An Irrelevant Variability Normalization Based Discriminative Training Approach for Online Handwritten Chinese Character Recognition

Jun Du[1]*, Qiang Huo[2]
[1]University of Science and Technology of China, Hefei, Anhui, P. R. China
[2]Microsoft Research Asia, Beijing, P. R. China
Emails: jundu@ustc.edu.cn, qianghuo@microsoft.com

***Abstract*—This paper presents a discriminative training approach to irrelevant variability normalization (IVN) based joint training of feature transforms and prototype-based classifier for recognition of online handwritten Chinese characters. A sample separation margin based minimum classification error criterion is adopted in IVN-based training, while an Rprop algorithm is used for optimizing the objective function. The IVN-trained recognizer can be made both compact and efficient by using a two-level fast-match tree whose internal nodes coincide with the labels of feature transforms. The effectiveness of the proposed approach is confirmed on an online handwritten character recognition task with a vocabulary of 9,306 characters.**

***Keywords*—irrelevant variability normalization, sample separation margin, minimum classification error, Rprop, handwritten Chinese character recognition**

I. Introduction

Using online handwritten Chinese character recognition as an input mode on a portable device has been becoming increasingly popular. Good solutions have been developed to build product engines for online handwritten Chinese character recognition (e.g., [1], [2]). However, the problem of *diversified* training data and/or possible *mismatch* between training and testing conditions has not been addressed explicitly in the above solutions. In this study, we adopt a so-called *irrelevant variability normalization* (IVN) [3] based training strategy to tackle the above problem.

In [4], a so-called speaker adaptive training (SAT) approach was proposed to normalize speaker variability in training hidden Markov models (HMMs) for automatic speech recognition (ASR). The concept of SAT training was generalized to deal with any variabilities irrelevant to phonetic classification in [3], therefore a term of IVN training was coined, where as an illustrative example, the IVN training was used to improve learning HMM state tying from data based on phonetic decision-tree. Since then, many variants of IVN training methods have been tried in ASR area. For example, IVN-based training of feature transforms and HMMs based on maximum likelihood [5] and discriminative training [6] has been verified to be effective for large vocabulary continuous speech recognition (LVCSR). A region-dependent feature transform (RDT) approach proposed in [7] is yet another example of IVN training in ASR area. Only recently, the concept of IVN training was tried in the area of handwriting recognition. For example, in [8], writer adaptive training (WAT) using constrained maximum likelihood linear regression (CMLLR) [9] based feature transform was studied for an HMM-based Arabic handwriting recognition task. RDT-based approach in [7] was also applied to HMM-based off-line handwriting recognition in [10]. More recently, a pattern field classification approach with style normalized transformation was proposed in [11] and demonstrated to be effective for several pattern recognition applications, including handwritten Chinese character recognition.

*This work was done when Jun Du worked at the Speech Group of Microsoft Research Asia, Beijing, China.

In this paper, we study the problem of IVN-based training for online handwritten Chinese character recognition. One of the state-of-the-art techniques to build a Chinese handwriting recognizer is to use a so-called sample separation margin (SSM) based minimum classification error (MCE) criterion [12], [13] to train a prototype-based classifier as reported in [1]. In spite of the large vocabulary of Chinese characters, such a classifier can be made both compact (e.g., [14]) and efficient (e.g., [15]) in the recognition stage. The main contribution of this work is to propose an approach to IVN-based joint training of feature transforms and prototype-based classifier parameters by using the SSM-MCE criterion and demonstrate its effectiveness for Chinese handwriting recognition as an illustrative example. An Rprop algorithm ([16], [17]) is used to optimize the objective function. Furthermore, the IVN-trained recognizer can be made both compact and efficient by using a two-level fast-match tree [15] whose internal nodes coincide with the labels of feature transforms.

Fig. 1 illustrates an overall system development flow of our work in this paper. In the first module, after feature extraction of training samples, an LBG clustering algorithm [18] is used to construct multiple prototypes for each character class. Then a baseline classifier is constructed by using the SSM-MCE training. In the second module, the clusters of feature space associated with feature transforms are generated via the baseline classifier, which are used for the IVN-based SSM-MCE joint training of feature transforms and prototype-based classifier parameters. Finally, with the IVN resources from the second module, at recognition stage (i.e., in the third module), the corresponding transform after cluster selection is used to transform the feature vector of the unknown sample, which is then fed to the IVN-based SSM-MCE trained classifier for recognition.

The remainder of the paper is organized as follows. In Sec-

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.22

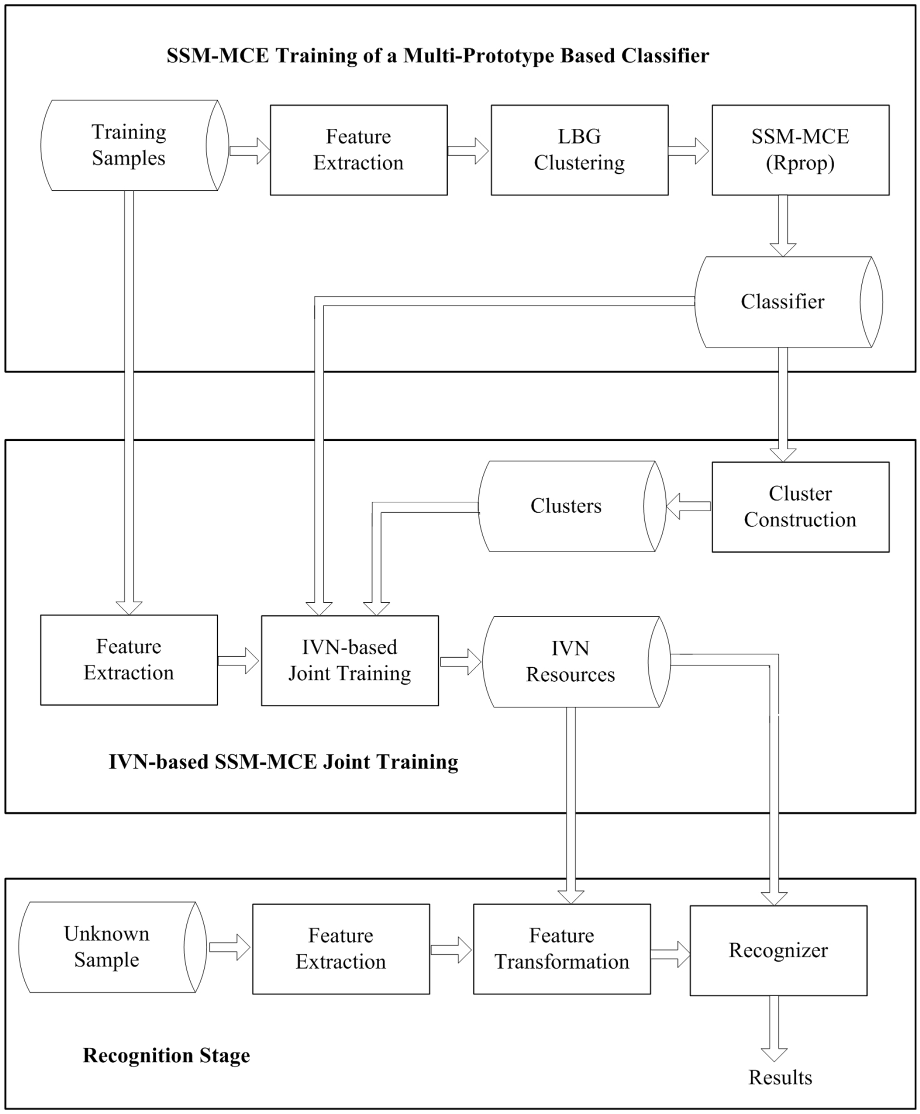

Fig. 1. Overall flow of system development.

tion II, we describe briefly how to construct a multi-prototype based classifier by using SSM-MCE training. In Section III, we present the detailed procedure for IVN-based SSM-MCE joint training of feature transforms and classifier parameters. The fast-match technique is introduced in Section IV. In Section V, we report experimental results. Finally we conclude the paper in Section VI.

II. SSM-MCE Trained Classifier

Suppose our classifier can recognize M character classes denoted as $\{C_i|i = 1, ..., M\}$. For a multi-prototype based classifier, each class C_i is represented by K_i prototypes, $\lambda_i = \{\mathbf{m}_{ik} \in \mathcal{R}^D|k = 1, ..., K_i\}$, where $\mathbf{m}_{ik}$ is the k^{th} prototype of the i^{th} class. Let's use $\Lambda = \{\lambda_i\}$ to denote the set of prototypes. In the classification stage, a feature vector $\mathbf{y} \in \mathcal{R}^D$ is first extracted. Then $\mathbf{y}$ is compared with each of the M classes by evaluating a Euclidean distance based discriminant function for each class C_i as follows

$$g_i(\mathbf{y}; \lambda_i) = -\min_k \parallel \mathbf{y} - \mathbf{m}_{ik} \parallel^2 \ . \tag{1}$$

The class with the maximum discriminant function score is chosen as the recognized class $r(\mathbf{y}; \mathbf{\Lambda})$, i.e.,

$$r(\mathbf{y}; \mathbf{\Lambda}) = \arg\max_i g_i(\mathbf{y}; \lambda_i) \ . \tag{2}$$

In the training stage, given a set of training feature vectors $\mathcal{Y} = \{\mathbf{y}_r \in \mathcal{R}^D|r = 1, ..., R\}$, first we initialize $\mathbf{\Lambda}$ by LBG clustering [18]. Then $\mathbf{\Lambda}$ can be re-estimated by minimizing the following SSM-MCE objective function:

$$l(\mathcal{Y}; \mathbf{\Lambda}) = \frac{1}{R}\sum_{r=1}^{R} \frac{1}{1 + \exp[-\alpha d(\mathbf{y}_r; \mathbf{\Lambda}) + \beta]} \tag{3}$$

where α, β are two control parameters, and $d(\mathbf{y}_r; \mathbf{\Lambda})$ is a misclassification measure defined by using a so-called sample separation margin (SSM) as follows [12]:

$$d(\mathbf{y}_r;\mathbf{\Lambda}) = \frac{-g_p(\mathbf{y}_r;\lambda_p) + g_q(\mathbf{y}_r;\lambda_q)}{2 \parallel \mathbf{m}_{p\hat{k}} - \mathbf{m}_{q\bar{k}} \parallel} \tag{4}$$

where

$$\hat{k} = \arg\min_{k} \parallel \mathbf{y}_r - \mathbf{m}_{pk} \parallel^2 \tag{5}$$

$$q = \arg\max_{i\in\mathcal{M}_r} g_i(\mathbf{y}_r;\lambda_i) \tag{6}$$

$$\bar{k} = \arg\min_{k} \parallel \mathbf{y}_r - \mathbf{m}_{qk} \parallel^2 \tag{7}$$

and $\mathcal{M}_r$ is the hypothesis space for the r^{th} sample, excluding the true label p.

To optimize the objective function in Eq. (3), the same implementation of Rprop algorithm as described in [1] is adopted here.

III. IVN-BASED SSM-MCE JOINT TRAINING

A. Feature Transformation

In this study of IVN-based training, the following feature transformation is used:

$$\mathbf{x}_r = \mathcal{F}(\mathbf{y}_r;\mathbf{\Theta}) = \mathbf{A}_{e_r}\mathbf{y}_r + \mathbf{b}_{e_r} \tag{8}$$

where $\mathbf{y}_r$ and $\mathbf{x}_r$ are the r^{th} D-dimensional input and transformed feature vectors, respectively; and e_r is the transform label for the r^{th} sample. Let's use $\mathbf{\Theta} = \{(\mathbf{A}_e, \mathbf{b}_e)|e = 1, ..., E\}$ to denote the set of transform parameters, where $\mathbf{A}_e$ is a $D\times D$ nonsingular matrix and $\mathbf{b}_e$ is a D-dimensional bias vector.

B. Cluster Construction and Selection

Suppose the feature space can be divided into E clusters and each cluster e is associated with one linear transform $(\mathbf{A}_e, \mathbf{b}_e)$, which is assumed to normalize the irrelevant variability of the feature vector belonging to this cluster. In this work, to construct the clusters, we divide all the prototypes of all classes into E groups by using k-means clustering approach. The centroid of each cluster is calculated as the sample mean of the prototypes belonging to the cluster. Then in both IVN-based training and recognition stage, given the clusters and for each feature vector, a transform label is assigned to the feature vector as the label of the cluster having the minimum Euclidean distance between the feature vector and the cluster centroid.

C. Training Procedure

The IVN-based SSM-MCE objective function is defined as follows:

$$l(\mathcal{Y};\mathbf{\Lambda},\mathbf{\Theta}) = \frac{1}{R}\sum_{r=1}^{R}\frac{1}{1+\exp[-\alpha d(\mathbf{y}_r;\mathbf{\Lambda},\mathbf{\Theta})+\beta]} \tag{9}$$

where

$$d(\mathbf{y}_r;\mathbf{\Lambda},\mathbf{\Theta}) = \frac{-g_p(\mathbf{x}_r;\lambda_p) + g_q(\mathbf{x}_r;\lambda_q)}{2 \parallel \mathbf{m}_{p\hat{k}} - \mathbf{m}_{q\bar{k}} \parallel}\ . \tag{10}$$

In the above equations, $\mathbf{x}_r$ is defined in Eq. (8), where the transform label e_r can be determined as described in Section III-B. The following *method of alternating variables* can then be used to jointly estimate $\mathbf{\Theta}$ and $\mathbf{\Lambda}$ by minimizing the above objective function:

Step 1: *Initialization*
First, the classifier parameters $\mathbf{\Lambda}$ are initialized by using SSM-MCE training described in Section II. The transform parameters $\mathbf{\Theta}$ are initialized as: $\mathbf{b}_e = \mathbf{0}$ and $\mathbf{A}_e = \mathbf{I}$.

Step 2: *Estimating the feature transform parameters $\mathbf{\Theta}$ by fixing the classifier parameters $\mathbf{\Lambda}$*
Given the fixed classifier parameters $\overline{\mathbf{\Lambda}}$, the SSM-MCE objective function $l(\mathcal{Y};\overline{\mathbf{\Lambda}},\mathbf{\Theta})$ can be optimized by using an Rprop algorithm with N_T iterations as described in [19], [20].

Step 3: *Estimating the classifier parameters $\mathbf{\Lambda}$ by fixing the feature transform parameters $\mathbf{\Theta}$*
Given the updated transform parameters $\overline{\mathbf{\Theta}}$ obtained in Step 2, we first transform each training feature vector $\mathbf{y}_r$ by using Eq. (8). Then an Rprop algorithm with N_C iterations is performed as described in [1] to re-estimate classifier parameters $\mathbf{\Lambda}$ by minimizing the objective function $l(\mathcal{Y};\mathbf{\Lambda},\overline{\mathbf{\Theta}})$.

Step 4: *Repeat Step 2 and Step 3 N_{IVN} times.*

In the above training procedure, the control parameters N_T, N_C, and N_{IVN} are set empirically.

IV. FAST MATCH TECHNIQUE

Our fast-match technique is based on a two-level tree [15]. To construct the tree, G clusters are first generated as described in Section III-B. Each cluster has a bucket consisting of character classes with their prototypes belonging to the cluster. Each training feature vector will then be classified into the cluster with the minimum Euclidean distance between the feature vector and the cluster centroid. The character class of the training sample will be added into the bucket if it is not in the bucket yet. In this way, we obtain a two-level tree with G buckets, each containing a number of character classes. In this work, to make the recognizer both compact and efficient, we share the clusters in IVN-based training and fast-match tree, i.e., we set $E = G$. In recognition stage, given the feature vector extracted from an unknown sample, we can find "Top N" candidates efficiently by using the following fast-match procedure:

- Compare the input feature vector with each cluster centroid and sort the result in ascending order of the Euclidean distances, which can be considered as the first-level recognition;
- If IVN is performed, the feature transform associated with the first cluster is applied to the input feature vector; Otherwise, skip this step;
- Merge all character classes in the top N_B buckets and use them to perform the second-level recognition as usual.

In the above procedure, a technique known as the *partial distance based elimination* has been used to speed up the process of identifying the "Top N" candidates.

V. Experiments and Results

A. Experimental Setup

The experiments are conducted on the task of recognizing isolated online handwritten characters with a vocabulary of 9,306 character classes including 9,143 Chinese characters, 62 alphanumeric characters, 101 punctuation marks and symbols. For training, we used about 1,000 samples per character class. Three testing sets are used for evaluation: 1) **Regular-1**: 97,221 samples from 6,903 character classes which are written in regular style; 2) **Regular-2**: 84,549 samples from 2,355 uncommon character classes in regular style; 3) **Cursive**: 383,064 samples from 3,755 frequently used character classes written in cursive style. Because there are much more regular-style training samples than cursive ones, the *re-sampling* of training samples is performed as in [14]. For feature extraction, a 512-dimensional raw feature vector is extracted as described in [21], which is followed by LDA (linear discriminant analysis) transformation to obtain a lower dimensional feature vector. As for the number of prototypes for each character, we use A prototypes for 3,755 most frequently used Chinese characters and B prototypes for the rest of character classes. For Rprop-based SSM-MCE training and IVN-based SSM-MCE joint training, the control parameters are set as described in [1] and [19], [20]. Other control parameters are set as: $D = 80$, $E = G = 128$, $N_T = 10$, $N_C = 10$, $N_{IVN} = 5$.

To handle large-scale training data, the tools for LBG clustering, SSM-MCE training, and IVN-based SSM-MCE joint training with the Rprop algorithm have been implemented based upon MSR Asia's MPI-based machine learning platform [22]. This platform was developed on top of Microsoft Windows HPC Server, and optimized for various machine learning algorithms. With this high-performance parallel computing platform, experiments can be run very efficiently for large-scale tasks.

B. Experimental Results

Table I summarizes a performance (recognition accuracies in %) comparison of baseline systems and IVN-trained systems on three testing sets under different settings of the number of prototypes and the number of buckets searched in fast-match tree. The footprint (in MB) and runtime latency (normalized by Baseline(2,1) without fast-match) of the corresponding recognizers are also compared. "Baseline" refers to an SSM-MCE trained system without IVN training while "IVN" refers to a system using our proposed IVN-based joint training. The second column of Table I indicates the top N_B buckets selected for second-level recognition in fast-match tree, where "N/A" means no fast-match is used. Three prototype configurations, namely (2,1), (4,2), and (8,4) are listed for comparison, because over-training would be observed if the number of prototypes was increased beyond (8,4) in current experiments. The runtime latency in the last column only includes the recognition time after feature extraction.

Based on those results, several observations can be made. First, IVN systems can achieve consistently significant improvements in recognition accuracy compared with the corresponding Baseline systems on all testing sets. Second, under the same prototype setting, the increased runtime latency from Baseline to IVN systems can be almost ignored, especially in cases with fast-match because the first-level recognition of fast-match tree and the cluster selection for each testing feature vector are shared completely. Third, IVN systems can still outperform Baseline systems with smaller footprints and less runtime latency, e.g., IVN(2,1) *vs.* Baseline(4,2) and IVN(4,2) *vs.* Baseline(8,4). Finally, with fast-match technique, the runtime latency of IVN systems can be reduced significantly while the footprint is only increased slightly. The tradeoff between recognition accuracy and efficiency can be made by setting different N_B. Compared with systems without using fast-match technique, $N_B = 5$ keeps the same recognition accuracy with reduced runtime latency while $N_B = 3$ degrades slightly recognition accuracy with a much more significant reduction of runtime latency.

Table II compares the "Top-N" recognition accuracies of Baseline systems and IVN-trained systems on three testing sets under different settings of the number of prototypes and a single setting of $N_B = 5$ for fast match. From the Top-5 and Top-10 results, IVN systems can achieve very high recognition accuracies already.

VI. Conclusion

In this paper, we have proposed an approach to IVN-based SSM-MCE joint training of feature transforms and a prototype-based classifier and demonstrated its effectiveness for online handwritten Chinese character recognition as an illustrative example. The IVN-trained recognizer can be made both compact and efficient by using a two-level fast-match tree whose internal nodes coincide with the labels of feature transforms. Given the consistent improvement of recognition accuracy compared with the corresponding SSM-MCE trained systems without using IVN training, the proposed approach in this study offers a good product solution to construct a handwritten character recognizer to be deployed on mobile devices with limited memory for East Asian languages such as Chinese, Japanese, and Korean. As future work, we will study the application of IVN training to other classifiers, e.g., modified quadratic discriminant function (MQDF) based classifiers [23], [24], [13], [2].

References

[1] J. Du and Q. Huo, "Designing compact classifiers for rotation-free recognition of large vocabulary online handwritten Chinese characters," *Proc. ICASSP-2012*, pp.1721-1724.

[2] Y.-Q. Wang and Q. Huo, "Building compact recognizers of handwritten Chinese characters using precision constrained Gaussian model, minimum classification error training and parameter compression," *International Journal on Document Analysis and Recognition*, Vol. 14, No. 3, pp.255-262, 2011.

[3] Q. Huo and B. Ma, "Irrelevant variability normalization in learning HMM state tying from data based on phonetic decision-tree," *Proc. ICASSP-1999*, pp.577-580.

[4] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," *Proc. ICSLP-1996*, pp.1137-1140.

[5] G.-C. Shi, Y. Shi, and Q. Huo, "A study of irrelevant variability normalization based training and unsupervised online adaptation for LVCSR," *Proc. Interspeech-2010*, pp.1357-1360.

[6] Y. Zhang, J. Xu, Z.-J. Yan, and Q. Huo, "A study of an irrelevant variability normalization based discriminative training approach for LVCSR," *Proc. ICASSP-2011*, pp.5308-5311.

TABLE I. PERFORMANCE (RECOGNITION ACCURACIES IN %) COMPARISON OF BASELINE SYSTEMS AND IVN-TRAINED SYSTEMS ON THREE TESTING SETS UNDER DIFFERENT SETTINGS OF THE NUMBER OF PROTOTYPES AND THE NUMBER OF BUCKETS SEARCHED IN FAST-MATCH TREE. THE FOOTPRINT (IN MB) AND RUNTIME LATENCY (NORMALIZED BY BASELINE(2,1) WITHOUT FAST-MATCH) OF THE CORRESPONDING RECOGNIZERS ARE ALSO COMPARED.

Method(A,B)	N_B	Regular-1	Regular-2	Cursive	Footprint	Latency
Baseline(2,1)	N/A	96.53	94.84	91.53	4.10	1
Baseline(4,2)	N/A	96.88	94.93	92.28	8.08	1.92
Baseline(8,4)	N/A	97.09	94.66	92.74	16.00	3.66
IVN(2,1)	N/A	96.89	95.32	92.18	7.30	1.06
IVN(4,2)	N/A	97.19	95.54	92.88	11.28	1.99
IVN(8,4)	N/A	97.38	95.32	93.23	19.2	3.75
Baseline(2,1)	5	96.52	94.84	91.53	4.44	0.52
Baseline(4,2)	5	96.88	94.92	92.28	8.41	0.89
Baseline(8,4)	5	97.09	94.65	92.74	16.33	1.58
IVN(2,1)	5	96.89	95.32	92.19	7.60	0.53
IVN(4,2)	5	97.19	95.53	92.88	11.57	0.90
IVN(8,4)	5	97.39	95.31	93.23	19.49	1.59
Baseline(2,1)	3	96.52	94.80	91.52	4.44	0.37
Baseline(4,2)	3	96.88	94.89	92.28	8.41	0.65
Baseline(8,4)	3	97.09	94.62	92.74	16.33	1.17
IVN(2,1)	3	96.89	95.29	92.18	7.60	0.38
IVN(4,2)	3	97.19	95.48	92.87	11.57	0.66
IVN(8,4)	3	97.39	95.27	93.22	19.49	1.18

TABLE II. PERFORMANCE ("TOP-N" RECOGNITION ACCURACIES IN %) COMPARISON OF BASELINE SYSTEMS AND IVN-TRAINED SYSTEMS ON THREE TESTING SETS UNDER DIFFERENT SETTINGS OF THE NUMBER OF PROTOTYPES AND A SINGLE SETTING OF $N_B = 5$ FOR FAST MATCH.

Method(A,B)	Top-N	Regular-1	Regular-2	Cursive
Baseline(2,1)	Top-1	96.52	94.84	91.53
	Top-5	99.49	99.45	97.89
	Top-10	99.67	99.73	98.72
Baseline(4,2)	Top-1	96.88	94.92	92.28
	Top-5	99.65	99.52	98.18
	Top-10	99.83	99.75	98.92
Baseline(8,4)	Top-1	97.09	94.65	92.74
	Top-5	99.71	99.53	98.39
	Top-10	99.86	99.80	99.05
IVN(2,1)	Top-1	96.89	95.32	92.19
	Top-5	99.65	99.65	98.20
	Top-10	99.83	99.82	98.94
IVN(4,2)	Top-1	97.19	95.53	92.88
	Top-5	99.73	99.70	98.45
	Top-10	99.87	99.85	99.11
IVN(8,4)	Top-1	97.39	95.31	93.23
	Top-5	99.77	99.73	98.60
	Top-10	99.89	99.87	99.20

[7] B. Zhang, S. Matasoukas, and R. Schwartz. "Discriminatively trained region dependent feature transforms for speech recognition," *Proc. ICASSP-2006*, pp.1520-1523.

[8] P. Dreuw, D. Rybach, C. Gollan, and H. Ney, "Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition," *Proc. ICDAR-2009*, pp. 21-25.

[9] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, Vol. 12, No. 2, pp.75-98, 1998.

[10] J. Chen, B. Zhang, H. Cao, R. Prasad, P. Natarajan, "Applying discriminatively optimized feature transform for HMM-based off-line handwriting recognition," *Proc. ICFHR-2012*, pp.219-224.

[11] X.-Y. Zhang, K. Huang, C.-L. Liu, "Pattern field classification with style normalized transformation," *Proc. IJCAI-2011*, pp.1621-1626.

[12] T. He and Q. Huo, "A study of a new misclassification measure for minimum classification error training of prototype-based pattern classifiers," *Proc. ICPR-2008*.

[13] Y.-Q. Wang and Q. Huo, "Sample-separation-margin based minimum classification error training of pattern classifiers with quadratic discriminant functions," *Proc. ICASSP-2010*, pp.1866-1869.

[14] Y.-Q. Wang and Q. Huo, "A study of designing compact recognizers of handwritten Chinese characters using multiple-prototype based classifiers," *Proc. ICPR-2010*, pp.1872-1875.

[15] Z.-D. Feng and Q. Huo, "Confidence guided progressive search and fast match techniques for high performance Chinese/English OCR," *Proc. ICPR-2002*, pp.III-89-92.

[16] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the Rprop algorithm," *Proc. International Conference on Neural Networks*, 1993, pp.586-591.

[17] C. Igel and M. Hüsken, "Improving the Rprop learning algorithm," *Proc. Int. Symp. on Neural Computation*, 2000, pp.115-121.

[18] Y. Linde, A. Buzo and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, Vol. 28, No. 1, pp.84-95, 1980.

[19] J. Du and Q. Huo, "A discriminative linear regression approach to OCR adaptation," *Proc. ICPR-2012*, pp.629-632.

[20] J. Du and Q. Huo, "A discriminative linear regression approach to adaptation of multi-prototype based classifiers and its applications for Chinese OCR," accepted by *Pattern Recognition*, 2013.

[21] Z.-L Bai and Q. Huo, "A study on the use of 8-directional features for online handwritten Chinese character recognition," *Proc. ICDAR-2005*, pp.262-266.

[22] Z.-J. Yan, T. Gao and Q. Huo, "Designing an MPI-based parallel and distributed machine learning platform on large-scale HPC clusters," *2012 International Workshop on Statistical Machine Learning for Speech Processing*, Kyoto, Japan, March 31, 2012 (http://www.ism.ac.jp/IWSML2012/).

[23] F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 9, No. 1, pp.149-153, 1987.

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