

Recognition Method for Cursive Japanese Word Written in Latin Characters

Kenichi Maruyama and Yasuaki Nakano
Dept. of Information Engineering, Shinshu University
4-17-1 Wakasato, Nagano, 380-8553, Japan
E-mail: {zmaru, nakano}@cs.shinshu-u.ac.jp

This paper proposes a recognition method for cursive Japanese words written in Latin characters. The method integrates multiple classifiers using duplicated candidates in multiple classifiers and orders of classifiers to improve the word recognition rate combining their results. In experiments using two classifiers, the word recognition rate was 68.4%, and the cumulative recognition rate among the ten best candidates was 92.5%.

1 Introduction

In Europe and America, cursive handwritten word recognition has proceeded on a large scale because it is an actual problem in regard to the recognition of postal addresses, checks and other handwritten items^{[1][2]}. In Japan, this market has been considered very small, but recently the recognition of postal addresses on mail from foreign countries has become a real problem.

In the general method of word recognition, strings that are combined from the results of the character recognition of patterns formed from a segmented word image are matched to each entry of a lexicon. Thus, in order to improve word recognition rate, character recognition rate must first be improved.

Recently, it was reported that multiple classifier integration is effective for improving word recognition rates^{[3][4]}. In the integration of multiple classifiers, different measures are generally used, so it is difficult to use the measures equally. But the orders in results of recognition behave rather equally in many recognition methods, so they can be used in the integration. In addition, a candidate which is duplicated in multiple classifiers is considered to be very probable.

In this paper, we propose a method to integrate multiple classifiers using duplicated candidates and orders. To evaluate the effectiveness of the proposed method, we tested the integration of two character classifiers, i.e., pattern matching based on directional features and HMM.

2 Cursive Word Samples

The cursive Japanese words written in Latin characters, such as shown in Figure 1, are investigated. These words were written by 4 subjects using felt-

tip pens. The number of words totaled 1,971, and they included cities, towns, villages in Nagano area and all the prefectures in Japan.

A sample of the word "Tokyo" written in a cursive, handwritten style.

Figure 1: Cursive word sample (Tokyo)

3 Approach and target

The system shown in Figure 2 is used in this research and is based on Yamada's work^[5].

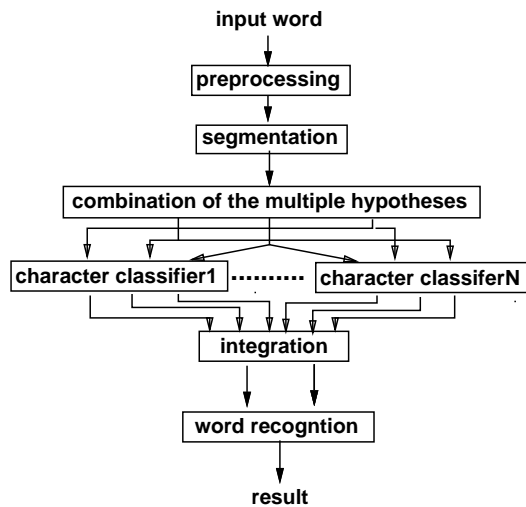


Figure 2: The flowchart of the proposed method

While Yamada used only one classifier at the character recognition level, our research is unique in that it integrates multiple classifiers.

3.1 Preprocessing

An input word image is binarized and normalized in the inclination. Then the contour is extracted for further analysis.

3.2 Segmentation

A cursive word is considered as the combination of character bodies and ligatures which combine character bodies. In our methods, candidates for segmentation points are estimated on the left and the right endpoints of character-like shapes (possibly character bodies) by contour analysis. Thus, a ligature which combines two candidates is extracted as an image fragment, as well as each character body. Since there is a possibility that a ligature could be divided into two, a middle point of the estimated ligature is added as a candidate for the segmentation point (Figure 3). A ligature is a thin pattern that has a horizontal direction.

3.3 Combination and Multiple Hypotheses

Neighboring pattern fragments segmented as candidates are combined to make a hypothesis for a character pattern. Combined patterns are assembled into a group.

A group is a set of pattern fragments which are combined around a major pattern other than a ligature. In Figure 3, the solid patterns show an example of major patterns.

Each combined pattern is given a number, called a pattern number. The pattern number is used at the character recognition and the word recognition stages. These numbers basically show which major fragment is used as the kernel of the combination.

All combinations of pattern fragments make segmentation hypotheses, as explained below. The most probable hypothesis is selected at the recognition and word matching stages.

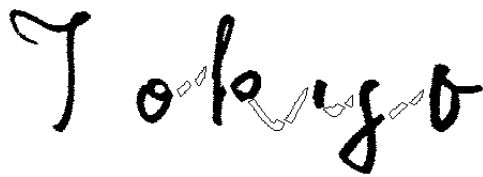


Figure 3: Segmented pattern fragments (a pattern painted in black shows a major one)

3.4 Target of the investigation

The target of the investigation consists of Japanese addresses written in Latin characters. Latin characters are classified into 52 classes (upper and lowercase).

4 Character Recognition

In this paper, two classifiers are used: pattern matching based on directional features and HMM. Details are stated in Sections 6 and 7.

Each method outputs three candidates for combination hypothesis, and the results belonging to the same group are merged. If the classes for the different patterns in the same group coincide, the result having the largest measure is adopted.

The word recognition stage uses the result obtained by integrating two classifiers. The word recognition stage does not distinguish uppercase and lowercase characters.

5 Word recognition

Word recognition uses the results of character recognition. For a set of segmentation points, a penalty measure is calculated for each word in a lexicon. Figure 4 shows an example of multiple hypotheses of segmentation with the recognition results. For the sake of simplicity, many results are not shown. The string (T, 1) in the rectangle shows that T is the first rank for the combination hypothesis. The penalty is the average of the rank of each rectangle.

For the word image in the example of Figure 4, the penalty of “TOKYO” is $(1+1+1+1+1)/5 = 1.0$, combining (T, 1)(O, 1)(K, 1)(Y, 1)(O, 1). This penalty is lower than those of any other words in the lexicon. Thus “TOKYO” holds the first rank in word recognition. In the same way, ten candidates are outputted and sorted in ascending order of penalties.

6 Pattern Matching Based on Directional Features

Pattern matching based on directional features (hereafter abbreviated as pattern matching) is known as an effective method in handwritten kanji recognition^[6].

The size of input pattern is normalized to 64×64 pixels, and inclination is corrected. The normalized pattern is partitioned into 8×8 blocks. Four patterns emphasizing four directions (vertical, horizontal, left slant, right slant) at every block are formed. The similarity of a pattern is the average of similarities in four directional patterns. The similarities are calculated for templates of all classes, and the best candidate classes are selected.

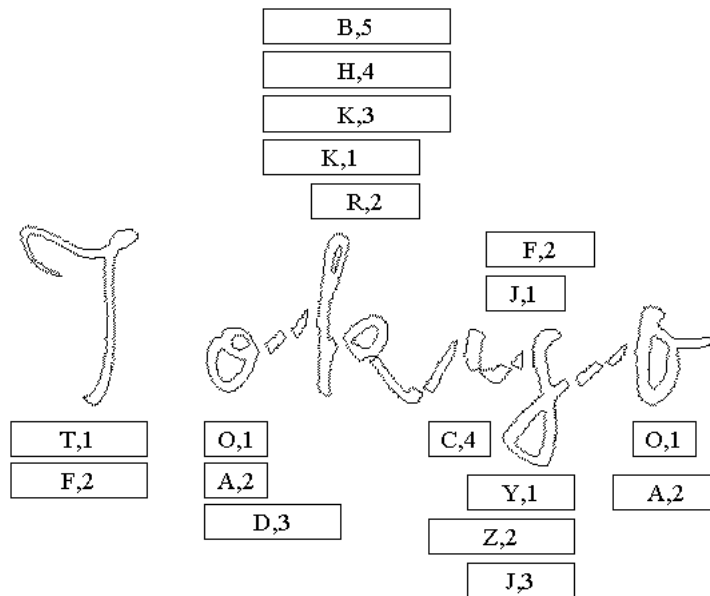


Figure 4: Example of word recognition

Three candidates are outputted in descending order of similarities for each segmented pattern. Candidates for patterns generated from the same group are merged and sorted in descending order of similarities.

6.1 Templates

Templates used in the matching are generated from averaging feature of learning samples. The total number of templates for 52 classes (Latin characters, lower and uppercase) is 5,200. The size of the learning sample set is 41,776. Uppercase and lowercase classes are merged after recognition.

7 Hidden Markov Models

We adopted 1-dimensional HMM, which is mature in speech recognition^[7]. The size of the input pattern is normalized to 16×16 pixels, and the inclination is normalized.

7.1 Feature Extraction

Features used in this research are shown in Figure 5.

By scanning the normalized pattern vertically from top to bottom, four features— f_1, f_2, f_3 and f_4 —are extracted. The first y ordinate of the black pixel is f_1 , and the run length of black pixels is f_2 . The second y ordinate of the black pixel is f_3 , and the run length of black pixels is f_4 . Next, from bottom to top, a similar operation extracts the other four features: f_5, f_6, f_7 and f_8 . The eight-dimensional vector (f_1, f_2, \dots, f_8) is used as the feature vector. Since the operation is repeated on each pixel on the abscissa, 16 vectors are generated.

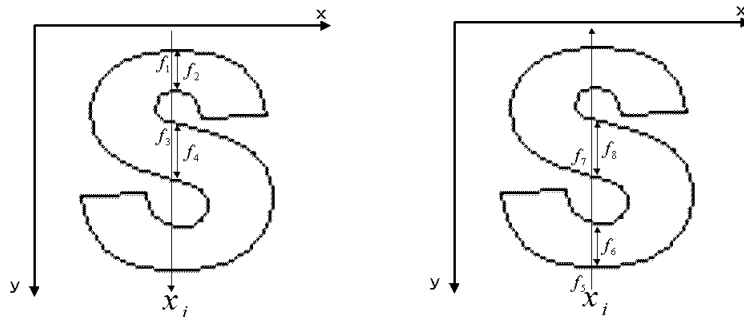


Figure 5: Feature extraction for HMM

7.2 Learning

An HMM is constructed for each character class. Therefore, 52 HMMs are constructed corresponding to A–Z and a–z. Though there are lower and uppercases, the recognition results are merged after recognition. Thus there are 26 truly different classes.

7.3 Recognition

Using HMMs constructed by learning, recognition is executed. For an input pattern, probability by each HMM is calculated. The recognition result is the class corresponding to the highest probability outputted from the HMMs.

Three candidates are outputted for each segmented pattern, merged, and then sorted in descending order of probabilities in the same group.

8 Integrating Two Classifiers

Results obtained by the pattern matching classifier and the HMM classifier are integrated. In this integration, only the orders in results of recognition are used.

It might be better that word recognition stage uses only reliable candidates and the number of them is smaller. In the case using multiple classifiers, it is considered that the candidates which have high measures in each method will be highly reliable. Of course, there is possibility that the candidates which have high measures are wrong. However, it is very probable that the candidates that both classifiers support will have high reliability. The candidates which have higher reliability might be obtained by adding measure of each recognition measure to these duplicated candidates. But, if we take duplicated candidates only, the number of candidates might be limited. The following procedure is proposed for such a case.

First, whether the same characters are duplicated in the results of the two classifiers is determined. For this purpose, a candidate is given the following penalty:

$$\frac{\textit{sum_candidate} - \textit{rank_pattern} - \textit{rank_hmm}}{\textit{sum_candidate}}$$

where the *sum_candidate* denotes the sum of orders of the candidates in the results of the two classifiers, *rank_pattern* is the order in the pattern matching and *rank_hmm* is that of the HMM candidate. These candidates are called “A” candidates.

Next, it is determined whether the candidates includes the characters A, C, F, G, H, I, J, K, S, T or Y. Though duplicated candidates in the two classifiers are considered to have high reliability, characters listed above are too exaggeratedly evaluated by this condition only. These characters are found from the observation of experimental results. Such a candidate is given the following penalty:

$$\frac{\textit{sum_candidate} - \textit{rank} - \textit{weight}}{\textit{sum_candidate}}$$

where *rank* denotes the order of the candidate (the pattern matching or the HMM), and *weight* is the maximum of the sum of duplicated candidates (the rank of the pattern matching + the rank of the HMM) /2. These candidates are called “B” candidates.

Finally, these candidates (“A” and “B”) are sorted in descending order of penalties. When these candidates have the same penalties, the priority is decided as follows: duplicated candidates first, pattern matching candidates second and HMM candidates third.

9 Result of Word Recognition

Table 1 shows the word recognition rate by integrating the pattern matching and HMM classifiers for the learning set as well as the rate obtained using the isolate method.

Table 2 shows the word recognition rate using the same conditions as above for the test set.

In Table 1, the total number of the words in the learning set is 659, and the lexicon size is 166. In Table 2, the size of the total number in the test set is 1,312, and the lexicon size is 166.

In Table 1, 2 and 3, the cumulative recognition rate within best N candidates shows the percentage that the correct word is included in the top N candidates.

Table 1: Word recognition rate for the learning set

	recognition rate	cumulative recognition rate within best N candidates	
	1	5	10
the pattern matching	54.0%	85.4%	92.0%
HMM	59.2%	79.5%	87.3%
integrating two classifiers	70.4%	89.7%	95.3%

Table 2: Word recognition rate for the test set

	recognition rate	cumulative recognition rate within best N candidates	
	1	5	10
the pattern matching	56.8%	83.2%	88.9%
HMM	59.2%	81.6%	88.4%
integrating two classifiers	68.4%	86.0%	92.5%

10 Discussion

The results mentioned in the previous section show that, using the proposed method, the word recognition rate is improved remarkably by integrating two classifiers compared to using either single classifier.

Table 3 shows the word recognition rate using only “ A ” candidates versus “ A ” and “ B ” candidates together, where “ A ” and “ B ” denote the set of classes

described in Section 8. The results show that adding the specified characters to the candidates helps to improve the recognition rate.

Table 3: Word recognition rate using candidates “*A*” versus candidates “*A*” and “*B*” mentioned in Section 8 for the learning set

	recognition rate	cumulative recognition rate within best N candidates	
	1	5	10
<i>A</i> candidates	67.8%	88.0%	93.2%
<i>A</i> and <i>B</i> candidates	70.4%	89.7%	95.3%

11 Conclusion

A method for the cursive word recognition by the integration of multiple classifiers is proposed. The idea was tested using two classifiers, pattern matching and HMM, and proved to be effective by an experiment is executed on cursive word written in Latin characters. Setting a lexicon size 166, 1,312 cursive word written in Latin characters are recognized. Though the first rank recognition rate using only the pattern matching is 56.8% and that using only HMM is 59.2%, the first rank recognition rate is improved to 68.4% by the integration. This result shows that the proposed method is very promising.

References

1. J. C. Simon, Off-line cursive word recognition, Proc. IEEE Vol.80, No.7, pp. 1150–1161 (1992)
2. R. M. Bozinovic and S. N. Srihari, Off-line cursive script word recognition, IEEE Trans. on PAMI, Vol 11, No.1, pp. 68–83 (1989)
3. P. Sinha and J. Mao, Combining Multiple OCRs for Optimizing Word Recognition, Proc.14th ICPR, Brisbane, Australia, pp.436–438 (1998)
4. Kenichi Maruyama, Makoto Kobayashi, Yasuaki Nakano and Hirobumi Yamada, “Cursive Handwritten Word Recognition by Integrating Multiple Classifiers”, IEICE Trans. vol. J82–D–II No.9 pp.1435–1443 (1999) (in Japanese)
5. Hirobumi Yamada and Yasuaki Nakano, Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis, IEICE Trans. INF&SYSTEM. Vol.E79–D, No.5 (1996)
6. S. Mori, C. Y. Suen and K. Yamamoto, Historical Review of OCR Research and Development, Proc. IEEE, Vol. 80, No. 7, pp.1029–1058

(1992)

7. Lawrence R. Rabiner, "A Tutrial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. IEEE, Vol.77, No.2, pp.257–286 (1989)