

# Keyword Spotting from Online Chinese Handwritten Documents Using One-Vs-All Trained Character Classifier

Heng Zhang, Da-Han Wang, Cheng-Lin Liu  
National Laboratory of Pattern Recognition,  
Institute of Automation, Chinese Academy of Sciences  
95 Zhongguang East Road, Beijing 100190, P.R. China  
{hzhang07,dhwang, liucl}nlpr.ia.ac.cn

## Abstract

*This paper presents a text query-based method for keyword spotting from online Chinese handwritten documents. The similarity between a text word and handwriting is obtained by combining the character similarity scores given by a character classifier. To overcome the ambiguity of character segmentation, multiple candidates of character patterns are generated by over-segmentation, and sequences of candidate characters are matched with the query word in beam search. The character classifier is trained by one-vs-all strategy so that it gives high similarity to the target class and low scores to the others. Particularly, we use a one-vs-all trained prototype classifier and a support vector machine (SVM) classifier for similarity scoring. The method yielded promising performance in experiments on a database containing 550 pages of 110 writers. For words of four characters, the recall, precision and F measure are 87.25%, 94.84% and 90.88%, respectively.*

## 1. Introduction

The availability of large volume of document images and the popular use of the Web demand the technology for efficient document retrieval. OCR-based text search relies on the character recognition accuracy, and so, has been restricted to printed documents. Despite the many works of document retrieval without OCR, the retrieval of handwritten documents still needs research efforts to handle the difficulty of character/word segmentation and the variability of writing styles.

Keyword spotting from handwritten documents is a concern because keywords partially satisfy the interest of information search, and the located keywords form the basis of vector space representation for document clustering and classification.

According to the word similarity scoring technique,

keyword spotting methods can be categorized into two groups: word shape matching [1][2] and model-based scoring [3-6]. Both can be applied to text (keyboard) query and handwriting query. For text query, the system stores/synthesizes shape/feature templates or learns probabilistic models for each word. While for handwriting query, the input handwritten word is directly matched with the words in the document. Text query-based retrieval is more convenient for use, and by learning character/word models with multi-writer samples, can better tolerate the shape variation.

This paper is concerned with keyword spotting from online Chinese handwriting. With the wide use of digitizing tablets, tablet PCs and digital pens (such the Anoto Pen), online handwritten documents are produced constantly. This entails efficient retrieval techniques to exploit the semantic information in the documents. In addition to the ambiguity of character segmentation and the shape variation, Chinese documents suffer from the large alphabet (over 5,000 characters are daily used) and the difficulty of word segmentation (there is no extra space between words).

Most previous works of Chinese document retrieval have focused on printed documents. A few works have been performed on word retrieval from online Chinese handwriting [7] and character retrieval from calligraphic Chinese archives [8] using approximate shape matching.

We aim to retrieve keywords from multi-writer documents using text query. The method can apply to single writer's documents as well. To overcome the difficulties of character and word segmentation, we search the query word from character candidates generated by over-segmentation. The word similarity is obtained by combining character similarity scores output by a character classifier, which is trained to separate target and non-target characters.

Text search using character segmentation and recognition candidates has been proven effective in Japanese document retrieval [9-11]. These methods are

similar to ours, but we have paid more attention to the training of character classifier for a better similarity measure, which has been shown to be an importance issue [4-6]. Previous works have rarely considered similarity optimization in training stage, except [6]. To cope with the large alphabet of Chinese, we use a one-vs-all (OVA) trained prototype classifier. In experiments of keyword spotting, the OVA trained prototype classifier is shown to outperform a prototype classifier trained with multi-class objective and perform comparably with a support vector machine (SVM) classifier.

## 2. System Overview

Fig. 1 shows the diagram of our system for keyword spotting. The input document is first segmented into text lines according to the time and space interval between consecutive strokes (the pre-segmentation step of [12]). Each line is over-segmented into primitive segments and consecutive segments combine to generate candidate character patterns [13]. Examples of text line segmentation and over-segmentation are shown in Fig. 2, and an example of candidate pattern generation is shown in Fig. 3. In over-segmentation, we added some rules to split connected strokes.

In text search, the query word is matched with sequences of candidate patterns (paths in the candidate lattice) with every primitive segment as the start. The word similarity is obtained by combining the character similarity scores output by a character classifier. When the word similarity is greater than a threshold, a word instance is located in the document.

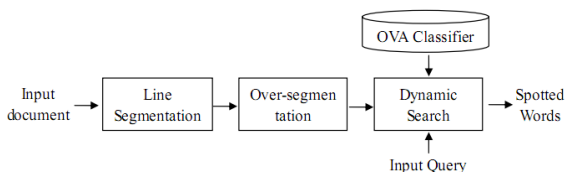


Figure 1. Block diagram of the keyword spotting system.

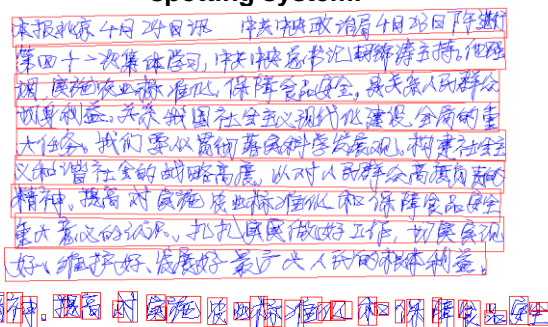


Figure 2. Text line segmentation and character over-segmentation of a line.

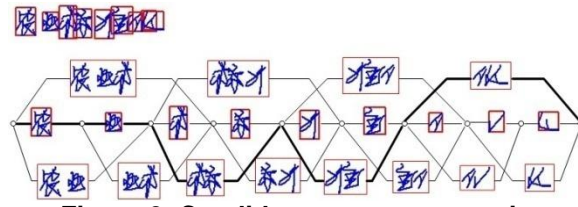


Figure 3. Candidate pattern generation.

## 3. Word Similarity Measure

In text search, the query word  $W = c_1 \cdots c_n$  is matched with each sequence of candidate patterns  $X = \mathbf{x}_1 \cdots \mathbf{x}_n$  (each pattern represented as a feature vector) in the document. The similarity measure is designed such that true words have high scores and imposters have low scores. A measure approximating the probability  $P(W|X)$  is preferred but to obtain an accurate estimate of probability is non-trivial. While this deserves close investigation in the future, we simply take the average of similarity (or distance) of characters:

$$Sim(W, X) = \frac{1}{n} \sum_{i=1}^n sim(c_i, \mathbf{x}_i) \quad (1)$$

where  $sim(c_i, \mathbf{x}_i)$  is the similarity between candidate pattern  $\mathbf{x}_i$  and character  $c_i$ , given by the classifier.

The character classifier is desired to give high similarity to the genuine class of the input pattern and low scores to all the other classes. On an input non-character pattern, the scores of all classes should be low. For large alphabet Chinese character recognition, the nearest prototype classifier is often used due to its good tradeoff between classification accuracy and complexity. We recently proposed an algorithm for training prototype classifier with one-vs-all objective such that the prototypes of each class perform as a binary classifier for separating the class from the others [14]. This property is desirable for retrieval. For comparison, we also use a one-vs-all SVM classifier and a prototype classifier trained with multi-class objective.

### 3.1. Prototype Classifiers

The nearest prototype classifier classifies the input pattern to the class of the nearest prototype under a distance metric (usually the Euclidean distance). Prototype learning methods based on learning vector quantization (LVQ) or empirical error minimization (such as minimum classification error (MCE)) have been demonstrated to give high classification accuracies. However, all these methods aim to optimize the multi-class accuracy. Recently, we

proposed a new prototype learning method which decomposes the multi-class problem into multiple binary classification problems and train the prototypes using a one-vs-all (OVA) criterion, the cross-entropy (CE) [14].

In the OVA formulation, each prototype  $\mathbf{m}_{ij}$  ( $j$ -th prototype of class  $i$ ) functions as a binary discriminant function, and the discriminant function of a class is the maximum over the prototypes:

$$f_i(\mathbf{x}) = \max_j f_{ij}(\mathbf{x}) = -\min_j (\|\mathbf{x} - \mathbf{m}_{ij}\|^2 - \tau_{ij}) \quad (2)$$

where  $\tau_{ij}$  is the threshold for prototype  $\mathbf{m}_{ij}$ . In training, the discriminant function is transformed to binary posterior probability using the logistic (sigmoidal) function:

$$p_i(\mathbf{x}) = \sigma[f_i(\mathbf{x})] \quad (3)$$

The training objective is to minimum the cross-entropy on a training dataset:

$$\min CE = -\sum_{i=1}^M \sum_{n=1}^N [y_i^n \log p_i + (1 - y_i^n) \log(1 - p_i)] \quad (4)$$

where  $M$  is the number of classes and  $N$  is the number of training samples.  $y_i^n = 1$  if the training sample  $\mathbf{x}^n$  belongs to class  $i$ , and 0 otherwise. Both the prototypes and their thresholds are optimized in training. Since the CE is a binary criterion, the classifier is trained to give high score to the genuine class of input pattern and low scores to the other classes.

Though the criterion (3) is class-decomposable, we train the prototypes of all classes simultaneously by stochastic gradient descent. For acceleration, on each training sample, we only update the parameters of the correct class and a few selected rival classes.

After training the OVA prototype classifier, we simply use the class discriminant function (2) as the similarity:  $sim(c_i, \mathbf{x}) = f_i(\mathbf{x})$ .

For comparison, we also trained a prototype classifier using a multi-class objective, the log-likelihood of margin (LOGM) [15], which was shown to perform superiorly in multi-class classification. By this method, the prototypes have no local thresholds, but we can simply set a global threshold to all prototypes.

### 3.2. SVM Classifier

Considering the large number of Chinese characters, we use a one-vs-all SVM classifier with linear kernel. The similarity of each class is a linear discriminant function:

$$f_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} - \tau_i, \quad (5)$$

The weight vector and the threshold of each class are calculated from support vectors selected by large

margin training. For the large number of classes and huge sample set, we use the successive overrelaxation (SOR) algorithm of [16] for SVM training.

### 3.3. Character Feature Extraction

For feature extraction of candidate character patterns, we use the local stroke direction histogram feature, which has been popularly used in both online and offline character recognition. Particularly, we adopt the implementation method of [17] for direction feature extraction from online characters. The coordinates of stroke trajectory are re-scaled to standard size using moment normalization. The line segments on the trajectory are decomposed into eight directions and mapped to eight direction planes of standard size. From each direction plane,  $8 \times 8$  feature values are extracted by Gaussian filtering. To reduce the complexity of classifier, the 512D feature vector is projected onto a 160D subspace learned by Fisher linear discriminant analysis (FLDA).

## 4. Word Spotting by Dynamic Search

After text line segmentation and character over-segmentation, the online handwritten document is represented as a sequence of primitive segments  $s_1 \cdots s_L$ . Each segment combines with its successors to form candidate patterns subject to constraints of maximum character width, width-to-height ratio and within-character gaps. The query word  $W = c_1 \cdots c_n$  is matched with subsequences of  $s_j \cdots s_{j+k-1}$  (partitioned into candidate patterns) with  $j=1, \dots, L-n+1$ . The word matching similarity is calculated by Eq. (1).

To accelerate text search, we use two thresholds to prune candidate matches: a character threshold  $T_c$  to prune those characters with  $sim(c_i, \mathbf{x}_i) < T_c$  and a word threshold  $T_w$  to prune those words with  $Sim(W, X) < T_w$ .

Text search can be performed in two modes: real-time matching mode and offline indexing mode. In real-time mode, the query characters are matched with the candidate patterns in real-time; While in offline mode, the similarity scores between each candidate pattern and multiple character classes (by setting a rather low threshold) are calculated offline and stored in an index file, and the search engine only has to compare the stored scores with thresholds. The former mode is suitable for retrieving a small number of documents generated online, and the latter mode is good for retrieving a large database indexed in advance. In terms of retrieval accuracy, the two modes have no difference. Except the online/offline calculation of

character similarity, the two modes also have the same search process.

We use a character-synchronous beam search algorithm for efficient search of query word from sequences of primitive segments. The algorithm is similar to that used in lexicon-driven character string recognition [18]. The search process repeats with every primitive segment as start. For a specific start segment  $s_j$ , the process is described in Algorithm 1, wherein a pair of matched character and candidate pattern is stored as a node in the state space.

#### Algorithm 1: Character-Synchronous Beam Search

**Input:** query word  $W = c_1 \cdots c_n$ , primitive segments from  $s_j$ .

**Output:** located word instance  $s_j \cdots s_{j+k-1}$ .

**Start:**  $i=1, k=j$ , root node (0,0) in OPEN.

**Step 1:** Match character  $c_i$  with all the candidate patterns  $\mathbf{x}$ , starting with  $s_k$ . Each pair  $(c_i, \mathbf{x}_i)$  satisfying

$$\text{sim}(c_i, \mathbf{x}_i) \geq T_c \quad \text{and} \quad \frac{1}{i} \sum_{p=1}^i \text{sim}(c_p, \mathbf{x}_p) \geq T_w \quad \text{is}$$

stored as an OPEN node.

**Step 2:** if OPEN is empty, go to End; otherwise, go to Step 3.

**Step 3:** label all the OPEN nodes as CLOSED and expand each of them.

For an OPEN node  $(c_i, \mathbf{x}_i)$ , with  $\mathbf{x}_i$  formed of primitive segments  $s_k \cdots s_{k+l-1}$ , if  $i=n$ , go to Step 4; else if  $k+l-1=L$ , go to End; otherwise, set  $i=i+1, k=k+l$ , go to Step 1.

**Step 4:** if there are multiple instances  $s_j \cdots s_{j+k-1}$  matched with the query, retain the one of maximum score.

**End**

Fig. 4 shows an illustrative example of dynamic search for spotting a word “农业标准化” from a sequence of primitive segments as shown in Fig. 3. The first character is matched with two candidate patterns, one of which is expanded to match with the second character, and another one is terminated. After matching the last character, two segmentation paths are matched, and the one with maximum word score is retained as the spotting result. After the beam search in Algorithm 1, some instances may have overlap shown in figure 5 and the one of lower score will be pruned. Fig. 6 shows that two instances of words are spotted for the query.

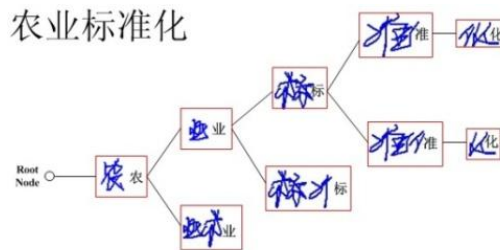


Figure 4. Dynamic search for retrieving a word “农业标准化”

农业标准化  
农业标准化

Figure 5. Two overlapping instances for retrieving a word “农业标准化”

本报北京4月20日讯 中共中央政治局4月20日下午进行第四十一次集体学习, 中共中央总书记胡锦涛主持。他强调, 实施农业标准化, 保障食品安全, 是关系人民群众切身利益, 关系我国社会主义现代化建设全局的重大任务。我们要以贯彻落实科学发展观, 构建社会主义和谐社会的战略高度, 以对人民群众高度负责的精神, 提高对实施农业标准化和保障食品安全重大意义的认识, 扎扎实实做好工作, 切实实现好、维护好、发展好最广大人民的根本利益。

Figure 6. Spotting results of word “农业标准化” in a document.

## 5. Experiment and Results

To evaluate the retrieval performance of the presented method, we tested with a database containing 550 handwritten pages of 110 writers. We trained three classifiers (one-vs-all prototype classifier, linear SVM classifier, and a prototype classifier trained with multi-class objective (LOGM)).

The dataset for classifier training contains 2,999,595 samples of 7,356 classes. This is a superset of the CASIA-OLHWDB1 database [19]. The 550 pages for retrieval were written by different persons from the training set. We selected the frequent words in these pages for retrieval, including 651 two-character word (occurring 27,342 times in total), 633 three-character words (occurring 9,162 times), and 617 four-character words (occurring 7,409 times).

The algorithms were implemented in C++ using Microsoft visual C++ 6.0 on PC.

## 5.1. Character Classification and Retrieval

On training the character classifiers, we evaluate their classification performance on the isolated characters segmented from the 550 test documents. The three classifiers are denoted by OVA-NPC (one-vs-all trained nearest prototype classifier), LOGM-NPC (NPC trained by LOGM) and SVM (one-vs-all linear SVM). The NPC has one prototype for each class. The test accuracies are given in Table 1. We can see that the NPC yields higher classification accuracies than the linear SVM, the two training methods of NPC yields comparable accuracies.

**Table 1. Character classification accuracy (%).**

	LOGM-NPC	OVA-NPC	SVM
Top rank	71.69	70.24	64.05
2 ranks	78.74	77.65	68.12
3 ranks	81.46	80.59	69.21

We then evaluate the character retrieval performance. By setting a threshold to the discriminant function of each class, the input pattern can be judged to belong to the class (accept, retrieved, detected) or not (reject, imposter). If a character pattern of class  $i$  is accepted by the discriminant function of class  $j$  ( $j \neq i$ ), it is a false positive of class  $j$ . If it is rejected by class  $i$ , it is a false negative of class  $i$ . We calculate the rates of recall, precision and the F-measure:

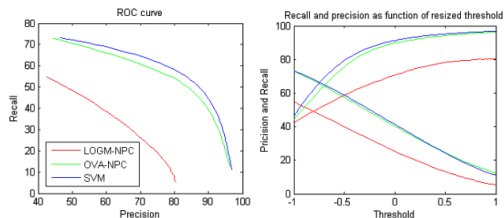
$$R = TP / (TP + FN), \quad (6)$$

$$P = TP / (TP + FP), \quad (7)$$

$$F = 2 / (1/R + 1/P), \quad (8)$$

where  $TP$ ,  $FN$  and  $FP$  are the numbers of true positives, false negatives and false positives, respectively.

By setting variable thresholds on re-scaled classifier outputs, we obtained the ROC curves of three classifiers as shown in Fig. 7. We can see the OVA classifiers (OVA-NPC and SVM) have much better retrieval performance than the LOGM-NPC, which was trained with a multi-class objective. With selected thresholds, the retrieval rates of three classifiers are shown in Table 2.



**Figure 7. Recall and precision rates of character retrieval.**

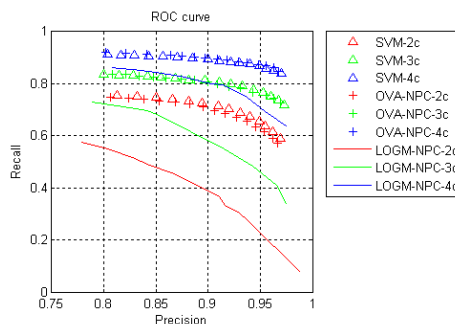
**Table 2. Character retrieval rates (%) of three classifiers.**

	LOGM-NPC	OVA-NPC	SVM
R	47.54	59.37	61.54
P	50.88	72.54	75.40
F	49.15	65.30	67.77

## 5.2. Word Spotting

In word spotting from the test documents, we set variable thresholds  $T_c$  and  $T_w$  to plot the ROC curves of three classifiers. For OVA-NPC and SVM, the threshold  $T_w$  was fixed at a low value and  $T_c$  was variable. For the LOGM-NPC, another way gave better ROC performance:  $T_c$  fixed and  $T_p$  variable. The word spotting ROC curves of three classifiers are shown in Fig. 8, where “-2C”, “-3C” and “-4C” means words of two, three and four characters, respectively.

From the results of Fig. 8, we have following observations: (1) One-vs-all trained classifiers are superior for retrieval purpose. Though the NPC trained with multi-class objective yields high classification accuracy, it is weak in rejecting negatives, and hence yield inferior retrieval performance. (2) When retrieving a word with more characters, the recall and precision rates can be higher. This is because the linguistic context in the word helps reduce the effect of noises and correct some matching errors/failures. Again, the performances of OVA-NPC and SVM are comparable.



**Figure 8. Recall and precision rates of word spotting.**

Selecting proper thresholds on re-scaled classifier outputs and word matching scores, we obtained the best spotting rates of three classifiers for words of different lengths as shown in Table 3, Table 4 and Table 5, respectively. For words of four characters, both the OVA-NPC and the SVM give F-measure higher than 90%. The recall rate is not high enough, because the documents were written cursively and many characters have very large shape variation. The performance should be improved by designing more powerful classifiers, training with non-character samples and integrating more contexts.

**Table 3. Word spotting results with OVA-NPC.**

length	$T_c$	$T_w$	R (%)	P (%)	F (%)
2	-9.5	-5.0	72.01	88.22	79.29
3	-14.0	-7.0	79.97	90.88	85.08
4	-18.0	-9.0	87.25	94.84	90.88

**Table 4. Word spotting results with SVM.**

length	$T_c$	$T_w$	R (%)	P (%)	F (%)
2	-1.90	-0.90	73.38	88.11	80.07
3	-2.60	-1.20	79.99	92.18	85.65
4	-3.40	-1.60	87.41	94.03	90.60

**Table 5. Word spotting results with LOGM-NPC**

length	$T_c$	$T_w$	R (%)	P (%)	F (%)
2	-5.00	-4.30	51.66	82.75	63.61
3	-6.20	-4.54	69.57	84.07	76.13
4	-6.50	-4.90	82.93	87.18	85.00

## 6. Conclusion

We presented a word spotting method for online Chinese handwritten documents employing the one-vs-all (OVA) trained character classifier. The query word is matched with sequences of candidate character patterns to locate the handwritten words with similarity greater than a threshold. The word similarity is obtained by combining the character similarity scores. Due to the merit of the OVA classifier that it gives high similarity to target class and low scores to the other classes, the proposed method has yielded high performance of word spotting in our experiments. Our future works aim to further improve the word matching similarity measure to make it better detect target words and reject imposters.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no. 60775004, 60825301 and 60933010.

## References

- [1] R. Manmatha, C. Han, E.M. Riseman, Word spotting: a new approach to indexing handwriting, *Proc. CVPR*, 1996, pp.631-637.
- [2] C.V. Jawahar, A. Balasubramanian, M. Meshesha, A.M. Namboodiri, Retrieval of online handwriting by synthesis and matching, *Pattern Recognition*, 42(7): 1445-1457, 2009.
- [3] T.M. Rath, R. Manmatha, V. Lavrenko, A search engine for historical manuscript images, *Proc. 27th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2004, pp.369-376.
- [4] H. Cao, A. Bhardwaj, V. Govindaraju, A probabilistic method for keyword retrieval in handwritten document images, *Pattern Recognition*, 42(12): 3374-3382, 2009.
- [5] J.A. Rodriguez-Serrano, F. Perronnin, Handwritten word-spotting using hidden Markov models and universal vocabularies, *Pattern Recognition*, 42(9): 2106-2116, 2009.
- [6] N.R. Howe, S. Feng, R. Manmatha, Finding words in alphabet soup: inference on freeform character recognition for historical scripts, *Pattern Recognition*, 42(12): 3338-3347, 2009.
- [7] D.P. Lopresti, M.Y. Ma, P.S.P. Wang, J.D. Crisman, Ink matching of cursive Chinese handwriting annotations, *Int. J. Pattern Recognition and Artificial Intelligence*, 12(1): 119-141, 1998.
- [8] Y. Zhuang, X. Zhang, J. Wu, X. Lu, Retrieval of Chinese calligraphic character image, *PCM 2004*, LNCS Vol.3331, Springer, pp.17-24.
- [9] S. Senda, M. Minoh, K. Ikeda, Document image retrieval system using character candidates generated by character recognition process, *Proc. 2nd ICDAR*, Tsukuba, Japan, 1993, pp.541-546.
- [10] H. Oda, A. Kitadai, M. Onuma, M. Nakagawa, A search method for on-line handwritten text employing writing-box-free handwriting recognition, *Proc. 9th IWFHR*, Tokyo, Japan, 2004, pp.157-162.
- [11] T. Nagasaki, T. Takahashi, K. Marukawa, Document retrieval system tolerant of segmentation errors of document images, *Proc. 9th IWFHR*, Tokyo, Japan, pp.280-285.
- [12] X.-D. Zhou, D.-H. Wang, C.-L. Liu, A robust approach to text line grouping in online handwritten Japanese documents, *Pattern Recognition*, 42(9): 2077-2088, 2009.
- [13] X.D. Zhou, J.L. Yu, C.L. Liu, T. Nagasaki, K. Marukawa, Online handwritten Japanese character string recognition incorporating geometric context, *Proc. 9th ICDAR*, Curitiba, Brazil, 2007, pp.23-26.
- [14] C.-L. Liu, One-vs-all training of prototype classifier for pattern classification and retrieval, submitted to *20th ICPR*, Istanbul, Turkey, 2010.
- [15] X. Jin, C.-L. Liu, X. Hou, Prototype learning by margin-based conditional log-likelihood loss, *Proc. 19th ICPR*, Tampa, USA, 2008.
- [16] O.L. Mangasarian, D.R. Musicant, Successive overrelaxation for support vector machines, *IEEE Trans. Neural Networks*, 10(5): 1032-1037, 1999.
- [17] C.-L. Liu, X.-D. Zhou, Online Japanese character recognition using trajectory-based normalization and direction feature extraction, *Proc. 10th IWFHR*, La Baule, France, 2006, pp.217-222.
- [18] C.-L. Liu, M. Koga, H. Fujisawa, Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(11): 1425-1437, 2002.
- [19] D.-H. Wang, C.-L. Liu, J.-L. Yu, X.-D. Zhou, CASIA-OLHWDB1: A database of online handwritten Chinese characters, *Proc. 10th ICDAR*, Barcelona, Spain, 2009, pp.1206-1210.