2014 22nd International Conference on Pattern Recognition

Improving Handwritten Chinese Character Recognition with Discriminative Quadratic Feature Extraction

Ming-Ke Zhou, Xu-Yao Zhang, Fei Yin, Cheng-Lin Liu
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
Email: {mkzhou, xyz, fyin, liucl}@nlpr.ia.ac.cn

***Abstract*—Discriminative feature extraction (DFE) is an effective linear dimensionality reduction method for pattern recognition. It improves the recognition performance via optimizing subspace projection axes and classifier parameters simultaneously. In this paper, we propose a nonlinear extension of DFE, called discriminative quadratic feature extraction (DQFE), for which feature vectors are firstly mapped to a high-dimensional nonlinear space and then projected to a low-dimensional subspace learned by DFE. The nonlinear mapping is obtained by adding quadratic (correlation or covariance) features computed directly on the original gradient feature maps with different region partition. In this way, both the structural information of the image and the correlation information of features are used to generate a nonlinear high-dimensional feature mapping (thousands of dimensions). Experimental results demonstrated that DQFE can improve the accuracy for different classifiers in handwritten Chinese character recognition.**

I. Introduction

Handwritten Chinese character recognition (HCCR) has been intensively studied. Traditional methods involve a flow of shape normalization, feature extraction, dimensionality reduction and classifier building. Shape normalization is to normalize the character image into a standard size and reduce the within-class shape variance. Gradient direction features have been proven effective for HCCR. For dimensionality reduction, Fisher linear discriminant analysis (FDA) [1] is a popular method which learns a linear subspace to maximize between-class variance and minimize within-class variance, while discriminative feature extraction (DFE) [2] [3] optimizes the subspace axes and classifier parameters simultaneously by minimizing the empirical loss in supervised learning. DFE is superior to FDA because it enhances the separability between confusion classes in the learned subspace while FDA tends to mix them [4]. Concerning classifier building, the modified quadratic discriminant function (MQDF) [5] and its discriminative learning version, namely discriminative learning quadratic discriminant function (DLQDF) [6], once yielded the state-of-the-art performance.

Deep neural network (DNN) is getting popular in recent years because of its record-breaking performances on many vision applications. Unlike traditional pattern classification methods, DNN operates on raw image pixels directly, and learns feature extraction, dimensionality reduction and classifier parameters at the same time in a supervised manner [7]. In spite of its superior performance, its time complexity is very high in both training and testing process.

We notice that the performance of nearest prototype classifier (NPC) [8] is far behind that of MQDF and DLQDF, and we conjecture that the reason lies in the complexity of classifiers: MQDF and DLQDF are quadratic models while NPC is linear (the class separation boundaries are hyperplanes). Therefore, nonlinear information is important for improving NPC. On the other hand, although DFE is effective for both NPC and MQDF as shown in [9], the subspace learned by DFE is a linear one, and hence, the performance is limited when the data are not linearly separable. To improve the performance of NPC and DFE, in this paper, we introduce nonlinear information into them for better feature extraction and classifier training. Specifically, a nonlinear feature extraction method is proposed through a nonlinear dimensionality promotion followed by a linear dimensionality reduction. In the dimensionality promotion procedure, quadratic information of original features is integrated to map the original feature vector to a high-dimensional nonlinear space. In the dimensionality reduction procedure, DFE is used to map it back to a low-dimensional space. Because of the use of quadratic information during DFE training, we call this model discriminative quadratic feature extraction (DQFE). DQFE is effective because of the integrated quadratic information as well as supervised class information. In the experiments of HCCR on a public dataset, compared with DFE, DQFE improves the test accuracy of NPC by about 2.5%. For MQDF and DLQDF classifiers, the improvements are also significant, both by about 1%. To further improve the test accuracy, we adopt the method of sample distortion for training set expansion. With expanded training set, the test accuracies of NPC, MQDF and DLQDF are further improved by 1.23%, 0.89% and 0.92%, respectively.

The rest of this paper is organized as follows. Section 2 reviews the DFE method; Section 3 details the proposed DQFE method; Section 4 briefly introduces the sample distortion method for training set expansion; Section 5 presents the experimental results and Section 6 concludes this paper.

II. Discriminative Feature Extraction

Discriminative feature extraction (DFE) [2] [3] is a linear dimensionality reduction method. It optimizes the subspace projection axes to minimize the classification error on the training set. As classification error is measured by a classifier in the reduced subspace, the training process of DFE is usually combined with that of a classifier. For example, the DFE combined with NPC is shown to be efficient and effective for HCCR [3] [9]. In NPC, each class is represented with one or several prototypes, and classification is done through nearest prototype search. Supervised prototype learning by Learning vector quantization (LVQ) [10] is effective to improve the performance of NPC. Using DFE for subspace learning and

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.51

IEEE computer society

LVQ for NPC learning, we denote this method as DFE+LVQ classifier.

The training process of DFE+LVQ is based on the minimum classification error (MCE) [11] criterion. It can also be trained under the conditional log-likelihood loss [8] or other similar criteria. For defining the loss function, we first define the misclassification measure. For a D-dimensional training sample x, the misclassification measure is:

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

where m_c and m_r are the prototypes of the genuine class and the closest rival class of x in the reduced subspace with dimensionality d. ϕ represents the dimensionality reduction matrix, and $d_E(\phi^T x, m_c)$ is the square Euclidean distance from $\phi^T x$ to the genuine prototype in the d-dimensional reduced subspace. We can see that $h(x) > 0$ signifies misclassification. Therefore, $-h(x)$ can be viewed as a discriminant function for a binary classification between the genuine class and the closest rival class, then the posterior probability of x belonging to genuine class can be approximated using a sigmoid function of $-h(x)$:

$$P(c|x) = \sigma(\xi[-h(x)]) = \frac{1}{1 + e^{\xi h(x)}}, \tag{2}$$

and the conditional log-likelihood loss is defined as

$$l(x) = -logP(c|x). \tag{3}$$

To constrain the excessive deviation of parameters from the maximum likelihood estimation, a regularization term is usually added, and the final loss function is

$$l'(x) = l(x) + \alpha d_E(\phi^T x, m_c). \tag{4}$$

Therefore, the empirical loss on the training set is

$$L = \frac{1}{N}\sum_{n=1}^{N}[l(x^n) + \alpha d_E(\phi^T x^n, m_c)]. \tag{5}$$

For training the NPC (LVQ alone) or DFE+LVQ, the empirical loss is minimized iteratively on a training sample set by stochastic or mini-batch gradient. FDA and sample means are used for initialization of subspace axes and prototypes, respectively.

III. Discriminative Quadratic Feature Extraction

In order to integrate quadratic information into DFE, we use a two-step scheme — quadratic dimensionality promotion followed by DFE linear dimensionality reduction. The block diagram of the proposed DQFE method is shown in Fig. 1. The input of DQFE is the feature vector extracted for representing a character image. At the first step, quadratic features are generated from the original feature vector (i.e. the histogram of gradient direction features); then, a new feature vector concatenating original features and quadratic features is fed into DFE for linear dimensionality reduction. Although DFE is a linear model, due to the quadratic features used, the whole process of DQFE is nonlinear (quadratic), which can extract much more discriminative features for the subsequent classification tasks. In the following subsections, we will first introduce the feature representation method, and then details the process of quadratic features generation.

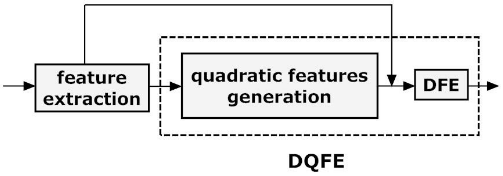

Fig. 1. The block diagram of DQFE.

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Fig. 2. The Sobel masks.

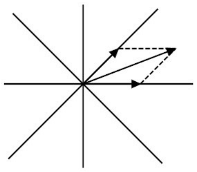

Fig. 3. Decomposition of gradient into components in two neighbouring standard directions.

A. Feature Representation

For feature representation, we use gradient direction histogram (GDH) features [12]. Before feature extraction, character images are normalized to standard size. We use a pseudo two-dimensional normalization method called line density projection interpolation (LDPI) [13], which is a simplified version of pseudo two-dimensional nonlinear normalization [14] in the spirit of stroke density equalization. After normalization, local GDHs are extracted on the normalized character images by zoning. The feature extraction procedure comprises three steps: First, gradients are computed for every pixel of the normalized image using the Sobel masks as illustrated in Fig. 2; Second, L standard directions are specified, and all the gradients are decomposed into components in the two neighboring standard directions (Fig. 3); Third, the normalized image is partitioned into $zn \times zn$ uniform zones, and in each zone, the decomposed gradients are summed by Gaussian blurring. Thus, a L-dimensional GDH is obtained for representing each zone, and the total dimensionality of the extracted feature vector is $zn \times zn \times L$.

In this paper, we use normalized-cooperated gradient feature extraction (NCGF) [12] which combines the procedure of normalization and GDH feature extraction. By this method, the normalized image is not generated, instead, gradients are computed on the original image, and mapped to direction planes of normalized size. The GDHs are computed from the direction planes by zoning and Gaussian blurring.

B. Quadratic Features Generation

As mentioned above, in the GDH feature representation, each image is partitioned into $zn \times zn$ zones, and each zone is represented by a L-dimensional GDH. If we consider each GDH as a feature point extracted from its corresponding image zone, then all these extracted feature points constitute a GDH feature map containing $zn \times zn$ feature points. The quadratic features we obtained are generated from this feature map. Specifically, they are quadratic terms of GDH features. For a feature point $x^k = (x_1^k, x_2^k, ..., x_L^k)$ in the GDH feature map, quadratic terms like $\{x_i^k \cdot x_j^k | 1 \leq i \leq j \leq L\}$ can be used. But there will be too many quadratic features if all of them are used, and they are not stable. Therefore, we turn to use their averages among a region of the feature map, and we call them the correlation features and denote them as $Corr_R$, where R is the region in which they are computed. The process of GDH feature map generation and quadratic features generation is shown in Fig. 4: the original image is first normalized and partitioned into zones, and in each zone, a GDH feature point is extracted. After that, the obtained feature map is partitioned into regions, and in every region, the quadratic features are generated.

Besides the correlation feature, we also use another type of quadratic feature named region covariance [15], and denote it as Cov_R. Region covariance is a powerful region descriptor and has bee applied successfully in object detection and recognition. It is defined as the covariance matrix of feature points in a region. The formulas of correlation features and covariance features are as follows:

$$Corr_R = \frac{1}{N_R} \sum_{i=1}^{N_R} x^i (x^i)^T, \tag{6}$$

$$Cov_R = \frac{1}{N_R} \sum_{i=1}^{N_R} (x^i - m_R)(x^i - m_R)^T, \tag{7}$$

where R is the region upon which we compute the quadratic features, N_R is the number of feature points in region R, x_i is the descriptor (L-dimensional vector of GDH) of a feature point in region R, m_R is the mean vector of all feature points in region R. The results of these two formulas are two symmetric matrices, so the number of unique features in a region is $L \times (L+1)/2$. In our implementation, in order to let the quadratic features be at the same scale as the GDH features, we use their signed square roots. The difference between (6) and (7) is that (7) is a mean-shifted quadratic term while (6) is not. We will compare their performance in latter experiments.

Now we have described how to generate quadratic features in a region of the GDH feature map, the next issue is how to partition the feature map in order to get regions. In order to generate quadratic features using regions of different scales, the feature map is partitioned in multiple levels from level 1 to level 4. On level l, the feature map is equally partitioned into $l \times l$ blocks, and one or several blocks are combined to form a region on which we will compute quadratic features. When the size zn of the feature map is not a multiple of l, some neighboring blocks are overlapped. The parameter zn we used in our experiments is 16. There are four types of regions according to the way of combination, namely small regions, strip regions, center regions and border regions. Small regions consist of only one block of the feature map, hence there are $l \times l$ small regions in level l. Strip regions consist of blocks in a row or a column, so there are $2 \times l$ strip regions in level l. Center regions consist of the blocks in the central area of the feature map, and border regions consist of the blocks at the boundary of the feature map. Here we use center region and border region only in level 4, therefore there is only one center region and one border region. Table I shows the numbers of regions in different partition levels. Fig. 5 illustrates the partitions of the feature map and four types of regions (represented by black blocks) in level 4.

TABLE I. NUMBERS OF FOUR TYPES OF REGIONS IN EACH PARTITION LEVEL.

level	small	strip	center	border	sum
1	1	0	0	0	1
2	4	4	0	0	8
3	9	6	0	0	15
4	16	8	1	1	26
sum	30	18	1	1	50

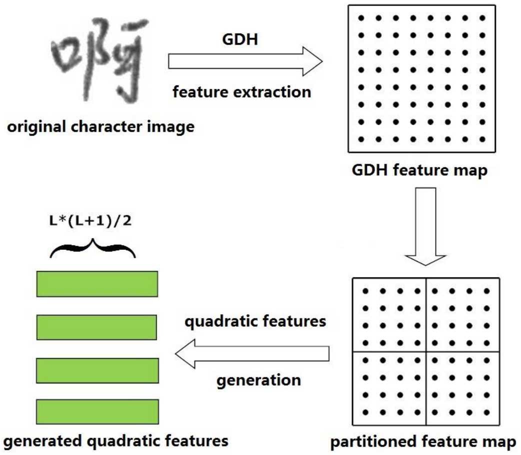

Fig. 4. The process of GDH feature extraction and quadratic features generation. In the feature map, every feature point is represented by a dot which is a L-dimensional GDH vector.

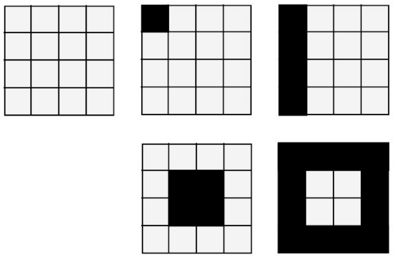

Fig. 5. Illustration of the partition of the feature map in level 4 and four types of regions. From top to bottom, left to right, are the partition of the feature map, the small region, the strip region, the center region and the border region.

After the processes of GDH feature extraction (III-A) and quadratic feature generation (III-B), these two types of features are concatenated to build the final feature vector, and the DFE is further used to reduce the features into a low-dimensional subspace for classification (Fig. 1).

IV. Training Set Expansion

HCCR is difficult because of two challenges, the first one is the large character set, i.e., the number of frequently used characters amounts to several thousands, and there are many similar characters which are hard to be discriminated; second, the shape variability within the same class is huge due to different writing styles and writing instruments. Therefore, to build a robust recognizer, we need a very large training set containing samples of many different styles for every class.

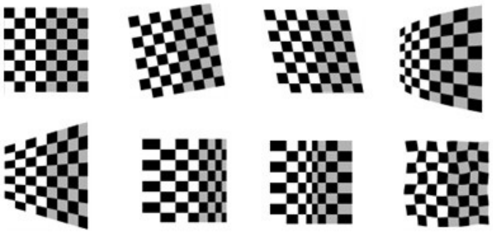

Fig. 6. Effects of different distortion models acted on a checkerboard image.

But in practice, to collect and label a big data set is difficult. In popular HCCR data sets, the number of samples for each class is several hundreds, which is much smaller than the number of classes. In order to alleviate this problem, sample synthesis is used by many researchers as a way of training set expansion. In this paper, we synthesize samples by distorting real ones.

We use three categories of distortion functions, namely geometric transform, local resizing [16] and elastic distortion [17]. For geometric transform, we adopt models proposed in [18]. These 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 the left/right, top/bottom or center/side parts. Readers can refer to [16] 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 total, there are 16 distortion functions in the three categories. Fig. 6 illustrates the effects of different distortion functions on a checkerboard image. In the top row are the original checkerboard image, the results of rotation, horizontal shearing and perspective transform in horizontal direction; in the bottom row are the results of shrink transform in horizontal direction, local resizing function w_1 in horizontal direction, local resizing function w_2 on horizontal direction and elastic distortion.

We adopted two schemes of distortion, the first one is using one randomly selected distortion function for each synthesis, the second one is using a combined distortion of shearing and local resizing in both horizontal and vertical directions [16]. We call these two distortion schemes the single model and the combined model respectively. During each synthesis process, the parameters of all distortion functions are randomly generated. Fig. 7 and Fig. 8 show some synthesized samples generated by the single model and the combined model, respectively. The first images in Fig. 7 and Fig. 8 are original images, and the rest are distorted images.

V. EXPERIMENTAL RESULTS

We used the CASIA-HWDB1.1 dataset [19] collected by the Institute of Automation of Chinese Academy of Sciences (CASIA) for experiments. The CASIA-HWDB1.1 dataset contains off-line handwritten Chinese character samples of 3,755 classes (GB2312-80 level-1 set) written by 300 writers. It is partitioned into a standard training set of 240 writers and a test set of 60 writers. There are totally 897,758 samples in the training set, and 223,991 samples in the test set.

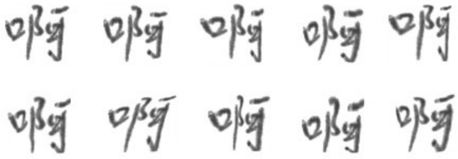

Fig. 7. Samples generated by randomly selected single distortion models.

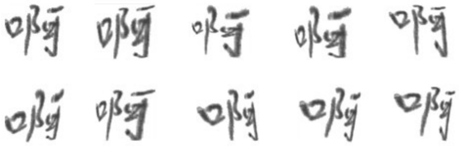

Fig. 8. Samples generated by the combined distortion model of shearing and local resizing.

The training process and testing process are illustrated in Fig. 9. In the training process, samples are firstly passed to NCGF feature extractor to extract GDH features; then quadratic features are generated from the obtained GDH feature maps, and the final feature vectors are composed of GDH features and quadratic features; After that, DFE+LVQ classifier is trained, and the transform basis and NPC are obtained; Finally, the transform basis is used for dimensionality reduction, and MQDF as well as DLQDF are trained in the reduced space. During testing, the GDH feature extraction and quadratic features generation are the same as in training; after that, samples are dimensionality-reduced using stored transform basis, and classifier parameters are loaded for classification.

During GDH feature extraction, the number of standard direction L is set to be 12. Two GDH feature maps of different sizes are generated for each sample. Their sizes are 8×8 and 16×16 respectively. The first feature map is used as our GDH features, and the second one is only used for quadratic features generation. The reason of using bigger feature map for quadratic features generation is that the computed quadratic features are more stable if using more feature points in each region. Therefore, there are $8 \times 8 \times 12 = 768$ GDH features, and $50 \times 12 \times 13/2 = 3900$ quadratic features if all the 50 regions are used.

In the dimensionality reduction procedure, besides DFE, we also use FDA for comparison. When using FDA, the transform basis is learned prior to LVQ training.

We conducted four experiments to evaluate different issues. The first experiment demonstrates the effect of quadratic features and the proposed DQFE method. The second experiment investigates the influence of the size of GDH feature map on the generated quadratic features. The third experiment investigates the influence of the number of regions used in quadratic features generation. The last experiment shows that training sample expansion with synthesized samples is promising.

For experimental efficiency, we implemented all the training and testing processes on our graphic processing units (GPU) server which contains four NVIDIA Tesla C2075 GPU

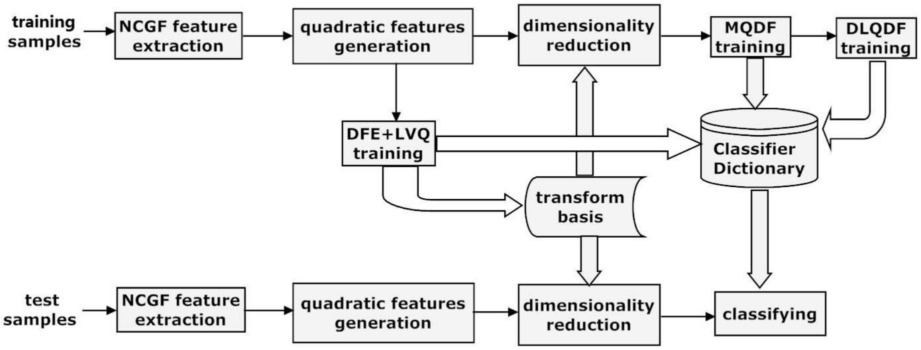

Fig. 9. Flowchart of the training and testing process.

TABLE II. EFFECTS OF QUADRATIC FEATURES UNDER DIFFERENT COMBINATIONS OF DIMENSIONALITY REDUCTION METHODS AND CLASSIFIERS.

Features	FDA			DFE		
	LVQ	MQDF	DLQDF	LVQ	MQDF	DLQDF
GDH	85.43	89.70	89.95	87.66	91.14	91.33
GDH+Corr	87.76	89.64	89.85	90.18	92.13	92.31
GDH+Cov	87.47	89.20	89.49	89.15	91.73	91.91

computing cards, and the programming language we used is NVIDIA's computing united device architecture (CUDA) [20]. This implementation is extended from our previous work [21] which implemented a parallelized training process of DLQDF with GPU.

A. *Effects of Quadratic Features and DQFE*

In this experiment, 12-dimensional GDH features and quadratic features extracted from all the 50 regions are used, hence the feature dimensionality is $768 + 3900 = 4668$. The dimensionality of reduced subspace is 160. The hyper-parameters of classifiers are set as follows: for LVQ, one prototype for each class is used; for MQDF and DLQDF, the number of principal eigenvectors is set to 50, and the minor eigenvalues are set to be class-independent, empirically as the average over all eigenvalues of all classes. No distorted samples were used in training. Table II shows the test accuracies of different combinations of dimensionality reduction methods and classifiers. The third row, fourth row and last row show the results of using GDH features only, using correlation features with GDH features and using covariance features with GDH features, respectively.

We can see that the performance is significantly improved when adding correlation features or covariance features particularly when using DFE for dimensionality reduction, and correlation features are more effective. When using DFE for dimensionality reduction, with correlation features, the test accuracy of LVQ is improved by 2.52%, and that of MQDF and DLQDF is also improved by about 1%. This indicates that the quadratic subspace learned by DQFE is more effective than the linear one of DFE. Whereas with FDA, quadratic features are only effective for LVQ classifier. This is because FDA cannot exploit the discriminative information in the nonlinear space for classification adequately. The improvement of LVQ classifier with FDA is due to the linear surface of classification, which benefits from the nonlinear features though linearly reduced by FDA. In contrast, MQDF and DLQDF are nonlinear classifiers which already utilize the quadratic features in classification.

TABLE III. TEST ACCURACIES WHEN USING CORRELATION FEATURES GENERATED FROM DIFFERENT-SIZED GDH FEATURE MAPS.

zn	FDA			DFE		
	LVQ	MQDF	DLQDF	LVQ	MQDF	DLQDF
8	87.31	88.94	89.09	89.94	91.78	91.90
12	87.72	89.52	89.67	90.17	92.09	92.18
16	87.76	89.64	89.85	90.18	92.13	92.31

TABLE IV. TEST ACCURACIES WITH DIFFERENT REGION NUMBERS FOR QUADRATIC FEATURES GENERATION.

level	#region	FDA			DFE		
		LVQ	MQDF	DLQDF	LVQ	MQDF	DLQDF
1	1	86.01	89.93	90.17	87.97	91.27	91.47
1-2	9	86.72	89.63	89.83	88.74	91.54	91.71
1-3	24	87.37	89.64	89.90	89.52	91.83	92.01
1-4	50	87.76	89.65	89.87	90.18	92.12	92.31

B. *Influence of Feature Map Size*

As mentioned above, two GDH feature maps of different sizes are generated. One is used as GDH features, and the other is used only for quadratic features generation. In this experiment, we study the influence of the size of the latter feature map on the generated quadratic features. The influence is measured by test accuracy. The features used in this experiment are the same as in section V-A, except for the different feature map sizes of the second feature maps. So the dimensionality of the feature vector is still 4668. We only use correlation features as quadratic features from here on as they lead to better results. Table III shows test accuracies of varying feature map sizes. We can see that with larger feature map size, the test accuracies are improved.

C. *Influence of Region Number*

To test the influence of the number of regions on quadratic features generation, this experiment uses different partition levels for feature maps. Table IV shows the results. The first column represents the partition levels used. 1 means only regions of level 1 are used, $1 - 2$ means regions of level 1 and level 2 are used, and so on. The second column represents the total number of regions used. The total number of regions in different partition levels has been illustrated in Table I in section III-B. From the results, we can see that with more regions, the performance is improved.

D. *Effect of Training Set Expansion*

To further improve the performance, we use sample synthesis method mentioned in section IV to generate training samples. We adopt two distortion schemes, namely the single distortion model and the combined distortion model, as men-

TABLE V. EFFECT OF TRAINING SET EXPANSION WITHOUT USING CORRELATION FEATURES.

redDim	FDA			DFE		
	LVQ	**MQDF**	**DLQDF**	**LVQ**	**MQDF**	**DLQDF**
160	86.62	90.38	90.92	88.47	91.44	91.76
200	86.75	90.64	91.16	88.65	91.66	91.98
250	86.70	90.82	91.46	88.77	91.80	92.18
300	86.60	90.82	91.53	88.77	91.83	92.24
400	86.27	90.59	91.41	88.78	91.77	92.20

TABLE VI. EFFECT OF TRAINING SET EXPANSION WITH CORRELATION FEATURES.

redDim	FDA			DFE		
	LVQ	**MQDF**	**DLQDF**	**LVQ**	**MQDF**	**DLQDF**
160	88.81	89.97	90.53	91.11	92.69	92.95
200	89.16	90.40	90.94	91.25	92.81	93.08
250	89.37	90.68	91.29	91.32	92.85	93.18
300	89.46	90.81	91.46	91.36	92.83	93.20
400	89.47	90.89	91.56	91.41	92.75	93.16

tioned before. With each scheme, we generate 10 samples from each real training sample with randomly generated parameters. We observed that the single distortion model is more effective for DFE training while the combined distortion model is more effective for MQDF and DLQDF training. This is possibly because MQDF and DLQDF are complex models compared to DFE+LVQ, thus samples with bigger shape variability are beneficial for their fitting. As DFE is trained with LVQ simultaneously, in the following experiments, FDA+LVQ classifier is also trained with training set expanded with the single distortion model, and MQDF and DLQDF are trained with training set expanded with the combined distortion model. We have tried different dimensionalities of the reduced subspaces. Test accuracies are shown in Table V and Table VI, for the case of classification without correlation features and the case with correlation features, respectively. We see that training set expansion with synthesized samples is effective to improve in both cases the generalization ability. Comparing results in Table VI and Table II, when using DFE for dimensionality reduction, training set expansion improves the best performances of LVQ, MQDF, DLQDF by 1.23%, 0.72%, and 0.89%, respectively. Comparing Table VI and Table V, on expanded training set, when using DFE for dimensionality reduction, correlation features improve the best performances of LVQ, MQDF, DLQDF by 2.63%, 1.02%, and 0.96% respectively. The best result of LVQ with correlation features and training set expansion is 91.41%. This exceeds DLQDF's 91.33% when not using correlation features and training set expansion.

VI. CONCLUSION

In this paper, we proposed a nonlinear feature extraction method called DQFE, which first uses some quadratic features for dimensionality promotion and then reduces the dimensionality using DFE. The quadratic features are generated from GDH features, and two types of quadratic features — correlation features and covariance features were evaluated. Experiments of HCCR using LVQ, MQDF and DLQDF classifiers demonstrated the effectiveness of the proposed DQFE method. With training sample expansion using synthesized samples, the performances of all the classifiers were further improved significantly.

ACKNOWLEDGMENT

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

REFERENCES

[1] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Academic Press, 1990.

[2] A. Biem, S. Katagiri, and B.-H. Juang, "Pattern recognition using discriminative feature extraction," *IEEE Trans. Signal Processing*, vol. 45, no. 2, pp. 500-504, 1997.

[3] C.-L. Liu, R. Mine, and M. Koga, "Building compact classifier for large character set recognition using discriminative feature extraction," in *Proc. 8th ICDAR*, pp. 846-850, 2005.

[4] X.-Y. Zhang and C.-L. Liu, "Evaluation of weighted Fisher criteria for large category dimensionality reduction in application to Chinese handwriting recognition," *Pattern Recognition*, vol. 46, no. 9, pp. 2599-2611, 2013.

[5] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.9, no.1, pp. 149-153, Jan. 1987.

[6] C.-L. Liu, H. Sako, and H. Fujisawa, "Discriminative learning quadratic discriminant function for handwriting recognition," *IEEE Trans. Neural Networks*, vol. 15, no. 2, pp. 430-444, Mar. 2004.

[7] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *IEEE Conf. on CVPR*, pp. 3643-3649, 2012.

[8] X.-B. Jin, C.-L. Liu, and X. Hou, "Regularized margin-based conditional log-likelihood loss for prototype learning," *Pattern Recognition*, vol. 43, no. 7, pp. 2428-2438, 2010.

[9] C.-L. Liu, "High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction," in *Proc. 18th ICPR*, pp. 942-945, 2006.

[10] C.-L. Liu, and M. Nakagawa, "Evaluation of prototype learning algorithms for nearest neighbor classifier in application to handwritten character recognition," *Pattern Recognition*, vol. 34, no. 3, pp. 601-615, 2001.

[11] B.-H. Juang, W. Chou, and C.-H. Lee, "Minimum classification error rate methods for speech recognition," in *IEEE Trans. Speech and Audio Processing*, vol. 5, no. 3, 257-265, 1997.

[12] C.-L. Liu, "Normalization-Cooperated gradient feature extraction for handwritten character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* val. 29, no. 8, pp. 1465-1469, Aug. 2007.

[13] C.-L. Liu and K. Marukawa, "Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition," *Pattern Recognition*, vol. 38, no. 12, pp. 2242-2255, 2005.

[14] T. Horiuchi, R. Haruki, H. Yamada, and K. Yamamoto, "Two-dimensional extension of nonlinear normalization method using line density for character recognition," in *Proc. 4th ICDAR*, pp. 589-600, 1997.

[15] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: a fast descriptor for detection and classification," in *Proc. 9th ECCV*, pp. 589-600, 2005.

[16] K. Leung and C. Leung, "Recognition of handwritten Chinese characters by combining regularization, Fisher's discriminant and distorted sample generation," in *Proc. 10th ICDAR*, pp. 1026-1030, 2009.

[17] P. Simard, D. Steinkraus, and J. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *Proc. 7th ICDAR*, pp. 958-963, 2003.

[18] T. Ha and H. Bunke, "Off-line, handwritten numeral recognition by perturbation method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 5, pp. 535-539, 1997.

[19] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, "CASIA online and offline Chinese handwriting databases," in *Proc. 11th ICDAR*, pp. 37-41, 2011.

[20] NVIDIA, *NVIDIA CUDA C programming guide 4.1*, 2011.

[21] M.-K. Zhou, F. Yin, and C.-L. Liu, "GPU-based fast training of discriminative learning quadratic discriminant function for handwritten Chinese character recognition," in *Proc. 12th ICDAR*, pp. 842-846, 2013.