Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition 41 (2008) 2916–2925
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

Building compact MQDF classifier for large character set recognition by subspace distribution sharing

Teng Long, Lianwen Jin*

School of Electronics and Information, South China University of Technology, Guangzhou 510641, PR China

Received 4 September 2007; received in revised form 23 February 2008; accepted 27 February 2008

Abstract

Quadratic classifier with modified quadratic discriminant function (MQDF) has been successfully applied to recognition of handwritten characters to achieve very good performance. However, for large category classification problem such as Chinese character recognition, the storage of the parameters for the MQDF classifier is usually too large to make it practical to be embedded in the memory limited hand-held devices. In this paper, we aim at building a compact and high accuracy MQDF classifier for these embedded systems. A method by combining linear discriminant analysis and subspace distribution sharing is proposed to greatly compress the storage of the MQDF classifier from 76.4 to 2.06 MB, while the recognition accuracy still remains above 97%, with only 0.88% accuracy loss. Furthermore, a two-level minimum distance classifier is employed to accelerate the recognition process. Fast recognition speed and compact dictionary size make the high accuracy quadratic classifier become practical for hand-held devices.

Keywords: Compact classifier; Modified quadratic discriminant function; Handwritten character recognition; Large character set; Subspace distribution sharing

1. Introduction

The emergence of personal hand-held devices, such as PDAs (personal digital assistants), Pocket PCs and mobile phones, with considerable computing power and touch panels, made the handwriting recognition technology widely used for their limited size of keyboards. These applications attract many researchers to embed the existing popular recognition approaches for handwritten characters into these hand-held devices. During the last decade, the statistical approaches have been widely developed and applied to handwritten character recognition. They became the most popular methods in the literature for their simplicity and robustness [1]. Among them, the modified quadratic discriminant function (MQDF) [2] based on Bayesian decision rule has been extremely successful and makes the very important part of the reported high accuracy classifiers [3–6]. With the power of such kind of quadratic classifier, the recognition accuracy reaches above 99% for handwritten digits and Kanji characters and above 98% for handwritten Chinese characters [4,5].

However, in real applications for hand-held devices, like other high accuracy classifiers such as neural classifier and support vector machine (SVM), the MQDF classifier also has a parameter complexity problem for large character set recognition. Take the work reported in Ref. [4] as an example, the parameters of the MQDF classifier need more than 140 M bytes of storage assuming 4-byte floating point number is used for each parameter. As the memory size of the hand-held devices is usually limited nowadays, it is not practical to embed the MQDF classifier directly into these devices. Thus at present, most applications for embedded systems usually employ simpler classifiers such as minimum distance classifier for limited storage consideration [7].

In order to embed the high accuracy MQDF classifier into the memory limited devices, compressing methods for reducing the parameter size are necessary. Although classifier error is usually the main concern in publications, in this paper, we suggest that the tradeoff between classifier's parameter size and error rate also be an important characteristic. Larger memory

* Corresponding author. Tel.: +86 20 87113540.
E-mail addresses: linky2003@gmail.com (T. Long), eelwjin@scut.edu.cn (L. Jin).

0031-3203/$30.00 © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.02.009

size directly causes the increase of products' cost. So in real world applications, different products with different memory size are designed for different markets. By seeking for an optimal tradeoff, we can customize the classifier for different situations to reach the best performance.

In previous works, de Ridder et al. [8] noticed the tradeoff between classifier error and evaluation complexity and proposed a simple economic model. The tradeoff can be then used to judge the necessity of increasing evaluation complexity for decreased classification error. The error–complexity curves were discussed for any classification problem. They mainly concerned about the computational complexity and the proposed method was based on prototype selection and feature extraction or dimension reduction by certain approaches such as principal component analysis (PCA). Nevertheless, there exist other methods such as cascaded classifiers to reduce the computational complexity. For example, the MQDF classifier was used to classify only the first 20 candidates selected by a simple minimum distance pre-classifier [4]. This kind of methods can heavily reduce the computational complexity for large character set recognition and is usually used in computational expensive classifiers such as SVM [9]. However, the storage of parameters cannot be reduced by such kind of techniques. Therefore, to our concern, the tradeoff between parameter size and classifier error becomes more important in practice.

Liu et al. [7] proposed a method of building compact classifier for large character set printed character recognition. They used the linear discriminant analysis (LDA) which is also known as Fisher discriminant analysis (FDA) to reduce the feature dimensionality and converted the 4-byte floating point parameters into 1 byte to form the final compact dictionary. But the converting technique is not discussed. Moreover, by using such kind of method for compressing MQDF classifier, the storage size of the parameters is not small enough for most memory limited embedded systems.

In this paper, we aim at building a compact and high accuracy MQDF classifier for memory limited hand-held devices. By using LDA for dimension reduction, the tradeoff between parameter size and classifier error is found to be improved. In order to further compress the parameters to reach a better tradeoff, a kind of vector quantization (VQ) technique is employed. We first split the original parameter's dimension to map the parameter space into multiple subspaces. Then we try to find a uniform statistical model to fit the distributions of the subspaces. By sharing the distribution of multiple subspaces of the parameters, the dictionary of the MQDF classifier can be greatly compressed with a slight recognition accuracy loss. Some preliminary results of this work were reported in Ref. [10]. Similar technique called split VQ is originally developed in the field of automatic speech recognition (ASR) [11,12] and also has been successfully used in compressing model parameters in the field of optical character recognition (OCR) [13,14] and handwriting recognition [15]. Ge and Huo described their split VQ method in detail for compressing model parameters of continuous-density hidden Markov model (CDHMM) in handwritten Chinese character recognition [16]. From the experimental results in Section 5, it is shown that even by using a shared distribution model for all subspaces of the parameters, the recognition accuracy only decreased by less than 0.2% for handwritten Chinese characters. This is why we choose the shared distribution model other than the model in original split VQ. A two-level minimum distance classifier is employed for coarse recognition to accelerate the recognition process. The fast recognition speed (1.8 ms/char for PC and 64 ms/char for Pocket PC) and compact dictionary size (2.06 MB) make the high accuracy MQDF classifier become practical for memory limited hand-held devices such as PDAs, mobile phones and Pocket PCs.

The rest of this paper is organized as follows. Section 2 describes the experiment database and the underlying feature extraction method. Section 3 reviews the MQDF classifier. In Section 4, we present our modeling techniques for building a compact MQDF classifier for large character set recognition. The experimental results on recognition of handwritten Chinese characters are shown in Section 5. Finally, we draw our conclusions in Section 6.

2. Database and feature extraction

In order to evaluate the performance of our classifier, the HCL2000 database [17], which is also used in Ref. [4], is used in this paper for training and testing. It is collected by Beijing University of Posts and Telecommunications for China 863 project and includes 3755 frequently used Chinese characters in GB2312-80 level 1 character set. For each Chinese character, 1000 samples written by 1000 different writers were collected in the database. In total, 3,755,000 handwritten Chinese character samples are included. A part of samples of the first Chinese character in the database is demonstrated in Fig. 1.

Each character sample in the HCL2000 database is a binary image of size 64×64, which was obtained by linear normalization on the original segmented handwritten character sample. We first apply an 8×8 global elastic meshing method [18] to partition the binary image into 8×8 meshes according to the image pixel projection onto the horizontal and vertical directions. Then the 8-directional gradient features [4,19] are extracted at each image pixel. For each sample point at the center of each mesh, an 8-dimensional vector is obtained from the 8-directional gradient features by passing a 20×20 Gaussian filter on the sample point. Thus, in total, 512 dimensional feature vectors are extracted for each character image. As the distribution of the features is like that of the directional elemental feature (DEF), we also performed the square root for each feature element to make the distribution of features Gaussian-like [20].

3. MQDF classifier

Based on Bayesian decision rule, which classifies the input pattern to the class of maximum a posteriori (MAP) probability out of classes, the quadratic discriminant function (QDF) is obtained under the assumption of multivariate Gaussian density

Fig. 1. Some samples of the first Chinese character in the HCL2000 database.

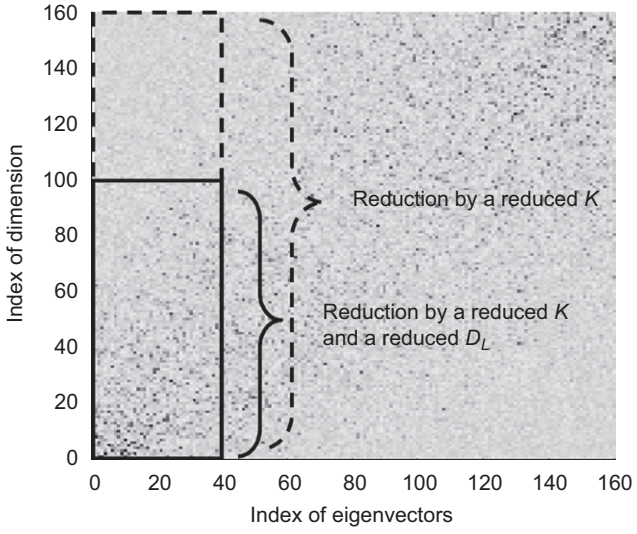

Fig. 2. The top-view of a typical eigenvector matrix's surface, which is obtained from a class's covariance matrix after LDA.

for each class. The MQDF proposed by Kimura et al. [2] makes a modification to the QDF by K–L transform and smoothing the minor eigenvalues to improve the computation efficiency and classification performance. We review it here.

According to the Bayes rule, the a posteriori probability is computed by

$$P(\omega_i|x) = \frac{P(\omega_i)p(x|\omega_i)}{p(x)}, \quad i = 1, \ldots, M \tag{1}$$

where M is the number of classes, $P(\omega_i)$ is the a priori probability of class ω_i, $p(x|\omega_i)$ is the class probability density function and $p(x)$ is the mixture density function. As $p(x)$ is independent of class, the nominator of (1) can be used as the discriminant function for classification:

$$g(x, \omega_i) = P(\omega_i)p(x|\omega_i) \tag{2}$$

Assume the probability density function of each class is multivariate Gaussian:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{\frac{D}{2}}|\Sigma_i|^{\frac{1}{2}}} \exp\left[-\frac{(x-\mu_i)^{\mathrm{T}}\sum_i^{-1}(x-\mu_i)}{2}\right] \tag{3}$$

where μ_i and Σ_i denote the mean vector and the covariance matrix of class ω_i, respectively, D is the dimension of μ_i. Inserting Eq. (3) into Eq. (2), taking the negative logarithm and omitting the common terms under equal a priori probabilities, the QDF is obtained:

$$g_0(x, \omega_i) = (x-\mu_i)^{\mathrm{T}} \sum_i^{-1}(x-\mu_i) + \log|\Sigma_i| \tag{4}$$

The QDF can be used as a distance metric in the sense that the class of minimum distance is assigned to the input pattern.

By K–L transform, the covariance matrix can be diagonalized as

$$\Sigma_i = \Phi_i \Lambda_i \Phi_i^{\mathrm{T}} \tag{5}$$

where $\Lambda = diag[\lambda_{i1}, \ldots, \lambda_{iD}]$ with λ_{ij}, $j = 1, \ldots, D$, being the eigenvalues (ordered in decreasing order) of Σ_i and $\Phi_i = [\phi_{i1}, \ldots, \phi_{iD}]$ with ϕ_{ij}, $j = 1, \ldots, D$, being the ordered eigenvectors. Φ_i is orthonormal (unitary) such that $\Phi_i^{\mathrm{T}}\Phi_i = I$.

According to Eq. (5), the QDF can be rewritten in the form of eigenvectors and eigenvalues:

$$\begin{aligned} g_0(x, \omega_i) &= [\Phi_i^{\mathrm{T}}(x-\mu_i)]^{\mathrm{T}}\Lambda_i^{-1}\Phi_i^{\mathrm{T}}(x-\mu_i) + \log|\Lambda_i| \\ &= \sum_{j=1}^{D} \frac{1}{\lambda_{ij}}[\phi_{ij}^{\mathrm{T}}(x-\mu_i)]^2 + \sum_{j=1}^{D} \log\ \lambda_{ij} \end{aligned} \tag{6}$$

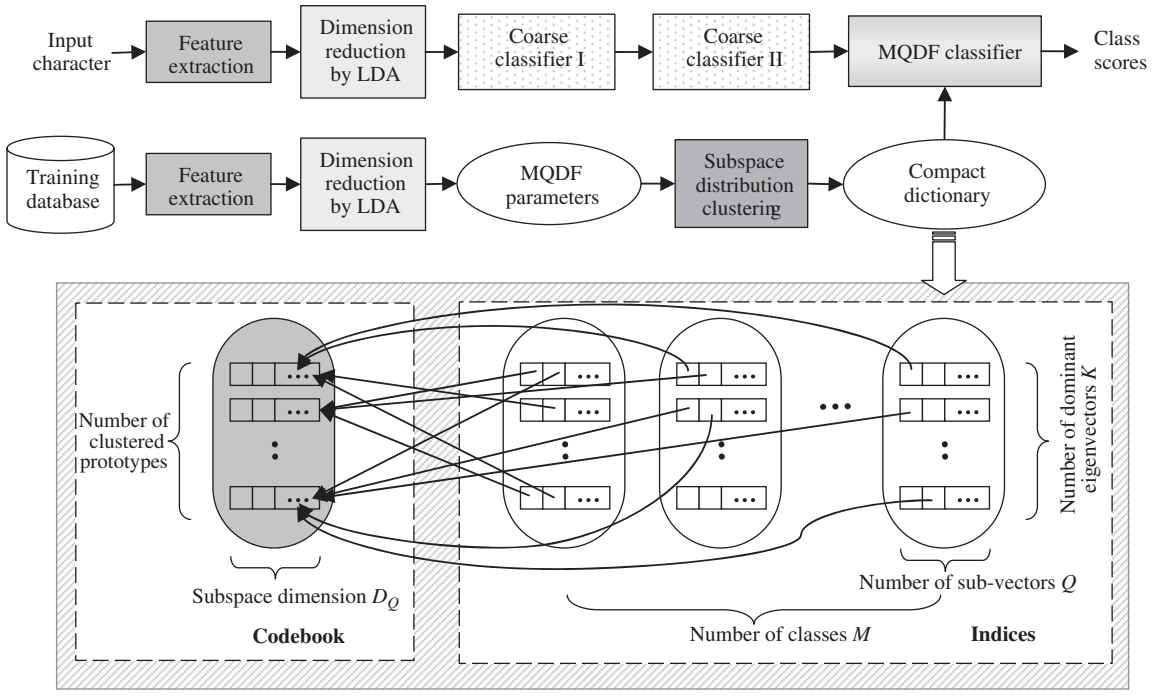

Fig. 3. Block diagram of the recognition system using the compact MQDF classifier.

By replacing the minor eigenvalues with a constant δ_i, the MQDF is obtained as

$$\begin{aligned} g_1(x,\omega_i) &= \sum_{j=1}^{K}\frac{1}{\lambda_{ij}}[\phi_{ij}^{\mathrm{T}}(x-\mu_i)]^2 \\ &\quad + \sum_{j=K+1}^{D}\frac{1}{\delta_i}[\phi_{ij}^{\mathrm{T}}(x-\mu_i)]^2 + \sum_{j=1}^{K}\log\lambda_{ij} \\ &\quad + (D-K)\log\delta_i \\ &= \frac{1}{\delta_i}\left\{\|x-\mu_i\|^2 - \sum_{j=1}^{K}\left(1-\frac{\delta_i}{\lambda_{ij}}\right)[\phi_{ij}^{\mathrm{T}}(x-\mu_i)]^2\right\} \\ &\quad + \sum_{j=1}^{K}\log\lambda_{ij} + (D-K)\log\delta_i \end{aligned} \tag{7}$$

where K denotes the number of dominant eigenvectors. The above utilizes the invariance of Euclidean distance:

$$d_E(x,\omega_i) = \|x-\mu_i\|^2 = \sum_{j=1}^{D}[\phi_{ij}^{\mathrm{T}}(x-\mu_i)]^2 \tag{8}$$

Since the training of the QDF classifier always underestimate the patterns' eigenvalues by limited sample set, the minor eigenvalues become some kind of unstable noises and affect the classifier's robustness. By smoothing them in the MQDF classifier, not only the classification performance is improved, but also the computation time and storage for the parameters are saved.

The parameter δ_i can be set to a class-independent constant as proposed by Kimura et al. [2] or a class-dependent constant calculated by the average of minor eigenvalues [21]. We set it to a class-independent constant and optimize its value by holdout cross validation on the training data set. In practice, we found the performance is superior when setting the constant class-independent rather than class-dependent. The same result was also found in Ref. [22].

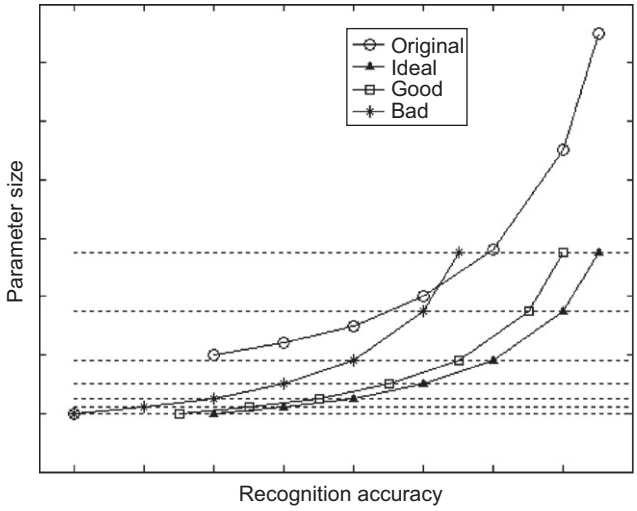

Fig. 4. A hypothesis of tradeoff curves between parameter size and recognition accuracy.

4. Building compact MQDF classifier

From the introduction of MQDF classifier in Section 3, we can see that, for each class, the classifier needs to train and store the parameters such as the mean vectors μ_i, the dominant eigenvalues λ_i and eigenvectors Φ_i of the covariance matrix Σ_i. The parameter size of μ_i, λ_i and Φ_i is D, K and $K \times D$ respectively. The storage problem mainly comes from

Table 1
Recognition accuracy (%) on different K and D_L

D_L	Dominant eigenvector number K										
	2	4	6	8	10	12	16	25	32	48	64
80	95.96	96.57	96.85	96.99	97.06	97.12	97.15	97.17	97.19	97.19	97.16
96	96.02	96.67	96.99	97.15	97.25	97.32	97.39	97.47	97.50	97.49	97.44
128	96.06	96.75	97.10	97.31	97.44	97.53	97.63	97.79	97.83	97.84	97.75
160	96.05	96.76	97.13	97.36	97.49	97.60	97.73	97.91	97.97	97.97	97.85

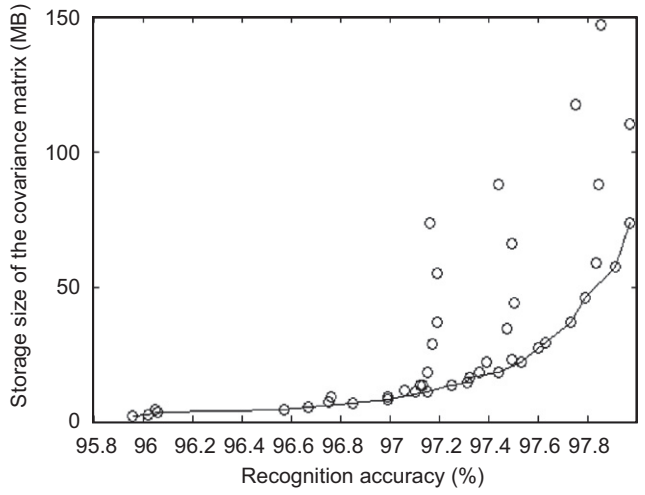

Fig. 5. The optimal tradeoff curve between parameter size and recognition accuracy by choosing different K and D_L.

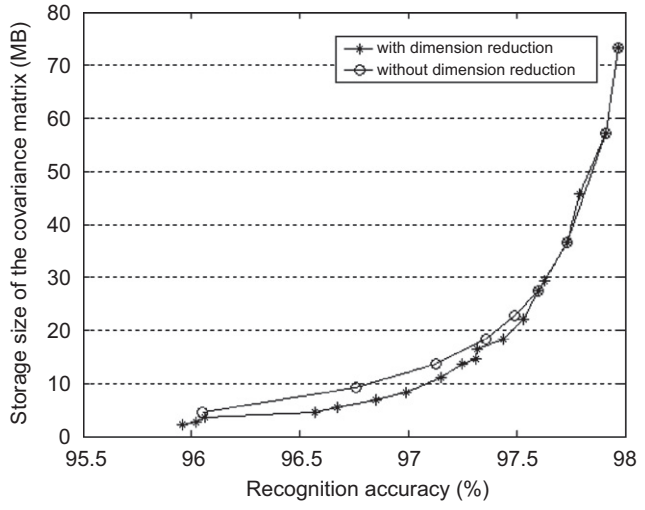

Fig. 6. The tradeoff curves with dimension reduction by choosing different D_L and without dimension reduction for eigenvectors.

the eigenvectors Φ_i of each class because K is less than D and usually larger than 10. Typically, the storage of the eigenvectors Φ_i $(i = 1, \ldots, M)$ is $4 \times D \times K \times M$ bytes when 4-byte floating point number is used. This requires about 293 MB storage space under a system setup of $D = 512$, $K = 40$, $M = 3755$.

In practice, before using MQDF, the feature vector's dimension D is usually compressed by LDA [4]. With the help of LDA, not only the parameter size and computation time are saved, but also the classifier's accuracy is improved. After dimension reduction from 512 to 160 by LDA, a typical eigenvector matrix of a class's covariance matrix is shown in Fig. 2. By choosing $K = 40$, the matrix can be reduced to the parameters in the dash line rectangle. When changing the parameter K, a tradeoff curve between classifier error and parameter size can be drawn by experiments. Meanwhile, by observation, the latter elements in the dominant eigenvectors are much smoother than the former elements. This provides us another way to further reduce the dominant eigenvectors by using only the former D_L $(D_L < D)$ elements in each eigenvector, which is shown in the bold line rectangle in Fig. 2. The latter elements in each eigenvector can be then substituted by their mean value.

In order to further compress the dominant eigenvectors of each class, a kind of VQ technique is used in our classifier's design. We first split the original parameter's dimension to map the parameter space into multiple subspaces. For each class ω_i, the dominant eigenvector matrix $\Phi_i^K = [\phi_{i1}, \ldots, \phi_{iK}]$ with ϕ_{ij} $(j = 1, \ldots, K)$ is partitioned into subspace eigenvector matrix, i.e. each D-dimensional eigenvector ϕ_{ij} is equally partitioned into Q D_Q-dimensional sub-vectors $\phi_{ij}^1, \phi_{ij}^2, \ldots, \phi_{ij}^Q$, where $D = D_Q \times Q$.

Then we try to find a general statistical model to fit the distributions of all the sub-vectors. By using LBG clustering algorithm [23] in the subspaces of the parameters, the sub-vectors ϕ_{ij}^q $(i = 1, \ldots, M, j = 1, \ldots, K, q = 1, \ldots, Q)$ are clustered into a small set of L prototypes. Each original subspace eigenvector is then presented by its nearest prototype. When the L is smaller than 2^r, where r is the number of bits for storing one D_Q-dimensional sub-vector, the storage of the eigenvector matrix of each class can be compressed.

The building process of the prototype set is similar to the split-dimension VQ for speech signals [12]. Typically, the smaller quantization error reached by the clustering process, the closer approximation obtained for the original sub-vectors. This can be ensured by using more prototypes. However, there is a tradeoff between the prototype size and the classifier error. By changing different parameter Q and L, the tradeoff curve can be drawn from experimental results.

After the clustering process is completed, the prototypes are clustered to approximate the distribution of the subspace eigenvectors. The dictionary for the eigenvector matrices of all

Table 2
Recognition accuracy (%) on different K and D_L when $D_Q = 1$, $L = 256$

D_L	Dominant eigenvector number K										
	2	4	6	8	10	12	16	25	32	48	64
96	96.04	96.68	97.00	97.17	97.24	97.32	97.38	97.45	97.48	97.49	97.43
128	96.09	96.78	97.15	97.35	97.43	97.53	97.62	97.78	97.82	97.83	97.74
160	96.09	96.82	97.18	97.35	97.49	97.60	97.73	97.90	97.96	97.96	97.85

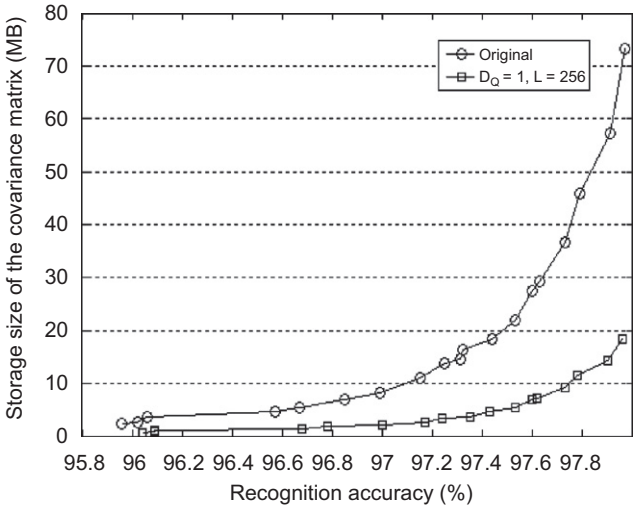

Fig. 7. The tradeoff curves of the original parameters and the compressed parameters by $D_Q = 1, L = 256$.

classes consists of two parts, the indices of the prototypes and the codebook. The storage size is computed as follows:

$$size(\Phi) = \frac{\log L}{8 \times D_Q} \times D_L \times K \times M + D_Q \times L \times 4 \qquad (9)$$

Fig. 3 demonstrates both the block diagram of our recognition system and the mapping relationship in the compact dictionary. The input character is first transformed to a binary image and normalized to size 64×64 by linear normalization. After the feature extraction which was described in Section 2, the dimension of the feature vector is reduced by LDA. Then a two-level minimum distance coarse classifier is designed for the purpose of speeding up the recognition process. The first-level coarse classifier uses first 16 dominant LDA features for classification and generates about 300 candidates for the second level coarse classifier. The second level coarse classifier uses 160 dominant LDA features and generates 20 candidates for MQDF classifier, which outputs the final recognition result. In the training stage, the training handwritten samples also performed the same feature extraction and dimension reduction by LDA. The MQDF parameters are then computed and the proposed VQ technique is performed to form the final compact dictionary. Fig. 3 only shows the mapping relationship of eigenvectors' indices and codebook in the compact dictionary. Other MQDF parameters, which have similar mapping relationship in the compact dictionary, are not shown here.

The same VQ technique can also be applied to other parameters such as the mean vectors μ_i, the dominant eigenvalues λ_i, and even the LDA transformation matrix to compress the final MQDF classifier's dictionary. Thus, the proposed method can be a general compressing method for any classifier and the tradeoff curve can be found to adapt the classifier's size to more situations.

In Fig. 4, we give a hypothesis to show how the tradeoff curve could be. If the VQ can reduce the classifier's parameter size to one half, the original tradeoff curve can be lowered down to the ideal one shown in Fig. 4 without any accuracy loss. However, VQs with different quantization error lead to different loss in the classifier's accuracy. We assume that if at each point on the curve, the recognition loss is approximately the same for one kind of quantization, the two tradeoff curves by good and bad quantizations under this assumption could be drawn in the figure. From Fig. 4, we can find out that even the bad quantization may fail to compress the parameters at high recognition accuracy level, it still may have chance to succeed at a lower recognition accuracy level. This could be proved by the experimental results in the next section.

5. Experiments

Experiments were carried out to test the performance of our compact MQDF classifier on recognition of handwritten Chinese characters. The recognition accuracies, parameter–size–accuracy tradeoff curves, error–reject tradeoff curves and timing analysis of the compact classifier are given by the experimental results.

5.1. Experimental setup

Both training and testing samples are from the HCL2000 database which has been introduced in Section 2. We use 700 sets (labeled as xx001–xx700) for training and the rest 300 sets (labeled as hh001–hh300) for testing. The testing platform is on a PC with Pentium4 2.8 G CPU and 1 G memory.

First, the dimension of each feature vector in mean vector template is reduced from 512 to 160 by LDA, which not only improves the classification performance but also arranges the dominant features for coarse classifier. After LDA, the baseline minimum distance classifier obtained a recognition rate of 94.23%. Then we compress all the parameters except the dominant eigenvectors by subspace distribution clustering and sharing with $D_Q = 1$, $L = 256$. All these parameters are then stored by only 1-byte index, which leads to a basic compact

Table 3
Recognition accuracy (%) on different K and D_L when $D_Q = 1$, $L = 16$

D_L	Dominant eigenvector number K										
	2	4	6	8	10	12	16	25	32	48	64
96	96.01	96.61	96.89	97.03	97.11	97.15	97.20	97.22	97.23	97.23	97.22
128	96.08	96.72	97.04	97.19	97.29	97.36	97.43	97.53	97.56	97.57	97.51
160	96.08	96.76	97.10	97.25	97.37	97.45	97.53	97.67	97.72	97.72	97.62

Table 4
Recognition accuracy (%) on different K and D_L when $D_Q = 2$, $L = 256$

D_L	Dominant eigenvector number K										
	2	4	6	8	10	12	16	25	32	48	64
96	96.02	96.64	96.94	97.09	97.17	97.22	97.26	97.31	97.32	97.32	97.29
128	96.09	96.74	97.09	97.27	97.35	97.43	97.50	97.61	97.65	97.65	97.59
160	96.09	96.77	97.15	97.33	97.41	97.50	97.60	97.74	97.80	97.79	97.69

Table 5
Recognition accuracy (%) on different K and D_L when $D_Q = 3$, $L = 256$

D_L	Dominant eigenvector number K										
	2	4	6	8	10	12	16	25	32	48	64
96	95.92	96.47	96.71	96.81	96.87	96.90	96.88	96.84	96.84	96.84	96.86
128	96.01	96.61	96.87	97.00	97.06	97.11	97.13	97.14	97.16	97.16	97.16
160	96.04	96.65	96.94	97.09	97.15	97.20	97.24	97.30	97.31	97.31	97.27

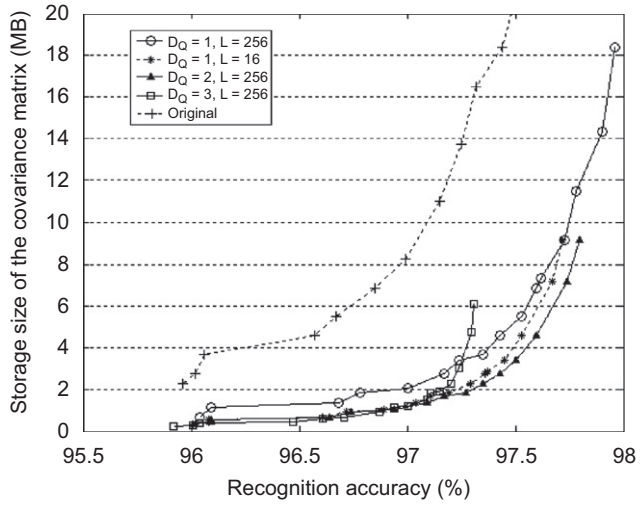

Fig. 8. The tradeoff curves of by choosing different D_Q and L.

dictionary whose size is only one fourth compared to the original dictionary. The following experiments were all based on these compressed parameters.

5.2. Speeding up the recognition process

Considering the MQDF is time consuming, a two-level minimum distance coarse classifier is employed in the recognition system. In order to avoid the sorting process for the candidates in the first-level coarse classifier, the distance between the input character and each prototype is recorded and a histogram of the distance distribution is calculated. By accumulating the number of candidates from the lowest distance in the histogram, a distance threshold is determined to give about 300 candidates by the recorded distances. The second level coarse classifier uses 160 dominant LDA features and generates 20 candidates for the MQDF classifier. The hit rate of the 20 candidates given by the two-level coarse classifier is 99.5%. The MQDF classifier calculates the MQDF score of the 20 candidates given by the coarse classifiers and gives the recognition result by choosing the one which has the minimum MQDF score.

5.3. Original tradeoff curve found by experiment

By choosing different parameter K and D_L, the open test recognition rates of MQDF are listed in Table 1. The recognition rate reaches 97.97% when $K = 32$ and $D_L = 160$. To our surprise, the MQDF classifier's recognition rate remains above 96% even when $K = 2$, which is about 1.8% higher than the conventional minimum distance classifier. In order to find the optimal tradeoff curve from the results, each point of different combination of K and D_L is plotted in Fig. 5. From the highest accuracy in Table 1, step by step, the optimal tradeoff curve can be found by the rule: lower recognition rate

and smaller parameter size. The tradeoff curve is demonstrated in Fig. 5.

5.4. Dimension reduction for eigenvectors

To find out whether the dimension reduction for eigenvectors is effective, the both curves without dimension reduction for eigenvectors and with dimension reduction by choosing different D_L are drawn in Fig. 6. It is shown that only after the recognition accuracy decreases to 97.5%, the dimension reduction is effective to further compress the parameters. For example, by choosing $K=4$, $D_L=80$, the recognition rate is 0.52% higher than that of the same size parameters without dimension reduction ($K=2$, $D_L=160$). Meanwhile, by choosing different D_L, there are more points on the curve which means the classifier's size is easier to be customized. Therefore, the proposed dimension reduction for eigenvectors of each class's covariance matrix is effective to optimize the tradeoff between parameter size and classifier accuracy.

5.5. Comparison between different VQ

To test the performance of the split VQ technique, we first choose $D_Q=1$, $L=256$. The recognition results are listed in Table 2. The tradeoff curve is drawn and compared with the original one in Fig. 7. From the results, it is seen the quantization is nearly perfect. The parameter size is reduced to one fourth with almost no accuracy loss. When choosing a small K less than 8, the accuracy is even improved. The reason may come from the noise reduction by quantization. By choosing some other different parameter D_Q and L, more experiments were performed to test the compact MQDF classifier's performance. The experimental results are listed in Tables 3–5. The tradeoff curves of different split VQ are also compared with each other in Fig. 8.

According to our hypothesis in Fig. 4, the split VQ with $D_Q=3$, $L=256$ is a bad quantization. However, its efficiency is promised at a lower recognition accuracy level such as less than 97%.

The comparison between the tradeoff curves shown in Fig. 8 indicates that slight compression by split VQ wins at high recognition accuracy level, while the medium one and the heavy one wins at medium and low accuracy level, respectively. From this observation, we can determine which kind of quantization is suitable for the specific requirement.

To our requirement, the best curve among them is found by setting $D_Q=2$, $L=256$. Five different parameter settings are selected to form five different scale compact dictionaries, which are shown in Table 6. The total dictionary size includes the size of parameters such as mean vectors, dominant eigenvalues and eigenvectors of covariance matrix, and the LDA transformation matrix. It is shown that the original dictionary size can be compressed from 76.4 to 7.9 MB with only 0.23% accuracy loss. The third compact MQDF classifier listed in the table is of less size compared with conventional LDA classifier but the recognition accuracy is 2.86% higher, which clearly

Table 6
Comparison between original classifier and compact MQDF classifiers

	Correct rate (%)	Dictionary size	
		Eigenvectors	Total
Original MQDF classifier ($K=32$, $D_L=160$)	97.97	$32\times160\times4\times3755=73.34$ MB	$[(160+32)\times3755+512\times160]\times4+73.34\,\text{MB}=76.4\,\text{MB}$
Compact MQDF classifier ($D_Q=2$, $L=256$)			
$K=25$, $D_L=160$	97.74	$160/2\times25\times3755+8\times256=7.16$ MB	$[(160+25)\times3755+512\times160]+8\times256+7.16\,\text{MB}=7.9\,\text{MB}$
$K=12$, $D_L=160$	97.50	$160/2\times12\times3755+8\times256=3.44$ MB	$[(160+12)\times3755+512\times160]+8\times256+3.44\,\text{MB}=4.13\,\text{MB}$
$K=8$, $D_L=96$	97.09	$96/2\times8\times3755+8\times256=1.38$ MB	$[(160+8)\times3755+512\times160]+8\times256+1.38\,\text{MB}=2.06\,\text{MB}$
$K=4$, $D_L=96$	96.64	$96/2\times4\times3755+8\times256=0.69$ MB	$[(160+4)\times3755+512\times160]+8\times256+0.69\,\text{MB}=1.35\,\text{MB}$
$K=2$, $D_L=96$	96.02	$96/2\times2\times3755+8\times256=0.34$ MB	$[(160+2)\times3755+512\times160]+8\times256+0.34\,\text{MB}=1.00\,\text{MB}$
Baseline LDA classifier	94.23	–	$(160\times3755+512\times160)\times4=2.60$ MB

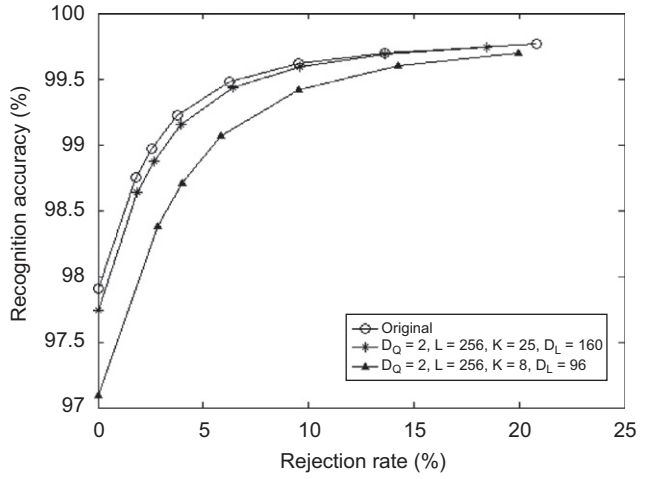

Fig. 9. The error–reject tradeoff curves of the original and compact MQDF classifier.

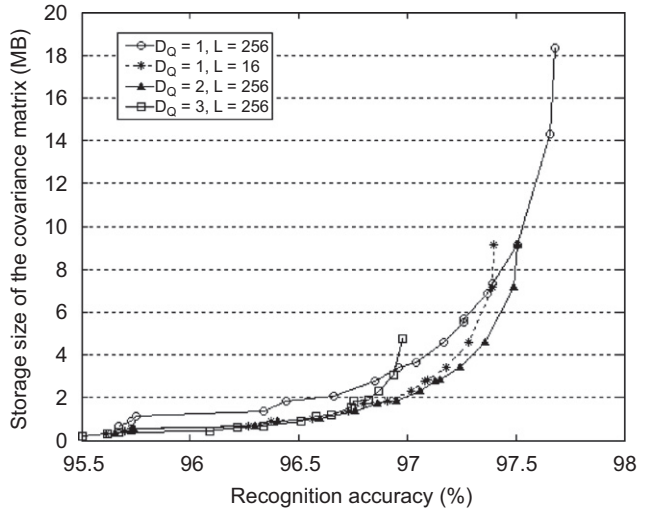

Fig. 10. The tradeoff curves of by choosing different D_Q and L based on the cleaned-up database.

demonstrates the efficiency of our proposed compact MQDF classifier.

5.6. Performance of the error–reject tradeoff curve

As a density model rather than a discriminant model, the outlier reject efficiency of MQDF is usually superior to other kinds of classifiers [24]. We also performed some experiments to test how the reject–error tradeoff curve will be affected by the compression of MQDF classifier. The score distance between the first two candidates is used and the classification result is rejected if the distance is smaller than a pre-defined threshold. By choosing different thresholds, the tradeoff curves are drawn in Fig. 9. It is found that higher reject rate makes the recognition performance of compact MQDF classifiers closer to the original classifier. For example, when the reject rate is about 10%, the recognition accuracy of the original classifier is 99.62%, and the ones of our compact classifiers listed in the figure reach 99.59% and 99.42%, with only 0.03% and 0.2% loss, despite their accuracy loss is 0.23% and 0.88% without rejection. Therefore, the compact MQDF classifier is particularly suitable to be used in the recognition with rejection.

Table 7
Result of timing analysis

	Process time (ms per character)	
	PC	Pocket PC
Feature extraction	0.6	32
Coarse classification	1	25
MQDF classification	0.2	7
Total	1.8	64

5.7. Results on a cleaned-up database

We have noticed that in the HCL2000 database, some labels and the corresponding images are not consistent. Many duplicated images are also found in the database. A cleaning-up process was performed on the original HCL2000 database. After that, 451,147 (12.01%) duplicated images and 29,075 (0.77%) mislabeled and heavy noised images were removed. As a result, there are 3,274,778 images remaining in the cleaned-up database. The sample number of each character varies from 756 to 902. We took 700 samples of each character for training and the remaining 646,278 samples for testing (i.e. about 172 samples on average for each character). The experimental results on tradeoff curves by choosing different VQ are shown in Fig. 10. From the results, no big difference is found compared to those on the original HCL2000 database except the recognition rate decreased about 0.3%. In order to keep our work comparable with others, we did not perform other experiments again on our cleaned-up database.

5.8. Timing analysis

We implemented the compact MQDF classifiers for both PC (P4 2.8 G CPU) and Pocket PC (Asus MyPal A730 Pocket PC). The timing analysis of the compact classifier when $D_Q = 2$, $L = 256$, $K = 8$ and $D_L = 96$ is shown in Table 7. The total recognition time for one Chinese character only consumes 1.8 ms on a PC and 64 ms on a Pocket PC, which promised the performance for the mainstream hand-held devices.

6. Conclusion

In this paper, we proposed a method for building compact and high accuracy MQDF classifier. We first reduce the dimension of vectors by LDA. After that, the original parameter space is mapped into multiple subspaces by splitting the dimension. A uniform distribution for the subspaces is then found by the clustering process. By using a small set of prototypes clustered from the original subspaces to represent the uncompressed sub-vectors, the storage of the MQDF parameters is greatly compressed. By seeking for the optimal tradeoff curves between parameter size and recognition accuracy, some sets of

parameter settings are discovered to form the optimal compact dictionary for MQDF parameters. From the experimental results, the compact MQDF classifier with even a smaller dictionary size, compared to the conventional LDA classifier, raises the recognition rate from 94.23% to 97.09%. To the best of our knowledge, no other research has attempted to solve the storage problem of the MQDF classifier for large scale character set recognition. With the reduced memory requirement of the compact dictionary and the accelerated coarse classification by a two-level coarse classifier, the high accuracy MQDF classifier now becomes practical to be used for the memory limited handheld devices such as PDAs, mobile phones and Pocket PCs.

Acknowledgment

The Authors sincerely thank Dr. Qiang HUO for his helpful discussions about the technique of split VQ. This work is supported in part by New Century Excellent Talent Program of MOE of China (Grant no. NCET-05-0736) and The University Fund of Microsoft Research Asia (No. FY07-RES-THEME-58).

References

[1] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. Mach. Intell. 22 (2000) 4–37.

[2] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, Modified quadratic discriminant functions and the application to Chinese character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1) (1987) 149–153.

[3] F. Kimura, M. Shridhar, Handwritten numeral recognition based on multiple algorithms, Pattern Recognition 24 (10) (1991) 969–981.

[4] H. Liu, X. Ding, Handwritten character recognition using gradient feature and quadratic classifier with multiple discrimination schemes, in: Proceedings of the 8th ICDAR, Seoul, Korea, 2005, pp. 19–23.

[5] C.-L. Liu, High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction, in: 18th International Conference on Pattern Recognition, 2006, pp. 942–945.

[6] F. Kimura, T. Wakabayashi, S. Tsuruoka, Y. Miyake, Improvement of handwritten Japanese character recognition using weighted direction code histogram, Pattern Recognition 30 (8) (1997) 1329–1337.

[7] C.-L. Liu, R. Mine, M. Koga, Building compact classifier for large character set recognition using discriminative feature extraction, in: Proceedings of the 8th ICDAR, Seoul, Korea, 2005, pp. 846–850.

[8] D. de Ridder, E. Pekalska, R.P.W. Duin, The economics of classification: Error vs. Complexity, in: Proceedings of the 16th ICPR, 2002, pp. 244–247.

[9] J.X. Dong, A. Krzyzak, C.D. Suen, An improved handwritten Chinese character recognition system using support vector machine, Pattern Recognition Lett. 26 (12) (2005) 1849–1856.

[10] T. Long, L.-W. Jin, Building compact MQDF classifier for off-line handwritten Chinese characters by subspace distribution sharing, in: Proceedings of the 9th ICDAR, vol. 2, 2007, pp. 909–913.

[11] E. Bocchieri, B. Mak, Subspace distribution clustering hidden Markov model, IEEE Trans. Speech Audio Process. 9 (3) (2001) 264–275.

[12] W. Law, C.F. Chan, Split-dimension vector quantization of Parcor coefficients for low bit rate speech coding, IEEE Trans. Speech Audio Process. 2 (1994) 443–446.

[13] Q. Huo, Y. Ge, Z.-D. Feng, High performance Chinese OCR based on gabor features, discriminative feature extraction and model training, in: Proceedings of the ICASSP-2001, May 7–11, Salt Lake City, USA, 2001, pp. III-1517–III-1520.

[14] Q. Huo, Z.-D. Feng, Improving Chinese/English OCR performance by using MCE-based character-pair modeling and negative training, in: Proceedings of the ICDAR-2003, August 3–6, Edinburgh, UK, 2003, pp. 364–368.

[15] Y. Ge, Q. Huo, A comparative study of several modeling approaches for large vocabulary offline recognition of handwritten Chinese characters, in: Proceedings of the ICPR-2002 August 11–15, Quebec, Canada, 2002, pp. III-85–III-88.

[16] Y. Ge, Q. Huo, A study on the use of CDHMM for large vocabulary off-line recognition of handwritten Chinese characters, in: Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition, 2002, pp. 334–338.

[17] J. Guo, Z.-Q. Lin, H.-G. Zhang, A new database model of off-line handwritten Chinese characters and its applications, Acta Electron. Sin. 28 (5) (2000) 115–116.

[18] L.-W. Jin, G. Wei, Handwritten Chinese character recognition with directional decomposition cellular features, J. Circuit Syst. Comput. 8 (4) (1999) 517–524.

[19] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: investigation of normalization and feature extraction techniques, Pattern Recognition 37 (2) (2004) 265–279.

[20] T. Wakabayashi, S. Tsuruoka, F. Kimura, Y. Miyake, On the size and variable transformation of feature vector for handwritten character recognition, Trans. IEICE Jpn J76-D-II (12) (1993) 2495–2503.

[21] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 696–710.

[22] C.-L. Liu, H. Sako, H. Fujisawa, Discriminative learning quadratic discriminant function for handwriting recognition, IEEE Trans. Neural Networks 15 (2) (2004) 430–444.

[23] Y. Linde, A. Buzo, R.M. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. COM-28 (1) (1980) 84–95.

[24] C.-L. Liu, H. Sako, H. Fujisawa, Performance evaluation of pattern classifiers for handwritten character recognition, Int. J. Doc. Anal. Recognition 2002(4) 191–204.

About the Author—TENG LONG received his B.S. degree in 2003, and pursuing his Ph.D. degree, in Communication and Information System at South China University of Technology, Guangzhou, China. His research interests include pattern recognition, machine learning, computer vision and signal processing.

About the Author—LIANWEN JIN obtained B. Engineering from University of Science and Technology of China and Ph.D. in Communication and Information System from South China University of Technology in 1991 and 1996, respectively. He visited Motorola China Research Center in 2000, the University of Hongkong in 2002 and 2006 as research fellow, respectively. He is now a professor at School of Electronic and Information Engineering, South China University of Technology. His current research fields include handwritten character recognition, pattern analysis and recognition, image processing, machine learning and intelligent system.