Pattern Recognition 49 (2016) 7–18

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Discriminative quadratic feature learning for handwritten Chinese character recognition

Ming-Ke Zhou*, Xu-Yao Zhang, Fei Yin, Cheng-Lin Liu

National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No. 95 Zhongguancun East Road, Beijing 100190, PR China

ARTICLE INFO

Article history:
Received 7 February 2015
Received in revised form
26 June 2015
Accepted 22 July 2015
Available online 1 August 2015

Keywords:
Handwritten Chinese character recognition
Discriminative feature learning
Quadratic correlation
Dimensionality promotion
Training set expansion

ABSTRACT

In this paper, we propose a feature learning method for handwritten Chinese character recognition (HCCR), called discriminative quadratic feature learning (DQFL). Based on original gradient direction feature representation, quadratic correlation between features is used to promote the feature dimensionality, then discriminative feature extraction (DFE) is used for dimensionality reduction. By combining dimensionality promotion and reduction, we can learn a much more discriminative and nonlinear feature representation, which can then boost the classification accuracy significantly. For dimensionality promotion, two types of correlation are exploited, namely, statistical correlation and spatial correlation. Statistical correlation is computed on multiple local feature vectors in different regions of the character image; while spatial correlation encodes the dependency between features of two positions. Feature correlation increases the dimensionality by over 40,000. DFE then reduces the dimensionality to less than 300 without losing discriminability. Classification is performed using nearest prototype classifier (NPC), modified quadratic discriminant function (MQDF) and discriminative learning quadratic discriminant function (DLQDF). In experiments on the CASIA-HWDB1.1 standard dataset, the proposed DQFL method improves the test accuracies of NPC, MQDF and DLQDF by 4.94%, 1.83%, and 1.82%, respectively. The test accuracy is further improved by training set expansion. On the ICDAR 2013 Chinese handwriting recognition competition dataset, the proposed DQFL+DLQDF classifier outperforms the best participating system based on deep convolutional neural network (CNN), while the test speed is much faster.

1. Introduction

Machine recognition of Chinese handwriting finds many applications, such as mail sorting [1], bank check reading, tax form processing, book and handwritten notes transcription, and so on. Handwritten Chinese character recognition (HCCR), which is an integral part of handwritten text recognition [2], is not solved yet despite that it has been studied for about fifty years [3]. Previous methods perform well only on constrained writings [3], but for free handwriting recognition, the performance is not satisfactory [4]. For example, the ICDAR 2013 Chinese handwriting recognition competition reported the best result of isolated character recognition accuracy 94.77% [5].

HCCR is difficult due to the large number of character classes, the presence of many confusing character pairs, and the variability of writing styles. There are over 30,000 Chinese characters in total, and the number of daily used ones is about 5000. Fig. 1 shows some confusing character pairs, and Fig. 2 shows some samples with different writing styles.

Traditional HCCR methods mostly use hand-crafted features, and in recent years, some works based on feature learning using deep neural networks, especially, deep convolutional neural networks (CNNs), have reported superior performance [6–8]. Traditional methods recognize a sample in a pipeline of shape normalization, feature extraction, dimensionality reduction and classification. Many effective methods have been proposed for these steps and have improved the performance of HCCR constantly. Some representative methods are nonlinear normalization [9,10] and pseudo-two-dimensional normalization [11,12], chaidcode [13] and gradient direction [14] feature extraction, modified quadratic discriminant function (MQDF) [15] and discriminative learning quadratic discriminant function (DLQDF) [16] classifiers. The performance of traditional methods has reached a bottleneck because of the restricted representability of handcrafted features. However, this pipeline of recognition is very efficient in computation.

Deep CNNs, which are comprised of multiple layers of convolution and pooling for automatic feature learning and fully connected layers for classification, have outperformed traditional methods by a large margin [7,5,8]. The deep hierarchical structure of CNNs

* Corresponding author. Tel.: +86 15210983082.
E-mail addresses: mkzhou@nlpr.ia.ac.cn (M.-K. Zhou), xyz@nlpr.ia.ac.cn (X.-Y. Zhang), fyin@nlpr.ia.ac.cn (F. Yin), liucl@nlpr.ia.ac.cn (C.-L. Liu).

http://dx.doi.org/10.1016/j.patcog.2015.07.007
0031-3203/

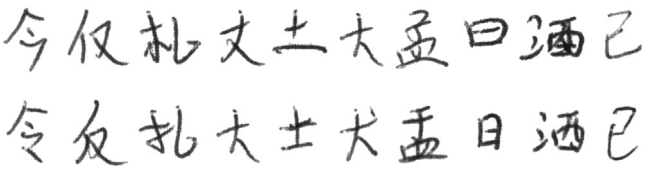

Fig. 1. Examples of confusing character pairs. Each column is a pair of different classes.

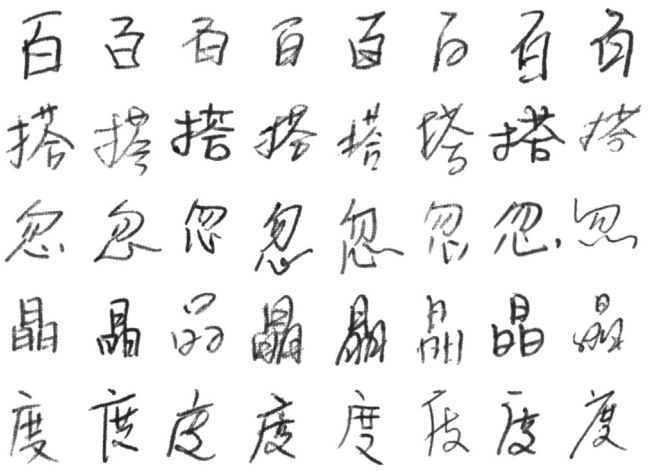

Fig. 2. Some characters with different writing styles. Each row shows the samples of one class.

enables it to learn high-level features relevant for recognition. Usually, a large training set produced by affine transform or elastic distortion is used to improve its generalization ability. In a recent HCCR competition [5], deep CNN committee reported test accuracies comparable to human observers [7,5,8]. However, the computation burden of deep CNNs is much higher than that of traditional methods due to multiple layers of convolution. Its training is feasible only when using massive parallel computing with graphics processing unit (GPU).

In this paper, within the traditional HCCR framework, we propose a feature learning method called discriminative quadratic feature learning (DQFL). The motivation is from three aspects. First, quadratic or nonlinear classifiers like MQDF, DLQDF, and polynomial classifier (PC) [17] perform much better than linear ones such as nearest prototype classifier (NPC) [18]. This signifies the importance of quadratic information embedded in data. Second, traditional methods usually use linear dimensionality reduction, which is not sufficient to HCCR because of the large class set. Third, high feature dimensionality has shown benefits in some vision recognition tasks [19] due to the strong discriminative capability brought by augmented feature vectors. In this paper, we adopt a simple strategy to learn discriminative quadratic features – deterministic dimensionality promotion followed by discriminatively learned linear dimensionality reduction. In dimensionality promotion, quadratic correlation between original features is sufficiently utilized to expand tens of thousands of quadratic features; while in dimensionality reduction, linear discriminative learning guarantees good class separation. The final obtained feature subspace is quadratic with respect to the original features. In this subspace, we train different classifiers, and further improve the classification performance by training set expansion. In experiments on the standard dataset of CASIA-HWDB1.1 [20] and the ICDAR 2013 Chinese handwriting recognition competition dataset, the proposed method achieved accuracies comparable to deep CNNs while the test speed is much faster.

Table 1
Some frequently used acronyms and their original names in this paper.

Acronym	Original name
DQFL	Discriminative quadratic feature learning
DFE	Discriminative feature extraction
HCCR	Handwritten Chinese character recognition
NPC	Nearest prototype classifier
MQDF	Modified quadratic discriminant function
DLQDF	Discriminative learning quadratic discriminant function
CNN	Convolutional neural network
LVQ	Learning vector quantization
MCE	Minimum classification error
GDH	Gradient direction histogram
NCGF	Normalization-cooperated gradient feature
GPU	Graphics processing unit

The rest of the paper is organized as follows. Section 2 briefly reviews related works. Section 3 describes the proposed DQFL method. Section 4 introduces the training set expansion method. Section 5 presents experimental results and Section 6 concludes the paper. To make this paper more readable, we list some frequently used acronyms in Table 1.

2. Related works

In this section, we review some related works on offline isolated character recognition. We focus on two important issues – feature extraction and classification.

Many types of features have been proposed for character recognition. Among them, the direction histogram feature, describing regional direction histograms of local stroke skeleton or edge, has been used popularly. The local direction can be extracted from chaincode of character contour or image gradient. Chaincode feature is extracted from binarized images while gradient feature can be extracted from both binarized images and gray-scale images. On assigning local direction to some bins (hardly or softly), the character image is partitioned into zones (hardly or softly), each obtaining a histogram of directions. Soft partitioning of zones has been shown to be equivalent to low-pass filtering with down-sampling on directional feature maps [13]. Among other features, the Gabor filter feature has yielded competitive performance in character recognition [21–23], but has higher complexity in computation and feature dimensionality [23].

Various classifiers can be used for character recognition, such as statistical classifiers, artificial neural networks and support vector machines (SVM). Considering the large class number of Chinese characters, HCCR usually uses statistical classifiers, including nearest mean, multi-prototype classifier, quadratic discriminant function (QDF), MQDF classifier. The MQDF is based on Gaussian density assumption like the QDF. It has the minor eigenvalues of each class regularized into a constant, so as to save parameters and improve the generalization performance. In the so-called DLQDF, the parameters of MQDF are optimized by discriminative training under the minimum classification error (MCE) criterion.

Among artificial neural networks, besides traditional multi-layer perceptron (MLP), radial basis function (RBF) network, and high order neural network (HONN), convolutional neural networks achieved the most remarkable results in numeral, English letter and Chinese character recognition [24,25,6]. It uses the technique of local connection and weights sharing to exploit the characteristics of images, and greatly reduces the number of connection weights. Our proposed method, DQFL, is a discriminative model

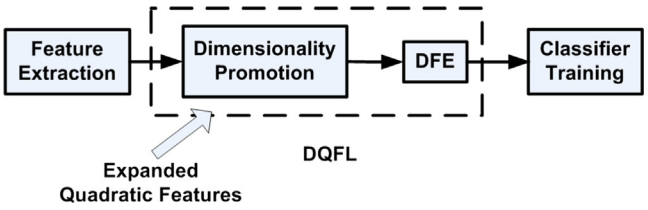

Fig. 3. Block diagram of DQFL.

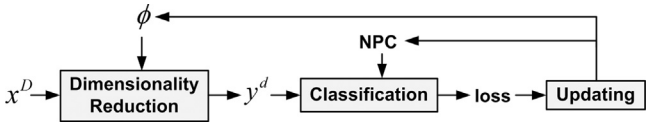

Fig. 4. DFE+LVQ learning with stochastic gradient descent.

like neural networks. The feature dimensionality promotion and training set expansion techniques used in DQFL are also inspired by previous works of neural networks.

3. DQFL

3.1. Overview of DQFL and DFE

Fig. 3 shows the block diagram of DQFL. First, feature extraction is conducted on the character image to obtain the original feature representation; second, feature dimensionality is promoted by quadratic feature expansion; third, discriminative feature extraction (DFE) [26] is used to reduce the feature vector back to a low-dimensional subspace; finally, the classifier is trained. The dimensionality promotion together with DFE dimensionality reduction is called DQFL.

Since DFE is a previously proposed method and plays an important role in DQFL, we give its outline in the following.

DFE is a learning-based linear dimensionality reduction method. It learns the subspace axes by minimizing the empirical loss. As empirical loss is measured by classification in the reduced subspace, the learning of DFE is combined with a specific classifier. NPC is often chosen due to its simplicity [27]. In NPC, each class is represented with one or several prototypes, and classification is done through nearest prototype searching. For its learning, learning vector quantization (LVQ) [28] is often adopted as an effective supervised method. Therefore, subspace axes and prototypes of NPC can be simultaneously learned with DFE and LVQ. As in [29], we denote this learning process with DFE+LVQ learning.

The loss function used for DFE+LVQ learning can be MCE [30] criterion, while conditional log-likelihood loss (log-loss) [31] and other similar criteria can also be chosen. Here we use the log-loss. For defining the log-loss function, at first, a misclassification measure is defined. For a D-dimensional training sample x, the misclassification measure is

$$h(x) = d_E(\phi^T x, m_c) - d_E(\phi^T x, m_r), \tag{1}$$

which represents the difference of two squared Euclidean distances in the d-dimensional subspace. The first distance is from the sample to the genuine prototype, and the second one is from the sample to the closest rival prototype. $h(x) > 0$ signifies misclassification. In this formula, ϕ represents the dimensionality reduction matrix, m_c and m_r are the prototypes from the genuine class and closest rival class of x in the reduced subspace. With $h(x)$, the log-loss is obtained as

$$l(x) = \log\left[1 + e^{\xi h(x)}\right], \tag{2}$$

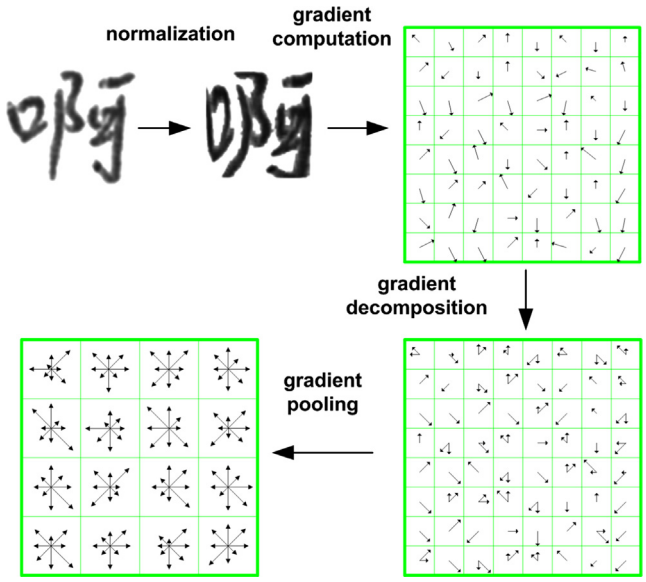

Fig. 5. The process of GDH feature extraction.

where ξ is a constant for constraining the absolute value of $\xi h(x)$. In addition, to constrain the excessive deviation of parameters from the maximum likelihood estimation, a regularization term is usually added, and the final loss function is

$$l'(x) = l(x) + \alpha d_E(\phi^T x, m_c). \tag{3}$$

Therefore, the empirical loss is

$$L = \frac{1}{N}\sum_{n=1}^{N}\left[l(x^n) + \alpha d_E(\phi^T x^n, m_c)\right]. \tag{4}$$

For DFE+LVQ learning, the empirical loss is minimized iteratively on a training set by stochastic or mini-batch gradient descent. Fisher discriminant analysis (FDA) [32] and k-means are usually used for initialization of subspace axes and prototypes, respectively.

The stochastic gradient descent training process is illustrated in Fig. 4. For each iteration, an input training sample x is dimensionality-reduced using current value of ϕ to produce the reduced feature vector y. By classifying y with NPC classifier, the classification loss is produced. Gradients for ϕ and prototypes of NPC can be obtained by computing the derivatives of the loss function. Then the gradients are used for updating all the parameters.

3.2. Feature extraction

We use gradient directional histogram (GDH) [13] for original feature representation of character images. In GDH representation, a grid of GDHs is used to represent the stroke direction distribution in different areas of a character image. Its computation is illustrated in Fig. 5. First, the character image is normalized to a given size to reduce the within-class variability; second, gradients are computed using Sobel masks for every pixel of the normalized image to produce the gradient map; third, each computed gradient is decomposed into components in two neighboring standard directions. Fig. 6 shows the gradient decomposition. Fourth, zoning is used to generate a grid of zones on the gradient map, and within each zone, one GDH is produced by pooling gradients. The final representation is a concatenation of all these GDHs, and we call it the GDH map. There are two important parameters for describing a GDH map – the size of the map (or the number of zones in a row/column during zoning) and the dimensionality of a GDH (or the number of standard directions), which are denoted by N_z and N_d. For implementation, we adopt Liu's normalization-cooperated gradient feature (NCGF)

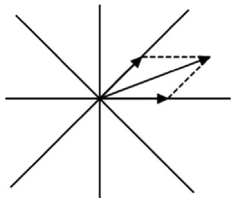

Fig. 6. Decomposing a gradient into two components in neighboring directions.

extraction [33]. In this method, normalized images are not generated. Instead, gradients are computed on original images, and then are mapped to the gradient map. The coordinate mapping function is produced during character normalization. In addition, for gradient pooling on the gradient map, Gaussian blurring and down-sampling are adopted to produce GDHs.

The quadratic feature expansion encodes quadratic correlation between original GDH features. Two types of quadratic correlation are exploited in this work, namely statistical correlation and spatial correlation. Statistical correlation encodes correlation of GDH features in different regions of the GDH map, while spatial correlation encodes correlation between neighboring GDH features. Details of these two quadratic feature expansion methods are presented in the following subsections.

3.3. Statistical correlation expansion

In statistical correlation expansion, quadratic features are generated to encode statistical correlation between GDH features. As each GDH in a map represents stroke direction distribution in a local area of a character image, the correlation of its elements corresponds to correlation between stroke directions in this local area. This correlation is informative and can be incorporated to enhance the representation capability of original GDH features. For a GDH represented by $x^k = \left(x^k_0, x^k_1, \ldots, x^k_{N_d-1}\right)^T$, quadratic terms like $\left\{x^k_i \cdot x^k_j \middle| 1 \le i \le j \le N_d - 1\right\}$ encode quadratic correlation between given directions, where N_d is the number of standard directions. However, these quadratic terms are not robust, and the number is very large. To overcome this problem, we use their statistics in regions of the GDH map. Specifically, average-pooling is utilized in each pooling region for each of these quadratic terms. In formula, for a pooling region R of N_R GDHs, an autocorrelation matrix is first computed as

$$Corr_R = \frac{1}{N_R}\sum_{i=1}^{N_R} x^i (x^i)^T. \tag{5}$$

Then all the $N_d \cdot (N_d+1)/2$ unique elements of $Corr_R$ are used as quadratic features. What's more, to make them at the same magnitude with original GDH features, we take their square roots. Similarly, other statistical features like region covariance [34] can also be adopted. For computing region covariance features, a covariance matrix is first computed, and signed square roots of its elements are used

$$Cov_R = \frac{1}{N_R}\sum_{i=1}^{N_R} (x^i - m^R)(x^i - m^R)^T, \tag{6}$$

where m^R is the mean GDH in region R.

To construct regions for statistical correlation expansion, we make a multi-level partition of the GDH map, and on each level of partition, neighboring regions are overlapped for utilizing correlation near region boundaries. The overlapping area is half of the region area. As partition level goes up, the region size shrinks at a ratio of 1/2. For example, when the size of the GDH map N_z is 16, on partition level 0, the whole GDH map is the only region; on partition level 1, the region size is 8, and the width of the overlapping area is 4, and so on. When $N_z = 16$, the region size, overlapping area width, and number of regions on each partition level are listed in Table 2.

Table 2
Region size, overlapping area width and number of regions on different partition levels.

Partition level	0	1	2
Region size	16	8	4
Overlap width	8	4	2
Region number	1	9	49

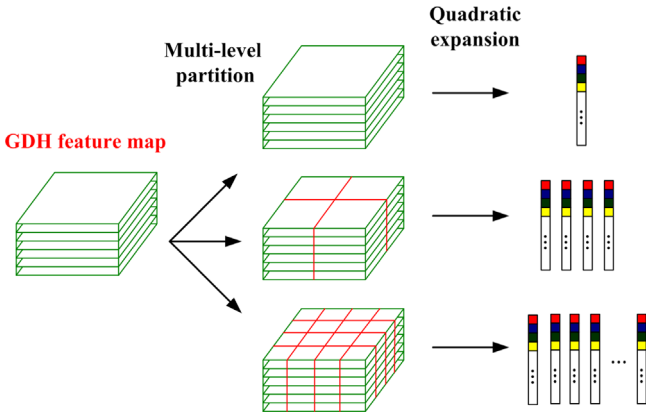

Fig. 7. The process of statistical correlation expansion.

The whole process of statistical correlation expansion is illustrated in Fig. 7. The GDH feature map is partitioned into overlapping regions in multiple levels, and corresponding to each region, $N_d \cdot (N_d+1)/2$ quadratic features are generated. All these quadratic features from different regions of different sizes are concatenated to constitute statistical correlation expansion.

3.4. Spatial correlation expansion

In spatial correlation expansion, the generated quadratic features encode correlation between GDH features of neighboring spatial positions. It is based on the observation that strong correlation exists between GDH features from close areas of a character image.

The GDH map can be considered as N_d directional feature planes with each plane holding features in its corresponding gradient direction. According to whether the GDH features used for expansion are in the same feature plane or not, spatial correlation expansion can be partitioned into intra-plane expansion and inter-plane expansion.

3.4.1. Intra-plane expansion

In intra-plane expansion, all neighboring features from the same feature plane are used. The set of expanded quadratic features can be represented as

$$\left\{\sqrt{f_d(p)\cdot f_d(q)} \middle| p \in Neighbor(q), 0 \le d \le N_d - 1\right\}, \tag{7}$$

where $f_d(p)$ represents the feature with coordinate p in directional plane f_d. $Neighbor(q)$ represents the collection of all neighboring coordinates of coordinate q. The neighborhood definition adopted is 8-neighborhood.

Spatial correlation expansion stated above is constrained within local areas, so only local correlation between features is used. Compared to local correlation, distant correlation contains higher level and more global information. In order to encode correlation of

different scales, as in deep CNNs, the directional feature planes are down-sampled in multiple levels to produce a multi-resolution representation. As the resolution goes down, the feature plane size is reduced by a half each time, and on each resolution level, the same spatial correlation expansion presented above is conducted. Therefore, spatial correlation expansion on a high resolution level encodes local correlation while that on a low resolution level encodes global correlation. With multi-resolution representation, the expansion can be formulated as

$$\left\{\sqrt{f_d^l(p)\cdot f_d^l(q)}\,|\,p\in Neighbor(q),\quad 0\le d\le N_d-1,\quad 0\le l\le L-1\right\},\tag{8}$$

where f_d^l represents the dth feature plane on resolution level l. The size of feature planes on the last resolution level $L-1$ is 2. So the number of resolutions L is decided by the size of original GDH feature planes N_z with $L=\log_2 N_z$. In Fig. 8, the process of intra-plane spatial correlation expansion is illustrated. The original GDH feature map is down-sampled to create the multi-resolution representation. On each resolution level, directional feature planes are obtained by separating GDH features in different directions, and on each feature plane, quadratic features are generated by expanding neighboring GDH feature pairs.

To construct the multi-resolution representation, following the multi-resolution high order neural networks (HONN) [35] of Liu et al., two down-sampling methods are adopted – QuadTree and Gaussian blurring. In QuadTree down-sampling, feature planes of level l are generated from those of level $l-1$. For each 2×2 block on level $l-1$, its average is used as the corresponding down-sampled value on level l:

$$f_d^l(x,y)=\tfrac{1}{4}\left[f_d^{l-1}(2x,2y)+f_d^{l-1}(2x+1,2y)\right.$$
$$\left.+f_d^{l-1}(2x,2y+1)+f_d^{l-1}(2x+1,2y+1)\right].\tag{9}$$

In Gaussian down-sampling, original (resolution level 0) GDH feature planes are first convolved with a Gaussian mask of deviation $\sigma_l=\sigma_0\times 2^{l-1}$:

$$G(x,y)=\frac{1}{2\pi\sigma_l^2}e^{-(x^2+y^2)/2\sigma_l^2},\tag{10}$$

and then down-sampled. Different values of σ_l and sampling intervals are used for different resolution levels.

3.4.2. Inter-plane expansion

Similar to intra-plane expansion, inter-plane expansion encodes correlation between neighboring features from different feature planes. In inter-plane expansion, a parameter G_d is defined to represent the direction gap between the two feature planes to be expanded. For example when $G_d=1$, the expansion is between planes that are close in direction; while $G_d=N_d/2$ represents the expansion between planes in opposite directions. The rule of inter-plane expansion is similar to that of intra-plane expansion. From a given pair of planes $(f_{d_1}^l, f_{d_2}^l)$, with two neighboring points (p,q), two quadratic features are generated – $\sqrt{f_{d_1}^l(p)\cdot f_{d_2}^l(q)}$ and $\sqrt{f_{d_1}^l(q)\cdot f_{d_2}^l(p)}$. Therefore, when G_d is fixed, the inter-plane expansion can be formulated as

$$\begin{cases}\left\{\sqrt{f_{d_1}^l(p)\cdot f_{d_2}^l(q)},\sqrt{f_{d_1}^l(q)\cdot f_{d_2}^l(p)}\,|\,p\in Neighbor(q),\right.\\ \left.d_1<d_2,(d_2-d_1+N_d)\%N_d=G_d,0\le l\le L-1\right\},\\ \quad \text{for } 1\le G_d<N_d/2-1,\\ \left\{\sqrt{f_{d_1}^l(p)\cdot f_{d_2}^l(q)},\sqrt{f_{d_1}^l(q)\cdot f_{d_2}^l(p)}\,|\,p\in Neighbor(q),\right.\\ \left.d_2-d_1=G_d,\ 0\le l\le L-1,\right\}\qquad \text{for } G_d=N_d/2.\end{cases}\tag{11}$$

To count the number of generated quadratic features by spatial correlation expansion in each resolution, we take resolution 0 as an example. For inter-plane expansion, from each feature plane, there are $N_z\cdot(N_z-1)$ pairs of neighboring GDH features in horizontal and vertical direction; while for each diagonal direction, there are $(N_z-1)\cdot(N_z-1)$ feature pairs. If we denote the number of quadratic features generated from each feature plane during intra-plane expansion by M, then M is $2\cdot N_z\cdot(N_z-1)+2\cdot(N_z-1)\cdot(N_z-1)$. Counting all planes, there are $N_d\cdot M$ quadratic features generated from intra-plane expansion. For inter-plane expansion, from Eq. (11), between each pair of planes, $2M$ quadratic features are produced. The total number of plane pairs is $N_d\cdot(N_d/2-1)+N_d/2$. On other resolution levels, the same expansion method is conducted. In Table 3, the numbers of quadratic features generated on different resolution levels are shown.

4. Training set expansion

Large training set is crucial for discriminative training of large models (such as deep CNNs and the proposed DQFL) to improve

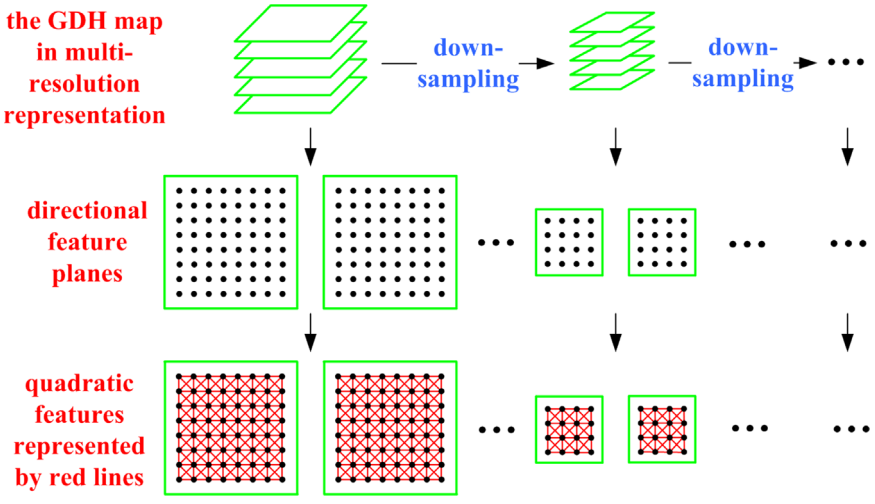

Fig. 8. Intra-plane spatial correlation expansion. Each red line at the bottom of the figure represents one expanded quadratic feature using two neighboring GDH features on the same directional feature plane. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Table 3
Number of quadratic features generated by spatial correlation expansion on each resolution level. The size of the GDH map N_z and the number of standard directions N_d are 16 and 12, respectively.

Resolution level	0	1	2	3	Sum
Plane size	16	8	4	2	
M	930	210	42	6	1188
# Intra-plane expansion	11,160	2520	504	72	14,256
# Inter-plane expansion	122,760	27,720	5544	792	156,816
Sum	133,920	30,240	6048	864	171,072

the generalization performance. However, in practice, collecting and labeling a big training set are very labor-consuming. Popular HCCR data sets have only hundreds of samples for each class, while the number of character classes is thousands. To deal with this problem, researchers usually expand training set by using synthesized samples. This scheme is especially successful for deep CNNs [36,6]. In this paper, to improve the performance of our system, we also tried training set expansion.

The synthesized samples are generated by distorting real ones. The distortion functions used in this paper can be partitioned into three categories, namely, geometric transform, local resizing [37] and elastic distortion [36]. For geometric transform, we adopt models proposed in [38]. They are rotation, shearing (slant transform), perspective transform and shrink transform. Among these geometric transforms, shearing can be done in horizontal and vertical directions, while perspective transform and shrink transform can be done in horizontal, vertical, left diagonal and right diagonal directions. Taking direction into consideration, there are eleven geometric distortion functions. Local resizing is a one-dimensional coordinate transform used to adjust the relative ratio of different parts of character images. Readers can refer to [37] for details. There are two one-dimensional local resizing functions, namely, w_1 and w_2, both of which can be done in horizontal and vertical directions. So combining the function type with direction, there are four local resizing distortion functions. Elastic distortion is a locally random distortion which simulates stroke distortion caused by hand muscle trembling during the writing process. In implementation, elastic distortion is represented by random coordinates shifting. So in total, there are 16 distortion functions from three distortion categories. Fig. 9 illustrates the effects of different distortion functions acted on a checkerboard image. The top row is the original checkerboard image, the results of rotation, horizontal shearing and perspective transform in horizontal direction; the bottom row is the results of horizontal shrink transform, local resizing function w_1, w_2 both in horizontal direction and elastic distortion.

To generate distortion of different degrees, we adopted two distortion schemes – the single model and the combined model. The single model uses only one randomly selected distortion function for each synthesis, while the combined model uses a combined distortion of shearing and local resizing in both horizontal and vertical directions [37]. For all the two distortion models, the parameters of all distortion functions are randomly generated. Figs. 10 and 11 show some synthesized samples generated by the single model and the combined model, respectively. The first images in Figs. 10 and 11 are original images, and the rest are distorted images.

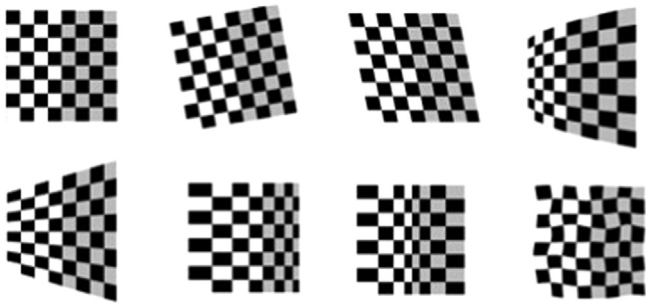

Fig. 9. Effect of distortion functions acted on a checkerboard image. The top row, from left to right, is the original checkerboard image, the results of rotation, horizontal shearing and perspective transform in horizontal direction. The bottom row, from left to right, is the results of horizontal shrink transform, local resizing function w_1 and w_2 both in the horizontal direction, elastic distortion.

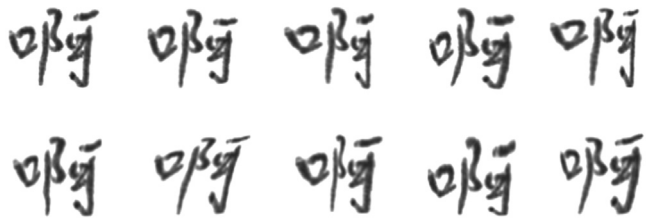

Fig. 10. Samples generated by randomly selected single distortion models.

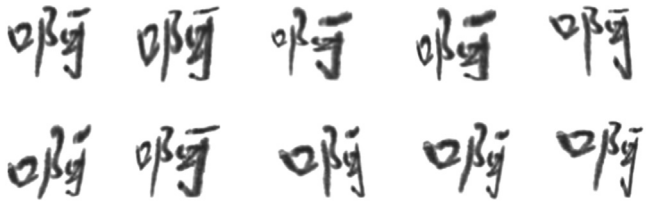

Fig. 11. Samples generated by the combined distortion model of shearing and local resizing.

5. Experimental results

For experiment, we used two datasets collected by the Institute of Automation of Chinese Academy of Sciences (CASIA). The first one is CASIA-HWDB1.1 [20]. It consists of off-line handwritten Chinese characters from GB2312-80 level-1 set containing 3755 characters, and each character is written by 300 writers. This dataset is partitioned into a training set of 240 writers and a test set of 60 writers. There are totally 897,758 training samples and 223,991 test samples. We also evaluate the performance on the test set of ICDAR 2013 Chinese handwriting recognition competition [5] (denoted as ICDAR-2013) for comparing with the published results in competition. This dataset contains 224,419 samples of 3755 classes written by 60 writers different from those of CASIA-HWDB1.1.

The training and test process of our HCCR system is illustrated in Fig. 12. During the training process, training samples are first feature-extracted, and then quadratic features are generated. After that, DFE+LVQ learning is conducted to produce the subspace basis and prototypes of NPC classifier. After dimensionality reduction, MQDF classifier and DLQDF classifier are trained. All the data including subspace basis and classifier (NPC, MQDF, or DLQDF) parameters are stored into a classifier dictionary. During test, the GDH feature extraction and quadratic feature expansion process are the same as training, the subspace basis and classifier parameters loaded from the classifier dictionary are used for dimensionality reduction and classification.

The GDH feature extraction actually consists of character normalization and GDH feature generation. We used NCGF [33] to combine these two processes. The normalization method used is line density projection interpolation (LDPI) [12] which is a simplified version of

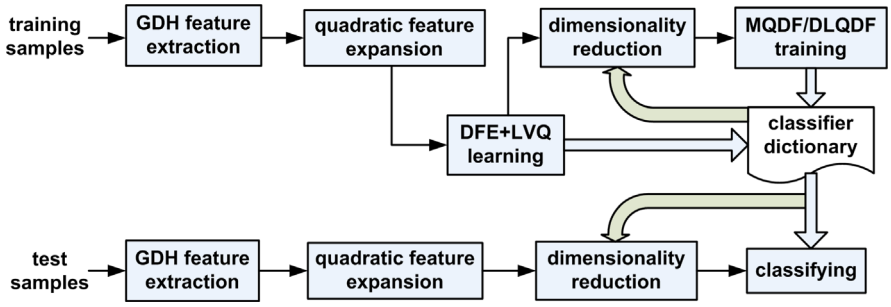

Fig. 12. The training and test process of our HCCR system.

two-dimensional nonlinear normalization [11]. For dimensionality reduction, three subspace dimensionalities are tested, namely, 160, 200 and 250. The configuration of classifiers are as follows: for NPC classifier, each class possesses one prototype; for MQDF and DLQDF classifier, the numbers of principal eigenvectors are 50, 50 and 80 for subspace dimensionality 160, 200 and 250, respectively. The minor eigenvalue is set to be class-independent, and it is the average of all the eigenvalues over all classes.

Due to the high dimensionality (after quadratic feature expansion), the large training set (especially after training set expansion), and the high time complexity of the training process, we implement some algorithms using CUDA GPU parallel computing [39]. During training, the dimensionality reduction, the DFE+LVQ learning, the training of MQDF and DLQDF are parallelized. For dimensionality reduction, a GPU accelerated matrix-matrix multiplication routine from CUBLAS library [40] is invoked. For MQDF training, we also use CUBLAS library for covariance matrices computation, and use an open-source GPU accelerated linear algebra library named MAGMA [41] for eigenvalues decomposition. For DLQDF training, we adopt mini-batch gradient descent for optimization. During each iteration, samples are classified in batches on GPU, while the computation of gradients and updating of parameters are conducted on CPU. Readers can refer to [42] for the details. The DFE+LVQ learning is similar to that of DLQDF, except that during each iteration, besides classifier parameters, the subspace basis is also updated. During test, both the dimensionality reduction and classification process are parallelized. All the experiments are conducted on our GPU server with 2 Intel 2.6 GHz CPUs and 4 NVIDIA Tesla C2075 GPUs.

In the following experiments, we will examine the effect of statistical correlation expansion, spatial correlation expansion, the combination of them, and finally training set expansion.

As a baseline, we first give the results without using DQFL. DFE+LVQ learning is thus applied on original GDH features to produce a linear subspace. The size of the GDH map N_z and the number of standard gradient directions N_d is set at 8 and 12. So the dimensionality of GDH features is $8 \times 8 \times 12 = 768$. The test accuracies of different classifiers on CASIA-HWDB1.1 dataset are shown in Table 4, where *sub_dim* is the subspace dimensionality.

5.1. Effect of statistical correlation expansion

In this subsection, we examine the effect of statistical correlation expansion, and study the influence of parameters on this expansion. The final feature vector consists of GDH features and quadratic features. For GDH features, the same 768-dimensional features as in the baseline experiment are used in this and all following experiments. For quadratic feature expansion, another GDH map with different size and number of standard directions is generated. As GDH features are fixed for all experiments, hereafter when parameter of the GDH map, N_z or N_d, are mentioned, we mean the second GDH map. We fix the subspace dimensionality at 160 for all experiments in this subsection.

Table 4
Test accuracies of classifiers on CASIA-HWDB1.1 dataset using only 768-dimensional GDH features.

Classifier	sub_dim		
	160	200	250
NPC	87.27	87.36	87.29
MQDF	91.01	91.18	91.27
DLQDF	91.13	91.27	91.37

Table 5
The influence on test accuracy caused by the number of standard gradient directions N_d in statistical correlation expansion. N_z is fixed at 16. Multi-level overlapped region partition is used for region construction.

N_d (dim_q)	8 (2124)		12 (4602)		16 (8024)	
Feature	corr	cov	corr	cov	corr	cov
NPC	90.18	89.21	90.47	89.42	90.39	89.45
MQDF	92.21	91.87	92.30	91.95	92.24	91.96
DLQDF	92.30	91.96	92.40	92.05	92.31	92.07

In the following four experiments of statistical correlation expansion, we inspect the influence of four factors on the test accuracy. They are the size of the GDH map N_z, the number of standard gradient directions N_d, the way of region partition (single-level or multi-level, overlapped or non-overlap), using statistical correlation expansion or inter-direction expansion.

In the first experiment, we fix N_z at 16, and vary the value of N_d (the number of standard gradient directions). So the number of generated quadratic features also varies. Table 5 shows the test accuracies of different classifiers on CASIA-HWDB1.1 dataset, where dim_q is the number of generated quadratic features. In this experiment, two kinds of statistical features, namely, covariance (cov) feature and autocorrelation (corr) feature are tested. We can see that autocorrelation feature is more effective than covariance feature. When comparing different values of N_d, for autocorrelation feature, the best results are obtained when N_d is 12; while for covariance feature, the performance is close when N_d is 12 and 16. For both statistical features, the worst performance appears when N_d is 8, which indicates that increasing direction resolution in a certain degree is beneficial. Compared to Table 4, both these two types of statistical quadratic features improve the performance significantly. In the following experiments, N_d is fixed at 12.

In the second experiment, we vary the value of N_z (the size of the GDH map). For all values of N_z, multi-level partition is used to generate 59 regions for statistical correlation expansion. So the number of quadratic features dim_q is fixed at 4602. The test accuracies on CASIA-HWDB1.1 dataset are shown in Table 6. We can see that performance falls behind when N_z is 8, and the difference between $N_z = 16$ and $N_z = 24$ is insignificant. The reason is possibly that a bigger

Table 6
The influence on test accuracy caused by the size of the GDH map N_z in statistical correlation expansion. N_d is fixed at 12. Multi-level overlapped region partition is used for region construction.

N_z (dim_q)	8 (4602)		16 (4602)		24 (4602)	
Feature	corr	cov	corr	cov	corr	cov
NPC	90.00	88.98	90.47	89.42	90.53	89.55
MQDF	91.92	91.57	92.30	91.95	92.36	91.97
DLQDF	92.00	91.69	92.40	92.05	92.43	92.09

Table 7
The influence on test accuracy caused by the number of partition levels as well as overlapped region partition used in statistical correlation expansion. N_z and N_d are fixed at 16 and 12, respectively.

Region partition	Overlap (2)		Overlap (2+1)		Overlap (2+1+0)		Non-overlap (2+1+0)	
#Regions (dim_q)	49 (3822)		58 (4524)		59 (4602)		21 (1638)	
Feature	corr	cov	corr	cov	corr	cov	corr	cov
NPC	90.14	89.15	90.44	89.39	90.47	89.42	89.71	88.69
MQDF	92.17	91.89	92.26	91.94	92.30	91.95	91.96	91.56
DLQDF	92.29	91.97	92.35	92.04	92.40	92.05	92.06	91.67

Table 8
Test accuracies of classifiers on CASIA-HWDB1.1 dataset using inter-direction expansion. N_d is fixed at 12.

N_z (dim_q)	8 (5760)	12 (12,000)	16 (20,736)
NPC	89.30	89.85	89.68
MQDF	91.43	91.87	91.74
DLQDF	91.52	91.97	91.79

GDH map produces bigger regions which leads to more stable statistical quadratic features. Considering computation complexity, we set N_z at 16 in all subsequent experiments.

In the third experiment, we inspect the influence of the number of partition levels as well as overlapped region partition. From Table 2, we see that as partition level goes up, the region size goes down, and the number of regions obtained is increased. At the same time, overlapped region partition also increases the region number. In this experiment, we first only use regions in the finest partition level, and then gradually add in regions in coarser partitions, and overlapped region partition is used for each level. Finally, we change to non-overlapped region partition. Table 7 shows the test results. It indicates that multi-level partition and overlapped partition are both beneficial.

In the last experiment of this subsection, we compare statistical correlation expansion with inter-direction expansion. In inter-direction expansion, without feature pooling, quadratic terms generated from GDHs are directly used as quadratic features. The number of quadratic features is thus $N_z^2 \cdot N_d \cdot (N_d+1)/2$. Table 8 shows the test results of inter-direction expansion with different values of N_z. Compared with Table 6, we see that, though more quadratic features are generated in inter-direction expansion, the performance is significantly inferior. It proves the robustness of statistical features.

5.2. Effect of spatial correlation expansion

In this subsection, experiments are conducted to examine the effect of intra-plane expansion, inter-plane expansion and their combination.

5.2.1. Effect of intra-plane expansion

There are three factors affecting spatial correlation expansion – the number of GDH feature planes (or the number of standard gradient directions) N_d, the size of GDH feature planes (or the GDH map which can be partitioned into multiple feature planes) N_z, using multi-resolution representation or not. As it is in statistical correlation expansion, besides quadratic features, 768-*dimensional* GDH features are included in the feature vector.

In the first experiment, we study the influence of N_d while other factors are fixed as follows: QuadTree multi-resolution representation is used, and N_z is fixed at 16. Test results in Table 9 show that the value of N_d has little influence on classification accuracy. To be coherent with statistical correlation expansion, in the following experiments, we fix N_d at 12.

In the second experiment, different values of N_z are tested while the other factors are fixed as above. From Table 10, we see that, $N_z=16$ produces much better results. This demonstrates that local spatial correlation obtained from high resolution representation can improve test accuracy.

In the third experiment, we compare single-resolution representation with multi-resolution representation, and two down-sampling

Table 9
The influence on test accuracy caused by the number of directional feature planes N_d in intra-plane expansion. N_z is fixed at 16. QuadTree multi-resolution representation is used.

N_d(dim_q)	8 (9504)	12 (14,256)	16 (19,008)
NPC	89.80	89.86	89.83
MQDF	91.90	91.89	91.87
DLQDF	91.95	91.95	91.93

Table 10
The influence on test accuracy caused by the size of directional feature planes N_z in intra-plane expansion. N_d is fixed at 12. QuadTree multi-resolution representation is used.

N_z (dim_q)	8 (3096)	16 (14,256)
NPC	88.77	89.86
MQDF	91.31	91.89
DLQDF	91.41	91.95

Table 11
Comparison between single-resolution and multi-resolution representation in intra-plane expansion. N_d and N_z are fixed at 12 and 16, respectively.

Feature map representation	Single-reso	QuadTree	Gaussian		
dim_q	11,160	14,256	14,256		
σ_0			1/3	2/3	1
NPC	89.27	89.86	90.08	90.10	89.99
MQDF	91.62	91.89	91.97	92.00	92.00
DLQDF	91.73	91.95	92.03	92.08	92.06

Table 12
Test accuracies of classifiers using inter-plane expansion with different direction gaps G_d. N_z and N_d are fixed at 8 and 12, respectively. Single-resolution representation is used.

G_d	0	1	2	3	4	5	6
dim_q	2520	5040	5040	5040	5040	5040	2520
NPC	88.32	89.33	89.54	89.29	89.47	89.40	88.58
MQDF	91.19	91.49	91.50	91.34	91.43	91.47	91.22
DLQDF	91.28	91.58	91.59	91.43	91.52	91.57	91.31

Table 13
Test accuracies of classifiers with the combination of intra-plane expansion and inter-plane expansion. N_z and N_d are fixed at 8 and 12, respectively. Single-resolution representation is used.

G_d	0, 1	0, 2	0, 3	0, 4	0, 5	0, 6	0–2	0–3	0–4	0–5	0–6
dim_q	7560	7560	7560	7560	7560	5040	12,600	17,640	22,680	27,720	30,240
NPC	89.55	89.92	89.76	89.89	89.79	89.10	90.18	90.45	90.63	90.79	90.86
MQDF	91.57	91.67	91.56	91.62	91.60	91.37	91.71	91.77	91.83	91.95	91.98
DLQDF	91.65	91.77	91.66	91.69	91.69	91.47	91.82	91.86	91.91	92.00	92.06

methods – QuadTree and Gaussian blurring are also compared. For Gaussian blurring, the influence of the deviation parameter σ_0 is tested. Table 11 shows the results. We see that, multi-resolution representation outperforms single-resolution representation especially for NPC classifier. Between two down-sampling methods, Gaussian blurring is better, but the advantage is not obvious especially for MQDF and DLQDF classifiers. The performance difference between different values of σ_0 is negligible. In subsequent experiments, by default, QuadTree down-sampling is chosen for its low computation burden.

5.2.2. Effect of inter-plane expansion

In this experiment, we inspect the effect of inter-plane expansion. Each time, only plane pairs with a given direction gap G_d are chosen for expansion. To constrain the computation complexity, we change N_z to 8, and use single-resolution representation. Table 12 gives the test accuracies on CASIA-HWDB1.1 dataset, where $G_d=0$ stands for intra-plane expansion. The results show that when G_d is between 1 and 5, inter-plane expansion produces twice the number of quadratic features as intra-plane expansion, and the test accuracies are also improved. The difference between different values of G_d is not obvious. When G_d is 6, the number of quadratic features is reduced by a half, because there are only 6 plane pairs with $G_d=6$. The accuracy with $G_d=6$ is slightly better than intra-plane expansion.

5.2.3. Combining intra-plane expansion and inter-plane expansion

Combining intra-plane expansion and inter-plane expansion can exploit more spatial correlation. In this experiment, at first, only inter-plane expansion with a given value of G_d is combined with intra-plane expansion; after that, on the basis of intra-plane expansion, inter-plane expansions with bigger G_d values are gradually added. Table 13 shows the results. Compared to Table 12, it shows that their combination improves test accuracies, and as more inter-plane expansions are used, the performance is continuously improved.

To get better results, we change N_z to 16, and use multi-resolution representation. For the highest resolution, only intra-plane expansion is used, while for other lower resolutions, both expansions are used. The total number of quadratic features is 48,312. Results in Table 14 show that the performance is further improved.

5.3. Combining statistical correlation expansion and spatial correlation expansion

In this subsection, we combine statistical correlation expansion with spatial correlation expansion. Then besides 768-dimensional GDH features, quadratic features generated by statistical correlation expansion and spatial correlation expansion are included in the feature vector. In statistical correlation expansion, with multi-level and overlapped region partition, 4602 quadratic features are generated. In spatial correlation expansion, two feature expansion settings are tested – only intra-plane expansion and the combination of intra-plane expansion and inter-plane expansion. For multi-resolution representation, QuadTree and Gaussian blurring (the deviation parameter σ_0 is set at 2/3) are also compared. Table 15 shows dimensionalities of feature vectors for these two settings of spatial correlation expansion. We tested three subspace dimensionalities, and the test results on CASIA-HWDB1.1 dataset are shown in Table 16. Compared with Tables 5 and 9, we see that combining these two quadratic feature expansion methods outperforms any of them individually. Comparing two multi-resolution representation methods, we see that when using only intra-plane spatial correlation expansion, Gaussian blurring multi-resolution representation produces slightly better results, while with both intra-plane and inter-plane expansion, its performance is inferior. So QuadTree multi-resolution representation is adopted in subsequent experiments. Comparing Table 16 with Table 4, when subspace dimensionality is 250, with quadratic feature expansion, test accuracies of NPC, MQDF and DLQDF are improved by 4.94%, 1.83%, 1.82%, respectively.

Table 14
The effect of combining intra-plane expansion with inter-plane expansion under multi-resolution representation. N_z and N_d are set at 16 and 12, respectively.

NPC	MQDF	DLQDF
91.45	92.56	92.64

Table 15
Number of features for two settings of spatial correlation expansion.

Setting	GDH_dim	statistical_dim	spatial_dim	total_dim
Intra	768	4602	14,256	19,626
Intra+inter	768	4602	48,312	53,682

As shown in Table 16, for MQDF and DLQDF classifiers, the difference of accuracies between these two feature settings is not very significant. This implies that much redundancy exists among generated quadratic features in the second setting. Therefore, feature selection is expected to reduce the complexity of DQFL.

To compare with deep CNNs, we use all data of CASIA-HWDB1.1 for training and ICDAR-2013 for test. The experimental settings are the same as the previous experiment, and QuadTree multi-resolution representation is adopted. The results are shown in Table 17. We can see that the best accuracy of DLQDF is 94.44%, which is comparable to the deep CNN's 94.47% announced in [7], but still inferior to relaxation CNN's 95.04% [8]. In [8], weights sharing restriction of convolutional layers is relaxed to enhance the expressive capability, and at the same time, the number of parameters is largely increased. Unlike these deep CNNs, no extra synthesized samples are used for training in this experiment.

5.4. Effect of training set expansion

To further improve the performance, we expand our training set with distortion functions introduced in Section 4. Two distortion models – the single model and the combined model are used. We use the single model to generate samples for DFE+LVQ learning, and the number of generated samples is five times the size of original training set. For the training of MQDF and DLQDF classifiers, the combined model is adopted, the synthesized data size is ten

Table 16
Test accuracies on CASIA-HWDB1.1 dataset when combining statistical correlation expansion and spatial correlation expansion. In spatial correlation expansion, two feature expansion settings are used – only intra-plane expansion, and both intra-plane and inter-plane expansion. The column named *dim* is the dimensionality of the feature vector (including GDH features and quadratic features).

Multi-resolution representation method	Feature	Dim	Classifier	sub_dim		
				160	200	250
QuadTree multi-resolution representation	GDH+ statistical+intra-plane	19,626	NPC	91.18	91.31	91.38
			MQDF	92.71	92.77	92.78
			DLQDF	92.78	92.84	92.88
	GDH+ statistical+ spatial	53,682	NPC	91.74	92.09	92.23
			MQDF	92.85	93.05	93.10
			DLQDF	92.92	93.12	93.19
Gaussian blurring multi-resolution representation	GDH+ statistical+ intra-plane	19,626	NPC	91.24	91.44	91.54
			MQDF	92.75	92.81	92.83
			DLQDF	92.82	92.90	92.89
	GDH+ statistical+ spatial	53,682	NPC	91.58	91.79	91.88
			MQDF	92.85	92.94	92.99
			DLQDF	92.92	93.03	93.07

Table 17
Test accuracies on ICDAR-2013 dataset when combining statistical correlation expansion and spatial correlation expansion.

Feature	Dim	Classifier	sub_dim		
			160	200	250
GDH+ statistical+ intra-plane	19,626	NPC	92.27	92.40	92.48
		MQDF	93.90	93.99	94.00
		DLQDF	93.99	94.09	94.11
GDH+statistical+ spatial	53,682	NPC	93.10	93.26	93.38
		MQDF	94.24	94.32	94.39
		DLQDF	94.32	94.37	94.44

Table 18
Test accuracies on CASIA-HWDB1.1 dataset with quadratic feature expansion and training set expansion.

Feature	Dim	Classifier	sub_dim		
			160	200	250
GDH+ statistical+ intra-plane	19,626	NPC	92.10	92.27	92.38
		MQDF	93.35	93.40	93.46
		DLQDF	93.55	93.66	93.72
GDH+ statistical+ spatial	53,682	NPC	92.93	93.12	93.23
		MQDF	93.66	93.73	93.80
		DLQDF	93.78	93.90	93.99

times the size of the original training set. For feature expansion, as it is in the previous experiment, two settings of spatial correlation expansion are tested. Results on CASIA-HWDB1.1 are shown in Table 18. Table 19 shows the results when training with CASIA-HWDB1.1, and test with ICDAR-2013. Comparing Table 18 with Table 16, we can see that with training set expansion, test accuracies of these three classifiers are improved by 1.00%, 0.70% and 0.80%. Comparing Table 18 with Table 4, by combining DQFL and training set expansion, the test accuracies of NPC, MQDF and DLQDF are improved by 5.94%, 2.53% and 2.62%, respectively; and meanwhile, the accuracy gap between NPC and DLQDF is reduced from 4.08% to 0.76%, which demonstrates that quadratic features are more beneficial for linear classifiers than for quadratic classifiers. As shown in Table 19, on the ICDAR-2013 test set, the best accuracy of DLQDF reaches 94.92%, which is close to relaxation CNN's 95.04% [8], but significantly inferior to CNN committee's

Table 19
Test accuracies on ICDAR-2013 dataset with quadratic feature expansion and training set expansion.

Feature	Dim	Classifier	sub_dim		
			160	200	250
GDH+ statistical+ intra-plane	19,626	NPC	93.10	93.22	93.36
		MQDF	94.23	94.33	94.40
		DLQDF	94.42	94.53	94.60
GDH+ statistical+ spatial	53,682	NPC	93.86	94.01	94.12
		MQDF	94.58	94.66	94.78
		DLQDF	94.72	94.81	94.92

96.06% [8]. When considering test speed, our system is much faster than CNN as analyzed in next subsection. Besides, the compact classifier NPC also produces a satisfactory test accuracy of 94.12%. NPC classifier owns the advantage of small dictionary size and low computation burden, which makes it suited for platforms with limited storage and computation capability, such as mobile devices.

In all experiments above, for NPC classifier, each class only owns one prototype. The following experiment inspects that if using multiple prototypes leads to better performance. In this experiment, three feature settings are used – besides GDH features, statistical correlation features, intra-plane spatial correlation features and inter-plane spatial correlation features are gradually included. The subspace dimensionality is fixed at 160. Table 20 shows the test accuracies on CASIA-HWDB1.1 dataset without and with training set expansion. We can see that, when the dimensionality is low, multi-prototype NPC performs slightly better, while when more quadratic features are included, it produces no better or even inferior results compared to one-prototype NPC. The reason is possibly that with high-dimensional feature space, multi-prototype NPC is more likely to bring about overfitting.

5.5. Time complexity analysis

As test speed matters for practical applications, we give the time complexity of our system for test. The test process consists of steps of GDH feature extraction, quadratic feature expansion, dimensionality reduction and classification. For classification, we choose DLQDF due to its superior performance. Usually a cascaded classification scheme is adopted for acceleration of DLQDF classification, in which a coarse classification with the nearest mean classifier is used to select N candidate classes, and quadratic discriminant functions are computed for each of these candidate classes [43] to produce the final decision. Here N is set at 100. We observed that this cascaded classification

Table 20
Test accuracies on CASIA-HWDB1.1 dataset using NPC with different prototype numbers for each class.

Without/with training set expansion	Feature	Dim	#prototype				
			1	2	3	4	5
Without training set expansion	GDH+statistical	5370	90.47	90.58	90.61	90.53	90.53
	GDH+statistical+intra-plane	19,626	91.18	91.17	91.24	91.21	91.17
	GDH+statistical+spatial	53,682	91.74	91.70	91.62	91.55	91.47
With training set expansion	GDH+statistical	5370	91.09	91.20	91.21	91.23	91.25
	GDH+statistical+intra-plane	19,626	92.10	92.10	92.17	92.18	92.14
	GDH+statistical+spatial	53,682	92.93	92.83	92.82	92.78	92.79

Table 21
Time consuming (in ms) for each process of test.

GDH_extr		quad_expand		dim_red	Classify	Total
$N_z=16$	$N_z=8$	stati	spatial			
1.36	1.12	0.35	0.32	13.49	3.54	20.18

scheme barely affects the classification accuracy. Table 21 shows the time consuming (in milliseconds) for each step of test. During feature extraction, two GDH maps with sizes 16 and 8 are extracted. In quadratic feature expansion, statistical correlation expansion generates 4602 quadratic features, while spatial expansion generates 48,312 features. The subspace dimensionality is 250, and the number of principal eigenvectors is 80. We can see that most of the time is spent on dimensionality reduction and DLQDF classification. If lower test time is required, the dimensionality reduction process can be accelerated by learning a sparse projection matrix to approximate the original dense one [19].

To analyze the time complexity of dimensionality reduction and DLQDF classification. We roughly count the number of multiplications occurred. For dimensionality reduction, it is

$$ftr\_dim \times sub\_dim, \tag{12}$$

where *ftr_dim* and *sub_dim* are the dimensionalities of feature vectors and subspace, which are 53,682 and 250, respectively in current setting. For DLQDF classification, it is

$$\#class \times sub\_dim + \#candidates \times sub\_dim \times \#principle\_eigenvectors \tag{13}$$

In the above setting, the numbers of multiplications are 13,420, 500 and 2,938,750, respectively. So together, there are approximately 1.64×10^7 multiplications.

To compare it with that of deep CNNs, we analyze a deep CNN structure proposed in [7]. This network achieved a recognition accuracy of 94.47% on ICDAR-2013 dataset. The network structure is represented by "48x48-150C3-MP2-250C2-MP2-350C2-MP2-450C2-MP2-1000N-3755N". Each substring separated by "-" represents a network layer. For example, "150C3" represents a convolutional layer with 150 output feature maps and kernel size 3×3, "MP2" represents a max-pooling layer with step size 2, and "1000N" represents a fully connected layer with 1000 output neurons. In the forward propagation of a deep CNN, most of the computation lies in convolutional layers and fully connected layers. So we only count the number of multiplications occurred in these two types of layers. For one convolutional layer, the number is

$$\#output\_plane \times \#input\_plane \times output\_plane\_size^2 \times kernel\_size^2, \tag{14}$$

while for a fully connected layer, it is

$$\#output\_neuron \times \#input\_neuron. \tag{15}$$

For this structure, the total number of multiplications is about 1.25×10^8. From Ciresan's paper [6], the time spent on classifying one sample with CPU is about 103.5 ms, which is 5 times slower compared with our system. When GPU is used in test, the time cost is reduced to 3.03 ms for a deep CNN [7], while in our system, when samples are classified in batches and one Tesla C2075 GPU is used, the time consuming of dimensionality reduction and DLQDF classification is reduced from 17.03 ms to 0.57 ms.

6. Conclusion

In this paper, an effective feature learning method named DQFL is proposed. It utilizes quadratic correlation between original features of gradient direction histograms for feature expansion. The quadratic feature expansion combined with discriminative dimensionality reduction makes the learned features nonlinear and discriminative. In feature expansion, statistical correlation and spatial relation of features are utilized. With this method, the classification accuracies of NPC, MQDF and DLQDF classifiers are improved significantly. After training set expansion, the performance is further improved. In experiments on the CASIA-HWDB1.1 database and the ICDAR 2013 Chinese handwriting recognition competition dataset, the proposed method yielded accuracies competitive with deep CNNs at much lower testing complexity. In the future, we aim to further reduce the complexity of DQFL by feature selection from expanded quadratic terms.

Conflict of interest

None declared.

Acknowledgment

This work has been supported by the National Basic Research Program of China (973 Program) Grant 2012CB316302, the National Natural Science Foundation of China (NSFC) Grant nos. 61175021 and 61403380.

References

[1] C.-L. Liu, M. koga, H. Fujisawa, Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading, IEEE Trans. Pattern Anal. Mach. Intell. 24 (11) (2002) 1425–1437.
[2] Q.-F. Wang, F. Yin, C.-L. Liu, Handwritten Chinese text recognition by integrating multiple contexts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (8) (2012) 1469–1481.
[3] R.-W. Dai, C.-L. Liu, B.-H. Xiao, Chinese character recognition: history, status and prospects, Front. Comput. Sci. China 1 (2) (2007) 126–136.

[4] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Online and offline handwritten Chinese character recognition: benchmarking on new databases, Pattern Recognit. 46 (1) (2013) 155–162.
[5] F. Yin, Q.-F. Wang, X.-Y. Zhang, C.-L. Liu, ICDAR 2013 Chinese handwriting recognition competition, in: Proceedings of 12th ICDAR, 2013, pp. 1464–1470.
[6] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: IEEE Conference on CVPR, 2012, pp. 3643–3649.
[7] D. Ciresan, J. Schmidhuber, Multi-column deep neural networks for handwritten Chinese character classification, Technical Report, no. IDSIA-05-13, August 2013.
[8] C. Wu, W. Fan, Y. He, J. Sun, S. Naoi, Handwritten character recognition by alternately trained relaxation convolutional neural network, in: Proceedings of 14th ICFHR, 2014, pp. 291–296.
[9] J. Tsukumo, H. Tanaka, Classification of handprinted Chinese characters using non-linear normalization and correlation methods, in: Proceedings of 9th ICPR, 1988, pp. 168–171.
[10] H. Yamada, K. Yamamoto, T. Saito, A nonlinear normalization method for handprinted Kanji character recognition-line density equalization, Pattern Recognit. 24 (9) (1990) 1023–1029.
[11] T. Horiuchi, R. Haruki, H. Yamada, K. Yamamoto, Two-dimensional extension of nonlinear normalization method using line density for character recognition, in: Proceedings of 4th ICDAR, 1997, pp. 589–600.
[12] C.-L. Liu, K. Marukawa, Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition, Pattern Recognit. 38 (12) (2005) 2242–2255.
[13] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, Handwritten digit recognition: benchmarking of state-of-the-art techniques, Pattern Recognit. 36 (10) (2003) 2271–2285.
[14] H. Fujisawa, C.-L. Liu, Directional pattern matching for character recognition revisited, in: Proceedings of 7th ICDAR, 2003, pp. 794–798.
[15] 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 (January (1)) (1987) 149–153.
[16] C.-L. Liu, H. Sako, H. Fujisawa, Discriminative learning quadratic discriminant function for handwriting recognition, IEEE Trans. Neural Netw. 15 (2) (2004) 430–444.
[17] J. Schurmann, Pattern Classification: A Unified View of Statistical and Neural Approaches, Wiley Interscience, New York, 1996.
[18] C.-L. Liu, M. Nakagawa, Evaluation of prototype learning algorithms for nearest-neighbor classifier in application to handwritten character recognition, Pattern Recognit. 34 (3) (2001) 601–615.
[19] D. Chen, X.-D. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: Proceedings of CVPR, 2013, pp. 3025–3032.
[20] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, CASIA online and offline Chinese handwriting databases, in: Proceedings of 11th ICDAR, 2011, pp. 37–41.
[21] Q. Huo, Y. Ge, Z.-D. Feng, High performance Chinese OCR based on Gabor features, discriminative feature extraction and model training, in: Proceedings of ICASSP, 2001, pp. 1517–1520.
[22] X. Wang, X. Ding, C. Liu, Optimized Gabor filter based feature extraction for character recognition, in: Proceedings of 16th ICPR, 2002, pp. 223–226.
[23] C.-L. Liu, M. Koga, H. Fujisawa, Gabor feature extraction for character recognition: comparison with gradient feature, in: Proceedings of 8th ICDAR, 2005, pp. 121–125.
[24] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[25] D. Ciresan, U. Meier, L. Gambardella, J. Schmidhuber, Convolutional neural network committees for handwritten character classification, in: Proceedings of 11th ICDAR, 2011, pp. 1135–1139.
[26] A. Biem, S. Katagiri, B.-H. Juang, Pattern recognition using discriminative feature extraction, IEEE Trans. Signal Process. 45 (2) (1997) 500–504.
[27] C.-L. Liu, R. Mine, M. Koga, Building compact classifier for large character set recognition using discriminative feature extraction, in: Proceedings of 8th ICDAR, 2005, pp. 846–850.
[28] T. Kohonen, The self-organizing map, Proc. IEEE 78 (9) (1990) 1464–1480.
[29] X.-Y. Zhang, C.-L., Liu, Style transfer matrix learning for writer adaptation, in: Proceedings of CVPR, 2011, pp. 393–400.
[30] B.-H. Juang, W. Chou, C.-H. Lee, Minimum classification error rate methods for speech recognition, IEEE Trans. Speech Audio Process. 5 (3) (1997) 257–265.
[31] X.-B. Jin, C.-L. Liu, X. Hou, Regularized margin-based conditional log-likelihood loss for prototype learning, Pattern Recognit. 43 (7) (2010) 2428–2438.
[32] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd edition, Academic Press, San Diego, 1990.
[33] C.-L. Liu, Normalization-cooperated gradient feature extraction for handwritten character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1465–1469.
[34] O. Tuzel, F. Porikli, P. Meer, Region covariance: a fast descriptor for detection and classification, in: Proceedings of 9th ECCV, 2005, pp. 589–600.
[35] C.-L. Liu, J. Kim, R. Dai, Multiresolution locally expanded HONN for handwritten numeral recognition, Pattern Recognit. Lett. 18 (10) (1997) 1019–1025.
[36] P. Simard, D. Steinkraus, J. Platt, Best practices for convolutional neural networks applied to visual document analysis, in: Proceedings of 7th ICDAR, 2007, pp. 958–963.
[37] K. Leung, C. Leung, Recognition of handwritten Chinese characters by combining regularization, Fisher's discriminant and distorted sample generation, in: Proceedings of 10th ICDAR, 2009, pp. 1026–1030.
[38] T. Ha, H. Bunke, Off-line, handwritten numeral recognition by perturbation method, IEEE Trans. Pattern Anal. Mach. Intell. 19 (5) (1997) 535–539.
[39] NVIDIA, NVIDIA CUDA C programming guide 4.1, 2011.
[40] NVIDIA, NVIDIA CUDA basic linear algebra subroutines (cuBLAS) library guide 4.1, 2011.
[41] Matrix Algebra on GPU and Multicore Architectures, http://icl.cs.utk.edu/magma/.
[42] M.-K. Zhou, F. Yin, C.-L. Liu, GPU-based fast training of discriminative learning quadratic discriminant function for handwritten Chinese character recognition, in: Proceedings of 12th ICDAR, 2013, pp. 842–846.
[43] C.-L. Liu, High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction, in: Proceedings of 18th ICPR, 2006, pp. 942–945.

Ming-Ke Zhou received the B.S. degree in computer science and technology from Central South University, Changsha, China, in 2009. He is currently working toward the PhD degree in pattern recognition and intelligent systems at the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include handwritten character recognition, document image analysis, discriminative feature learning, and deep learning.

Xu-Yao Zhang received the B.S. degree in computational mathematics from Wuhan University, Wuhan, China, in 2008, the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013. He is now an assistant professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include pattern recognition, machine learning, and especially large category classification, dimensionality reduction and classifier adaptation.

Fei Yin received the B.S. degree in computer science from Xidian University of Posts and Telecommunications, Xi'an, China, the ME degree in pattern recognition and intelligent systems from Huazhong University of Science and Technology, Wuhan, China, the PhD degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1999, 2002, and 2010, respectively. He is an associate professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include document image analysis, handwritten character recognition, and image processing.

Cheng-Lin Liu is a Professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation of Chinese Academy of Sciences, Beijing, China, and is now the director of the laboratory. He received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, the M.E. degree in electronic engineering from Beijing Polytechnic University, Beijing, China, the Ph.D. degree in pattern recognition and intelligent control from the Chinese Academy of Sciences, Beijing, China, in 1989, 1992 and 1995, respectively. He was a postdoctoral fellow at Korea Advanced Institute of Science and Technology (KAIST) and later at Tokyo University of Agriculture and Technology from March 1996 to March 1999. From 1999 to 2004, he was a research staff member and later a senior researcher at the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. His research interests include pattern recognition, image processing, neural networks, machine learning, and especially the applications to character recognition and document analysis. He has published over 200 technical papers at prestigious international journals and conferences. He is a fellow of the IAPR and the IEEE.