Face Recognition With Radial Basis Function (RBF) Neural Networks

Meng Joo Er, *Member, IEEE*, Shiqian Wu, *Member, IEEE*, Juwei Lu, *Student Member, IEEE*, and Hock Lye Toh, *Member, IEEE*

***Abstract*—A general and efficient design approach using a radial basis function (RBF) neural classifier to cope with small training sets of high dimension, which is a problem frequently encountered in face recognition, is presented in this paper. In order to avoid overfitting and reduce the computational burden, face features are first extracted by the principal component analysis (PCA) method. Then, the resulting features are further processed by the Fisher's linear discriminant (FLD) technique to acquire lower-dimensional discriminant patterns. A novel paradigm is proposed whereby data information is encapsulated in determining the structure and initial parameters of the RBF neural classifier before learning takes place. A hybrid learning algorithm is used to train the RBF neural networks so that the dimension of the search space is drastically reduced in the gradient paradigm. 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.**

***Index Terms*—Face recognition, Fisher's linear discriminant, ORL database, principal component analysis, radial basis function (RBF) neural networks, small training sets of high dimension.**

I. Introduction

MACHINE recognition of human face from still and video images has become an active research area in the communities of image processing, pattern recognition, neural networks and computer vision. This interest is motivated by wide applications ranging from static matching of controlled format photographs such as passports, credit cards, driving licenses, and mug shots to real-time matching of surveillance video images presenting different constraints in terms of processing requirements [1]. Although researchers in psychology, neural sciences and engineering, image processing and computer vision have investigated a number of issues related to face recognition by human beings and machines, 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 two-fold: 1) Face images are highly variable and 2) Sources of variability include individual appearance, three-dimensional (3-D) 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?

For 1), many successful face detection and feature extraction paradigms have been developed [3]–[12]. The frequently used approaches are to use geometrical features, where the relative positions and shapes of different features are measured [3], [4]. At the same time, several paradigms have been proposed to use global representation of a face, where all features of a face are automatically extracted from an input facial image [5]–[12]. It has been indicated in [4] that these algorithms with global encoding of a face are fast in face recognition. In [5], singular value decomposition (SVD) of a matrix 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, in which each coordinate is actually an image that is called an eigenpicture. However, the eigenface paradigm, which uses principal component analysis (PCA), yields projection directions that maximize the total scatter across all classes, i.e., across all face images. In choosing the projection which maximizes the total scatter, the PCA retains unwanted variations caused by lighting, facial expression, and other factors [7]. Accordingly, the features produced are not necessarily good for discrimination among classes. In [7], [8], the face features are acquired by using the fisherface or discriminant eigenfeature paradigm. This paradigm aims at overcoming the drawback of the eigenface paradigm by integrating Fisher's linear discriminant (FLD) criteria, while retaining the idea of the eigenface paradigm in projecting faces from a high-dimension image space to a significantly lower-dimensional feature space. Instead of using statistical theory, neural-networks-based feature extraction has been reported recently [9]–[12]. The goal of face processing using neural networks is to develop a compact internal representation of faces, which is equivalent to feature extraction. Therefore, the number of hidden neurons is less than that in either input or output layers, which results in the network encoding inputs in a smaller dimension that retains most of the important information. Then, the hidden units of the neural net-

Manuscript received March 29, 1999; revised March 5, 2001 and December 5, 2001.

M. J. Er is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore.

S. Wu and H. L. Toh are with the Centre for Signal Processing, Innovation Centre, Singapore 637722, Singapore.

J. Lu is with Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada.

Publisher Item Identifier S 1045-9227(02)03984-X.

work can serve as the input layer of another neural network to classify face images.

In many pattern recognition systems, the methodology frequently used is the statistical approach, whereby decision theory derived from statistics of input patterns is used to design a classifier [13]. Although this paradigm has been successfully applied to solve various problems in pattern classification, it has difficulty in expressing structural information unless an appropriate choice of features is made possible. Furthermore, this approach requires much heuristic information to design a classifier [14]. Neural-networks-based paradigms, as new means of implementing various classifiers based on statistical and structural approach, have been proven to possess many advantages for classification because of their learning ability and good generalization [9]–[12], [14]–[16]. Generally speaking, multilayered networks (MLNs), usually coupled with the backpropagation (BP) algorithm, are most widely used in face recognition [9]. 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, RBF neural networks have recently attracted extensive interests in the community of neural networks for a wide range of applications [17]–[29]. The salient features of RBF neural networks are as follows.

- They are universal approximators [17].
- They possess the best approximation property [18].
- Their learning speed is fast because of locally tuned neurons [19].
- They have more compact topology than other neural networks [20].

Normally, RBF neural networks are widely used for function approximation and pattern recognition wherein the pattern dimension in these applications is usually small. As pointed out by Moody and Darken [19], "RBF neural networks are best suited for learning to approximate continuous or piecewise continuous, real-valued mapping where the input dimension is sufficiently small." When RBF neural networks are implemented in face recognition, such systems possess the following characteristics:

- High dimension. For example, a 128×128 image will have 16 384 features.
- Small sample sets. The sample patterns are very few for each class, say, only one–ten images per person so that $n \ll r$ (n is the number of training patterns, r is the number of features), which is more severe than the case shown in [16].

Therefore, face recognition is substantially different from classical pattern recognition problem, for instance, character recognition [14], in which there are a limited number of classes with a large number of training patterns in each class. This situation leads to the following challenges in designing an RBF neural classifier:

1) *Overfitting problem*. It has been indicated that if the dimension of the network input is comparable to the size of the training set, the system is liable to overfitting and result in poor generalization [16].
2) *Overtraining problem*. High dimension of the network input results in complex optimal processing and slow convergence. Hence, it is likely to cause overtraining.

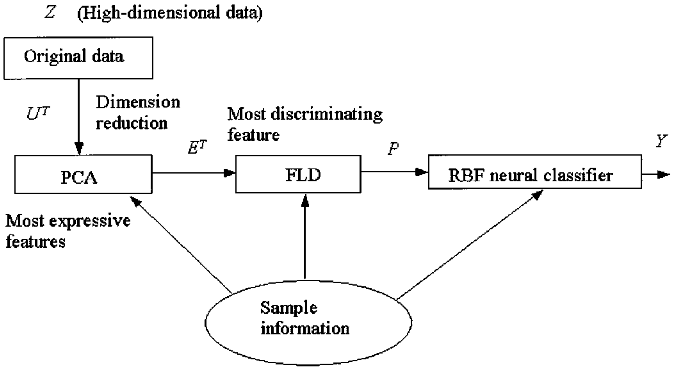


Fig. 1. Schematic diagram of RBF neural classifier for small training sets of high dimension.

3) *Small-sample effect*. It has been indicated that small sample can easily contaminate the design and evaluation of a proposed system [30]. For applications with a large number of features and a complex classification rule, the training sample size must be quite large [30]. It has been further pointed out that the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities as the dimension increases [31].
4) *Singular problem*. If n is less than $r+1$, the sample covariance matrix is singular, and therefore unusable regardless of the true value of the covariance matrix [32].

To circumvent the aforementioned problems, a systematic methodology for RBF neural classifier design to deal with small training sets of high-dimensional feature vectors is presented, as shown in Fig. 1. The proposed methodology comprises the following 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 FLD 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 clustering algorithm concerning category information of training samples is proposed so that homogeneous data could be clustered and a compact structure of an RBF neural classifier with limited mixed data could be achieved; 3) Two important criteria are proposed to estimate the initial widths of RBF units which control the generalization of RBF neural classifier; and 4) A hybrid learning algorithm is presented to train the RBF neural networks so that the dimension of the search space is significantly reduced in the gradient paradigm.

The rest of this paper is organized as follows. Section II presents the architecture of RBF neural networks and the related design problems when they are used as a classifier. Section III provides the procedure of extracting face features. In Section IV, we propose a systemic approach for structure determination and initialization of RBF neural networks. A hybrid learning algorithm is developed in Section V. Experimental results are demonstrated in Section VI. In Section VII, we discuss some important issues concerning performances of the proposed approach and provide more insights into several paradigms, which are closely related to our proposed paradigm. Finally, conclusions are drawn in Section VIII.

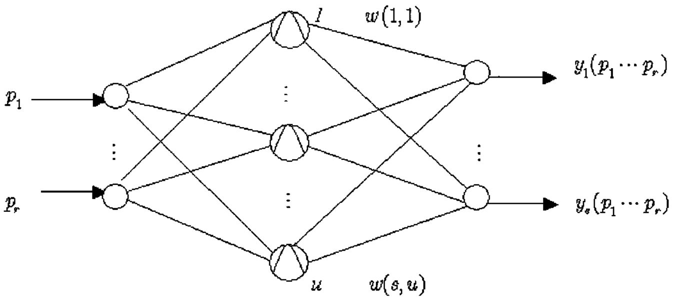


Fig. 2. RBF neural networks.

II. RBF Neural Networks

An RBF neural network, shown in Fig. 2, can be considered as a mapping: $\Re^r \rightarrow \Re^s$.

Let $P \in \Re^r$ be the input vector and $C_i \in \Re^r$ $(1 \leq i \leq u)$ be the prototype of the input vectors. The output of each RBF unit is as follows:

$$R_i(P) = R_i(||P - C_i||) \quad i = 1, \ldots, u \tag{1}$$

where $|| \cdot ||$ indicates the Euclidean norm on the input space. Usually, the Gaussian function is preferred among all possible radial basis functions due to the fact that it is factorizable. Hence

$$R_i(P) = \exp\left[-\frac{||P - C_i||^2}{\sigma_i^2}\right] \tag{2}$$

where σ_i is the width of the ith RBF unit. The jth output $y_j(P)$ of an RBF neural network is

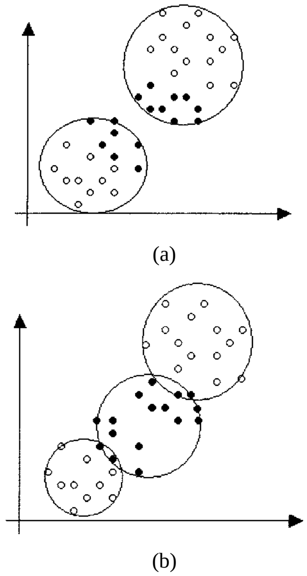
$$y_i(P) = \sum_{i=1}^{u} R_i(P) \times w(j,i) \tag{3}$$

where $R_0 = 1$, $w(j,i)$ is the weight or strength of the ith receptive field to the jth output and $w(j,0)$ is the bias of the jth output. In order to reduce the network complexity, the bias is not considered in the following analysis.

We can see from (2) and (3) that the outputs of an RBF neural classifier are characterized by a linear discriminant function. They generate linear decision boundaries (hyperplanes) in the output space. Consequently, the performance of an RBF neural classifier strongly depends on the separability of classes in the u-dimensional space generated by the nonlinear transformation carried out by the u RBF units.

According to Cover's theorem on the separability of patterns wherein a complex pattern classification problem cast in a high-dimensional space nonlinearly is more likely to be linearly separable than in a low-dimensional space [33], the number of Gaussian nodes $u \geq r$, where r is the dimension of input space. On the other hand, the increase of Gaussian units may result in poor generalization because of overfitting, especially, in the case of small training sets [16]. It is important to analyze the training patterns for the appropriate choice of RBF hidden nodes.

Geometrically, the key idea of an RBF neural network is to partition the input space into a number of subspaces which are in the form of hyperspheres. Accordingly, clustering algorithms (k-means clustering, fuzzy k-means clustering and hierarchical clustering) which are widely used in RBF neural networks [19], [21], are a logical approach to solve the problems [19], [22]. However, it should be noted that these clustering approaches are inherently unsupervised learning algorithms as no category information about patterns is used. As an illustrative example, consider a simple training set (x_k, y_k) illustrated in Fig. 3. The black and white data points reflect the corresponding values assumed by the dependent variable y_k. If we simply use k-means clustering approach without considering y_k, two evident clusters as shown in Fig. 3(a) are achieved. This brings about significant misclassification initially. Although the clustering boundaries are modified in the subsequent learning phase, this could easily lead to an undesired and highly dominant averaging phenomenon as well as to make the learning less effective [21]. To preserve homogeneous clusters, three clusters as depicted in Fig. 3(b) should be created. In other words, a supervised clustering procedure which takes into consideration the category information of training data should be considered.



(a)

(b)

Fig. 3. Two-dimension patterns and clustering: (a) conventional clustering, (b) clustering with homogeneous analysis.

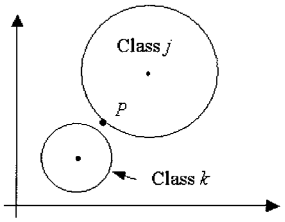


Fig. 4. Effect of Gaussian widths in clustering.

While considering the category information of training patterns, it should be emphasized that the class memberships are not only depended on the distance of patterns, but also depended on the Gaussian widths. As illustrated in Fig. 4, P is near to the center of class k in Euclidean distance, but we can select different Gaussian widths for each cluster so that the point p has greater class membership to class j than that to class k. Therefore, the use of class membership implies that we should propose a supervised procedure to cluster the training patterns and determine the initial Gaussian widths, and this work will be elaborated in Section IV.

III. Extraction of Face Features

A. Principal Component Analysis (PCA)

Let a face image Z_i be a two-dimensional $m \times m$ array of intensity values. An image may also be considered as a vector of dimension m^2. Denote the training set of n face images by $Z = (Z_1, Z_2, \ldots, Z_n) \subset \Re^{m^2 \times n}$, and we assume that each image belongs to one of c classes. Define the covariance matrix as follows [6], [13]:

$$\begin{aligned}\Gamma =& \frac{1}{n}\sum_{i=1}^{n}(Z_i - \bar{Z})(Z_i - \bar{Z})^T \\ =& \Phi\Phi^T\end{aligned} \tag{4}$$

where $\Phi = (\Phi_1, \Phi_2, \ldots, \Phi_n) \subset \Re^{m^2 \times n}$ and $\bar{Z} = (1/n)\sum_{i=1}^{n} Z_i$. Then, the eigenvalues and eigenvectors of the covariance Γ are calculated. Let $U = (U_1, U_2, \ldots, U_r) \subset \Re^{m^2 \times n}$ $(r < n)$ be the r eigenvectors corresponding to the r largest eigenvalues. Thus, for a set of original face images $Z \subset \Re^{m^2 \times n}$, their corresponding eigenface-based feature $X \subset \Re^{r \times n}$ can be obtained by projecting Z into the eigenface space as follows:

$$X = U^T Z. \tag{5}$$

B. Fisher's Linear Discriminant (FLD)

Actually, the PCA paradigm does not provide any information for class discrimination but dimension reduction. Accordingly, the FLD is applied to the projection of the set of training samples in the eigenface space $X = (X_1, X_2, \ldots, X_n) \subset \Re^{r \times n}$. The paradigm finds an optimal subspace for classification in which the ratio of the between-class scatter and the within-class scatter is maximized [7], [8], [13]. Let the between-class scatter matrix be defined as

$$S_B = \sum_{i=1}^{c} n^i(\bar{X}^i - \bar{X})(\bar{X}^i - \bar{X})^T \tag{6}$$

and the within-class scatter matrix be defined as

$$S_W = \sum_{i=1}^{c}\sum_{X_i \in n^i}\left(X_k - \bar{X}^i\right)(X - \bar{X}^i)^T \tag{7}$$

where $\bar{X} = (1/n)\sum_{j=1}^{n} X_j$ is the mean image of the ensemble, and $\bar{X}^i = (1/n^i)\sum_{j=1}^{n^i} X_j^i$ is the mean image of the ith class, n^i is the number of samples in the ith class, and c is the number of classes. The optimal subspace, E_{optimal} by the FLD is determined as follows [7], [8], [13]:

$$E_{\text{optimal}} = \arg\max_{E} \frac{|E^T S_B E|}{|E^T S_W R|} = [e_1, e_2, \ldots, e_{c-1}] \tag{8}$$

where $[e_1, e_2, \ldots, e_{c-1}]$ is the set of generalized eigenvectors of S_B and S_W corresponding to the $c-1$ largest generalized eigenvalues $\lambda_i,\ i = 1, 2, \ldots, c-1$, i.e.,

$$S_B E_i = \lambda_i S_W E_i \quad i = 1, 2, \ldots, c-1. \tag{9}$$

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

$$P = E_{\text{optimal}}^T \cdot U^T \cdot Z. \tag{10}$$

Remarks:

1) From (7), we see $\operatorname{rank}(S_w) \le \min\{r, c(n^i - 1)\}$. In order to prevent S_w from becoming singular, the value of r should be no more than $n - c$.
2) Also we can see from (6) that $\operatorname{rank}(S_b) \le \min\{r, c-1\}$. Accordingly, there are at most $c-1$ nonzero generalized eigenvectors. In other words, the FLD transforms the r-dimension space into $(c-1)$-dimension space to classify c classes of objects.
3) It should be noted that the FLD is a linear transformation which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples. The results are globally optimal only for linear separable data. The linear subspace assumption is violated for the face data that have great overlappings [34]. Moreover, the separability criterion is not directly related to the classification accuracy in the output space [34].
4) Several researchers have indicated that the FLD method achieved the best performance on the training data, but generalized poorly to new individuals [35], [36].

Therefore, RBF neural networks, as a nonlinear alternative with good generalization, have been proposed for face classification. In the sequel, we will use the feature vectors P instead of their corresponding original data Z in Sections IV–VIII.

IV. Structure Determination and Initialization of RBF Neural Networks

A. Structure Determination and Choice of Prototypes

From the point of view of face recognition, a set of optimal boundaries between different classes should be estimated by RBF neural networks. Conversely, from the point of view of RBF neural networks, the neural networks are regarded as a mapping from the feature hyperspace to the classes. Each pattern is represented by a real vector and each class is assigned for a suitable code. Therefore, we set:

- the number of inputs to be equal to that of features (i.e., the dimension of the input space);
- the number of outputs to be equal to that of classes (see Fig. 2).

It is cumbersome to select the hidden nodes. Different approaches revolving around increasing or decreasing the complexity of the architecture have been proposed [19]–[28]. Many researchers have illustrated that the number of hidden units depends on the geometrical properties of the training patterns as well as the type of activation function [24]. Nevertheless, this is still an open issue in implementing RBF neural networks. Our proposed approach is as follows.

1) Initially, we set the number of RBF units to be equal to that of the output, $u = s$, i.e., we assume that each class has only one cluster.

2) For each RBF unit k, $k = 1, 2, \ldots, u$, the center is selected as the mean value of the sample patterns belonging to class k, i.e.,

$$C^k = \frac{1}{n^k} \sum_{i=1}^{n^k} P_i^k \quad k = 1, 2, \ldots, u \tag{11}$$

where P_i^k is the ith sample belonging to class k and n^k is the total number of training patterns in class k.

3) For any class k, compute the Euclidean distance d_k from the mean C^k to the furthest point $P^k(f)$ belonging to class k, i.e.,

$$d_k = \|P^k(f) - C^k\|. \tag{12}$$

4) For any class k:
   - Calculate the distance $dc(k, j)$ between the mean of class k and the mean of other classes as follows:

$$dc(k, j) = \|C^k - C^j\| \quad j = 1, 2, \ldots, s, \ j \neq k \tag{13}$$

   - Find

$$d_{\min}(k, l) = \arg\min_{l}(dc(k, j)) \quad j = 1, 2, \ldots, s, \ j \neq k \tag{14}$$

   - Check the relationship between $d_{\min}(k, l)$ and d_k, d_l.

     Case 1) *No overlapping.* If $d_k + d_l \le d_{\min}(k, l)$, class k has no overlapping with other classes [see Fig. 5(a)].

     Case 2) *Overlapping.* If $d_k + d_l > d_{\min}(k, l)$, class k has overlapping with other classes and misclassification may occur in this case. Fig. 5(b) represents the case that $d_k + d_l > d_{\min}(k, l)$ and $|d_k - d_l| < d_{\min}(k, l)$, while Fig. 5(c) depicts the case that $d_k + d_l > d_{\min}(k, l)$ and $|d_k - d_l| \ge d_{\min}(k, l)$.

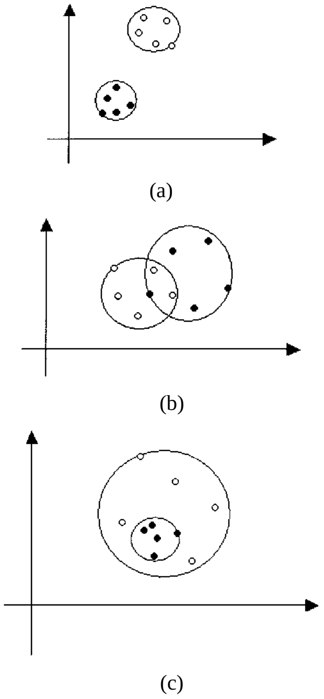
5) Splitting Criteria:

   i) ***Embody Criterion***: If class k is embodied in class l completely, i.e., $d_k + d_l > d_{\min}(k, l)$ and $|d_k - d_l| \ge d_{\min}(k, l)$, class l will be split into two clusters, see Fig. 6.

   ii) ***Misclassified Criterion***: If class k contains many data of other classes (in the following experiment, this implies that if the number of misclassified data in class k is more than one), then class k will be split into two clusters.

   If class k satisfies one of the above conditions, class k will be split into two clusters in which the centers are calculated based on their corresponding sample patterns according to (11).

6) Repeat (2)–(5), until all the training sample patterns meet the above two criteria.


(a)

(b)

(c)

Fig. 5. Clusters and distribution of sample patterns: (a) $d_k + d_l \le d_{\min}(k, l)$. (b) $d_k + d_l > d_{\min}(k, l)$ and $|d_k - d_l| < d_{\min}(k, l)$. (c) $d_k + d_l > d_{\min}(k, l)$ and $|d_k + d_l| \ge d_{\min}(k, l)$.

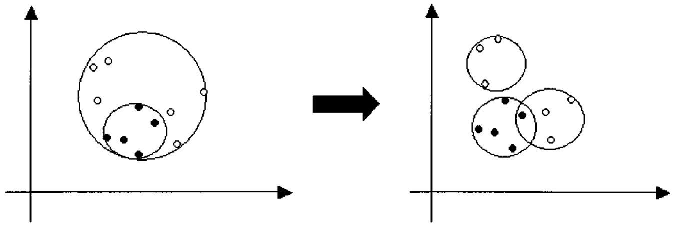


Fig. 6. Splitting of one class into two clusters.

B. Estimation of Widths

Essentially, RBF neural networks overlap localized regions formed by simple kernel functions to create complex decision regions while the amount of overlapping is controlled by the widths of RBF units [22], [24]. If no overlapping occurs, the system will not give meaningful outputs for inputs between the inputs for which the system is designed, i.e., the RBF units do not generalize well. However, if the widths are too large, the interaction of different classes will be great and the output belonging to the class will not be so significant [25], while the output of other classes may be large so that it will lead to misclassification greatly. Hence, our goal is to select the widths in such a way that they would minimize overlapping of nearest neighbors of different classes to preserve local properties, as well as maximize the generalization ability of the network [25]. Here, we present a general approach to select the widths of an RBF neural classifier according to two criteria.

1) ***Majority Criterion***: The majority criterion can be described as follows: In any class, each datum should have more than 50% confidence level for the class it belongs to. The detailed calculations are presented as follows:

   First, d_k, the distance from the mean to the furthest point belonging to class k, is calculated according to (11) and (12).

Next, define the width σ_w^k of class k considering the confidence level as

$$\sigma_W^k = \frac{d_k}{\sqrt{|\ln \beta|}} \tag{15}$$

where β is called the confidence coefficient, which lies in the range $0.5 \leq \beta < 1$.

2) ***Overlapping Criterion***: The overlapping criterion can be described as follows: For any class k, the choice of σ_B^k considering the overlapping of the nearest class l is determined by

$$\sigma_B^k = \eta \times d_{\min}(k, l) \tag{16}$$

where η is an overlapping factor that controls the overlap of different classes, $d_{\min}(k, l)$ is the minimum distance between the center of class k and centers of other classes.

Then, the width of class k is finally determined by

$$\sigma^k = \max\left(\sigma_W^k, \sigma_B^k\right). \tag{17}$$

The key idea of this approach is to consider not only the intra-data distribution but also the inter-data variations.

In order to efficiently determine the width σ^k, the parameter η could be approximately estimated as follows:

$$\eta \approx \frac{\sum_{i=1}^{c} \frac{d_k}{\sqrt{|\ln \beta|}}}{\sum_{k=1}^{c} d_{\min}(k, l)}. \tag{18}$$

The choice of β is determined by the distribution $(dc(k, j), d_k)$ of sample patterns. If the data are scattered largely, but the centers are close, a small β should be selected as demonstrated in Table IV. Normally, η lies in the range $0.7 \leq \eta \leq 1.8$. The best values of β and η are selected when the best performance is achieved for training patterns.

V. Hybrid Learning Algorithm

The adjustment of RBF unit parameters is a nonlinear process while the identification of weight $w(i, j)$ is a linear one. Though we can apply the gradient paradigm to find the entire set of optimal parameters, the paradigm is generally slow and likely to become trapped in local minima. Here, a hybrid learning algorithm, which combines the gradient paradigm and the linear least square (LLS) paradigm to adjust the parameters, is presented.

A. Weight Adjustment

Let r and s be the number of inputs and outputs respectively, and suppose that u RBF units are generated according to the above clustering algorithm for all training patterns. For any input P_i, the jth output y_j of the system is

$$y_j^i(P_i) = \sum_{k=1}^{u} w(j, k) \times R_k \tag{19}$$

or

$$Y = WR. \tag{20}$$

Given $R \in \Re^{u \times n}$ and $T = (T_1, T_2, \ldots, T_n)^T \in \Re^{s \times n}$, where n is the total number of sample patterns, T is the target matrix consisting of "1's" and "0's" with exactly one per column that identifies the processing unit to which a given exemplar belongs, find an optimal coefficient matrix $W^* \in \Re^{s \times u}$ such that the error energy $\tilde{E}^T \tilde{E} = (T - Y)^T (T - Y)$ is minimized. This problem can be solved by the LLS method [15]

$$W^* = T(R^T R)^{-1} R^T \tag{21}$$

where R^T is the transpose of R, and $R^+ = (R^T R)^{-1} R^T$ is the pseudoinverse of R.

B. Modification of Parameters of RBF Units

Here, the parameters (centers and widths) of the prototypes are adjusted by taking the negative gradient of the error function E^l

$$E^l = \frac{1}{2} \sum_{k=1}^{s} \left(t_k^l - y_k^l\right)^2 \quad l = 1, 2, \ldots, n \tag{22}$$

where y_k^l and t_k^l represent the kth real output and the target output at the lth pattern, respectively. The error rate for each output y_k^l can be calculated readily from (22)

$$\frac{\partial E^l}{\partial y_k^l} = -\left(t_k^l - y_k^l\right). \tag{23}$$

For the internal nodes (center C and width σ), the error rate can be derived by the chain rule as follows [15]:

$$\begin{aligned} \Delta C^l(i, j) &= -\xi \frac{\partial E^l}{\partial C^l(i, j)} \\ &= -\xi \frac{\partial E^l}{\partial y_k^l} \frac{\partial y_k^l}{\partial R_j^l} \frac{\partial R_j^l}{\partial C^l(i, j)} \\ i &= 1, 2, \ldots, r, \quad j = 1, \ldots, u \\ &= 2\xi \sum_{k=1}^{s} \left(t_k^l - y_k^l\right) \cdot w^l(k, j) \cdot R_j^l \\ &\quad \cdot \frac{P(i, l) - C^l(i, j)}{\left(\sigma_j^l\right)^2} \end{aligned} \tag{24}$$

$$\begin{aligned} \Delta \sigma_j^l &= -\xi \frac{\partial E^l}{\partial \sigma_j^l} \\ &= -\xi \frac{\partial E^l}{\partial y_k^l} \frac{\partial y_k^l}{\partial R_j^l} \frac{\partial R_j^l}{\partial \sigma_j^l} \quad j = 1, 2, \ldots, u \\ &= 2\xi \sum_{k=1}^{s} \left(t_k^l - y_k^l\right) \cdot w^l(k, j) \cdot R_j^l \cdot \frac{\|P_l - C^j\|^2}{\left(\sigma_j^l\right)^3} \end{aligned} \tag{25}$$

where $\Delta C^l(i, j)$ is the central error rate of the ith input variable of the jth prototype at the lth training pattern, $\Delta \sigma_j^l$ is the width error rate of the jth prototype at the lth pattern, $P(i, l)$ is the ith input variable at the lth training pattern and ξ is the learning rate.

C. Learning Procedure

In the forward pass, we supply input data and functional signals to calculate the output R_j of the jth RBF unit. Then, the weight W is modified according to (21). After identifying the weight, the functional signals continue going forward till the error measure is calculated. In the backward pass, the errors propagate from the output end toward the input end. Keeping the weight fixed, the centers and widths of RBF nodes are modified according to (24) and (25). The learning procedure is illustrated in Table I.

Remarks:

1) If we fix the parameters of the RBF units, the weights found by the LLS are guaranteed to be global optimum. Accordingly, the dimension of the search space is drastically reduced in the gradient paradigm so that this hybrid learning algorithm converges much faster than the gradient descent paradigm.
2) It is well known that the learning rate ξ is sensitive to the learning procedure. If ξ is small, the BP algorithm will closely approximate the gradient path, but the convergence speed will be slow since the gradient must be calculated many times. On the other hand, if ξ is large, convergence speed will be very fast initially, but the algorithm will oscillate around the optimum value. Here, we propose an approach so that ξ will be reduced gradually. We compute

$$\xi = \max(lr_{\max} \cdot \gamma^i, lr_{\min}) \tag{26}$$

where $lr_{\max}$, $lr_{\min}$ are maximum and minimum learning rates, respectively, i is the number of epochs, and γ is a descent coefficient which lies in the range $0 < \gamma < 1$.
3) As the widths are sensitive to the generalization of an RBF neural classifier, a larger learning rate is adopted for width adjustment than for center modification (twice as that for center modification).
4) As the system with high dimension is liable to overtraining, the early stop strategy in [24] is adopted.

TABLE I
TWO PASSES IN THE HYBRID LEARNING PROCEDURE

	Forward pass	Backward pass
Weight	LLS	Fixed
Parameters of RBF units	Fixed	Gradient descent
Signals	Node outputs	Error rate

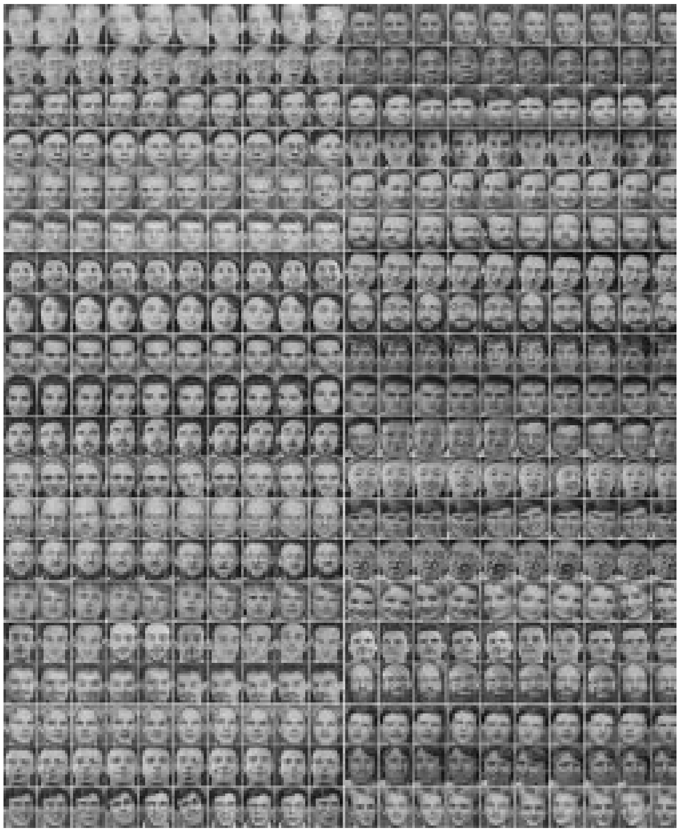


Fig. 7. The ORL face database.

VI. Experimental Results

A. ORL Database

Our experiments were performed on the face database which contains a set of face images taken between April 1992 and April 1994 at the Olivetti Research Laboratory (ORL) in Cambridge University, U.K. There are 400 images of 40 individuals. For some subjects, the images were taken at different times, which contain quite a high degree of variability in lighting, facial expression (open/closed eyes, smiling/nonsmiling etc), pose (upright, frontal position etc), and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position, with tolerance for some tilting and rotation of up to 20°. The variation in scale is up to about 10%. All the images in the database are shown in Fig. 7.[1]

In the following experiments, a total of 200 images were randomly selected as the training set and another 200 images as the testing set, in which each person has five images. Next, the training and testing patterns were exchanged and the experiment was repeated one more time. Such procedure was carried out several times.

B. Clustering Error Before Learning

The structure of RBF neural networks and parameters of prototypes are obtained according to the algorithm shown in Section IV. In order to test how the clustering algorithm works, the data distributions on six simulations are illustrated in Table II and the misclustering number based on different dimensions of feature patterns in six runs of simulations before learning are listed in Table III. We see from Table II that there are a total of five classes which are in well-separated distribution as depicted in Fig. 5(a) and 235 classes are distributed as shown in Fig. 5(b) when the dimension of the feature vectors is 39. The separation of data becomes better and better as the dimension decreases. Correspondingly, the clustering performance is better when the number of feature vectors reduces, as shown in Table III. However, as we will see later, it does not imply that the recognition performance will improve along with reduction in the dimension. We also find from Table III that the maximum misclus-

[1] The ORL database is available from http//www.cam-orl.co.uk/face-database.html.

TABLE II
DATA DISTRIBUTION BASED ON THE PROPOSED APPROACH BEFORE LEARNING (THE RESULTS ARE OBTAINED BASED ON SIX SIMULATIONS)

Feature dimension	Data distribution		
	No of clusters	No of well-separated classes	No of overlapping classes
39	40	5	235
35	40	19	221
30	40	48	192
25	40	96	144
20	40	140	100
10	40	187	53

TABLE III
CLUSTERING ERRORS FOR TRAINING PATTERNS BEFORE LEARNING (THE RESULT IS THE SUM OF SIX SIMULATIONS)

Feature dimension	No. of clusters	No. of misclustering	Maximum number of misclustering within a class
39	40	18	2
35	40	11	1
30	40	5	1
25	40	2	1
20	40	0	1
10	40	0	1

TABLE IV
SPECIFIED PARAMETERS AND CLASSIFIED PERFORMANCE

Feature dimension	Training phase					Testing phase
	No of RBF units	β	η	Epochs	RMSE *	NOM **
39	40	0.5~0.6	1.0~1.2	20~200	0.049~0.046	5
35	40	0.55~0.65	1.0~1.2	20~200	0.048~0.045	5
30	40	0.55~0.7	0.78~1.1	20~120	0.047~0.041	2
25	40	0.7~0.85	0.99~1.4	20~150	0.052~0.045	2
20	40	0.8 ~0.9	1.1~1.6	20~150	0.052~0.042	6
10	40	0.8~0.95	0.9~1.9	20~150	0.058~0.041	9

* RMSE—Root Mean Squared Error

** NOM—Number of Misclassifications

TABLE V
CLASSIFIED PERFORMANCES ON SIX SIMULATIONS

Dimension	39	35	30	25	20	10
Total NOM	36	29	23	23	27	58

TABLE VI
BEST PERFORMANCES ON 6 SIMULATIONS (DIMENSION $r = 25$ OR 30)

Simulation	1	2	3	4	5	6
NOM	2	6	6	2	1	6

tering number within a class for each simulation is one except for one case which occurred when the dimension is 39. We can conclude that the clustering paradigm presented is good for all cases.

C. Error of Classification After Learning

After the structure of RBF neural networks and parameters of prototypes are selected, the hybrid learning algorithm presented in Section V is employed to train the network. One run of the recognition results is shown in Table IV.

From Table IV, we can see that:

1) If the information is sufficient (feature dimension is larger than 20), the results are stable in each case for different choice of initial parameters β and η in terms of the number of misclassifications. Otherwise, the error rate will increase drastically.
2) On the other hand, it does not mean that more information (dimension is larger than 30) will result in higher performance. The reason may be that high dimension will lead to complexity in structure and increase difficulty in learning. Moreover, the addition of some unimportant information may become noise and degrade the performance. The best results are achieved when the dimension is 25–30.
3) Along with the increase in the feature dimension, the training patterns have more overlapping, and a small β should be selected.[2]

The total results based on six simulations are summarized in Tables V and VI.

D. Comparisons With Other Approaches

Recently, a number of researchers use the ORL database to verify their algorithms [11], [12], [29], [37]–[45]. Here, we adopt the same definition of average error rate, E_{ave} used in [11], [37], which is given by

$$E_{\text{ave}} = \frac{\sum_{i=1}^{q} n_{\text{mis}}^{i}}{q n_{\text{tot}}} \tag{27}$$

where q is the number of experimental runs, each one being performed on random partitioning of the database into two sets, n_{mis}^{i} is the number of misclassifications for the ith run, and n_{tot} is the number of total testing patterns of each run. Using the criterion of E_{ave}, comparisons with CNN [11], NFL [37] and M-PCA [38] performed on the same ORL database are shown in Table VII.

Here, the best value of E_{ave} for the CNN is based on three runs of experiments, and the SOM size is 8 and 9. For NFL, the best error rate is obtained when the number of feature vectors is 40, and the average error rate is evaluated based on four runs of experiments. For M-PCA, it was reported that the overall performance is the average of ten runs of experiments. For our proposed paradigm, the best error rate is based on six runs, and the feature dimension is 25 and 30, respectively. The face features are the same as [37], and the way to partition the training set and query set is the same as the methods in [11] and [37].

[2] It should be noted that β lies in the range $0.5 \leq \beta < 1$. Even the RMSE is smaller in the case of $\beta < 0.5$ when the dimension is 39, the generalization will be very bad.

TABLE VII
ERROR RATES OF DIFFERENT APPROACHES

Approach	No. of simulations	E_{ave} (%)
CNN	3	3.83
NFL	4	3.125
M-PCA	10	2.4
Our proposed RBF	6	1.92

Some other results recently performed on the ORL database are listed in Table VIII as references (these results are tabulated separately from Table VII because we are not aware of how their experiments are exactly performed). It should be noted that some approaches used different number of training data (for example, only one training pattern per person is used in [39], [40], and eight patterns per person in [29]); some results were evaluated based on the best performance of one run, such as [41], [42]; some experiments were performed based on part of the database [40]. It is not clear how the experiments were carried out and how the performances were evaluated in [12], [43]–[45]. It is not fair to compare the performances under different experimental conditions.

VII. DISCUSSION

In this paper, a general and efficient approach for designing an RBF neural classifier to cope with high-dimensional problems in face recognition is presented. For the time being, many algorithms have been proposed to configure RBF neural networks for various applications including face recognition, as shown in [19]–[29]. Here, we would like to provide more insights into these algorithms and compare their performances with our proposed method.

A. Face Features, Classifiers, and Performances

Here, the face features are first extracted by the PCA paradigm so that the resulting data are compressed significantly. Then, the information is further condensed via the FLD approach. Corresponding to Tables II and III in which the patterns are obtained from the PCA + FLD, the data distribution resulting from the PCA and the clustering errors for training patterns based on our proposed approach are tabulated in Tables IX and X. Comparing Tables II and III with Tables IX and X, we have the following observations: 1) Class overlapping gradually reduces along with decrease in the number of feature vectors for the data resulting from both the PCA and the PCA + FLD methods; 2) For the data from the PCA, the clustering errors increase along with decrease in the feature dimension, but the clustering errors decrease for the data from the PCA + FLD; and 3) The data from the PCA + FLD are still overlapping without complete separation unless the feature dimension is less than 20. However, the FLD indeed alleviates the class overlapping as evidenced in comparing Tables IX and II.

Different face features are then used for testing by different classifiers. Figs. 8 and 9 illustrate the effect of data dimension resulted from the PCA and the PCA + FLD methods on performance classified by the nearest neighbor method. We see that more information (more dimensions) result in higher performance in the PCA. However, the performance resulting from the PCA + FLD is not monotonically improved along with increase in the feature dimension, and the best performance is a little decrease in PCA + FLD because of information loss.[3]. Table XI illustrates the performances by using different face features and classifiers.

As foreshadowed earlier, the FLD is a linear transformation and the data resulting from this criterion are still heavily overlapping. It is also indicated in [34] that this separability criterion

TABLE VIII
OTHER RESULTS RECENTLY PERFORMED ON THE ORL DATABASE

Approach	Error rate (%)	Year
PDBNN [12]	4	1997
Point-matching [39]	16	1998
Pseudo2-D HMM +DCT [43]	0	1999
LVQ+RBF+FEC [42]	0.5	1999
PCA+RBF [29]	4.9	2000
FND [44]	1.1	2000
UDT [41]	2.5	2001
Wavelet+RBF [45]	3.7	2001
PCA+moment invariant [40]	4	2001

TABLE IX
DATA DISTRIBUTION RESULTED FROM THE PCA BASED ON THE PROPOSED APPROACH (THE RESULTS ARE OBTAINED BASED ON SIX SIMULATIONS)

Feature	Data distribution		
dimension	No. of clusters	No. of well-separated classes	No. of overlapping classes
170	40	0	240
140	40	0	240
110	40	0	240
80	40	0	240
50	40	2	238
30	40	5	235

TABLE X
CLUSTERING ERRORS FOR TRAINING PATTERNS RESULTED FROM THE PCA (THE RESULT IS THE SUM OF SIX SIMULATIONS)

Feature dimension	No. of clusters	No. of misclustering	Maximum number of misclustering within a class
170	40	8	2
140	40	10	2
110	40	17	2
80	40	17	2
50	40	22	2
30	40	35	2

[3] This is also indicated in several papers, for example, [35] and [36]

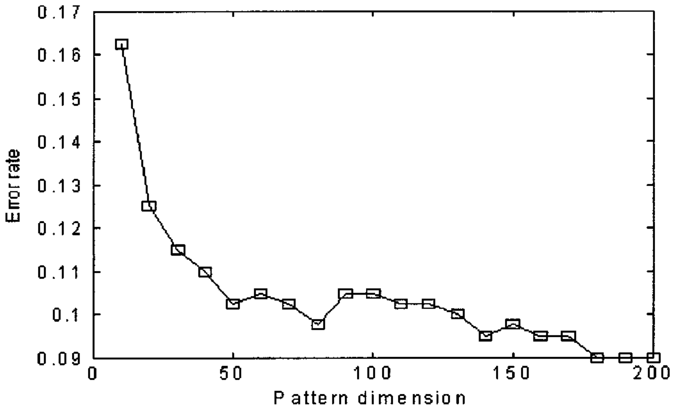


Fig. 8. Error rate as a function of pattern dimension in PCA (this is the average result of two runs).

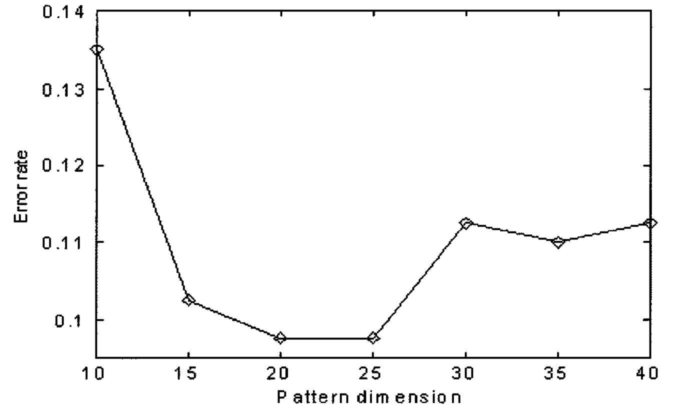


Fig. 9. Error rate as a function of pattern dimension in PCA + FLD (this is the average result of two runs).

is not directly related to the classification accuracy in the output space. Accordingly, nonlinear discriminant analysis is necessary for classification among which neural networks are one of the popular approaches [15].

The advantage of neural classifiers over linear classifiers is that they can reduce misclassifications among the neighborhood classes as shown in Fig. 10. However, this ability will gradually decrease along with increase in the feature dimension. We can see from Table XI that the performance gained by the PCA + FLD is better than that obtained from the PCA. This is because the FLD can alleviate data overlapping, and reduction in the number of feature dimension moderates the architecture complexity of the RBF neural classifier and reduces the computational burden significantly in order to avoid overtraining and overfitting. However, for those data falling into nonnearest classes as shown in Fig. 10, the neural classifier still cannot classify correctly.

It has been reported in [11] that error rates of the multilayer networks (MLNs) classifier are 41.2% and 39.6% respectively on the ORL database when the features are extracted by the PCA and the SOM respectively. Our proposed approach, which is different from the MLN, wherein a particular supervised learning paradigm is employed, is a tremendous improvement over the results of MLN, CNN [11] and the RBF method shown in [29].

TABLE XI
PERFORMANCE COMPARISONS OF VARIOUS FACE FEATURES AND CLASSIFIERS

Classifier	Face Features	No of clusters	Feature dimension	Error rate (%)
Nearest neighbor Classifier	PCA	—	170	9
	PCA+FLD	—	25	9.75
RBF neural classifier	PCA	40	170	4.75
	PCA+FLD	40	25	1.92

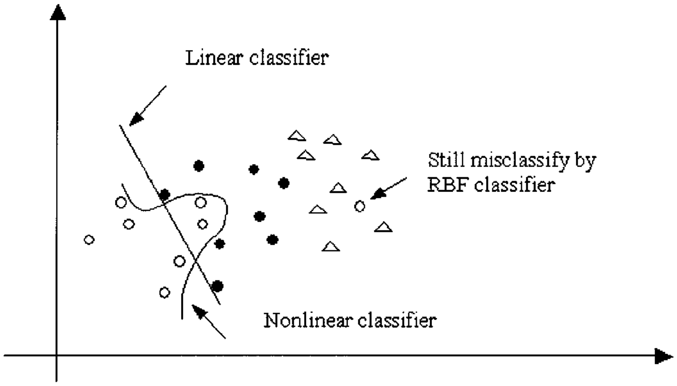
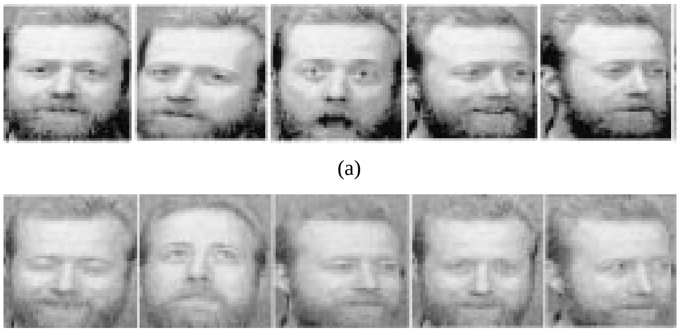


Fig. 10. An RBF neural classifier versus a linear classifier.

B. Training Samples versus Performances

Due to the fact that there are very small sample patterns for each face in the ORL database, and further, as mentioned in Section I that similarities between the different face images with the same pose are almost always larger than those between the same face image with different poses, the choice of training data is consequently very crucial for generalization of RBF neural networks. If the training data are representative of face images, the generalization of RBF neural classifier implies to interpolate the testing data. Otherwise, it means to predict the testing data.

From the viewpoint of images, it is shown that the proposed approach is not as sensitive to illumination (see Fig. 11), as other paradigms do [2], [6]. Usually, the proposed method also discounts the variations of facial pose, expression and scale when such variations are presented in the training patterns. If the training patterns are not representative of image variations which appear during the testing phase, say, upward, then the face turning up in the testing phase will not be recognized correctly, as shown in Fig. 11. According to this principle, another database consisting of 600 face images of 30 individuals, which comprise different poses (frontal shots, upward, downward, up-right, down-left and so on), and high degree of variability in facial expression, has been set up by us. All the images were taken under the same background with the resolution of 160 × 120 pixels. A total of 300 face images, in which each person has ten images, were selected to represent different poses and expressions as the training set. Another 300 images were used as the testing set. Our results demonstrated that the success rate of recognition is 100%.



(a)

(b)

Fig. 11. An example of incorrect classification: (a) training images and (b) test images.

C. Initialization versus Performances

1) Selection of Gaussian Centers : Several paradigms have been proposed for kernel location estimation. The simplest approach is to select from the training data as shown in [20], [23], [25], [28]. Other typical techniques can be found by using clustering algorithms [19], [21] or median operation [26]. If we also select the same six groups of face data resulting from the PCA + FLD with a feature dimension of 40, the initial clustering errors for training data by other clustering algorithms are tabulated in Table XII.

We can see from Table XII that many data are misclassified by the unsupervised k-means clustering method and the regression clustering method [23]. It also implies that these data are significantly overlapped. However, the clustering error will be remarkably reduced if the category information about patterns is used, for example, the MRBF paradigm [26] shown in Table XII, and our proposed method achieves the best clustering performance as shown in Table III.

2) Determination of Gaussian Widths: The appropriate estimation of widths of Gaussian units is very significant for generalization [20], [22], [24]–[29]. Frequently, the widths are selected by heuristics [19], [21], [23], [29]. Also many researchers choose the widths as the common variance (CV) (i.e., calculated over all sample patterns) [20], [33] or the class-related variance (CRV) (i.e., calculated over all the patterns belonging to the corresponding class) [24], [27]. Recently, some new methods have been proposed to estimate the widths, for example, the sample covariance (i.e., the class-related variance) plus common covariance (SCCC) [32], the minimum distance between cluster centers (i.e., using σ_β) [25], the median operation (MO) [26], or the evolutionary optimization [28]. If we use the same six groups of data resulting from the PCA + FLD, where the centers are determined by our proposed clustering method, and the cluster number is still 40, the initial clustering errors in different widths chosen by different methods are tabulated in Table XIII. Table XIV illustrates the generalization performance for testing patterns performed on the ORL database.

It is shown from Table XIII that the SCCC method is the best method to describe the training patterns. However, a method for good description of training patterns does not imply that it has good generalization, as we see from Table XIV. On the other hand, the testing errors before learning for the MO and CRV approaches are very high (the total NOM's are 95 and 117, respectively). But their final performances after learning are comparable to other paradigms (CV, SCCC and σ_W). Theoretically, the final results should be the same regardless of initial parameters if the learning algorithm is good enough for optimization. The discrepancies are mainly caused by overfitting and overtraining due to small sets of patterns with high dimension.

Two unsupervised algorithms with growing structure, i.e., ORBF [23] and D-FNN [25] have been employed to test the ORL database. Tables XV and XVI illustrate the generalization results for the first group data with different clusters. We see that the loss of category information will be at the cost of more clusters for the comparable performance. However, it should be noted that the increase of clusters is likely to result in overfitting, as shown in Table XVI.

TABLE XII
CLUSTERING ERRORS FOR TRAINING PATTERNS BY OTHER CLUSTERING ALGORITHMS (THE RESULT IS THE SUM OF SIX SIMULATIONS)

Feature dimension	No. of clusters	*K*-means clustering [19]	Median operation [26]	ORBF [23]
39	40	1050	22	464
35	40	968	16	446
30	40	756	3	388
25	40	508	3	378
20	40	378	2	324
10	40	300	0	18

TABLE XIII
CLUSTERING ERRORS FOR TRAINING PATTERNS CONSIDERING WIDTHS CHOSEN BY DIFFERENT METHODS (THE RESULT IS THE SUM OF SIX SIMULATIONS)

Feature dimension	CV	CRV	SCCC	MO	σ_B	σ_W
39	4	42	2	70	39	23
35	4	25	2	41	18	13
30	2	12	2	20	12	4
25	2	4	1	13	10	1
20	0	0	0	5	6	0
10	0	0	0	2	1	0

*The widths are best chosen for each case

* The widths are best chosen for each case

TABLE XIV
GENERALIZATION ERRORS FOR TESTING DATA BY DIFFERENT INITIAL WIDTHS (THE RESULT IS THE SUM OF SIX SIMULATIONS)

Method	NOM						Total NOM	E_{ave} (%)
	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6		
CV	5	9	11	6	4	7	42	3.5
CRV	5	10	11	13	4	8	51	4.25
SCCC	5	8	10	9	4	7	43	3.58
MO	3	10	9	10	4	7	43	3.58
σ_B	1	8	8	5	5	4	31	2.58
σ_W	5	6	9	9	2	7	38	3.17

* The results are obtained when the feature dimension is 30

* The results are obtained when the feature dimension is 30

TABLE XV
GENERALIZATION ERRORS FOR TESTING DATA BY THE ORBF METHOD

No. of clusters	40	52	80	123	150	172
NOM	15	11	10	7	7	7

TABLE XVI
GENERALIZATION ERRORS FOR TESTING DATA BY THE D-FNN METHOD

No. of clusters	40	45	52	62	89	106	112	125	158
NOM	18	16	11	8	8	8	9	9	10

D. The Problem of Small Sample Sets

One of the difficulties for neural networks to train in face recognition is the small sample data. This severe limitation always results in poor generalization. An alternative paradigm to improve generalization is to adopt the technique of regularization [15], [33], i.e., to encourage smoother network mappings by adding a penalty term Ω to the error function. But, the appropriate choice of regularization parameter is time consuming.

Another direct method to improve generalization is to use more data patterns, i.e., by adding some patterns with noise [46]. According to the procedure proposed in [46], another set of training patterns with noise randomly chosen from the uniform distribution is replenished. The learning algorithm is executed with and without adding noise to the inputs. Our experiments show that if the variance of noise is small, there is no effect on generalization, whereas large variance of noise will deteriorate the performance.

High dimension may be one of the reasons that lead to poor generalization. As the values in each dimension vary greatly and different features have different influences for face recognition, a uniformly distributed noise may affect some features substantially and has no influence on some other features. On the other hand, different features may have different weights for different face features. Therefore, normalization of the inputs should be taken when the noise is injected into the inputs.

Another reason may be due to the presupposition that the acquisition of generalization capability by noise injection into the inputs relies on the assumption that the mapping from the input space to the output space should be smooth [46]. For high-dimensional classifications, it is not easy to determine whether the assumption could be satisfied in advance.

VIII. CONCLUSION

It is well known that if the dimension of the network input is comparable to the size of the training set, which is the usual case in face recognition, the system will easily bring about overfitting and result in poor generalization. In this paper, a general design approach using an RBF neural classifier for face recognition to cope with small training sets of high-dimensional problem is presented. Firstly, face features are first extracted by the PCA. Then, the resulting features are further projected into the Fisher's optimal subspace in which the ratio of the between-class scatter and the within-class scatter is maximized. A novel paradigm, whereby training data information is used in the choice of structure and parameters of RBF neural networks before learning takes place, is presented. Finally, a hybrid learning algorithm is proposed to train the RBF neural networks. Simulation results show that the system achieves excellent performance both in terms of error rates of classification and learning efficiency.

In this paper, the feature vectors are only extracted from grayscale information. More features extracted from both grayscale and spatial texture information and a real-time face detection and recognition system are currently under development.

ACKNOWLEDGMENT

The authors wish to gratefully acknowledge Dr. S. Lawrence and Dr. C. L. Giles for providing details of their experiments on their CNN approach and Dr. A. Bors for his discussions and comments. Also, the authors would like to express sincere thanks to the anonymous reviewers for their valuable comments that greatly improved the quality of the paper.

REFERENCES

[1] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *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] 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.

[10] 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.

[11] 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.

[12] S.-H. Lin, S.-Y. Kung, and L.-J. Lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Trans. Neural Networks*, vol. 8, pp. 114–132, Jan. 1997.

[13] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. San Diego, CA: Academic Press, 1990.

[14] 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.

[15] C. M. Bishop, *Neural Networks for Pattern Recognition*, New York: Oxford Univ. Press.

[16] J. L. Yuan and T. L. Fine, "Neural-Network design for small training sets of high dimension," *IEEE Trans. Neural Networks*, vol. 9, pp. 266–280, Jan. 1998.

[17] J. Park and J. Wsandberg, "Universal approximation using radial basis functions network," *Neural Comput.*, vol. 3, pp. 246–257, 1991.

[18] F. Girosi and T. Poggio, "Networks and the best approximation property," *Biol. Cybern.*, vol. 63, pp. 169–176, 1990.

[19] J. Moody and C. J. Darken, “Fast learning in network of locally-tuned processing units,” *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[20] S. Lee and R. M. Kil, “A Gaussian potential function network with hierarchically self-organizing learning,” *Neural Networks*, vol. 4, pp. 207–224, 1991.

[21] W. Pedrycz, “Conditional fuzzy clustering in the design of radial basis function neural networks,” *IEEE Trans. Neural Networks*, vol. 9, pp. 601–612, July 1998.

[22] M. T. Musavi, W. Ahmed, K. H. Chan, K. B. Faris, and D. M. Hummels, “On the training of radial basis function classifiers,” *Neural Networks*, vol. 5, pp. 595–603, 1992.

[23] S. Chen, C. F. N. Cowan, and P. M. Grant, “Orthogonal least squares learning algorithm for radial basis function network,” *IEEE Trans Neural Networks*, vol. 2, pp. 302–309, 1991.

[24] G. Bors and M. Gabbouj, “Minimal topology for a radial basis functions neural networks for pattern classification,” *Digital Processing*, vol. 4, pp. 173–188, 1994.

[25] S. Wu and M. J. Er, “Dynamic fuzzy neural networks: A novel approach to function approximation,” *IEEE Trans. Syst, Man, Cybern*, pt. B: Cybern, vol. 30, pp. 358–364, 2000.

[26] A. G. Bors and I. Pitas, “Median radial basis function neural network,” *IEEE Trans. Neural Networks*, vol. 7, pp. 1351–1364, Sept. 1996.

[27] N. B. Karayiannis and G. W. Mi, “Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques,” *IEEE Trans. Neural Networks*, vol. 8, pp. 1492–1506, Nov. 1997.

[28] A. Esposito, M. Marinaro, D. Oricchio, and S. Scarpetta, “Approximation of continuous and discontinuous mappings by a growing neural RBF-based algorithm,” *Neural Networks*, vol. 12, pp. 651–665, 2000.

[29] E.-D. Virginia, “Biometric identification system using a radial basis network,” in *Proc 34th Annu. IEEE Int. Carnahan Conf. Security Technol.*, 2000, pp. 47–51.

[30] S. J. Raudys and A. K. Jain, “Small sample size effects in statistical pattern recognition: Recommendations for practitioners,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 252–264, 1991.

[31] L. O. Jimenez and D. A. Landgrebe, “Supervised classification in high-dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data,” *IEEE Trans. Syst, Man, Cybern. C*, vol. 28, pp. 39–54, 1998.

[32] J. P. Hoffbeck and D. A. Landgrebe, “Covariance matrix estimation and classification with limited training data,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 763–767, 1996.

[33] S. Haykin, *Neural Networks, A Comprehensive Foundation*, New York: Macmillan, 1994.

[34] R. Lotlikar and R. Kothari, “Fractional-step dimensionality reduction,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 623–627, 2000.

[35] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski, “Classifying facial actions,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 974–989, 1999.

[36] C. Liu and H. Wechsler, “Learning the face space—Representation and recognition,” in *Proc. 15th Int. Conf. Pattern Recognition*, Spanish, 2000, pp. 249–256.

[37] S. Z. Li and J. Lu, “Face recognition using the nearest feature line method,” *IEEE Trans. Neural Networks*, vol. 10, pp. 439–443, Mar. 1999.

[38] V. Brennan and J. Principe, “Face classification using a multiresolution principal component analysis,” in *Proc. IEEE Workshop Neural Networks Signal Processing*, 1998, pp. 506–515.

[39] K.-M. Lam and H. Yan, “An analytic-to-holistic approach for face recognition based on a single frontal view,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 673–686, 1998.

[40] T. Phiasai, S. Arunrungrusmi, and K. Chamnongthai, “Face recognition system with PCA and moment invariant method,” in *Proc. IEEE Int. Symp. Circuits Syst.*, 2001, pp. II165–II168.

[41] Z. Jin, J.-Y. Yang, Z.-S. Hu, and Z. Lou, “Face recognition based on the uncorrelated discriminant transformation,” *Pattern Recognition*, vol. 34, pp. 1405–1416, 2001.

[42] A. S. Tolba and A. N. Abu-Rezq, “Combined classifiers for invariant face recognition,” in *Proc. Int. Conf. Inform. Intell. Syst.*, 1999, pp. 350–359.

[43] S. Eickeler, S. Mueller, and G. Rigoll, “High quality face recognition in JPEG compressed images,” in *Proc. IEEE Int. Conf. Image Processing*, 1999, pp. 672–676.

[44] T. Tan and H. Yan, “Object recognition using fractal neighbor distance: Eventual convergence and recognition rates,” in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 781–784.

[45] B.-L. Zhang and Y. Guo, “Face recognition by wavelet domain associative memory,” in *Proc. Int. Symp. Intell. Multimedia, Video, Speech Processing*, 2001, pp. 481–485.

[46] K. Matsuoka, “Noise injection into inputs in back-propagation learning,” *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 436–440, 1992.

Meng Joo Er (S’82–M’85) received the B.Eng. and M.Eng. degrees in electrical engineering from the National University of Singapore, Singapore, and the Ph.D. degree in systems engineering from the Australian National University, Canberra, in 1985, 1988, and 1992, respectively.

From 1987 to 1989, he was as a Research and Development Engineer in Chartered Electronics Industries Pte Ltd, Singapore, and a Software Engineer in Telerate Research and Development Pte Ltd, Singapore, respectively. He is currently an Associate Professor in the School of Electrical and Electronic Engineering (EEE), Nanyang Technological University (NTU). From February to April 1998, he was invited as a Guest Researcher at the Precision Instrument Development Center of National Science Council, Taiwan. He served as a consultant to Growth Venture Pte Ltd., Singapore, from February 1999 to May, 2000. He was invited as a panelist to the IEEE-INNS-ENNS International Conference on Neural Networks held from 24 to 27 July, 2000, Como, Italy. He has been Editor of IES Journal on Electronics and Computer Engineering since 1995. He is currently a member of the editorial board of the *International Journal of Computer Research* and the guest editor of the special issue on intelligent control of mobile robots. He has authored numerous published works in his research areas of interest, which include control theory and applications, robotics and automation, fuzzy logic and neural networks, artificial intelligence, biomedical engineering, parallel computing, power electronics and drives and digital signal processors applications.

Dr. Er was the recipient of the Institution of Engineers, Singapore (IES) Prestigious Publication (Application) Award in 1996 and the IES Prestigious Publication (Theory) Award in 2001. He received the Teacher of the Year Award for the School of EEE for his excellent performance in teaching in 1999. He was also a recipient of a Commonwealth Fellowship tenable at the University of Strathclyde, U.K., from February to October 2000. He is a member of IES. He has served in these professional organizations in various capacities, including Chairman of Continuing Education Subcommittee of IEEE Control Chapter in 1993, Honorary Secretary of the IEEE Singapore Section from 1994 to 1996, Vice-Chairman of IEEE Singapore Section and Chairman of Continuing Education Subcommittee, IEEE Singapore Section from March, 1998 to December 1999. From February 1997 to February, 1999, he also served as a member of the Singapore Science Centre Board. Currently, he is serving as the First Vice-Chair of the Action Group on Educational Institutions, Uniform Groups and Youth of Publication Education Committee on Family. He has also been very active in organizing international conferences. He was a member of the main organizing committees of International Conference on Control, Automation, Robotics and Vision (ICARCV) for four consecutive years, 1994, 1996, 1998, and 2000, and the Asian Conference on Computer Vision (ACCV) in 1995. He was the Cochairman of Technical Programme Committee and person-in-charge of invited sessions for ICARCV’96 and ICARCV’98. He was in charged of International Liaison for ICARCV’2000. He is currently serving as the Technical Program Chair for ICARCV’2002. He has also served as a member of the International Scientific Committee of the International Conference on Circuits, Systems, Communications and Computers (CSC) since 1998. He was a member of the International Scientific Committee of a unique three conferences series on Soft Computing consisting of 2001 World Scientific and Engineering Society (WSES) International Conference on Neural Networks and Applications, 2001 WSES International Conference on Fuzzy Sets and Fuzzy Systems, and 2001 WSES International Conference on Evolutionary Computation. He has also been invited to serve as General Chair of the 2001 WSES Conference on Robotics, Distance Learning and Intelligent Communications Systems (RODLICS), International Conference on Speech, Signal and Image Processing 2001 (SSIP’2001), International Conference on Multimedia, Internet, Video Technologies (MIV’2001) and International Conference on Simulation (SIM’2001) organized by WSES. He has been listed in Who’s Who in the World since 1998 and was nominated Man of the Year 1999 by the American Biographical Institute Board of International Research.

Shiqian Wu (M'02) received the B.S. and M.Eng. degrees, both from Huazhong University of Science and Technology, Wuhan, China, in 1985 and 1988, respectively. He received the Ph.D. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, Singapore, in June 2001.

From 1988 to 1997, he was a Lecturer and then Associate Professor in Huazhong University of Science and Technology. Since August 2000, he has been with the Centre for Signal Processing, Singapore. Currently, his research interests include neural networks, fuzzy systems, computer vision, face detection and recognition, and infrared image analysis. He has published more than 20 papers.

Juwei Lu (M'99–S'01) received the Bachelor of Electronic Engineering degree from Nanjing University of Aeronautics and Astronautics, China, in 1994, and the Master of Engineering degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore, in 1999.Currently, he is pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada.

From July 1999 to January 2000, he was with the Centre for Signal Processing, Singapore, Singapore, as a Research Engineer.

Hock Lye Toh (S'85–M'88) received the B.Eng. degree from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore, in 1988.

He joined the Signal Processing Laboratory of Defence Science Organization in 1988. Prior to joining Centre for Signal Processing as a Senior Research Engineer in 1997, he was a Senior Engineer holding Group Head appointment. From 1988 to 1997, he has worked on algorithm development projects on image enhancement and restoration, design, and development of real-time computation engine using high-performance digital signal processors, and programmable logic devices. He was Project Leader for the development of a few real-time embedded systems for defense applications. He was Group Head, Imaging Electronics Group from 1993 and Group Head, Image Processing Group from 1995. He was appointed Program Manager, Centre for Signal Processing from 1998. He is spearheading a few research and development projects on embedded video processing and human thermogram analysis for multimedia, biometrics, and biomedical applications. He has led numerous in-house and industry projects, filed a patent and submitted a few invention disclosures. His research interests are embedded processing core development, thermogram video signal processing, and 3-D signal reconstruction.

Mr. Toh is a member of the Institution of Engineers, Singapore.