Handwritten Chinese Text Recognition by Integrating Multiple Contexts

[1]M. Sruthi, [2]GVNKV SUBBA RAO
[1,2]Sree Dattha Institute of Engineering and Science, Sheriguda, AP, India

Abstract
This paper presents an effective approach for the offline recognition of unconstrained handwritten Chinese texts. Under the general integrated segmentation-and-recognition framework with character over segmentation, we investigate three important issues: candidate path evaluation, path search, and parameter estimation. For path evaluation, we combine multiple contexts (character recognition scores, geometric and linguistic contexts) from the Bayesian decision view, and convert the classifier outputs to posterior probabilities via confidence transformation. In path search, we use a refined beam search algorithm to improve the search efficiency and, meanwhile, use a candidate character augmentation strategy to improve the recognition accuracy. The combining weights of the path evaluation function are optimized by supervised learning using a Maximum Character Accuracy criterion. We evaluated the recognition performance on a Chinese handwriting database CASIA-HWDB, which contains nearly four million character samples of 7,356 classes and 5,091 pages of unconstrained handwritten texts. The experimental results show that confidence transformation and combining multiple contexts improve the text line recognition performance significantly. On a test set of 1,015 handwritten pages, the proposed approach achieved character-level accurate rate of 90.75 percent and correct rate of 91.39 percent, which are superior by far to the best results reported in the literature.

Keywords
???? Is Missing ??????

Existing system
In the context of handwritten text (character string1) recognition, many works have contributed to the related issues of oversegmentation, character classification, confidence transformation, language model, geometric model, path evaluation and search, and parameter estimation. For oversegmentation, connected component analysis has been widely adopted, but the splitting of connected (touching) characters has been a concern. After generating candidate character patterns by combining consecutive primitive segments, each candidate pattern is classified using a classifier to assign similarity/dissimilarity scores to some character classes. Character classification involves character normalization, feature extraction, and classifier design. For classification of Chinese characters with large number of classes, the most popularly used classifiers are the modified quadratic Discriminant function (MQDF) and the nearest prototype classifier (NPC). The MQDF provides higher accuracy than the NPC but suffers from high expenses of storage and computation.

Proposed System
This system focuses on the recognition of text lines, which are assumed to have been segmented externally. For the convenience of academic research and benchmarking, the text lines in our database have been segmented and annotated at character level. First, the input text line image is over segmented into a sequence of primitive segments using the connected component-based method. Consecutive primitive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice. After that, each candidate pattern is classified to assign a number of candidate character classes, and all the candidate patterns in a candidate segmentation path generate a character candidate lattice.

I. Introduction
Handwritten Chinese text recognition (HCTR) is a challenging problem due to the large character set, the diversity of writing styles, the character segmentation difficulty, and the unconstrained language domain. Figure 1 shows an example of a Chinese handwritten page. The large set of Chinese characters (tens of thousands of classes) brings difficulties to efficient and effective recognition. The divergence of writing styles among different writers and in different geographic areas aggravates the confusion between different classes. Hand written text recognition is particularly difficult because the characters cannot be reliably segmented prior to character recognition. The difficulties of character segmentation originate from the variability of character size and position, character touching and overlapping. A text line of Chinese handwriting must be recognized as a whole because it cannot be trivially segmented into words (there is no more extra space between words than between characters). Last, handwritten text recognition is more difficult than bank check recognition and mail address reading because the lexical constraint is very weak: Under grammatical and semantic constraints, the number of sentence classes is infinite.

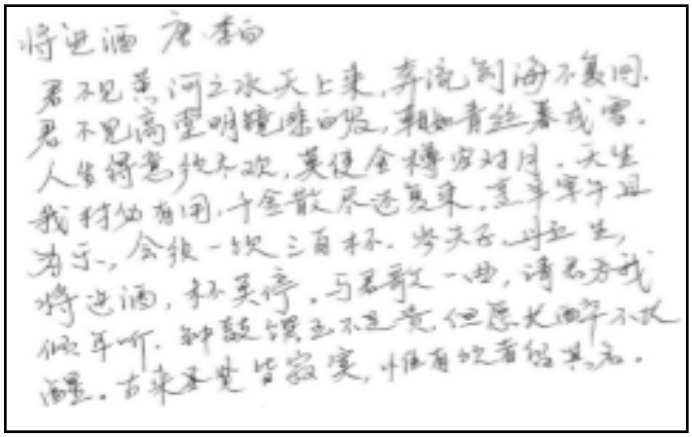

Fig. 1: A page Hand Written Chinese Text

Due to the large number of character classes and the infinite sentence classes of Chinese texts, HCTR can only be solved by segmentation-based approaches using character models [6], preferably by explicit segmentation, also called oversegmentation, which can take advantage of the character shape and overlapping and touching characteristics to better separate the characters at their boundaries. The result of oversegmentation is a sequence of primitive segments, each corresponding to a character or a part of a character, such that candidate characters can be generated by concatenating consecutive segments [1]. The candidate character sequences can be represented in a network called a candidate

lattice [7], and each candidate segmentation path in the lattice can be split into many segmentation- recognition paths by assigning character classes to the candidate characters. The result of character segmentation and recognition is obtained by evaluating the paths in the lattice and searching for the optimal path.

II. Related Work

In the context of handwritten text (character string1) recognition, many works have contributed to the related issues of oversegmentation, character classification, confidence transformation, language model, geometric model, path evaluation and search, and parameter estimation. For oversegmentation, connected component analysis has been widely adopted, but the splitting of connected (touching) characters has been a concern [1], [10], [11]. After generating candidate character patterns by combining consecutive primitive segments, each candidate pattern is classified using a classifier to assign similarity/dissimilarity scores to some character classes. Character classification involves character normalization, feature extraction, and classifier design. The state-of-the-art methods have been reviewed in [12-13]. For classification of Chinese characters with large number of classes, the most popularly used classifiers are the modified quadratic discriminant function (MQDF) [14] and the nearest prototype classifier (NPC) [15]. The MQDF provides higher accuracy than the NPC but suffers from high expenses of storage and computation.

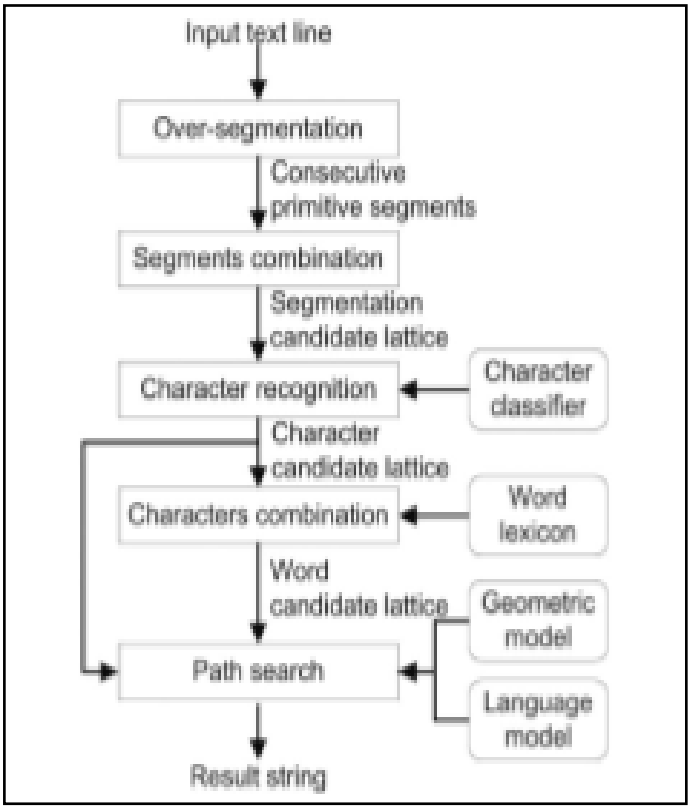

Fig. 2: Implementation of Text Line Recognition

A key issue in character string recognition is to design an objective function evaluating each candidate segmentation- recognition path. The path evaluation function is hoped to be insensitive to the path length (number of characters on the path). The summation of classifier output similarity/dissimilarity scores or product of class probabilities is not appropriate since this is biased to short paths. Normalizing the summation or product by the path length overcomes the bias problem, but this normalized form does not enable optimal path search by Dynamic Programming (DP). Beam search can be used instead, but does not guarantee optimality . Another way to overcome the path length bias is to add a compensative constant in the summated path evaluation function [8], but the constant needs to be estimated empirically. Wuthrich et al. called this constant a word insertion penalty, and Quiniou et al. also used this constant to control the deletion and insertion of words. Another effective way is to weight the character classification score with the number of primitive segments forming the character pattern [3, 5], motivated by the variable duration. This not only makes the number of summated terms in the path evaluation function equal the number of primitive segments (and thus independent of the path length), but also preserves the summation form and enables optimal path search by DP.

III. System Overview

This study focuses on the recognition of text lines, which are assumed to have been segmented externally. For the convenience of academic research and benchmarking, the text lines in our database have been segmented and annotated at character level [49]. Fig. 2 shows the block diagram of our system for text line recognition.

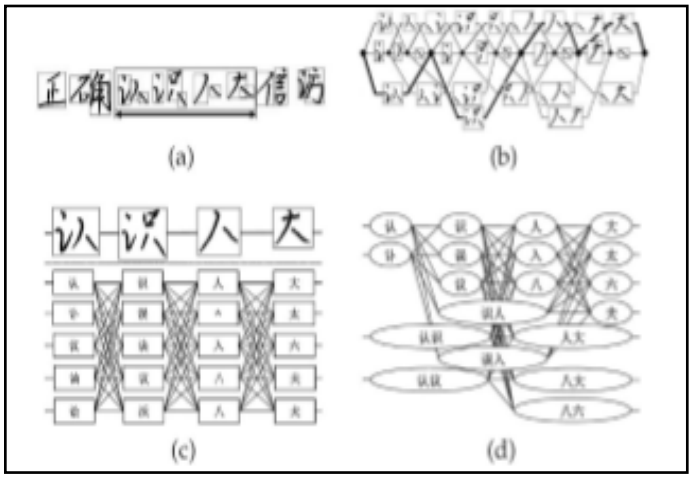

Fig. 3:(a) Oversegmentation to a sequence of primitive segments (each is bounded by a small box), (b) segmentation candidate lattice of the arrowed part of (a), (c) character candidate lattice of the thick path in (b), (d) word candidate lattice of (c).

First, the input text line image is over segmented into a sequence of primitive segments (Fig. 3(a)) using the connected component-based method [1]. Consecutive primitive segments are combined to generate candidate character patterns, forming a segmentation candidate lattice (Fig. 3(b)). After that, each candidate pattern is classified to assign a number of candidate character classes, and all the candidate patterns in a candidate segmentation path generate a character candidate lattice (Fig. 3(c)). If a word- level language model is used, each sequence of candidate characters is matched with a word lexicon to segment into candidate words, forming a word candidate lattice (Fig. 3(d)). All of these character (or word) candidate lattices are merged to construct the segmentation-recognition lattice of text line image. Each path in this lattice is constructed by a character sequence paired with a candidate pattern sequence, and this path is called a candidate segmentation-recognition path. Finally, the task of string recognition is to find the optimal path in this segmentation-recognition lattice. Considering that the text lines are segmented from text pages, we utilize the linguistic dependency between consecutive lines to improve the recognition accuracy by concatenating multiple

top-rank recognition results of the previous line to the current line for recognition.

IV. Bayesian Equation

We formulate the problem of handwritten Chinese text recognition from Bayesian decision view. According to Bayesiandecision underthe0/1loss,maximizingaposterior probability of character sequence (string class) C = <c1 ...cm> given a text line image X is the optimal criterion for recognition. This posterior probability is formulated by

$$P(C|X) = \sum_s P(C,s|X) = \sum_s P(s|X)P(C|s,X) = \sum_s P(s|X)P(C|X^s), \quad (1)$$

where s is the segmentation path index, P(s|X) denotes the posterior probability of the sth segmentation path given the text line image, and P(C|Xs) represents the posterior probability of string class given the sth segmentation path. P(s|X) is formulated by

$$P(s|X) = \prod_{i=1}^{m} p(z_i^p = 1|g_i^{ui}) p(z_i^g = 1|g_i^{bi}) \quad (2)$$

V. Path Evaluation Function

Note that all the terms m, ci, xi, g^{uc}, g^{bc}, gu^{ii}, gb^{ii}, z^p, z^g, h_i are related to the sth segmentation path, and the index s is dropped for simplification. However, the probability formulation (3) is still insufficient, because it does not consider the different contribution and reliability of differ- ent models (character recognition, geometric, and language models). In the following, we take the logarithm of probability and incorporate the weights of different models to get a generalized likelihood function $f(X^s,C)$ for the segmentation-recognition path evaluation:

$$C^* = \arg\max_{s,C} \frac{1}{P^m} \prod_{i=1}^{m} [p(c_i|\mathbf{x}_i) p(c_i|g_i^{uc}) p(c_{i-1}c_i|g_i^{bc}) p(z_i^p = 1|g_i^{ui}) p(z_i^g = 1|g_i^{bi}) p(c_i|h_i)]. \quad (3)$$

VI. Path Search

On defining a score for each path in the segmentation- recognition lattice, the next issue is how to efficiently find the path of maximum score. In addition, to alleviate the loss that the candidate classes assigned by character classifier do not contain the true class, we propose an augmentation technique to supplement candidate classes in the lattice.

A. Search Algorithm

If the segmentation-recognition path is evaluated by the accumulated score (WIP, WSN, and WCW), it satisfies the principle of optimality, and the optimal path with maximum score can be found by dynamic programming. Nevertheless, when binary or higher order contexts are used, the complexity of DP search is high. For the NPL function, which does not satisfy the principle of optimality, DP search does not guarantee finding the optimal path, and the beam search strategy can better find an approximately optimal solution. In beam search, it is critical to retain the correct partial path in fewer survived paths. A simple strategy of beam search is to retain the multiple top-rank partial paths ending at each primitive segment [6]. This simple strategy, though it works efficiently, is too rough, particularly when high-order context models are used. A refined beam search algorithm was presented in our previous work (called pruned DP there) [3], which is suitable for using high-order context models. after oversegmentation, the text line image is represented as a sequence of primitive segments. A candidate pattern composed of k consecutive segments and ending at the ith segment is denoted by (i,k). A node in the search space is represented as a quadruple SN = {CP;CC;AS;PN}, where SN denotes a search node, CP is a candidate pattern, CC is a candidate character of CP, and AS is the accumulated score from the root node (calculated by (11)-(14), where m is the length of the current partial path), and PNis a pointer to the parent node of SN. All nodes are stored in a list named LIST to backtrack the final path.

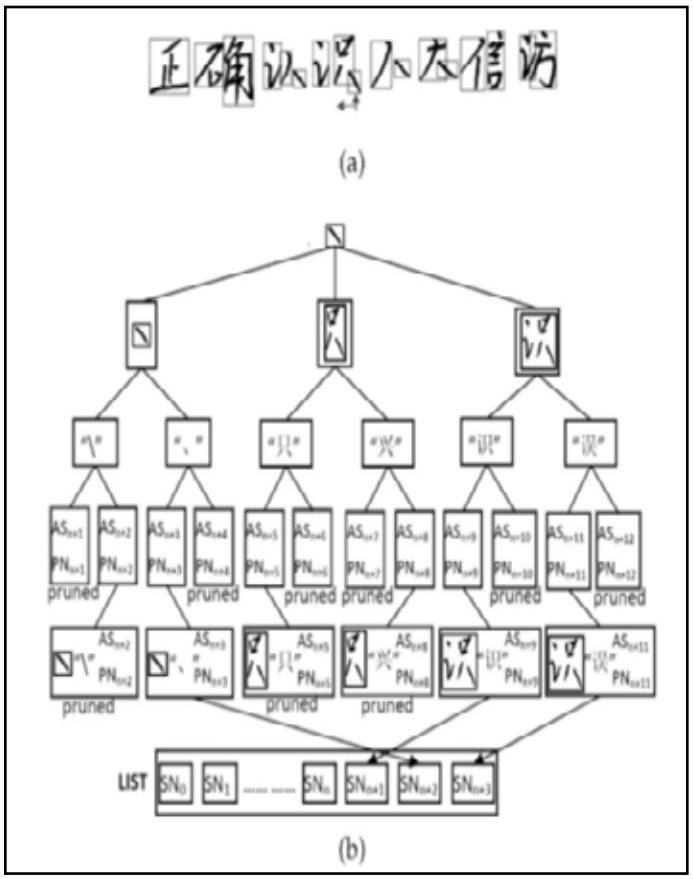

Fig. 4: An Illustrative Example of Refined Beam Search (K = 3, CN = 2, BW = 3) at a Primitive Segment. (a) A Sequence of Consecutive Primitive Segments (The Upward Arrow Points to Current Primitive Segment and the Leftward Arrow Points to the Direction of Segments Combination to Generate Candidate Patterns), (b) Search Space Expansion at the Pointed Primitive Segment of (a) (the Pruned Nodes are Labelled).

Refined Beam Search in frame-synchronous fashion:

1. Initialize the first search node (i.e., the root) of LIST, SN_0 = {null;null;0;null}, set i = 1.
2. Generate nodes of CP ¼ði;kÞ over k (the second level nodes in Figure. 4b, i > = k , k <= K, K is the maximum number of segments to be concatenated). For each CP, the top CN (Candidate Number) candidate characters are assigned by the character classifier (the third level nodes in fig. 4(b). In total, at most K CN nodes are generated.
3. Link to parent nodes for current nodes (CP = (i,k) , CC = c_{ik}). For multiple such parent nodes (CP^i = (i-$k_i$$k^j$, CC^1 = ci - $k_i$$k^j$), the current node generates multiple copies, each linked to a respective Parent Node (PN) and associated to an accumulated score (AS) (the fourth level nodes in fig. 4(b). In these copies, only the node with maximum AS over (k^j, c_i k^j) is retained (the fifth level nodes in fig. 4(b).

4. Sort the retained nodes in above in decreasing order according to AS over (k^j, c_i k^j), and the leading BW (Beam Width) nodes are retained and added to LIST, while the others are pruned to accelerate search.
5. Set i = i + 1, back to Step 2 and iterate until the last primitive segment is reached (such nodes called terminal nodes).
6. Backtrack the terminal node in LIST of maximum score along the element PN, and obtain the result character string.

B. Candidate Character Augmentation

The character classifier assigns a number of candidate classes to each candidate pattern with the risk of missing the true class. In Chinese handwriting recognition, even assigning hundreds of classes cannot guarantee 100 percent inclusion of the true class. Therefore, we propose a Candidate Character Augmentation (CCA) method, as diagrammed in fig. 5, to supplement candidate classes during search. The CCA method exploits both confusion information of the classifier and linguistic context. First, a candidate pattern xi is classified to assign a number of candidate classes, called the Original Candidate Set (OCS). Then, the confusion in formation and the linguistic context are used to supplement two types of candidate forming the Augmented Candidate Set (ACS). Last, the Extended Candidate Set (ECS), as the union of the OCS and the ACS, is used to generate candidate nodes at Step 2 of the search process.

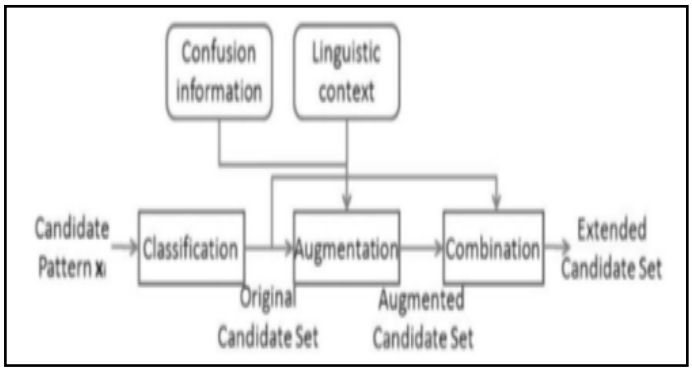

Fig. 5: Candidate Character Augmentation

VII. Experimental Results

We evaluated the performance of our approach on a large database of unconstrained Chinese handwriting, CASIA- HWDB [9], and on a small data set, HIT-MW .

A. Database and Experimental Setting

The CASIA-HWDB database contains both isolated characters and unconstrained handwritten texts, and is divided into a training set of 816 writers and a test set of 204 writers. The training set contains 3,118,477 isolated character samples of 7,356classes (7,185Chinesecharacters,109 frequently used symbols, 10 digits, and 52 English letters) and 4,076 pages of handwritten texts. The text pages have a few miswritten characters and characters beyond the 7,356 classes, which we call non characters and outlier characters, respectively. The characters in the training text pages (except for the non characters and outlier characters, 1,080,017 samples) were also segmented and used together with the isolated samples for training the character classifier. We evaluated the text line recognition performance on the 1,015 handwritten pages of 204 test writers, which were segmented into 10,449 text lines containing 268,629 characters (including 723 non characters and 368 outlier characters). To compare our results with those reported in the literature [2-4], we also tested on the data set HIT- MW , from which a test set of 383 text lines contains 8,448 characters (7,405 Chinese characters, 780 symbols, 230 digits, eight English letters, 16 non characters, and nine outlier characters). To build the character classifier, we extract features from gray-scale character images (background eliminated) using the normalization-cooperated gradient feature (NCGF) method . Before feature extraction, the gray levels of foreground pixels in each image are normalized to a standard mean and deviation. The 512D feature vector obtained is reduced to 160D by Fisher linear discriminant analysis (FLDA), and then input into a modified quadratic discriminant function classifier. The classifier parameters were learned on 4/5 samples of the training set, and the remaining 1/5 samples were used for confidence parameter estimation and confusion matrix construction. For parameter estimation of the geometric models, we extracted geometric features from 41,781 text lines of training text pages. The statistical language models were trained on a text corpus containing about 50 million characters (about 32 million words) [3]. On obtaining the context models, the combining weights of path evaluation function were learned on 300 training text pages. Table 1 shows some statistics of character samples segmented from the test text pages of CASIA-HWDB. The "number" row gives the numbers of different types of characters (including non characters and outlier characters). We can see that the majority of segmented characters are:

Table 1: Statistics of Character Types, Recognition, and Segmentation Correct Rates on the Test Set

	All	Chinese	symbol	digit	letter	non	outlier
number	268,629	233,329	26,583	6,879	747	723	368
rec (%)	83.78	87.28	60.34	69.40	77.24	0	0
rec20 (%)	98.24	98.55	99.36	98.90	97.86	0	0
rec200 (%)	99.18	99.58	99.64	99.40	98.93	0	0
seg (%)	95.54	95.69	96.84	86.97	83.53	94.05	92.66

Table 2: Recognition Results of Different Path Evaluation Functions

	AR (%)	CR (%)	ch (%)	sb (%)	dg (%)	lt (%)	time (h)
w/o	74.63	74.72	77.97	57.45	46.10	46.85	12.46
WIP	88.96	89.68	91.31	81.77	80.69	74.97	11.88
NPL	89.07	90.67	92.29	83.07	81.31	75.64	11.65
WSN	89.54	90.49	92.17	82.27	81.25	76.04	11.85
WCW	90.20	90.80	92.94	79.10	79.63	74.43	11.73

B. Text Line Recognition Results

We evaluated the effects of different techniques. First, we compared the effects of different path evaluation functions. Second, the effects of different confidence transformation methods, combinations of geometric models and language models were evaluated. Last, we show the results of different numbers of candidate character classes, beam widths, and candidate character augmentation methods in path search. We report the recognition rates of different techniques on the CASIA-HWDB test set, and give the processing time on all test pages (1,015 pages) consumed on a desktop computer of 2.66 GHz CPU, programming using Microsoft Visual C++. With several selected combinations of techniques, we also report results on the HIT-MW test set.

C. Comparing Path Evaluation Functions

In evaluating the effects of path evaluation functions and CT methods, the character trigram language model and all geometric models were used. The search algorithm was the refined beam search with K = 4, CN = 20, and BW = 10,but CCA methods were not used in the search process. In evaluating the path evaluation functions, the D-S evidence confidence was taken. The recognition results of different path evaluation functions (11)-(14) are shown in Table 2, where "w/o" denotes the path evaluation function without word insertion penalty ((11) removing the last term). We can see that by considering the balance of path length using different heuristics, the string recognition performance is largely improved. Among the four strategies, the one of weighting with character width performs best with respect to both AR and CR. The normalized path function gives a little lower CR but significantly lower AR. This is because NPL tends to generate more oversegmentation. The performance of weighting with primitive segment number is higher than that of NPL, but lower than that of WCW. We hence used the strategy WCW for all the following experiments.

VIII. Conclusion

This paper presented an approach for handwritten Chinese text recognition under the character oversegmentation and candidate path search framework. We evaluate the paths from the Bayesian decision view by combining multiple contexts, including the character classification scores, geometric and linguistic contexts. The combining weights of path evaluation function are optimized by a string recognition objective, namely, the Maximum Character Accuracy criterion. In path search, we use a refined beam search algorithm to improve the accuracy and efficiency. In experiments on the unconstrained Chinese handwriting database CASIA-HWDB, the proposed approach achieved the character-level accurate rate of 90.75 percent and correct rate of 91.39 percent. The experimental results justify the benefits of confidence transformation of classifier outputs, geometric context models, and language models. Nevertheless, the effect of candidate character augmentation is limited. We also evaluated performance on the HIW-MW test set and achieved an accuracy rate of 91.86 percent and correct rate of 92.72 percent, which are significantly higher than those reported in the literature.

References

[1] C.-L. Liu, M. Koga, H. Fujisawa,"Lexicon-Driven Segmentation and Recognition of Handwritten Character Strings for Japanese Address Reading", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 24, No. 11, pp. 1425-1437, Nov. 2002.

[2] T.-H. Su, T.-W. Zhang, D.-J. Guan, H.-J. Huang, "Off-Line Recognition of Realistic Chinese Handwriting Using Segmenta- tion-Free Strategy", Pattern Recognition, Vol. 42, No. 1, pp. 167-182, 2009.

[3] Q.-F. Wang, F. Yin, C.-L. Liu,"Integrating Language Model in Handwritten Chinese Text Recognition", Proc. 10th Int'l Conf. Document Analysis and Recognition, pp. 1036-1040, July 2009.

[4] N.-X. Li, L.-W. Jin,"A Bayesian-Based Probabilistic Model for Unconstrained Handwritten Offline Chinese Text Line Recognition", Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics, pp. 3664- 3668, 2010. Analysis and Recognition, pp. 521-525, July 2009.

[5] B. Zhu, X.-D. Zhou, C.-L. Liu, M. Nakagawa,"A Robust Model for On-Line Handwritten Japanese Text Recognition", Int'l J. Document Analysis and Recognition, vol. 13, no. 2, pp. 121-131, 2010.

[6] M. Cheriet, N. Kharma, C.-L. Liu, C.Y. Suen,"Character Recognition Systems: A Guide for Students and Practitioners", John Wiley & Sons, Inc., 2007.

[7] H. Murase,"Online Recognition of Free-Format Japanese Hand- writings", Proc. Ninth Int'l Conf. Pattern Recognition, pp. 1143-1147, 1988.

[8] S. Senda, K. Yamada,"A Maximum-Likelihood Approach to Segmentation-Based Recognition of Unconstrained Handwriting Text", Proc. Sixth Int'l Conf. Document Analysis and Recognition, pp. 184-188, Sept. 2001.

[9] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang,"CASIA Online and Offline Chinese Handwriting Databases", Proc. 11th Int'l Conf. Document Analysis and Recognition, pp. 37-41, Sept. 2011.

[10] L.Y. Tseng, R.C. Chen,"Segmenting Handwritten Chinese Characters Based on Heuristic Merging of Stroke Bounding Boxes and Dynamic Programming", Pattern Recognition Letters, Vol. 19, No. 10, pp. 963-973, Aug. 1998.

[11] Z. Liang, P. Shi,"A Metasynthetic Approach for Segmenting Handwritten Chinese Character Strings", Pattern Recognition Letters, Vol. 26, No. 10, pp. 1498-1511, July 2005.

[12] C.-L. Liu,"Handwritten Chinese Character Recognition: Effects of Shape Normalization and Feature Extraction", Proc. Conf. Arabic and Chinese Handwriting Recognition, S. Jaeger and D. Doermann, eds., pp. 104-128, 2008.

[13] C.-L. Liu, H. Fujisawa,"Classification and Learning in Character Recognition: Advances and Remaining Problems", Machine Learning in Document Analysis and Recognition, S. Marinai and H. Fujisawa, eds., pp. 139-161, Springer, 2008.

[14] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, "Modified Quadratic Discriminant Functions and the Application to Chinese Character Recognition", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 9, No. 1, pp. 149-153, Jan. 1987.

[15] C.-L. Liu, 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, Mar. 2001.

ISSN : 0976-8491 (Online) | ISSN : 2229-4333 (Print)

M Sruthi pursuing M.Tech (CSE) in Sree Dattha Institute of Engineering and Science, Sheriguda, RR Dist, AP, India.

GVNKV SUBBA RAO Pursuing PHD from JNTU Hyd. Received M.Tech Degree from JNTU Ananthapur and working as Vice Principal HOD of CSE in Sree Dattha Institute of Engineering and Science, Sheriguda, RR Dist, AP, India.