Improving Chinese/English OCR Performance by Using MCE-based Character-Pair Modeling and Negative Training *

Qiang HUO and Zhi-Dan FENG
Department of Computer Science and Information Systems,
The University of Hong Kong, Pokfulam Road, Hong Kong, China
(Email: qhuo@csis.hku.hk)

Abstract

In the past several years, we've been developing a high performance OCR engine for machine printed Chinese/English documents. We have reported previously (1) how to use character modeling techniques based on MCE (minimum classification error) training to achieve the high recognition accuracy, and (2) how to use confidence-guided progressive search and fast match techniques to achieve the high recognition efficiency. In this paper, we present two more techniques that help reduce search errors and improve the robustness of our character recognizer. They are (1) to use MCE-trained character-pair models to avoid error-prone character-level segmentation for some trouble cases, and (2) to perform a MCE-based negative training to improve the rejection capability of the recognition models on the hypothesized garbage images during recognition process. The efficacy of the proposed techniques is confirmed by experiments in a benchmark test.

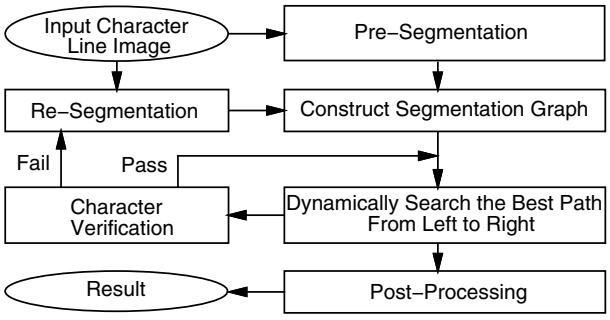

Figure 1. Architecture of Search Engine

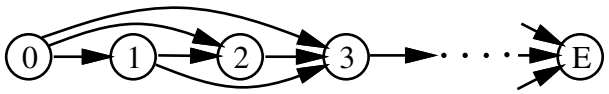

Figure 2. Segmentation Graph of a Text Line

1. Introduction

In the past several years, we've been developing a high performance OCR engine for machine printed Chinese/English documents [6, 4]. The overall architecture of our character line recognizer is shown in Figure 1 and works as follows.

Given the binary image of a horizontal input character line, a conservative pre-segmentation step is first performed to segment the character line into a sequence of blocks. From the pre-segmentation result, we can construct a segmentation graph *dynamically*, with each node representing a potential segmentation point, and each arc representing a hypothesized character candidate with an associated dissimilarity score, a confidence score for recognition result generated by the *character verification* module, and information for other top candidates. Figure 2 shows a schematic example of such a segmentation graph for a text line. For the arc with a confidence score below a pre-specified threshold, we need to re-segment, using an over-segmentation strategy, the part of the image associated with the arc that may consist of one or several blocks into a sequence of sub-blocks. Consequently, a new segmentation graph can be constructed dynamically and the above search process continues to the end of the character line. 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 final segmentation graph. The recognition result can be refined further through a post-processing module to resolve the problems that can not be solved by character classification only. In [6], we have described how we can use MCE-based character modeling techniques to provide the character candidates and the associated dissimilarity scores for each arc in the

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

Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)
0-7695-1960-1/03 $17.00 © 2003 IEEE

above search graph. Other details about pre-segmentation, re-segmentation, character verification, fast matching, and search algorithm are described in [4].

Although very promising results have been achieved as we reported in [4], a detailed error analysis of the recognition results in our benchmark test reveals the following problems:

Problem 1: Our re-segmentation algorithm is not good enough to deal with ligatures and some complicated touching character patterns in English scripts;

Problem 2: Some search errors are caused by a winning arc of the hypothesized garbage image.

To mitigate the above problems, we have developed two new MCE-based techniques, namely character-pair modeling and negative training using rubbish samples. In the following, we describe the details of these two new techniques and report the benchmark results of our updated OCR engine by using more training and testing data.

2. What's New

2.1. Using More Training Data

To construct our OCR engine, three character corpora are used. The first one is a Chinese character corpus constructed in our lab with in total $3,035,571$ character image samples from 6977 character classes that includes 6720 meaningful simplified Chinese characters in GB2312-80, 12 frequently used GBK Chinese characters, 62 alphanumeric characters, 183 punctuation marks and symbols. The second corpus is *NIST Special Database 8* [13]. We extracted $514,871$ plain (non-italic) ASCII character image samples from this corpus to enrich their coverage. The experimental results reported in [4] was based on the above two corpora by randomly choosing about 20% of character samples for each character class to form a testing set and the remaining samples to form a training set. In this study, we further enrich the coverage of ASCII character image samples by using the third corpus, namely, *UW English Document Image Database I* [3] . A total of $52,201$ character lines were extracted from the real images of document pages in UW-I database. We reserve randomly about 20% of character lines for testing ($10,864$ lines in total) and the remaining ones ($41,337$ lines in total) for training. From the above character lines, we extracted $1,638,974$ and $401,212$ isolated character image samples for training and testing respectively.

2.2. MCE-based Character-Pair Modeling

Our character recognizer is a multiple (16 for ASCII characters and 4 for non-ASCII characters in our experiments) prototype based nearest-neighbor classifier using Euclidean distance [6]. Given a hypothesized character image, it is normalized to a 40×40 image from which a 196-dimensional raw Gabor feature vector is first extracted and then transformed via LDA into a 48-dimensional feature vector. Theoretically speaking, this feature vector would be compared with all the prototypes of all the classes to identify the top N nearest character classes in terms of Euclidean distances. In [4], a practical fast match technique was presented for finding these top N candidates efficiently. The top 1 class label will be the recognition result used for the relevant arc in the search graph.

To mitigate the above-mentioned **Problem 1**, instead of developing a complicated over-segmentation algorithm as many researchers did, we adopt a simple approach of introducing some character-pairs as our recognition units as suggested in [12]. To determine the set of pairs to be used, we first construct a recognizer without using character-pair modeling. Then, we recognize all the character lines in the training set of UW-I database. After a detailed error analysis, we identify the following 44 character-pairs

```
Fl In Li Ll Th Ti al ca ci cl co
el fa fe ff fi fl fr ft in it li
ll ln ni nt oo ra rc re ri rj rl
rn ro rt ru rv th ti to tt ur vi
```

that are either difficult to segment or prone to recognition errors. Once we identify the above set of pairs, the corresponding image samples are automatically extracted from the character lines in UW-I training set. After several runs of manual screening and corrections, we obtained eventually a total of $208,751$ image samples for 44 character-pair classes. In this way, the number of classes of our *pattern classifier* becomes $6977 + 44 = 7021$. The character-pair models are then trained the same way as for the single character models by using the MCE training approach described in [6].

2.3. MCE-based Negative Training

The above mentioned **Problem 2** is well-known in character recognition community. Negative training using hypothesized rubbish image samples has been applied to neural network based character classifiers and its efficacy has been demonstrated in a number of studies, e.g., [1, 11, 5, 7, 9]. We are not aware of any study that uses MCE-based negative training for character recognition though. In the following, we describe how we can exploit this idea to improve the performance and robustness of our character recognizer.

Negative samples refer to rubbish images that are hypothesized during the search process and do not belong to any valid recognition unit. We used the following procedure to generate the set of negative samples:

Table 1. A summary of the number of relevant items in our benchmark testing set. The number in ( ) is the number of corresponding character classes observed in testing set.

Line Type	Lines	All Characters	Chinese	Alphanumeric	Symbols
Chinese	1107	16993 (1766)	14318 (1692)	800 (49)	1875 (25)
Our English	400	21970 (90)	0 (0)	21245 (60)	725 (30)
UW-I English	1825	75226 (120)	0 (0)	71700 (62)	3526 (58)
Mixed Chinese/English	435	16219 (884)	6201 (774)	8259 (62)	1759 (48)

(1) For each line of character images,

- Do a force-alignment using the previously trained models and the given text transcription to generate the pseudo ground-truth of character-level segmentation;
- Recognize the character line as usual:
  - for each arc on the best path, if it is mismatched with the above pseudo ground-truth, yet it is not a valid character-pair, the image associated with this arc is selected as a potential negative sample,
  - for each remaining arc not on the best path, if it passes the threshold in character verification module, and neither be it one of the characters in a valid character-pair nor be it a valid character-pair itself, the image associated with this arc is also selected as a potential negative sample;

(2) After the set of potential negative samples are collected, they will be screened manually to remove the possible correct samples of the valid recognition units.

In this way, we generated a total of $166,237$ negative training samples from the character lines in the training sets of both Chinese corpus and UW-I database. We then treat these negative training samples as belonging to a *dummy class* with a large number of prototypes. Therefore, the same MCE training approach as described in [6] can be used to train the prototype parameters for both valid recognition units and the dummy class. After the completion of MCE training, the dummy class will be ignored and only models of valid recognition units will be used to construct our character recognizer. It is expected that such trained models will have a better capability to reject the similar hypothesized rubbish image patterns during the recognition of an unknown character line. Please note that the dummy class and negative training samples are not used in estimating the LDA transformation.

3. Experiments and Results

3.1. Benchmark Testing Set

In order to verify the efficacy of the above techniques for Chinese/English OCR, a series of comparative experiments are conducted. To form a testing set, we collected 1107 Chinese, 400 English, and 435 mixed Chinese/English character lines from varied sources such as newspapers, magazines, journals, books, etc. To enrich the English part, we further chose randomly another 1825 English character lines from the testing set of UW-I database. The detailed statistics of the benchmark testing set are summarized in Table 1. Every time we construct a recognizer, we will perform a benchmark test using the above data set. OCR performance is measured using the *Percentage Accuracy* defined as

$$(1-\frac{substitutions+deletions+insertions}{total\ number\ of\ characters\ in\ reference})\times 100\%\,,$$

where the reference is ground truth.

3.2. A Comparison of Character Recognition Accuracies of Three Recognition Systems

The first recognizer we constructed is our baseline system. For this system, each recognition unit is a single character, thus the vocabulary of our recognition system includes 6977 characters that is the same as the number of internal recognition units. We used a total of $4,472,910$ isolated single-character image samples for MCE training as described in [6].

The second recognizer was constructed by adding 44 character-pair models into the inventory of our recognition units (in total $6977+44=7021$). Consequently, we used a total of $4,472,910+208,751=4,681,661$ image samples for MCE training of both single-character and character-pair models.

The third recognizer was constructed by using the MCE-based negative training as described previously. During MCE training, the parameters of 6977 single-character models, 44 character-pair models, and 1 dummy-class

Table 2. A comparison of character recognition accuracies (%) for three recognition systems tested on different subsets of character lines.

Test Subset	Systems	Subs.	Del.	Ins.	% Accuracy	R.E.R.(%)
Chinese	Baseline	57	18	4	99.54	N/A
1107 lines	+ Character-Pair	68	17	4	99.48	-13.0
16993 characters	+ Negative Training	55	8	3	99.61	15.2
Our English	Baseline	70	10	37	99.47	N/A
400 lines	+ Character-Pair	55	16	9	99.64	32.1
21970 characters	+ Negative Training	49	9	8	99.70	43.4
UW-I English	Baseline	753	155	355	98.32	N/A
1825 lines	+ Character-Pair	747	180	211	98.49	10.1
75226 characters	+ Negative Training	671	170	217	98.59	16.1
Mixed Chinese/English	Baseline	93	15	19	99.22	N/A
435 lines	+ Character-Pair	84	16	17	99.28	7.7
16219 characters	+ Negative Training	75	14	12	99.38	20.5

model are adjusted. The number of prototypes for dummy-class model is 1000. The total amount of training data is $4,681,661 + 166,237 = 4,847,898$ image samples.

After we constructed the above three recognition systems, we performed a benchmark test and the result is summarized in Table 2. In this table, the rightest column (labeled as "R.E.R.") indicates the relative error reduction (in %) of the corresponding systems in comparison with that of the baseline system. From Table 2, we observed that the use of character-pair modeling technique is beneficial to all the testing cases except for Chinese character lines. After further using the negative training technique, we observed the performance improvement for all the testing cases, with a relative error reduction ranging from 15.2% to 43.4%. This clearly demonstrates the power and usefulness of the negative training technique. A much lower recognition accuracy on UW-I English lines can be explained by the fact of that the image quality of character lines in UW-I database is much more diversified than that of English character lines we collected.

3.3. A Comparison of Recognition Speed

In contrast with our approach, a more traditional approach to solve the mixed Chinese/English OCR problem is to first segment a character line image into small segments, each being a character or part of a character, and then to search through a segmentation graph constructed from the above segmentation points for deriving the recognition result (e.g., [2, 8]). In order to cope with the problem of touched and/or overlapped characters, especially for English part, an over-segmentation strategy is typically used. This makes the segmentation graph unnecessarily dense thus leads to a less efficient search process. In order to make a good sense of the difference between these two approaches in terms of recognition speed, we also implemented a recognizer based on the traditional over-segmentation strategy. For both systems, we used 7021 recognition units including character-pairs, that were trained using the above MCE-based negative training technique. We then performed a benchmark test for each system on a Pentium III 733 MHz PC running Windows 98SE OS. For different benchmark tests, we made sure the same number of applications were running in background so that the measured recognition time in different sessions can be fairly compared.

Table 3 compares the recognition time of our system (labeled as "Verification-based") and the one using the traditional over-segmentation strategy (labeled as "Over-segmentation"), along with the comparison of recognition accuracies. The "Total Time" in Table 3 refers to the response time of recognizing the amount of testing character lines in respective benchmark test subsets. In terms of recognition accuracy, it is observed that our verification-based recognition system performs better than the traditional over-segmentation-based system except for the case of UW-I English subset. In terms of recognition speed, our system can recognize $134 \sim 204$ characters per second while the traditional system can only recognize $48 \sim 70$ characters per second. Our system is much more efficient in this regard.

As a remark, in another round of benchmark test for systems without using negative training, we observed that our verification-based recognition system performs much better than the over-segmentation-based system in all the cases. This in turn indicates that the negative training truly helps improve the rejection capability of the trained models such that the recognition accuracy of the over-segmentation-based system can be improved even without using an explicit character verification module.

Table 3. A comparison of recognition accuracies (%) and speed (in second) for two recognition systems tested on different subsets of character lines.

Test Subset	Systems	Subs.	Del.	Ins.	% Accuracy	Total Time (sec.)	Chars/s
Chinese	Verification-based	55	8	3	99.61	83.1	204
16993 characters	Over-segmentation	101	1	47	99.12	355.5	48
Our English	Verification-based	49	9	8	99.70	157.7	139
21970 characters	Over-segmentation	226	20	93	98.46	327.2	67
UW-I English	Verification-based	671	170	217	98.59	559.8	134
75226 characters	Over-segmentation	638	188	186	98.65	1073.8	70
Mixed Chinese/English	Verification-based	75	14	12	99.38	107.4	151
16219 characters	Over-segmentation	198	19	99	98.05	303.1	54

4. Discussions and Conclusion

In this paper, we have described two techniques that help reduce search errors and improve the robustness of our character recognizer. They are (1) to use MCE-trained character-pair models to avoid error-prone character-level segmentation for some trouble cases, and (2) to perform a MCE-based negative training to improve the rejection capability of the recognition models on the hypothesized garbage images during recognition process. The efficacy of the proposed techniques is confirmed by experiments in a benchmark test. As a final remark, we want to emphasize that all of the reported benchmark results are obtained by using a prototype system with a very compact implementation that requires less than 3MB ROM for storage and 150KB RAM for execution under the assumption of recognizing a line of characters with a binary image of $100(heigh) \times 1000(width)$ pixels. Our OCR engine is ready for being ported to those embedded systems with very limited computational resources and storage capacities.

References

[1] J. Bromley and J. S. Denker, "Improving rejection performance on handwritten digits by training with rubbish," *Neural Computation*, Vol. 5, pp.367-370, 1993.

[2] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in Character segmentation," *IEEE Trans. on PAMI*, Vol. 18, No. 7, pp.690-760, 1996.

[3] S. Chen, M.Y. Jaisimha, J. Ha, R.M. Haralick, and I.T. Phillips, "Reference Manual for UW English Document Image Database I," University of Washington, August 1993.

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

[5] P. D. Gader, M. Mohamed, and J.-H. Chiang, "Handwritten word recognition with character and inter-character neural networks," *IEEE Trans. on SMC, Part B: Cybernetics*, Vol.27, No.1, pp.158-164, 1997.

[6] Q. Huo, Y. Ge, and Z.-D. Feng, "High performance Chinese OCR based on Gabor features, discriminative feature extraction and model training," in *Proc. ICASSP-2001*, May 2001, pp.III-1517-1520.

[7] H.-Y. Kim, K.-T. Lim, and Y.-S. Nam, "Handwritten numeral string recognition using neural network classifier trained with negative data," in *Proc. IWFHR-2002*, 2002, pp.395-400.

[8] S.-W. Lee, D.-J. Lee and H.-S. Park, "A new methodology for gray-scale character segmentation and recognition," *IEEE Trans. on PAMI*, Vol. 18, No. 10, pp.1045-1050, 1996.

[9] C.-L. Liu, H. Sako, and H. Fujisawa, "Integrated segmentation and recognition of handwritten numerals: comparison of classification algorithms," in *Proc. IWFHR-2002*, 2002, pp.303-308.

[10] J.-H. Liu and P. Gader, "Neural networks with enhanced outlier rejection ability for off-line handwritten word recognition," *Pattern Recognition*, Vol. 35, pp.2061-2071, 2002.

[11] R. F. Lyon and L. S. Yaeger, "On-line handprinting recognition with neural networks," in *Proc. MicroNeuro-1996*, 1996, pp.201-212.

[12] G. Nagy, T. A. Nartker, and S. V. Rice, "Optical character recognition: an illustrated guide to the frontier," in *Proc. SPIE Conference on Document Recognition and Retrieval VII*, SPIE Vol. 3967, 2000, pp.58-69.

[13] R. A. Wilkinson, "NIST Special Database 8: Machine Print Database," Manual, NIST, October 1, 1992.