's hidden layer is different from multi-layer perceptron neural network. The RBF neural network has a better performance when solving complex problems of small sample training sets and, classification of nonlinear separables.

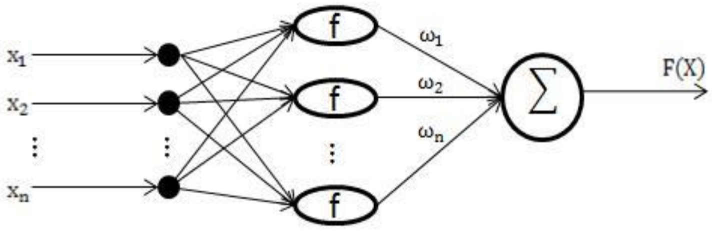

Figure 1. RBF Neural Network Structure

3.2. Radical Basis Function

We define X as the input vector of r-dimension, $X = [x_1, x_2 \dots x_\gamma] \in R'$, the center of i-th RBF neuron is C_i, the Gaussian width is σ_i, so the output of every hidden layer:

$$R_i(X) = R_i(\|X - C_i\|) \tag{3}$$

Function (3) is the general form of Radical Basis Function, the parameter is the Euclidean distance from vector X to center. In this paper, we use the Gauss function as basis function:

$$R_i(X) = \exp\left(-\frac{\|X-C_i\|^2}{\sigma_i}\right) \tag{4}$$

Gaussian function has the advantage of having a simple form, good analytical and any order derivative [12]. The Gaussian function partial response to the input signal is, if the input signal is closer to the center of a hidden layer, output layer unit would get larger.

3.3. RBF Neural Network Designing

We define the number of input layer and output layer neuron as r and s.

(1) Initialize the number of neurons in the hidden layer as the number of model categories: u=s

(2) Calculate the mean of various types of samples, define the mean as the initial center:

$$C_k = \frac{1}{n_k}\sum_{i=1}^{n_k} p_i^k, \quad k = 1,2,\dots,u \tag{5}$$

n_k is the total of k-class sample; p_i^k is the i-th k-class sample.

(3) Find out the point the farthest from k-class center in the k-class sample, define the point , and remember this distance:

$$d_k = \left\| P^k(f) - C_k \right\| \tag{6}$$

(4) Calculate the Euclidean distance among the center of k-class and the other classes, find out the shortest distance and remember this distance:

$$\begin{gathered} dc(k,j) = \left\| C_k - C_j \right\| \\ d_{min}(k,l) = \underbrace{\arg}_{l} \min\big(dc(k,l)\big) \\ j = 1,2,\ldots,u,\ \ j \neq k \end{gathered} \tag{7}$$

(5) If $d_k + d_l > d_{min}(k,l)$ and $d_k - d_l < d_{min}(k,l)$, k-class needs to divide into two classes.

(6) Repeat the step (2) to (5) until all the classes do not need to divide.

3.4. RBF Neural Network Hybrid Learning Algorithm

The parameter of RBF neural network hidden layer node is a nonlinear process, while the adjust of output layer connection weights is a linear process. Given the difference between these two learning processes, we use a hybrid-learning algorithm [13] for RBF neural network training:

(1) Using linear least square method, to adjust output layer connection weights.
(2) Using Gradient descent method, to adjust the center and the Gaussian width of the hidden layer neurons.

When the input matrix is $X \in R^{s \times N}$, the output matrix of hidden layer is $R \in R^{u \times N}$, the output matrix of output layer is $Y \in R^{s \times N}$, N is the number of training samples.

The pending output layer weights of RBF neural network is $X \in R^{s \times u}$, so the relationship among the hidden layer output matrix, the output layer weights and network output matrix is:

$$Y = W \times R \tag{8}$$

Let the target output of sample be $T = [t_1, t_2, \ldots, t_n] \in R^{s \times N}$; in order to make the error between the target output and the actual network output minimum, we can use linear least squares method, use the pseudo-inverse R^+ of R to determine W:

$$W = R^+ T \tag{9}$$

Center and width of the hidden layer using supervised learning algorithm for training, all of the parameters have gone through a learning process of error correction. In this paper, we use the Gradient descent method [14-16] for training, the function is:

Define the error function

$$E = \frac{1}{2} \sum_{n=1}^{N} E^n \tag{10}$$

N is the number of sample; E^n is the error after the n-th input of sample, define E^n is:

$$E^n = \sum_{k=1}^{N} (t_k^n - y_k^n)^2 \tag{11}$$

In order to make the error function minimum, the correction of parameter should be proportional to its negative gradient:

$$\Delta C_j = -\vartheta \frac{\partial E}{\partial C_j}, \Delta\sigma_j = -\vartheta \frac{\partial E}{\partial \sigma_j} \tag{12}$$

ϑ is the learning rate, and the calculation function obtained after expansion:

$$\Delta C_j = 2\vartheta \sum_{n=1}^{N} \sum_{k=1}^{s} (t_k^n - y_k^n) \cdot \omega^n(k,j) \cdot R_j^n \cdot \frac{P^n - C_j^n}{\left(\sigma_j^n\right)^2} \tag{13}$$

$$\Delta\sigma_j = 2\vartheta \sum_{n=1}^{N} \sum_{k=1}^{s} (t_k^n - y_k^n) \cdot \omega^n(k,j) \cdot R_j^n \cdot \frac{\left\|P^n - C_j^n\right\|^2}{\left(\sigma_j^n\right)^2} \tag{14}$$

After entering all the samples, using the modified function to adjust the parameters, the iterative function is

$$C_j(m+1) = C_j(m) + C_j \tag{15}$$
$$\sigma_j(m+1) = \sigma_j(m) + \sigma_j \tag{16}$$

m is the number of iterations.

4. Experimental Results

The proposed face recognition system is running on the hardware environment of Inter(R) Core(TM)2 (2.93GHz) and the software environment of Windows 7 and Matlab R2009a.

The experiment uses the sample face images from the ORL database showed in Figure 2, using Matlab simulation experiments. The database contains 40 individuals per 10 images in total 400 facial images. In this experiment, we will divide these face images between a training set and test set in a total of 2 groups, where 200 are used for training, and the other 200 used for testing.

Figure 2. Sample Face Images from the ORL Database

After using the method in this paper to initialize RBF hiddenlayer and learning center, we set different parameters for the classification of the performance analysis. We fix training times for the 200, the center of learning rete is 0.025, Gaussian width learning rate is 0.05. For different confidence levels, we have multiple sets of experiments; we have a statistics of the number of error identification and the cutoff error after the network training, and then we get the results in Table 1. Consolidated results under different feature vector, it is not difficult to find in the confidence level 0.6-0.7 that the RBF classifier has better recognition results. In Figure 3, we show a part of the results of the tests are given in erroneous identification, with the rightmost one per line for line misrecognized face actual output. Here, we can find that having a larger angle change, under appendages (such as glasses), *etc.*, are easier to correctly identify that the classifier has better robustness.

Table 1. Performance Analysis under Different Confidence Intervals and Eigenvectors

β	Misclassification Number			End Error			Recognition Rate		
	25	30	35	25	30	35	25	30	35
0.5	6	5	5	0.020	0.002	0.002	97	97.5	97.5
0.6	5	5	5	0.003	0.003	0.004	97.5	97.5	97.5
0.65	5	5	5	0.004	0.005	0.006	97.5	97.5	97.5
0.7	5	5	5	0.005	0.006	0.008	97.5	97.5	97.5
0.8	6	6	5	0.013	0.015	0.019	97.5	96.5	97.5
0.9	7	7	6	0.055	0.069	0.090	96.5	96.5	96.5

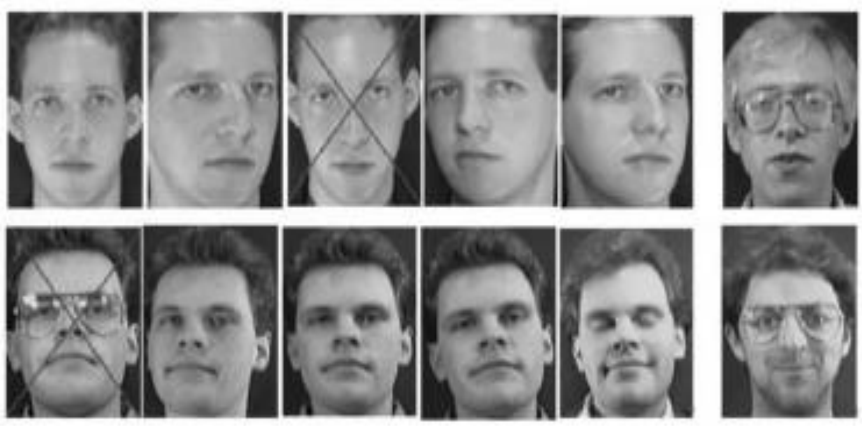

Figure 3. A Part of Identification Test Results

We fix the learning rate at 0.1, use the same set of samples to emulate, the learn times of 20, 100, 150, 200, 500, 800, 6 times emulate in total, with the results showed in Table 2 and Figure 4. We can find that when the learning time is beyond 200, although the error decreased, it did not reduce the number of misclassifications. In fact, higher training times will reduce the generalization performance of the classifier; it will also cause performance degradation of classifier [17].

Table 2. The Effect of Learning Times Epochs for the Number of Misclassified and Network Errors

Learning Times	Error Recognition	Error Value	Recognition Rate
30	42	0.0564	96.5%
45	36	0.0348	97.0%
46	36	0.0339	97.0%
47	30	0.0339	97.5%
48	30	0.0320	97.5%
49	30	0.0311	97.5%
50	30	0.0303	97.5%
80	30	0.0154	97.5%
110	30	0.0097	97.5%
140	30	0.0068	97.5%

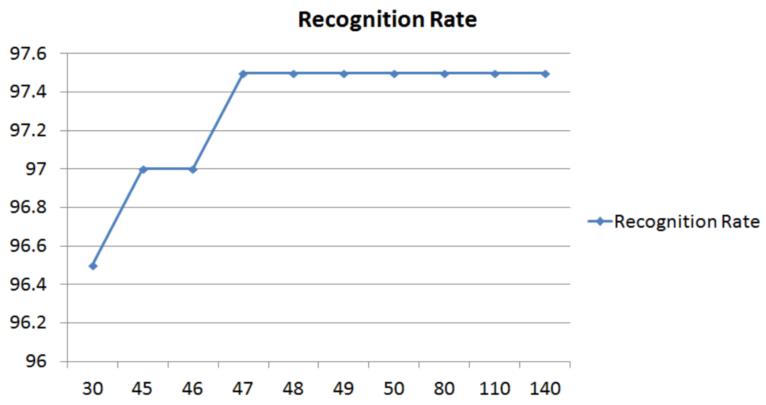

Figure 4. The Effect of Learning Times Epochs for the Number of Misclassified and Network Errors

In order to test the classifier performance, we added random noise for the test set of face images, and the learning time is 200, learning rate is 0.1, confidence is 0.6 for the experiment, we receive the result in Figure 5. The result shows that the classifier has good anti-noise performance, although at high intensity noise, it also to maintains a high recognition rate.

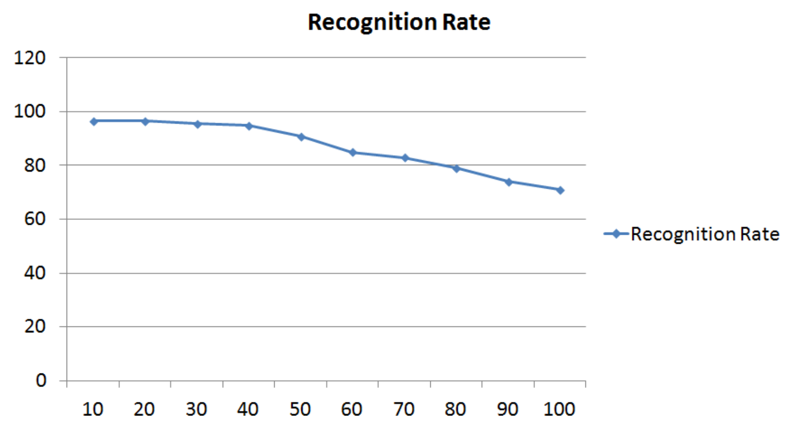

Figure 5. The Impact of Noise Intensity on Recognition Rate

5. Conclusion

In this paper, a face recognition methodology based on a combination algorithm of Discrete Cosine Transform (DCT) and Radical Basis Function (RBF) neural network was proposed. First, face features are extracted through the DCT algorithm. The method based on DCT can compress the information of original signals efficiently. Then using the RBF neural network introduced in this paper, which combined the Gradient Descent algorithm with the Linear Least Square algorithm, it had both local performance and entire performance.

The experimental results showed that the method in this paper worked well on ORL face database, and also had a good anti-noise performance, although at high intensity noise, it also needed to maintain a high recognition rate. So, this method has good capability of resolving high-dimension problems, small size samples problems and no-linear separable problem. The feature vectors are only extracted from gray-scale, so a real-time recognition system needs to be researched continuously.

Acknowledgments

This research was supported by the Tongmyong University of Research Grants 2014A002.

References

[1] M. Turk and A. Pentland, “Eigenfaces for Recognition”, Journal of Cognitive, Neuroscience, vol. 3, no. 1, **(1991)**, pp. 71-86.
[2] T.-J. Chin and D. Suter, “A Study of the Eigenface Approach for Face Recognition”, Technical Report of Monash University, Dept of Elect & Computer Systems Engineering, **(MECSE-6-2004)**, pp. 1-18.
[3] D. Blackburn, M. Bone and P. Phillips, “Face Recognition Vendor Test 2000: Evaluation Report”, National Institute of Science and Technology, Gaithersburg, USA, **(2000)**.
[4] P. J. Phillips, H. Moon, S. Rizvi and P. Rauss, “FERET Evaluation Methodology for Face Recognition Algorithms”, IEEE Transaction on Pattern Analysis and Machine Itelligence, vol. 22, no. 10, **(2000)**, pp. 1090-1103.
[5] D. Bryliuk and V. Starovoitoy, “Access Control by Face Recognition using Neural Networks and Negative Examples”, Proceeding of 2nd International Conference on Artificial Intelligence, Crimea, Ukraine, **(2002)**, pp. 428-436.
[6] S. A. Nazeer, N. Omar and M. Khalid, “Face Recognition System using Artificial Neural Networks Approach”, IEEE International Conference on Signal Processing, Communications and Networking, **(2007)**, pp. 428-436.

[7] X. Shao Huang and H. Wechsler, "Face Pose Discrimination Using Support Vector Machines", Technical Report of George Mason University and University of Minnesota, Minneapolis Minnesota, United states, vol. 1, **(1998)**, pp. 154-156.
[8] R. Brunelli and T. Poggio, "Face Recognition: Features versus Templates", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 10, **(1993)**, pp. 1042-1052.
[9] W. Chen, T. Sun, X. Yang and L. Wang, "Face detection based on half face template", Proceeding of the IEEE Conference on Electronic Measurement and Instrumentation, **(2009)**, pp. 54-58.
[10] W. Xu and E.-J. Lee, "Human Face Recognition Based on Improved D-LDA and Integrated BRNNs Algorithms", International Journal of Security & Its Applications, vol. 6, no. 2, **(2012)**, pp. 121.
[11] W. Chen, M. Joo Er and S. Wu, "PCA and LDA in DCT domain", Pattern Recognition Letters, vol. 26, **(2005)**, pp. 2474-2482.
[12] H. Li-Qun, "Artificial Neural Network Tutorial", Beijing: Beijing University of Posts Telecommunications Press, **(2006)**.
[13] Z. Xiu-Qing, "Face Recognition System based on Hopgield Neural Network", Electronic Science and Technology University, Chengdu, China, **(2002)**.
[14] M. Joo Er, S. Wu and J. Lu, "Face Recognition With Radial Basis Function (RBF) Neural Networks", IEEE Transactions on Neural Networks, vol. 13, no. 3, **(2002)**, pp. 697-709.
[15] T. Sergios and K. Konstantios, "Rattern Recognition", Beijing: Electronic Industry Press, **(2004)**.
[16] K. Satish, "Neural Networks", Beijing: Tsinghua University Press, **(2006)**.
[17] W. Zhong, "Face Recognition Technology Based on Gabor Wavelet and RBF Neural Network", Fuzhou University, Fuzhou, China, **(2005)**.

Authors

Ning Zhang, received his B. S. at Dalian Polytechnic University in China (2008-2012). Currently, he is studying in the Department of Information and Communications Engineering in Tongmyong University, Korea for his Masters degree. His main research areas are image processing and pattern recognition.

Eung-Joo Lee, received his B. S., M. S. and Ph. D. in Electronic Engineering form Kyungpook National University, Korea, in 1990, 1992, and Aug. 1996, respectively. Since 1997, he has been with the Department of Information & Communications Engineering, Tongmyong University, Korea, where he is currently a professor. From 2000 to July 2002, he was a president of Digital Net Bank Inc. From 2005 to July 2006, he was a visiting professor in the Department of Computer and Information Engineering, Dalian Polytechnic University, China. His main research interests include biometrics, image processing, and computer vision.