Face Recognition Using DCT and Hybrid Flexible Neural Tree

Yuehui Chen
School of Information Science
and Engineering
Jinan University
Jinan 250022, China
E-mail: yhchen@ujn.edu.cn

Shuyan Jiang
Department of Automation
Luoyang Technology College
Luoyang 471003, China
E-mail: jsyff88@163.com

Ajith Abraham
School of Information Science
and Technology
Chung-Ang University
Seoul, Republic of Korea
E-mail: ajith.abraham@ieee.org

Abstract—**This paper proposes a new face recognition approach by using the Discrete Cosine Transform (DCT) and hybrid flexible neural tree (FNT) classification model. The DCT is employed to extract the input features to build a face recognition system, and the flexible neural tree is used to identify the faces. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. This framework allows input features selection, over-layer connections and different activation functions for the various nodes involved. The FNT structure is developed using an evolutionary algorithm and the parameters are optimized by a particle swarm optimization algorithm. Empirical results indicate that the proposed framework is efficient for face recognition.**

I. INTRODUCTION

Face recognition has become a very active research area in recent years mainly due to increasing security demands and its potential commercial and law enforcement applications. Face recognition is a very challenging problem and up to date, there is no technique that provides a robust solution to all situations and different applications that face recognition may encounter. Face recognition approaches on still images can be broadly grouped into geometric and template matching techniques. In the first case, geometric characteristics of faces to be matched, such as distances between different facial features, are compared. This technique provides limited results although it has been used extensively in the past. In the second case, face images represented as a two dimensional array of pixel intensity values are compared with a single or several templates representing the whole face. More successful template matching approaches use Principal Components Analysis (PCA) or Linear Discriminant Analysis (LDA) to perform dimensionality reduction achieving good performance at a reasonable computational complexity/time. Other template matching methods use neural network classification and deformable templates, such as Elastic Graph Matching (EGM). Recently, a set of approaches that use different techniques to correct perspective distortion are being proposed. These techniques are sometimes referred to as view-tolerant. For a complete review on the topic of face recognition the reader is referred to [1] and [2].

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 classifiers [2]. Multilayered networks (MLNs), usually employing the backpropagation (BP) algorithm, are widely used in face recognition [3]. Recently, RBF neural networks have been applied in many engineering and scientific applications including face recognition [4][5].

This paper proposes a Flexible Neural Tree (FNT) [7][8] for selecting the input variables and face recognition. Based on the pre-defined instruction/operator sets, a flexible neural tree model can be created and evolved. FNT allows input variables selection, over-layer connections and different activation functions for different nodes. In our previous work, the hierarchical structure was evolved using Probabilistic Incremental Program Evolution algorithm (PIPE) with specific instructions. In this research, the hierarchical structure is evolved using tree structure based evolutionary algorithm. The fine tuning of the parameters encoded in the structure is accomplished using particle swarm optimization (PSO). The proposed method interleaves both optimizations. Starting with random structures and corresponding parameters, it first tries to improve the structure and then as soon as an improved structure is found, it fine tunes its parameters. It then goes back to improving the structure again and, fine tunes the structure and rules' parameters. This loop continues until a satisfactory solution is found or a time limit is reached. The novelty of this paper is in the usage of flexible neural tree model for selecting the important features and for face recognition.

The paper is organized as follows. The DCT is presented in Section 2. Section 3 presents the FNT classifier for face recognition. The simulation results are given in Section 4, and finally the conclusion is drawn in Section 5.

II. DISCRETE COSINE TRANSFORM

0-7803-9422-4/05/$20.00 ©2005 IEEE

Like other transforms, the Discrete Cosine Transform (DCT) attempts to decorrelate the image data [11]. After decorrelation each transform coefficient can be encoded independently without losing compression efficiency. This section describes the DCT and some of its important properties.

The 2-D DCT is a direct extension of the 1-D case and is given by

$$C(u,v)=\alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}f(x,y)\cos\frac{\pi(2x+1)u}{2N}\cos\frac{\pi(2y+1)}{2N} \qquad (1)$$

for u, v = 0, 1,2,...,N-1, and $\alpha(u)$ and $\alpha(v)$ are defined as follows, $\alpha(u)=\sqrt{1/N}$ for $u=0$,and $\alpha(u)=\sqrt{2/N}$ for $u\neq 0$. The inverse transform is defined as

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{N-1}\alpha(u)\alpha(v)C(u,v)\cos\frac{\pi(2x+1)u}{2N}\cos\frac{\pi(2y+1)}{2N} \qquad (2)$$

for x, y = 0, 1, 2, . . . , N-1.

The DCT possess some fine properties, i.e., de-correlation, energy compaction, separability, symmetry and orthogonality. These attributes of the DCT have led to its widespread deployment in virtually every image/video processing standard of the last decade [11].

For an $M\times N$ image, we have $M\times N$ an DCT coefficient matrix covering all the spatial frequency components of the image. The DCT coefficients with large magnitude are mainly located in the upper-left corner of the DCT matrix. Accordingly, we scan the DCT coefficient matrix in a zig-zag manner starting from the upper-left corner and subsequently convert it to a one-dimensional (1-D) vector. Detailed discussions about image reconstruction errors using only a few significant DCT coefficients can be found in [9]. As a holistic feature extraction method , the DCT converts high-dimensional face images into low-dimensional spaces in which more significant facial features such as outline of hair and face, position of eyes, nose and mouth are maintained. These facial features are more stable than the variable high-frequency facial features. As a matter of fact, the human visual system is more sensitive to variations in the low-frequency band.

In this paper, we investigate the illumination invariant property of the DCT by discarding its several low-frequency coefficients. It is well-known that the first DCT coefficient represents the dc component of an image which is solely related to the brightness of the image. Therefore, it becomes DC free (i.e., zero mean) and invariant against uniform brightness change by simply removing the first DCT coefficient.

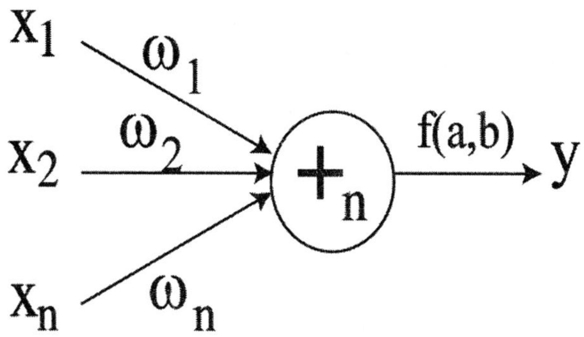

Fig. 1. A flexible neuron operator in which the terminal instruction set T = $\{x_1, x_2,...,x_n\}$ represents input variables.

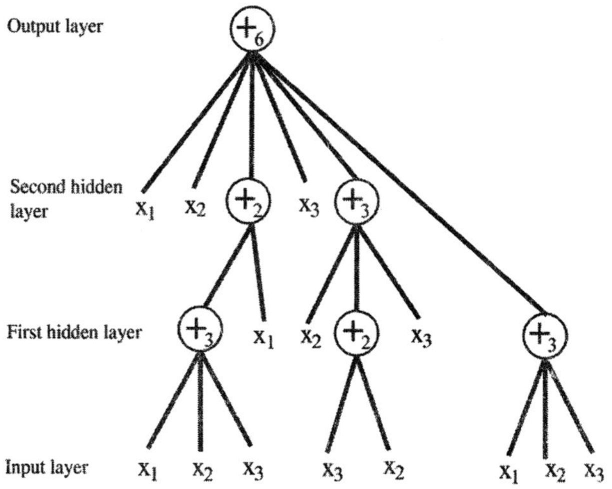

Fig. 2. A typical representation of the FNT with function instruction set F = $\{+_2, +_3, +_4, +_5, +_6\}$ and terminal instruction set T = $\{x_1, x_2, x_3\}$.

III. THE FLEXIBLE NEURAL TREE (FNT) MODEL

The function set F and terminal instruction set T used for generating a FNT model are described as

$$S=F\cup T=\{+_2,+_3,...,+_N\}\cup\{x_1,...,x_n\}$$

where $+_i$ (i = 2, 3, . . . , N) denote non-leaf nodes' instructions and taking i arguments. x_1, x_2, . . . , x_n are leaf nodes' instructions and taking no other arguments. The output of a non-leaf node is calculated as a flexible neuron model (see Fig.1 and Fig.2). From this point of view, the instruction $+_i$ is also called a flexible neuron operator with i inputs.

In the creation process of neural tree, if a nonterminal instruction, i.e., $+_i\ (i=2,3,4,...,N)$ is selected, i real values are randomly generated and used for representing the connection strength between the node $+_i$ and its children. In addition, two adjustable parameters a_i and b_i are randomly created as flexible activation function parameters. For developing the face classifier, the flexible activation function used is as follows,

$$f(a_i,b_i,x)=e^{-\left(\frac{x-a_i}{b_i}\right)^2} \tag{3}$$

The total excitation of $+_n$ is

$$net_n=\sum_{j=1}^{n}\omega_j * x_j \tag{4}$$

where $x_j\left(j=1,2,...,n\right)$ are the inputs to node $+_n$. The output of the node $+_n$ is then calculated by

$$out_n=f(a_n,b_n,net_n)=e^{-\left(\frac{net_n-a_n}{b_n}\right)^2} \tag{5}$$

The overall output of flexible neural tree can be computed from left to right by depth-first method, recursively.

A. Tree Structure Optimization

Finding an optimal or near-optimal neural tree is formulated as a product of evolution. In this study, the crossover and selection operators used are same as those of standard GP. A number of neural tree mutation operators are developed as follows:

1) Changing one terminal node: randomly select one terminal node in the neural tree and replace it with another terminal node;

2) Changing all the terminal nodes: select each and every terminal node in the neural tree and replace it with another terminal node;

3) Growing: select a random leaf in hidden layer of the neural tree and replace it with a newly generated subtree.

4) Pruning: randomly select a function node in the neural tree and replace it with a terminal node.

B. Parameter Optimization with PSO

The Particle Swarm Optimization (PSO) [12] conducts searches using a population of particles which correspond to individuals in evolutionary algorithm (EA). A population of particles is randomly generated initially. Each particle represents a potential solution and has a position represented by a position vector x_i. A swarm of particles moves through the problem space, with the moving velocity of each particle represented by a velocity vector v_i. At each time step, a function *fi* representing a quality measure is calculated by using x_i as input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far in a vector p_i. Furthermore, the best position among all the particles obtained so far in the population is kept track of as p_g. In addition to this global version, another version of PSO keeps track of the best position among all the topological neighbors of a particle. At each time step *t*, by using the individual best position, p_i, and the global best position, $p_g(t)$, a new velocity for particle *i* is updated by

$$\begin{aligned} v_i(t+1)=v_i(t)+c_1\phi_1\left(p_i(t)-x_i(t)\right)\\ +c_2\phi_2\left(p_g(t)-x_i(t)\right) \end{aligned} \tag{6}$$

where c_1 and c_2 are positive constant and Φ_1 and Φ_2 are uniformly distributed random number in [0,1]. The term v_i is limited to the range of $\pm v_{\max}$. If the velocity violates this limit, it is set to its proper limit. Changing velocity this way enables the particle *i* to search around its individual best position, p_i, and global best position, p_g. Based on the updated velocities, each particle changes its position according to the following equation:

$$x_i(t+1)=x_i(t)+v_i(t+1). \tag{7}$$

C. Procedure of the general learning algorithm

The general learning procedure for constructing the FNT model can be described as follows.

1) Create an initial population randomly (FNT trees and its corresponding parameters);

2) Structure optimization is achieved by the neural tree variation operators as described in subsection 2.

3) If a better structure is found, then go to step 4), otherwise go to step 2);

4) Parameter optimization is achieved by the PSO algorithm as described in subsection 2. In this stage, the architecture of FNT model is fixed, and it is the best tree developed during the end of run of the structure search. The parameters (weights and flexible activation function parameters) encoded in the best tree formulate a particle.

5) If the maximum number of local search is reached, or no better parameter vector is found for a significantly long time then go to step 6); otherwise go to step 4);

6) If satisfactory solution is found, then the algorithm is stopped; otherwise go to step 2).

IV. FEATURE SELECTION AND FACE RECOGNITION USING FNT PARADIGMS

A. The ORL face database

The database consists of 400 images acquired from 40 persons with variations in facial expression (e.g. open / close eyes, smiling / non-smiling), and facial details (e.g. wearing glasses / not wearing glasses). All images were taken under a dark background, and the subjects were in an upright frontal position, with tilting and rotation tolerance up to 20 degree, and tolerance of up to about 10%. All images are of grey scale with a 92*112 pixels resolution. For each person, 5 images are generated randomly to form the training data set and the remaining were chosen as test data set.

B. Feature/Input Selection with FNT

It is often a difficult task to select important variables for any problem, especially when the feature space is large. A fully connected neural network (NN) classifier usually cannot do this. In the perspective of FNT framework, the nature of model construction procedure allows the FNT to identify important input features in building a face classifier that is computationally efficient and effective. The mechanisms of input selection in the FNT constructing procedure are as follows.

(1) Initially the input variables are selected to formulate the FNT model with same probabilities;

(2) The variables which have more contribution to the objective function will be enhanced and have high opportunity to survive in the next generation by a evolutionary procedure;

(3) The evolutionary operators i.e., crossover and mutation, provide a input selection method by which the FNT should select appropriate variables automatically.

C. Face Recognition Using FNT with 55 Input Variables

For this simulation, the DCT is employed to training and testing data sets, respectively. The extracted 55 input features are used for constructing a FNT model. A FNT classifier was constructed using the training data and then the classifier was used on the test data set to classify the data as an face ID or not. The instruction sets used to create an optimal FNT classifier is

$$S = F \bigcup T = \{+_5,\ldots,+_{20}\} \bigcup \{x_1, x_2, \ldots, x_{55}\}. \tag{8}$$

Where $x_i (i = 1,2,\ldots,55)$ denotes the 55 features.

A comparison of different feature extraction methods and different face classification methods is shown in Table 1. Table 2 depicts the face recognition performance of the FNT by using the 55 features for test data set. The FNT method helps to reduce the features from 55 to 8-25.

TABLE I

COMPARISON OF DIFFERENT APPROACHES FOR ORL FACE RECOGNITION

Method	**Recognition rate**
PCA+RBFN [10]	94.5%
LDA+RBFN [10]	94.0%
FS+RBFN [10]	92.0%
DCT+FNT (this paper)	98.13%

V. CONCLUSIONS

In this paper DCT based feature extraction method is proposed for the extraction of features from facial images and recognition experiments were carried out by using FNT classifier. The ORL database images are used for conducting all the experiments. The obtained results are compared. Facial features are first extracted by the DCT which greatly reduces dimensionality of the original face image as well as maintains the main facial features. Compared with the well-known PCA approach, the DCT has the advantages of data independency and fast computational speed. The presented Flexible Neural Tree (FNT) model for face recognition with a focus on improving the face recognition performance by reducing the input features. Simulation results on ORL face database also show that the proposed method achieves high training and recognition speed, as well as high recognition rate. More importantly, it is insensitive to illumination variations.

ACKNOWLEDGMENT

This research was partially supported by the National High Technology Development Program of China (863 Program) under contract number 2002AA4Z3240, and The Provincial Science and Technology Development Program of Shandong under contract number SDSP2004-0720-03.

REFERENCES

[1] Zhao, W., Chellappa R., Rosenfeld A. and Phillips P.J.: Face Recognition: A literature survey. *Technical Report CART-TR-948*. University of Maryland, Aug. 2002.

[2] Chellappa R., Wilson, C.L., and Sirohey, S.: Human and machine recognition of faces: A survey, *Proc. IEEE*, 83(5): 705C740, 1995.

[3] Valentin D., Abdi H., Toole, A. J. O., and Cottrell, G. W.: Connectionist models of face processing: A survey, *Pattern Recognit.*, 27: 1209C1230, 1994.

[4] Er, M. J., Wu, S., Lu, J., and Toh, H. L.: Face recognition with radial basis function (RBF) neural networks, *IEEE Trans. Neural Netw.*, 13(3): 697C710, 2002.

[5] Yang F. and Paindavoine M.: Implementation of an RBF neural network on embedded systems: Real-time face tracking and identity verification, *IEEE Trans. Neural Netw.*, 14(5): 1162C1175, 2003.

[6] Kwak., K.-C., and Pedrycz., W.: Face Recognition Using Fuzzy Integral and Wavelet Decomposition Method, *IEEE Trans. on SMC-B*, 34(4): 1667-1675, 2004.

[7] Chen, Y., Yang, B., Dong, J., Nonlinear systems modelling via optimal design of neural trees, *International Journal of Neural Systems*, **14**, pp. 125-138, 2004.

[8] Chen Y., Yang B., Dong J. and A. Abraham: Time-series Forecasting using Flexible Neural Tree Model. *Information Science*, Vol.174, Issues 3/4, pp.219-235, 2005.

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

[10] Su, H., Feng D., Zhao R.-C.: Face Recognition Using Multi-feature and Radial Basis Function Network, *Proc. of the Pan-Sydney Area Workshop on Visual Information Processing (VIP2002)*, Sydney, Australia, 183-189, 2002.

[11] Sorwar G., Abraham A. and Dooley L., Texture Classification Based

on DCT and Soft Computing, *The 10th IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'01*, IEEE Press, Volume 2, pp. 545 -548, 2001.

[12] Kennedy, J. and Eberhart, R.C., Particle Swarm Optimization, *Proc. Of IEEE International Conference on Neural Networks*, **IV**, pp. 1942-1948, 1995.

TABLE II

THE TRUE POSITIVE (TP), FALSE POSITIVE (FP) RATES FOR FNT FACE RECOGNITION.

Face ID	TP	FP	Accuracy
S1	80.00%	1.54%	98.00%
S2	80.00%	1.54%	98.00%
S3	80.00%	4.61%	95.00%
S4	100.00%	2.56%	97.50%
S5	80.00%	0.51%	99.00%
S6	100.00%	4.10%	96.00%
S7	40.00%	0.51%	98.00%
S8	80.00%	0.51%	99.00%
S9	80.00%	1.53%	98.00%
S10	100.00%	0.00%	100.00%
S11	60.00%	3.59%	95.50%
S12	80.00%	0.00%	99.50%
S13	80.00%	0.51%	99.00%
S14	80.00%	0.51%	99.00%
S15	80.00%	3.07%	96.50%
S16	80.00%	0.51%	99.00%
S17	100.00%	4.62%	95.50%
S18	100.00%	0.51%	99.50%
S19	100.00%	2.05%	98.00%
S20	60.00%	0.00%	99.00%
S21	100.00%	0.00%	100.00%
S22	80.00%	0.00%	99.50%
S23	80.00%	0.51%	99.00%
S24	80.00%	2.05%	97.50%
S25	100.00%	2.05%	98.00%
S26	80.00%	0.51%	99.00%
S27	60.00%	1.02%	98.00%
S28	60.00%	1.54%	97.50%
S29	100.00%	3.07%	97.00%
S30	60.00%	0.00%	99.00%
S31	100.00%	0.51%	99.50%
S32	80.00%	1.03%	98.50%
S33	100.00%	0.51%	99.50%
S34	100.00%	0.51%	99.50%
S35	60.00%	2.05%	97.00%
S36	60.00%	2.05%	97.00%
S37	80.00%	1.54%	98.00%
S38	80.00%	3.07%	96.50%
S39	100.00%	0.51%	99.50%
S40	80.00%	4.62%	95.00%
Average	82.00%	1.50%	98.13%