HIGH PERFORMANCE CHINESE OCR BASED ON GABOR FEATURES, DISCRIMINATIVE FEATURE EXTRACTION AND MODEL TRAINING

Qiang Huo, Yong Ge and Zhi-Dan Feng

Department of Computer Science and Information Systems,
The University of Hong Kong, Pokfulam Road, Hong Kong, China
(E-mail: qhuo@csis.hku.hk)

ABSTRACT

We've been developing a Chinese OCR engine for machine printed documents. Currently, our OCR engine can support a vocabulary of 6921 characters which include 6707 simplified Chinese characters in GB2312-80, 12 frequently used GBK Chinese characters, 62 alphanumeric characters, 140 punctuation marks and symbols. The supported font styles include Song, Fang Song, Kai, He, Yuan, LiShu, WeiBei, XingKai, etc. The averaged character recognition accuracy is above 99% for newspaper quality documents with a recognition speed of about 250 characters per second on a Pentium III-450MHz PC yet only consuming less than 2MB memory. In this paper, we describe the key technologies we used to construct the above recognizer. Among them, we highlight three key techniques contributing to the high recognition accuracy, namely the use of Gabor features, the use of discriminative feature extraction, and the use of minimum classification error as a criterion for model training.

1. INTRODUCTION

In the last two decades, many advances have been achieved in OCR (optical character recognition) area (e.g., [1, 2]). During the same period, a great deal of efforts have also been made towards Chinese OCR (e.g., [3, 4]). Consequently, many Chinese OCR products became available on the market. In the past, Chinese OCR products can only support 3755 level 1 Chinese characters in GB2312-80 which is a national Chinese character set standard, established in 1981 by Chinese government. Only very recently, several products supporting the full set of 6763 GB2312-80 Chinese characters become available on the market. In the past several years, we've also been developing a Chinese OCR engine for machine printed documents. Currently, our OCR engine can support a vocabulary of 6921 characters which include 6707 meaningful simplified Chinese characters derived by removing 56 meaningless single-radical characters from 6763 characters defined in GB2312-80, 12 frequently used GBK Chinese characters, 62 alphanumeric characters, 140 punctuation marks and symbols. The supported font styles include Song, Fang Song, Kai, He, Yuan, LiShu, WeiBei, XingKai, etc. The averaged character recognition accuracy is above 99% for newspaper quality documents. The main purpose of this paper is to describe the key technologies we used to construct our Chinese character recognizer. Among them, we highlight three key techniques contributing to the high recognition accuracy, namely (1) the use of Gabor features to characterize the original character image, (2) the use of discriminative feature extraction methods to reduce the dimension of the feature vector, and (3) the use of minimum classification error (MCE) as a criterion for model training to achieve the high performance in a maximum discriminant function based character recognizer.

This work was supported by a grant from the RGC of the Hong Kong SAR (Project No. HKU7020/98E) and two internal HKU CRCG grants.

2. METHODOLOGY AND ARCHITECTURE

Our Chinese OCR system works roughly as follows: Given a binary image of a document page, after skew correction, page segmentation can be conducted manually. After selecting a text region and determining whether the text is horizontally or vertically aligned, line segmentation based on projection profile analysis is conducted automatically. Once a character line is located, the OCR problem becomes similar to the automatic speech recognition (ASR) problem. Similar to what have been used successfully in the ASR area, we construct our character recognizer by adopting a powerful *statistical pattern recognition* paradigm and a strategy of *dynamic programming search* over a structural network representation of character image models and other possible linguistic knowledge sources. By considering the characteristics of printed Chinese documents, an innovative confidence-guided integrated search technique has been developed for the recognition of the whole character line.

More specifically, given a line of character image, we first segment it into a sequence of sub-segments using the projection profile, aspect ratio statistics, minimum and maximum character width constraints, and other heuristics. From these sub-segments, we can construct dynamically a search graph, with each node representing a potential segmentation point, and each arc representing a hypothesized character with an associated dissimilarity score, a confidence measure for recognition result and other information. For the arc with a confidence score below a pre-specified threshold, we need to re-segment the part of the image associated with the arc and dynamically update the search graph. So the recognition of the whole line of characters can be cast as finding the shortest path from the starting node to the ending node in the search graph. Other knowledge sources such as language model, insertion penalty, etc. can be easily incorporated into the above search strategy. The principle of *dynamic programming* can be used to design an efficient implementation of the above search method. We found that this approach works quite well for dealing with the difficult problems of the broken strokes, touching and overlapping characters.

In the following, we first describe a Chinese character database

constructed for supporting our research. Then, we describe in detail the character modeling techniques we used to provide the character candidates and the associated dissimilarity scores for each arc in the search graph. This is essentially a traditional character classification problem for a given unknown character image.

3. DATA COLLECTION

In order to achieve a high recognition accuracy for documents of different types and with different quality, it is critical to collect a sufficient amount of representative training data which follow as faithly as possible the sample distribution of the testing data to be recognized. Over the years, a Chinese character corpus has been constructed in our lab with in total $3,024,043$ character image samples from 6921 character classes as described in the introduction section. The original documents are from varied sources, such as newspapers, magazines, journals, books, printed lists of characters generated from many popular font libraries available on the market. The document quality and font sizes vary widely among the various sources. Many font styles as described in the introduction section are observed in our character corpus. Depending on the font size of the documents, the document pages have been digitized at a resolution ranging from 300 DPI to 500 DPI on several flatbed scanners. In our database, each character sample is stored as a normalized $N \times N$ ($N = 40$ here) binary image. We have chosen randomly about 20% of character samples for each character class to form a testing set and the remaining samples to form a training set. By this partition, there are 2,412,898 character samples in the training set and 611,145 character samples in the testing set.

4. MAXIMUM DISCRIMINANT BASED CHARACTER CLASSIFICATION AND MCE TRAINING

We adopt a maximum discriminant function based approach to construct our printed Chinese character classifier. Suppose there are M character classes $\{C_i\}_{i=1}^{M}$, each being modeled by K_i prototypes, $\lambda_i = \{m_{ik}\}_{k=1}^{K_i}$, where each prototype m_{ik} is a D dimensional vector. We use $\Lambda = \{\lambda_i\}_{i=1}^{M}$ to denote the set of prototype parameters. The aim of our character recognizer is to classify an input binary character image $f(x,y)$ ($x, y = 1, 2, \cdots, N$; $f(x,y)$ takes the value of either 1 or 0) as one of the M classes. This is done in two steps: a feature analysis step and a pattern classification step. In the feature analysis step, the input image $f(x,y)$ is analyzed and D_1 raw features are measured to form a feature vector X. The D_1-dimensional vector X is then transformed into a new feature vector Y of dimension D by using a $D_1 \times D$ linear transformation matrix W, i.e., $Y = W^t X$, where $D \leq D_1$. If $D < D_1$, an effect of dimension reduction is achieved. In the pattern classification step, the feature vector Y is compared with each of the M character models and a discriminant function is computed for each class C_i as follows:

$$g_i(X;\Lambda,W) = -\min_k ||Y - m_{ik}||^2 = -\min_k ||W^t X - m_{ik}||^2$$
$$= -\min_k \sum_{d=1}^{D} (\sum_{e=1}^{D_1} W_{ed} X_e - m_{ikd})^2 \quad (1)$$

The class that gives the maximum discriminant function is considered to be the recognized class, i.e.,

$$X \in C_i \quad \text{if } i = \arg\max_j g_j(X;\Lambda,W) \quad (2)$$

and the value of $-g_i(X;\Lambda,W)$ serve as the dissimilarity score for the hypothesized character C_i in the search graph described above. Both sets of parameters Λ and W can be trained from a set of training samples.

4.1. Using Gabor Features as Raw Features

Although many types of features have been proposed for Chinese OCR, we decide to use Gabor features to serve as the raw features extracted from each character image. Gabor features have been widely used in computer vision, texture analysis, face recognition, fingerprint recognition, and more recently, character recognition as well. An important property of Gabor features is that they can achieve a joint optimal resolution in both the spatial and the spatial-frequency domains. Gabor features have also been found less sensitive to noises, small range of translation, rotation, and scaling. Specifically, we adopt the following complex 2-D Gabor filter originally reported in [5] for face recognition:

$$G(x,y;\kappa,\vartheta_k) = G_1(x,y)[\cos(R) - \exp(-\frac{\sigma^2}{2})]$$
$$+iG_1(x,y)\sin(R) \quad (3)$$

where $G_1(x,y) = \frac{\kappa^2}{\sigma^2}\exp[-\frac{\kappa^2(x^2+y^2)}{2\sigma^2}]$ with $\sigma = \pi$, $R = \kappa x\cos\vartheta_k + \kappa y\sin\vartheta_k$, $\kappa = \frac{2\pi}{\iota}$, and $\vartheta_k = \frac{\pi k}{\mathcal{M}}$ with $k = 0, 1, 2, ..., \mathcal{M}-1$. The parameters ι and ϑ_k are the wavelength and orientation of the above plane wave respectively. Given a binary character image $f(x,y)$, at a sampling point (x_0, y_0), $\mathcal{M}$ Gabor features can be derived as the magnitudes of the $\mathcal{M}$ Gabor filter outputs as follows:

$$f_{\iota,k}(x_0,y_0) = |\sum_{x=-x_0}^{N-x_0-1}\sum_{y=-y_0}^{N-y_0-1} f(x_0+x, y_0+y)G(x,y;\kappa,\vartheta_k)|, \quad (4)$$

where $k = 0, 1, 2, ..., \mathcal{M}-1$. Consequently, for a given wavelength ι and by uniformly choosing $N_1 \times N_1$ spatial sampling points of (x_0, y_0)'s, we can derive $D_1 = N_1 \times N_1 \times \mathcal{M}$ Gabor features from a given image $f(x,y)$ and use them to form the raw feature vector X. The optimal values of controlling parameters ι, N_1, and $\mathcal{M}$ are task- and data-dependent, thus can only be determined by experiments. By applying the above feature extraction method to each character image in our training data set, we can derive a training set of feature vectors $\mathcal{X} = \{\mathcal{X}_i\}_{i=1}^{M}$ where $\mathcal{X}_i = \{X_i^{(j)}\}_{j=1}^{n_i}$ represents the set of n_i training samples for class C_i with $X_i^{(j)}$ being the j-th training sample. Let's use N_{tr} to denote the total number of training samples, i.e., $N_{tr} = \sum_{i=1}^{M} n_i$.

4.2. LDA-Based Feature Extraction

There are many ways to estimate the linear transformation W for discriminative feature extraction and/or dimension reduction. One traditional way is to use the linear discriminant analysis (LDA) [6]. From $\mathcal{X}$, we can calculate a within-class scatter matrix

$$S_w = \sum_{i=1}^{M}\sum_{X_i^{(j)}\in\mathcal{X}_i} (X_i^{(j)} - m_i)(X_i^{(j)} - m_i)^t$$

and a between-class scatter matrix

$$S_b = \sum_{i=1}^{M} n_i (m_i - m)(m_i - m)^t$$

where $m_i = \frac{1}{n_i} \sum_{X_i^{(j)} \in \mathcal{X}_i} X_i^{(j)}$ and $m = \frac{1}{N_{tr}} \sum_{i=1}^{M} n_i m_i$. For LDA, W can then be derived to maximize the following objective function

$$J(W) = \frac{|W^t S_b W|}{|W^t S_w W|} \tag{5}$$

The solution can be shown to correspond to the generalized eigenvectors of the following equation:

$$S_b \mathbf{w}_i = \gamma_i S_w \mathbf{w}_i \tag{6}$$

By sorting the eigenvalues $\gamma_i, i = 1, 2, \cdots, \min\{D_1, M-1\}$ in descending order, we can then use the corresponding first D eigenvectors $\mathbf{w}_i$ to form the columns of the matrix W. Using W, $\mathcal{X}$, and the transformation $Y = W^t X$, we can derive another form of training data set $\mathcal{Y} = \{\mathcal{Y}_i\}_{i=1}^{M}$ where $\mathcal{Y}_i = \{Y_i^{(j)}\}_{j=1}^{n_i}$ represents the set of n_i training samples for class C_i with $Y_i^{(j)}$ being the j-th training sample. From $\mathcal{Y}$, we can use a likelihood based clustering approach to estimate the prototype parameters Λ.

4.3. MCE-Based Feature Extraction and Model Training

In addition to the LDA and clustering-based training, in the past decade, an MCE-based training technique has been developed for the estimation of classifier parameters Λ [7] and feature extractor parameters W [8, 9], either separately or jointly (e.g. [10]). This technique was primarily established in the ASR area, but recently was also studied in OCR area (e.g. [11, 12]). We also adopted this technique in constructing our Chinese character recognizer and we describe some details in the following.

Given the definition of the discriminant function in Eq. (1) and the classification rule in Eq. (2), a misclassification measure [7] can be defined as

$$d_i(X_i^{(j)}; \Lambda, W) = \frac{g_i(X_i^{(j)}; \Lambda, W) - g_q(X_i^{(j)}; \Lambda, W)}{g_i(X_i^{(j)}; \Lambda, W) + g_q(X_i^{(j)}; \Lambda, W)} \tag{7}$$

for each training sample $X_i^{(j)}$, where

$$q = \arg\max_{n, n \neq i} g_n(X_i^{(j)}; \Lambda, W) \tag{8}$$

The above specific misclassification measure was proposed in [11] and also used in [12]. The loss function for $X_i^{(j)}$ is then defined as:

$$\ell_i(X_i^{(j)}) = \frac{1}{1 + \exp[-\alpha(d_i(X_i^{(j)}; \Lambda, W) + \beta)]} \tag{9}$$

where α and β are two control parameters. An empirical average loss can then be defined on the training set $\mathcal{X}$ as

$$L_0(\Lambda, W) = \frac{1}{N_{tr}} \sum_{i=1}^{M} \sum_{j=1}^{n_i} \ell_i(X_i^{(j)}) \tag{10}$$

Then the parameters Λ and W can be estimated by minimizing the $L_0(\Lambda, W)$. In practice, we use the following approximate sequential gradient descent algorithm to solve this problem.

Given $\mathcal{X}$, we first randomize the ordering of $\{X_i^{(j)}\}$ and then we present the training samples sequentially. Upon the presentation of each training sample $X_i^{(j)}$, in addition to calculate the $g_i(X_i^{(j)}; \Lambda, W)$, we also calculate

$$g_i^{(1)}(X_i^{(j)}; \Lambda, W) = -\min_{k, k \neq \hat{k}} ||W^t X_i^{(j)} - m_{ik}||^2$$

where $\hat{k} = \arg\min_k ||W^t X_i^{(j)} - m_{ik}||^2$.

If $g_i^{(1)}(X_i^{(j)}; \Lambda, W) > g_q(X_i^{(j)}; \Lambda, W)$, then we will not update Λ, W; otherwise, W, $m_{i\hat{k}}$ and $m_{q\bar{k}}$ will be updated as follows:

$$W_{cd}^{(t+1)} = W_{cd}^{(t)} + \epsilon_t v \times \left[\frac{g_q (Y_{id}^{(j,t)} - m_{i\hat{k}d}^{(t)}) X_{ic}^{(j)}}{(g_i + g_q)^2} - \frac{g_i (Y_{id}^{(j,t)} - m_{q\bar{k}d}^{(t)}) X_{ic}^{(j)}}{(g_i + g_q)^2} \right]$$

$$m_{i\hat{k}}^{(t+1)} = m_{i\hat{k}}^{(t)} - \epsilon_t v \frac{g_q}{(g_i + g_q)^2} (Y_i^{(j,t)} - m_{i\hat{k}}^{(t)})$$

$$m_{q\bar{k}}^{(t+1)} = m_{q\bar{k}}^{(t)} + \epsilon_t v \frac{g_i}{(g_i + g_q)^2} (Y_i^{(j,t)} - m_{q\bar{k}}^{(t)})$$

where $g_i = g_i(X_i^{(j)}; \Lambda^{(t)}, W^{(t)})$, $g_q = g_q(X_i^{(j)}; \Lambda^{(t)}, W^{(t)})$, $Y_{id}^{(j,t)} = \sum_{e=1}^{D_1} W_{ed}^{(t)} X_{ie}^{(j)}$, $v = \ell_i(X_i^{(j)})(1 - \ell_i(X_i^{(j)}))$, and the index $\bar{k}$ is defined as follows:

$$\bar{k} = \arg\min_k ||W^t X_i^{(j)} - m_{qk}||^2 \tag{11}$$

Furthermore, the index "t" in the superscript of the relevant variables in the above equations represents the cumulative number of training samples presented so far. One pass of the training samples is called an epoch. After the completion of each epoch, we need to randomize the ordering of $\{X_i^{(j)}\}$ again. Let's use N_{epoc} to denote the total number of epoches being performed. Then the following schedule is used for learning rate ϵ_t:

$$\epsilon_t = \epsilon_0 \left(1 - \frac{t}{N_{tr} \cdot N_{epoc}}\right)$$

where ϵ_0 is a control parameter need to be carefully determined by experiments. In practice, we also observed that using different learning rates for Λ and W respectively is helpful.

5. EXPERIMENTS AND RESULTS

In order to demonstrate the efficacy of the above techniques for Chinese OCR, a series of comparative experiments are conducted on the task of character classification. The vocabulary of the character classifier and the experimental setup for training and testing sets have been described in sections 1 and 3 respectively. As for Gabor feature extraction from a binary character image, after a series of comparative experiments, we set the relevant control parameters as follows: $N_1 = 7$, $\mathcal{M} = 4$, $\iota = 8$. Consequently, from each character image, we derive a 196-dimensional (i.e., $D_1 = 196$) feature vector X. For all of the experiments, we use 4 prototypes (i.e, $K_i = 4$) for each character.

In the first experiment, we set W to be an identity matrix, i.e., no feature transformation is applied, thus $D = D_1 = 196$.

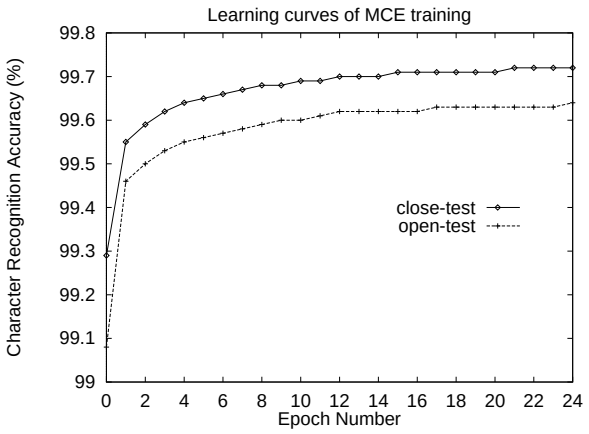

Fig. 1. Learning curves for MCE training: close-test and open-test recognition accuracies (%) as a function of Epoch number.

Table 1. A summary of character recognition accuracies (%) for several recognizers constructed with different methods.

Method	Close-test	Open-test
Baseline, $D = 196$	99.35	99.24
LDA (W), $D = 196$	99.48	99.33
LDA (W), $D = 48$	99.29	99.08
LDA (W), MCE (Λ)	99.72	99.64
MCE (W & Λ)	99.73	99.64

We use a k-means clustering method to estimate the set of prototype parameters $\{m_{ik}\}$ for each character class C_i. This recognizer achieves character classification accuracies of 99.35% on the training set (close-test) and 99.24% on the testing set (open-test) respectively. We treat this classifier as our baseline system.

In the second set of experiments, we estimate W by LDA. We still use the k-means clustering method to estimate $\{m_{ik}\}$'s. When $D = 196$ (i.e. no dimension reduction), the recognizer achieves accuracies of 99.48% on close-test and 99.33% on open-test respectively. This clearly shows the usefulness of the LDA for feature extraction. In order to reduce the number of parameters, we can use a smaller value for D. Even with $D = 48$, the recognizer performance degrades not so significantly with a close-test accuracy of 99.29% and an open-test accuracy of 99.08% respectively. So we use $D = 48$ in the remaining experiments.

In the third set of experiments, we use a fixed LDA-derived 196×48 transformation matrix for W. Then, we perform MCE training for estimating $\{m_{ik}\}$'s. The control parameters in Eq. (9) are set as $\alpha = 1.0$ and $\beta = 0$. Fig. 1 shows the learning curves for MCE training in terms of close-test and open-test recognition accuracies as a function of epoch number. The resultant recognizer can upgrade the open-test and close-test recognition accuracies to 99.72% and 99.64% respectively. The power of MCE training is clearly demonstrated here.

Finally, in the fourth set of experiments, we use MCE training for estimating both the transformation matrix W and prototype parameters $\{m_{ik}\}$'s. This recognizer achieves character recognition accuracies of 99.73% on close-test and 99.64% on open-test respectively. It is observed that in comparison with the case of the LDA-derived W and MCE-trained $\{m_{ik}\}$, the joint MCE training of W and $\{m_{ik}\}$ improves the performance slightly on the training set, but remains the same on the testing set. So, in our real system, we use the LDA-derived W and the MCE-trained $\{m_{ik}\}$'s. Table 1 summarizes the results of the above experiments.

6. SUMMARY

In this paper, we have described several key techniques we used to construct a large vocabulary highly accurate Chinese OCR engine. By further using fast match techniques for efficient character classification, and advanced coding techniques to represent economically our model parameters, our Chinese OCR engine can recognize the lines of Chinese characters at a speed of about 250 characters per second on a Pentium III-450MHz PC while only consumes less than 2MB memory and achieves a character recognition accuracy well above 99% for newspaper quality documents. We are currently improving our pre-processing techniques for skew detection and correction, page layout analysis, text orientation detection, and line segmentation. We are also studying how the above techniques work for a much more difficult task of the recognition of handwritten Chinese characters. We will report these results elsewhere.

7. REFERENCES

[1] Special Issue on Optical Character Recognition, *Proceedings of the IEEE*, Vol. 80, No. 7, 1992.

[2] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE Trans. on PAMI*, Vol. 22, pp.38-62, 2000.

[3] Special Issue on Oriental Character Recognition, *Pattern Recognition*, Vol. 30, No. 8, 1997.

[4] Special Issue on Advances in Oriental Document Analysis and Recognition Techniques, Part I, Part II, *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 12, Nos. 1-2, 1998.

[5] M. Lades et al., "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. on Computer*, Vol. 42, No. 3, pp.300-311, 1993.

[6] R. Duda and P. Hart, *Pattern Classification and Scene Analysis*, 1973.

[7] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. on Signal Processing*, Vol. 40, pp.3043-3054, 1992.

[8] A. Biem and S. Katagiri, "Feature extraction based on minimum classification error / generalized probabilistic descent method," *Proc. ICASSP-93*, 1993, pp.II-275-278.

[9] H. Watanabe, T. Yamaguchi, S. Katagiri, "Discriminative metric design for robust pattern recognition," *IEEE Trans. on Signal Processing*, Vol. 45, pp.2655-2662, 1997.

[10] K. K. Paliwal et al., "Simultaneous design of feature extractor and pattern classifier using the minimum classification error training algorithm," *Proc. NNSP-95*, 1995, pp.67-76.

[11] A. Sato and K. Yamada, "Generalized learning vector quantization," *Advances in Neural Information Processing 8*, pp.423-429, 1996.

[12] M.-K. Tasy, K.-H. Shyu, P.-C. Chang, "Feature transformation with generalized learning vector quantization for handwritten Chinese character recognition," *IEICE Trans. Information & Systems*, Vol. E82-D, No.3, pp.687-692, 1999.