

# Face Recognition Based on Constructive Neural Networks Covering Learning Algorithm

<sup>1</sup>HUANG Guohong, XIONG Zhihua and SHAO Huihe

**Abstract**—A general and efficient design approach using covering neural classifier to cope with the high-dimensional and small sample size problem is proposed. For alleviating the computational burden, face features are first extracted by the principal component analysis (PCA). In order to avoid the influence of outlier classes and reduce the large overlapping of neighboring classes, a new weighted Fisher linear discriminant (WFLD) criterion is presented by weighting contributions of individual class pairs according to the Euclidian distance of the respective class means. A new learning algorithm is used to train the neural networks classifier, which uses the “sphere neighborhoods” to cover the input samples and draw up their distributions in the original space. Thus, the training problem of neural networks may be transformed into the covering problem of a point set, which avoids the iterative process, and overcomes the problem of longtime training of classical neural networks. Simulation results conducted on the ORL database show that the system achieves excellent performance both in terms of error rates of classification and learning efficiency.

## I. INTRODUCTION

Human face recognition has become an active research area in the image processing, pattern recognition, networks and computer vision. This interest is motivated by wide applications such as passports, credit cards, driving licenses and mug shots, and so on [1]. Although researchers in psychology, neural sciences and engineering, image processing and computer vision have investigated a number of issues related to face recognition, it is still difficult to design an automatic system for this task, especially when real-time identification is required. The reasons for this difficulty are in two aspects: 1) Face images are highly variable and 2) Sources of variability include individual appearance, pose, facial expression, facial hair, makeup, and so on and these factors change from time to time. Furthermore, the lighting, background, scale, and parameters of the acquisition are all variables in facial images acquired under real-world scenarios [1]. As stated by Moses *et al.* [2], “The variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to changes in the face identity.” This makes face recognition a great

challenging problem. In our opinion, two issues are central to face recognition: 1) what features can be used to represent a face under environmental changes? 2) How to classify a new face image based on the chosen representation?

The geometric feature-based methods [3, 4] are frequently used, where the relative positions and shapes of different features are measured. Although being insensitive to variations in illumination and viewpoint, this class of methods rely heavily on the extraction and measurement of facial features. Unfortunately, feature extraction and measurement techniques and algorithms developed to data have not been reliable enough to cater to this need. At the same time, several approaches have been proposed to use global representation of a face [5-8], where all features of a face are automatically extracted from an input facial image. In [5], singular value decomposition (SVD) was used to extract features from the patterns. It has been illustrated that singular values of an image are stable and represent the algebraic attributes of an image, being intrinsic but not necessarily visible. The eigenface approach of describing the features of a face was presented in [6]. The key idea is to calculate the best coordinate system for image compression. The eigenface uses principal component analysis (PCA) to yield projection directions that maximize the total scatter. However, in choosing the projection directions, the PCA retains unwanted variations caused by lighting, facial expression, and other factors [7]. Accordingly, the features are not necessarily good for discrimination among classes. In [7, 8], the face features are acquired by using the fisherface. This approach aims at overcoming the drawback of the eigenface by integrating Fisher’s linear discriminant (FLD) criteria, while retaining the idea of the eigenface in projecting faces from a high-dimensional feature space to a significantly lower-dimensional feature space. Furthermore, neural-networks-based feature extraction has been reported recently [9, 10]. The goal of face processing using neural networks is to develop a compact internal representation of faces, which is equivalent to feature extraction.

In many pattern recognition systems, the methodology frequently used is the statistical approach, whereby decision theory derived from statistical information of input patterns is used to design a classifier [11]. Although this

<sup>1</sup>. The Institute of Automation, Shanghai Jiao tong University, Shanghai, P. R. China (e-mail: h\_guohong@163.com)

paradigm has been successfully applied to solve various problems in pattern classification, it has difficulty in expressing structural information. Furthermore, this approach requires much heuristic information to design a classifier [12]. Neural-networks-based methods have been proven to have many advantages for classification because of their learning ability and good generalization. Generally speaking, multilayered networks, usually coupled with the backpropagation (BP) algorithm, are most widely used in face recognition [13]. Yet, two major criticisms are commonly raised against the BP algorithm: 1) It is computationally intensive because of its slow convergence speed and 2) there is no guarantee at all that the absolute minima can be achieved. On the other hand, covering learning algorithms based on constructive neural networks recently have attracted extensive interests in the community of neural networks [14–19], which have the best approximation property, fast learning speed, and more compact topology. In this paper, we also present a new covering algorithm called max-density covering learning algorithm to design the classifier. The algorithm directly uses the “sphere neighborhoods” belonging to the different classes to cover the input samples and draw up their distributions in the original space. Thus, the training problem of neural networks may be transformed into the covering problem of a point set, which avoids the iterative process, and overcomes the problem of longtime training of classical neural networks. Moreover, there is no need to guess the size, depth, and connectivity pattern of the network in advance. A reasonably small net is built automatically.

Particularly, face recognition is a typical high dimensional and small sample sets problem. For example, a 128\*128 image will have 16384 features, whereas the sample patterns are very few for each class. This situation can lead to complex computation in designing a classifier. As point out by Meng [20], “It does not mean that more information will result in higher performance.” Moreover, the addition of some unimportant information may become noise and degrade the performance. Along with the increase in the feature dimension, the training patterns have more overlapping, so a small should be selected. A systematic methodology to deal with small training sets of high-dimensional feature vectors is presented as shown in Fig. 1, which comprises the two parts: 1) The number of input variables is reduced through feature selection, i.e., a set of the most expressive features is first generated by the PCA and the WFLD (proposed in this paper) is then implemented to generate a set of the most discriminant features so that different classes of training data can be separated as far as possible and the same classes of patterns are compacted as close as possible; 2) A new covering neural classifier is proposed, which is circumstantiated in this paper.

The rest of this paper is organized as follows. Section 2 presents the architecture of max-density covering learning algorithm and the related design problem when they are used as a classifier. Section 3 provides the procedure of extracting face features. Experimental results are demonstrated in Section 4.

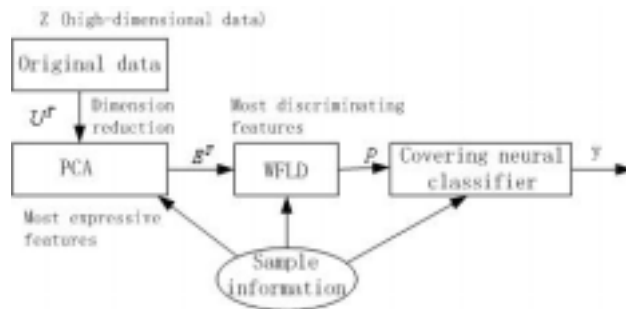


Fig.1. Schematic of max-density covering neural classifier for small training sets of high dimension.

## II. THE MAX-DENSITY COVERING LEARNING ALGORITHM OF CONSTRUCTIVE NEURAL NETWORKS

### A. The Geometrical Interpretation of the Neuron Model

A basic M-P neuron model [14, 21] is an element with  $n$  input and one output. The general form of its function is

$$y = \text{sgn}(W^T X - \theta) \quad (1)$$

where  $X = (x_1, x_2, \dots, x_n)^T$  is an input vector,  $W = (w_1, w_2, \dots, w_n)^T$  is a weight vector, and  $\theta$  is the threshold.

$$\text{sgn}(v) = \begin{cases} 1, & v \geq 0 \\ -1, & v < 0 \end{cases} \quad (2)$$

Note that  $W^T X - \theta = 0$  can be interpreted as a hyperplane  $P$  in an  $n$ -dimensional space. When  $(W^T X - \theta) > 0$ , input vector  $X$  falls into the positive half-space of the hyperplane  $P$ . Meanwhile,  $y = \text{sgn}(W^T X - \theta) = 1$ . When  $(W^T X - \theta) < 0$ , input vector  $X$  falls into the negative half-space of  $P$ , and  $y = -1$ . In summary, the function of an M-P neuron can geometrically be regarded as a spatial discriminator of an  $n$ -dimensional space divided by the hyperplane.

Assume each input vector  $X$  has an equal length [14] (which can be obtained by projecting the original data upward on the  $n$ -dimensional sphere  $S^n$ ). Thus all input vectors will be restricted to an  $n$ -dimensional sphere  $S^n$ . Then,  $(W^T X - \theta) > 0$  represents the intersection

between the positive half-space  $P$  and  $S^n$ , i.e., a “sphere neighborhood” as shown in Fig.2. When input vector  $X$  falls into the “sphere neighborhood”, output  $y = \text{sgn}(W^T X - \theta) = 1$ , otherwise,  $y = -1$ . An M-P neuron acts as a discriminator of an input class.

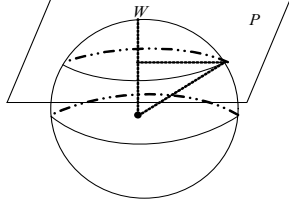


Fig.2. A sphere neighborhood

If the weight vector  $W$  has the same length as input  $X$ , then  $W$  becomes the center of the sphere neighborhood, and  $r(\theta)$ , a monotonically decreasing function of  $\theta$ , becomes its radius, i.e., an M-P neuron corresponds to a sphere neighborhood in an  $n$ -dimensional sphere. Thus, the first design stage of a neural classifier can be transformed to an input vectors covering problem. Namely, a set of sphere neighborhoods is chosen to cover the inputs having the same output, i.e., belonging to the same class. Different classes of inputs are covered by different sets of sphere neighborhoods. Namely, an M-P neuron also corresponds to a covering on  $S^n$ . Therefore, a network consisting of these neurons has the same function as a classifier.

### B. The Max-density Covering Learning Algorithm for Neural Classifiers

As a classifier, the function of a neural network can be stated as follows. Given a set of training samples  $\{(X_{i,j}, y_i), 1 \leq i \leq c, 1 \leq j \leq \text{num}_i\}$ , assume each input vector  $X_{i,j}$  has an equal length, the output  $y_i$  has  $c$  different values, i.e., the input vectors will be classified into  $c$  classes, and  $\text{num}_i$  is the number of the input vectors belonging to the class  $i$ . After learning, the network should store the input pairs  $(X_{i,j}, y_i)$ . Moreover, when the input becomes  $X_{i,j} + \Delta$ , the output  $y_i$  still remains the same, where  $\Delta$  is regarded as a noise or an error. For an M-P neuron  $(W, \theta)$ , when  $W = X_{i,j}$  and  $r(\theta) = \Delta$ , as mentioned before, it geometrically corresponds to a sphere neighborhood on  $S^n$  with  $X_{i,j}$  as its center and  $\Delta$  as its radius. Thus, the proposed max-density learning algorithm of neural classifiers can be stated as follows.

Firstly, for the input vector  $X_{i,j}$ , we define a density function:

$$D_{i,j} = \sum_{n=1}^{\text{num}_i} \exp \left[ - \frac{\|X_{i,j} - X_{i,n}\|}{(r_{i,j}/2)^2} \right] \quad (3)$$

$$r_{i,j} = \min_{i \neq k} (\|X_{i,j} - X_{k,n}\|) \quad (4)$$

where positive number  $r_{i,j}$  (the minimum distance between the input vector  $X_{i,j}$  and the input vectors belonging to different classes) denotes a region with  $X_{i,j}$  as its center. Thus the input vectors out of the radius  $r_{i,j}$  have little effect on the density value  $D_{i,j}$  of the input vector  $X_{i,j}$ . Obviously, if some sample vector has a higher density value, it must have more sample vectors around it.

Let

$$d_{i,j}^1 = \max_{i \neq k} \{ \langle X_{i,j}, X_{k,n} \rangle \} \quad (5)$$

$$d_{i,j}^2 = \min \{ \langle X_{i,j}, X_{i,n} \rangle > d_{i,j}^1 \} \quad (6)$$

where  $d_{i,j}^1$  is the maximum inner product, namely minimum distance between the input vector  $X_{i,j}$  and the other vectors belonging to the different classes;  $d_{i,j}^2$  is the minimum inner product, namely maximum distance between the input vector  $X_{i,j}$  and the input vectors belonging to the same class.  $d_{i,j}^1$  and  $d_{i,j}^2$  both show the similarity degree between the training vectors.

Then the training process consisted of the following steps.

Step 1. Compute density values  $D$  of all the uncovered training samples on the  $S^n$ , and find the sample vector  $X'$  with the maximum density value. At the beginning of training, all the flags of training samples are initialized to 0.

Step 2. Evaluate the parameters  $d^1$  and  $d^2$  of the sample vector  $X'$  related in step 2, then get the parameter  $d = a \cdot d^1 + b \cdot d^2$ , where parameters  $a$  and  $b$  with the constraint  $a + b = 1$ .

Step 3. Regard the sample vector  $X'$  as the weight vector  $W$  of the constructive neural network and the parameter  $d$  as the threshold  $\theta$ , then get a hyperplane  $P: (W^T X - \theta) = 0$  across the sphere as shown in

fig.2.

Step 4. Find all the sample data meeting the condition  $(W^T X - \theta) > 0$  and then label their flag values 1. Then go back step 1 until all the flag values are 1.

An M-P neuron  $(W, \theta)$  described by step 3 corresponds to a covering, a neighborhood on  $S^n$ . The number of coverings determines the number of neurons needed in the first layer (hidden layer) of a neural classifier. The inputs are classified into different classes by the first layer. The second layer is used to form the given output. The framework of the neural classifier is shown in Fig. 3.

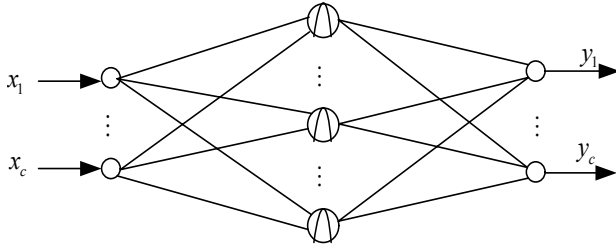


Fig. 3. The framework of the covering neural classifier

### III. EXTRACTION OF FACE FEATURES

#### A. Principal Component Analysis (PCA)

Let a face image  $Z_i$  be a two-dimensional  $m \times n$  array of intensity values. An image may also be considered as a vector of dimension  $mn$ . Denote the training set of  $L$  face images by  $Z = (Z_1, Z_2, \dots, Z_L) \subset R^{mn \times L}$ , and we assume that each image belongs to one of  $c$  classes. Define the covariance matrix as follows [6, 11]:

$$\Gamma = \frac{1}{L} \sum_{i=1}^L (Z_i - \bar{Z})(Z_i - \bar{Z})^T = \Phi \Phi^T \quad (7)$$

where  $\bar{Z} = (1/L) \sum_{i=1}^L Z_i$ . Then, the eigenvalues and eigenvectors of the covariance  $\Gamma$  are calculated. Here, we preserve all positive principal components of  $\Gamma$  so that no statistical information of original images is lost. Let  $U = (U_1, U_2, \dots, U_r) \subset R^{mn \times L}$  ( $r < L$ ) be the  $r$  eigenvectors corresponding to all the  $r$  positive eigenvalues. Hence, for a set of original face images  $Z \subset R^{mn \times L}$ , their corresponding eigenface-based feature  $F \subset R^{r \times L}$  can be obtained by projecting  $Z$  into the eigenface space as follows:

$$F = U^T Z \quad (8)$$

#### B. The Weighted Fisher Linear Discriminant (WFLD)

Actually, the PCA paradigm does not provide any information for class discrimination but dimension reduction. Accordingly, the Fisher linear discriminant (FLD)

is always applied to the projection of the set of training samples in the eigenface space  $F = (f_1, f_2, \dots, f_L) \subset R^{r \times L}$  to find an optimal subspace for classification in which the ratio of the between-class scatter and the within-class scatter is maximized [7,8,11,22,23].

However, the traditional FLD hardly any considers the distribution of the sample data and the influence of outlier classes (assume that one class is located remotely from the other classes and can be considered an outlier). In maximizing the squared distances, pairs of classes, between which there are large distances, completely dominate the eigenvalue decomposition. Hence, the direction to project on found by optimizing the Fisher criterion is the one that separates the outlier as much from the remaining classes as possible. As a consequence, there are large overlaps among the remaining classes, which lead to an overall low and suboptimal classification rate. Therefore, we present the weighting Fisher linear discriminant (WFLD) to extract discriminant features so as to weaken the influence of outlier classes.

As a part of our approach, the between-class scatter matrix  $S_b$  is rewritten as follows (see [24] for a proof):

$$\begin{aligned} S_b &= \sum_{i=1}^c P_i (m_i - m_0)(m_i - m_0)^T \\ &= \sum_{i=1}^{c-1} \sum_{j=i+1}^c P_i P_j (m_i - m_j)(m_i - m_j)^T \end{aligned} \quad (9)$$

where  $m_i, m_0, P_i (i=1, 2, \dots, c)$  are the mean vector, the overall mean, and a prior probability of class  $i$ , respectively. Notice that the decomposition enables us to write the between-class scatter matrix in terms of class-mean differences and that the term  $(m_i - m_j)(m_i - m_j)^T$  is actually the between-class scatter matrix for the classes  $i$  and  $j$  in a two-class model. In order to reduce the influence of the outlier classes, the between-class scatter matrix can be redefined as follows:

$$S_B = \sum_{i=1}^{c-1} \sum_{j=i+1}^c P_i P_j \omega(\Delta_{ij}) (m_i - m_j)(m_i - m_j)^T \quad (10)$$

where  $\omega$  is a weighting function depends on the Euclidean distance  $\Delta_{ij} := \sqrt{(m_i - m_j)^T (m_i - m_j)}$  between the classes  $i$  and  $j$  in the original model. For the simplicity, in this paper, we define the weighting function  $\omega(\Delta_{ij}) = 1 / \Delta_{ij}^2$ .

Then the weighting Fisher criterion can be defined as follows:

$$\begin{aligned}
J(\varphi) &= \sum_{i=1}^{c-1} \sum_{j=i+1}^c P_i P_j \omega(\Delta_{ij}) \frac{\varphi^T (m_i - m_j)(m_i - m_j)^T \varphi}{\varphi^T S_w \varphi} \\
&= \frac{\varphi^T S_B \varphi}{\varphi^T S_w \varphi}
\end{aligned} \tag{11}$$

where  $S_w = \sum_{i=1}^c P_i \sum_{k=1}^L \frac{1}{L} [(f_k - m_i)(f_k - m_i)^T]$  is the within-class scatter matrix. The optimal subspace,  $E_{opt}$  by the WFLD is determined as follow:

$$E_{opt} = \arg \max_E \frac{|E^T S_B E|}{|E^T S_w E|} = [\varphi_1, \varphi_2, \dots, \varphi_{c-1}] \tag{12}$$

where  $[\varphi_1, \varphi_2, \dots, \varphi_{c-1}]$  is the set of generalized eigenvectors of  $S_B$  and  $S_w$  corresponding to the  $(c-1)$  largest generalized eigenvalues  $\lambda_i, i=1, 2, \dots, c-1$ , i.e.,

$$S_B \varphi_i = \lambda_i S_w \varphi_i \quad i=1, 2, \dots, c-1 \tag{13}$$

Thus, the feature vectors  $X$  for any query face images  $Z$  in the most discriminant sense can be calculated as follows:

$$X = E_{opt}^T \cdot U^T \cdot Z \tag{14}$$

#### IV. EXPERIMENTAL RESULTS

Face recognition is a typical high-dimensional and small sample size problem. For testing the efficiency of our algorithm, experiments have been done on ORL face image database. ORL face image database is a widely used database. There are 10 different images of 40 distinct subjects and variations in facial expression (open/closed eyes, smiling/non-smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an up-right, frontal position, with tolerance for some tilting and rotation of up to about  $20^\circ$ . There is some variation in scale of up to about 10%. The size of each image is  $92 \times 112$ , which are shown by Fig.4.



Fig. 4 Face images from ORL database

Many researchers always use the ORL database to verify their algorithms. Here, we adopt the same definition of average error rate,  $E_{ave}$  used in [10,25,26], which is given by

$$E_{ave} = \sum_{i=1}^q n_{mis}^i / qn_{tot} \tag{15}$$

where  $q$  is the number of experiment times, each one being performed on random partitioning of the database into two sets,  $n_{mis}^i$  is the number of misclassifications for the  $i$ th time, and  $n_{tot}$  is the number of total testing patterns of each time. Using the criterion of  $E_{ave}$ , comparisons with CNN [11], NFL [25] and M-PCA [26] performed in the same ORL database are shown in Table 1. For our proposed method, the best error rate is based on five times, and the feature dimension is 25 and 30, respectively.

TABLE 1 ERROR RATE OF DIFFERENT APPROACHES

Approach	Times of simulations	$E_{ave}$ (%)
CNN	3	3.83
NFL	4	3.125
M-PCA	10	2.4
Our proposed	5	1.98

#### V. CONCLUSIONS

In this paper, an efficient approach for designing a neural classifier to cope with high-dimensional problems in face recognition is presented. The face features are first extracted by the PCA paradigm so that the resulting data are compressed significantly. Here, we preserve all positive principal components so that no statistical information of original images is lost. Then, the information is further condensed via the WFLD approach, which fully considers the influence of outlier classes and reduces the overlapping between the patterns. We present a new covering algorithm called max-density covering learning algorithm to design the classifier. The training problem of neural networks may be transformed into the covering problem of a point set, which avoids the iterative process, and overcomes the problem of longtime training of classical neural networks. At the same time, there is no need to guess the size, depth, and connectivity pattern of the network in advance. A reasonably small net is built automatically.

#### REFERENCES

- [1] R.,Chellappa Wilson C. L. and Sirohey S, "Human and machine recognition of faces: A survey," in *Proc. IEEE*, vol.83, pp. 705-740, 1995.
- [2] Y. Moses, Y. Adini, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," in *Proc. EuroP. Conf. Comput. Vision*, vol. A, 1994, pp. 286-296.

- [3] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1042–1053, 1993.
- [4] P. J. Phillips, "Matching pursuit filters applied to face identification," *IEEE Trans. Image Processing*, vol. 7, pp. 1150–1164, 1998.
- [5] Z. Hong, "Algebraic feature extraction of image for recognition," *Pattern Recognition*, vol. 24, pp. 211–219, 1991.
- [6] M. A. Turk and A. P. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, pp. 71–86, 1991.
- [7] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces versus fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 711–720, 1997.
- [8] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 831–836, 1996.
- [9] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivariate data projection," *IEEE Trans. Neural Networks*, vol. 6, pp. 296–317, Mar. 1995.
- [10] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Networks*, vol. 8, pp. 98–113, Jan. 1997.
- [11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic Press, 1990.
- [12] H. H. Song and S. W. Lee, "A self-organizing neural tree for large-set pattern classification," *IEEE Trans. Neural Networks*, vol. 9, pp. 369–380, Mar. 1998.
- [13] D. Valentin, H. Abdi, A. J. O'Toole, and G. W. Cottrell, "Connectionist models of face processing: A Survey," *Pattern Recognition*, vol. 27, pp. 1209–1230, 1994.
- [14] L. Zhang and B. Zhang, "A geometrical representation of McCulloch-Pitts neural model and its application," *IEEE Trans. Neural Networks*, vol. 10, pp. 925–929, 1999.
- [15] L. Zhang and B. Zhang, "A geometrical representation of M-P neural model and its applications," *Journal of Software*, vol. 9(5), pp. 334–338, 1998.
- [16] L. Zhang and B. Zhang, "An alternative covering design algorithm of multi-layer neural networks," *Journal of Software*, vol. 10(7), pp. 737–742, 1999.
- [17] M. Wu and B. Zhang, "A neural network algorithm for large scale pattern recognition problems," *Journal of Software*, vol. 12(6), pp. 851–855, 2001.
- [18] P. Tao, B. Zhang and Z. Ye, "An incremental bi-covering learning algorithm for constructive neural networks," *Journal of Software*, vol. 14(2), pp. 194–201, 2003.
- [19] B. Zhang, M. Wu and W. B. Zheng, "A fuzzy classifier based on the constructive covering approach in neural networks" *Journal of Software*, vol. 14(3), pp. 429–434, 2003.
- [20] J. Meng, "Face recognition with radial basis function (RBF) neural networks," *IEEE Trans. Neural Networks*, vol. 13(3), pp. 296–317, 2002.
- [21] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in neurons activity. *Bull. Math. Biophys.*, 1943, 5: 115–133.
- [22] Z. Jin and J.Y. Yang, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, vol. 34(7), pp. 1405–1416, 2001.
- [23] J. Yang and J.Y. Yang, "Why can LDA be performed in PCA transformed space?" *Pattern Recognition*, vol. 36, pp. 563–566, 2003.
- [24] M. Loog. Approximate Pairwise accuracy criteria for Multiclass linear dimension reduction: generalization of the Fisher criterion. Delft Univ. Press, 1999.
- [25] S. Z. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Networks*, vol. 10, pp. 439–443, 1999.
- [26] V. Brennan and J. Principe, "Face classification using a multiresolution principal component analysis," in *Proc. IEEE Workshop Neural Networks signal Processing*, pp. 506–515, 1998.