

# Cursive Handwritten Word Recognition by Integrating Multiple Classifiers

Kenichi Maruyama<sup>1</sup>, Makoto Kobayashi<sup>1</sup>, Hirobumi Yamada<sup>2</sup>, and Yasuaki Nakano<sup>1</sup>

<sup>1</sup> Dept. of Information Engineering, Shinshu University, Nagano, Japan

<sup>2</sup> Faculty of Engineering, Toyohashi University of Technology  
Tenpakuchou, Toyohashi, Aichi, Japan

**Abstract.** This paper proposes a method for cursive handwritten word recognition. In the traditional research, many cursive handwritten word recognition systems used a single method for character recognition. In this research, we propose a method integrating multiple character classifier to improve word recognition rate combining the results of them. As a result of the experiment using two classifiers, word recognition rate is improved than from those using a single character classifier.

## 1 Introduction

In a traditional method of word recognition, strings that are combined from the results of character recognition for patterns formed by segmenting a word image are matched to each entry of a lexicon. Thus, to improve a word recognition rate, it is most important to improve character recognition rate because it is most important for correct classes to be at higher rank. But it is difficult to improve the recognition rate of a classifier because usually the performance of the classifier reaches to its highest potential.

To improve word recognition rate fixing a single character classifier, increasing the number of candidates obtained from the classifier may be thought as effective. But if candidates from character recognition increase, wrong candidates will increase as well as the correct ones. There is possibility that the increase of wrong candidates has a bad influence on a word recognition rate.

On the other hand, in recognition of printed English characters and handwritten numerals, it is reported that recognition rate is improved by integrating multiple recognition methods [1] [2]. In this research, we propose a method to integrate multiple classifiers at the character recognition stage in the word recognition system. To evaluate the effectiveness of the proposed method, we test the integration of two character classifiers, i.e., the pattern matching based on directional feature patterns [5] and HMM (Hidden Markov Models) [7].

Recently, a similar approach is proposed by Sinha et al. [3] and discusses the optimal combination by a statistical method. Our paper proposes a more practical combination method.

## 2 Approach

The system used in this research is based on Yamada's research [4]. As illustrated in Figure 1, this paper proposes a method using multiple classifiers at the character recognition stage and integrating the results. A newly developed character classifier using HMM is used as well as the one used in the previous paper [4].

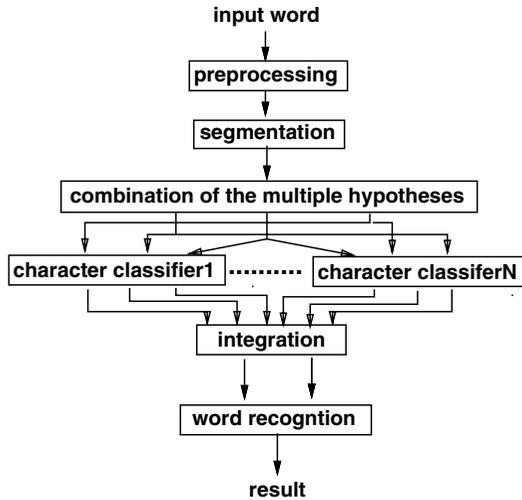


Fig. 1. Flowchart of the proposed system

### 2.1 Preprocessing

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

### 2.2 Segmentation

Candidates of segmentation points are estimated on the left and the right ends of character-like shapes by a contour analysis. Since there is possibility that the simple curve line linking two candidates is a ligature, a middle point of the estimated ligature is added as a candidate of segmentation point.

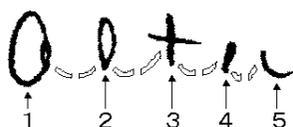
### 2.3 Combination and Multiple Hypotheses

Neighboring pattern fragments segmented as candidates are combined to make a hypothesis of a character pattern. Combination of pattern fragments is limited

by the number of fragments and the total length of contour. Combined patterns are assembled into a group.

A group is a set of pattern fragments which are combined around a major pattern which is not a ligature. Example of major patterns are painted black in Figure 2. Figure 3 shows an example of groups generated from pattern fragments shown in Figure 2.

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



**Fig. 2.** Segmented pattern fragments (a pattern painted black shows a major)



**Fig. 3.** Example of groups (The rectangles show the range of combination in each group)

## 2.4 Character Recognition

Two character classifiers are used in this paper. They are the pattern matching based on directional feature patterns [5] and HMM [7]. The details of two classifiers will be stated in chapters 4 and 5.

Each method outputs three candidates for a combination hypothesis and the results for all hypotheses generated from a same group are merged. If the classes for the different patterns in a group coincide, the result having the largest similarity is adopted.

Word recognition stage uses the result obtained by integrating two methods and does not distinguish uppercase and lowercase characters. So, the outputs from both character classifiers are case insensitive.

## 2.5 Word Recognition

Word recognition uses the results of character recognition. A set of segmentation points having the smallest penalty is calculated to each word in a lexicon. Figure 4 shows an example but many results are not shown for the simplicity. In Figure 4, each rectangle shows a combination of patterns. For example the string (5, C) in a rectangle shows that "C" is the fifth rank for the combination hypothesis. The combination is given a penalty calculated as the average of the rank of each rectangle.

For the word in the example of Figure 4, the penalty of “ALTA” is  $(2 + 2 + 1 + 1)/4 = 1.5$  combining (2, A)(2, L)(1, T)(1, A) and lower than the penalties of all other words in the lexicon. Thus “ALTA” is the first rank in the word recognition. In the same way, ten candidates are outputted and sorted in the ascending order of penalties.

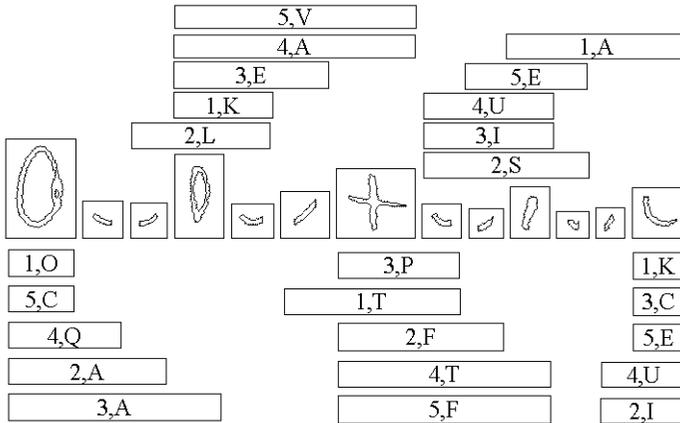


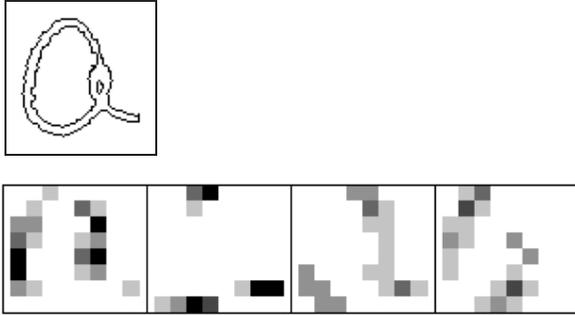
Fig. 4. Example of word recognition

### 3 Character Classifier Using the Pattern Matching Based on Directional Feature Patterns

The pattern matching based on directional feature patterns (abbreviated as the pattern matching hereafter) is known as an effective method in handwritten kanji recognition [5].

The size of the input pattern sent to the recognition unit is normalized to  $64 \times 64$  pixels. At the same time, the inclination of the input pattern is corrected. The normalized pattern is partitioned into  $8 \times 8$  blocks. Four patterns emphasizing in four directions (vertical, horizontal, left slant, right slant) at every block are formed (Figure 5). Pattern matching using these four patterns is executed. A similarity for a pattern is the average of similarities on four directional patterns. The similarities are calculated with templates of all classes and the several similarities from the top are selected.

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



**Fig. 5.** Patterns at the top row is an original. Those are at the second row are emphasized in four direction (vertical, horizontal, left slant and right slant)

### 3.1 Templates

Templates used in matching are made from averaging features of learning samples. Some characters have variant shapes, so multiple templates are generated for such a character class by a clustering technique. Total number of templates for 52 alphabets (uppercase and lowercase characters) is 426. In the outputs of recognition uppercase and lowercase are not distinguished in the output, so the number of different classes is 26.

## 4 Character Recognition Using Hidden Markov Models

In this research, same as the pattern matching method explained in the previous chapter, each input pattern is normalized to  $64 \times 64$  pixels and the inclination is corrected.

### 4.1 HMM

Though 2-dimensional HMM methods are proposed and tested for character recognition [6] [12], we adopted 1-dimensional HMM, which has become mature in speech recognition [7].

The elements of HMM consist of several states and arcs of state transition shown in Figure 6. Each arc has state transition probability and output probability where  $A = \{a_{ij}\}$ : set of probabilities of transition from  $S_i$  to  $S_j$ ,  $B = \{b_{ij}(k)\}$ : set of probabilities of outputting symbol  $k$  when state is transited from  $S_i$  to  $S_j$ ,  $\pi = \{\pi_i\}$ : set of initial state probability. For convenience, we use the compact notation  $\lambda = (A, B, \pi)$  to indicate the complete parameter set of the model.

Given the observation sequence  $O = O_1 O_2 \cdots O_T$  and HMM  $\lambda$ , we can compute  $P(O | \lambda)$ , the probability of the observation sequence outputted from the HMM  $\lambda$ .

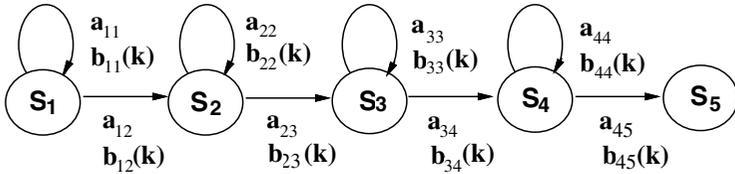


Fig. 6. Hidden Markov Model

## 4.2 Learning of HMM

Given the parameters of HMM  $\lambda = (A, B, \pi)$  and the observation sequences  $\mathcal{O} = \{\mathcal{O}^{(1)}, \mathcal{O}^{(2)}, \dots, \mathcal{O}^{(K)}\}$ , we can estimate the parameters which maximize

$$P(\mathcal{O} | \lambda) = \prod_{k=1}^K P(\mathcal{O}^{(k)} | \lambda),$$

where  $K$  is the number of different symbols [7].

Using this learning method, we construct an HMM to each character class. Thus, 52 HMMs are constructed corresponding to A–Z and a–z. The sample number used in learning is 12,050.

## 4.3 Recognition

Using HMMs constructed by the learning, recognition is executed. Given the observation sequence obtained from an unknown character pattern, probability of outputting this sequence at each HMM is calculated. The recognition result is a set the classes corresponding to high probabilities outputted from the HMMs.

When the result of character recognition is outputted, uppercase letters are substituted to the corresponding lowercase letters to make the result case insensitive. Same as the pattern matching, three candidates for a group are outputted by sorting the results in descending order of probabilities.

## 4.4 Feature Extraction

In this research, we adopt 1-dimensional HMM using a symbol string extracted from thin rectangles obtained by slicing a character pattern. Features used in the research are shown in Figure 7.

By scanning the normalized pattern vertically up to down four features  $f_1, f_2, f_3, f_4$  are extracted. Here  $f_1$  is the first  $y$  coordinate of black pixel and  $f_2$  is the run length of black pixels starting from  $f_1$ . Similarly  $f_3$  is the second  $y$  coordinate of black pixel and  $f_4$  is the run length of black pixels starting from  $f_3$ . Next, down to up, similar operation extracts other four features  $f_5, f_6, f_7, f_8$ . The eight-dimensional vector whose elements are obtained

as above is used as the feature vector. Since the operation is repeated on each pixel on the abscissa, 64 vectors are generated. When any black pixel does not exist in the searching, the coordinate is set to (maximum + 1 = 64) or (minimum - 1 = -1) and the length is set to 0.

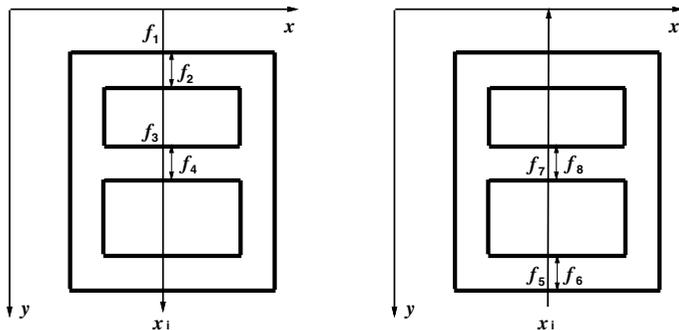


Fig. 7. Feature extraction of HMM

#### 4.5 Vector Quantization

A feature vector is converted into a symbol using vector quantization [7].

We compare a feature vector with the codebook which is the set of vectors generated in advance and represent the vector by the index of the codebook vector whose distance is minimal.

The codebook itself is generated by clustering feature vectors from a learning sample set. Size of codebook is 128. In this manner, a character sample is changed into a symbol sequence.

### 5 Character Recognition Rate

Table 1 shows character recognition rate using the pattern matching and HMM. The size of the test set is 2,744. These samples are segmented manually from the test set used for word recognition. Uppercase and lowercase letters are not distinguished. The result of the HMM classifier is on the learning pattern set, or, in other words, HMMs are generated from the test set.

### 6 Character Recognition Integrating Two Classifiers

Results obtained by the pattern matching classifier and HMM classifier are integrated. In this integration, it must be noted that two classifiers use different

**Table 1.** Character recognition rate using the pattern matching and HMM

	size of test set	recognition rate	cumulative recognition rate	
	2,744		2	3
the pattern matching		56.6%	71.9%	78.1%
HMM		66.0%	83.7%	90.9%

measures: the pattern matching uses similarity and HMM uses probability. As the result, the measures cannot be used in the integration, but only the orders in a result of recognition can be used.

The method of integration is as follows. Assume that an array  $combi[i]$  is the table of results (orders) after integration of two classifiers,  $pattern[j]$  that of the pattern matching and  $hmm[k]$  that of HMM. (Here,  $i, j, k$  denotes the size of array,  $i = j + k$ .)

An element of the array  $combi$  shows the order after integration which represents penalty for each candidate class.

1. Substitute each element of  $pattern$  into an odd number element of  $combi$  in turn:  $combi[2m - 1] = pattern[m]$  ( $m = 1, 2, \dots, j$ ).
2. Substitute each element of  $hmm$  into an even number element of  $combi$  in turn:  $combi[2n] = hmm[n]$  ( $n = 1, 2, \dots, k$ ).
3. If either array ends before the last element comes, the rests of opposite array are added to the bottom.

In the present status, the pattern matching has better performance, so it is given the higher priority.

Next, it is tested whether the same characters are duplicated in  $combi$ . The candidate having the higher penalty is deleted in the case that candidates have the same character class and the same pattern number. In these cases, it is considered that these candidates have high reliability because both classifiers support them. To reflect this conjecture, the remained candidate (a candidate having low penalty) is promoted. The promotion is done by interchanging the candidate with the one rank higher candidate. But, the interchange is not done in the case that the one rank higher candidate has already been promoted.

## 7 Result of Word Recognition

To improve word recognition rate fixing a single character classifier, increasing the number of candidates obtained from the classifier may be thought as effective. To test this conjecture, an experiment is executed by changing the number of the candidates at the result of the recognition stage.

Table 2 shows the change of word recognition rate when the number of candidates obtained from the pattern matching classifier is increased. It can be seen

from the result that even if candidates from character recognition increase, word recognition rate is not improved but sometimes deteriorated. The reason may be as follows. The increase of the candidates may sometimes increase “imposters” and deteriorate the word recognition.

Table 3 shows word recognition rate using the pattern matching, HMM and integration of two classifiers. For the pattern matching and HMM classifiers, output number of the character recognition stage is set to 3, by the consideration on Table 2.

In Table 2 and 3, the size of the test set is 175 and the lexicon size is 115.

**Table 2.** A variation of word recognition rate depending on candidates of character recognition using the pattern matching

	test set size	recognition rate	cumulative recognition rate	
	175		5	10
three candidates		62.9%	88.6%	93.7%
five candidates		61.7%	85.7%	92.0%

**Table 3.** Word recognition rate of single classifiers and integration

	test set size	recognition rate	cumulative recognition rate	
	175		5	10
the pattern matching		62.9%	88.6%	93.7%
HMM		57.1%	86.3%	90.3%
integrating two classifiers		79.4%	92.6%	95.4%

## 8 Discussion

From a result mentioned in the previous chapter using the proposed method, it is shown that word recognition rate is improved by integrating two character classifiers in comparison with word recognition rate using a single usage of either classifier.

The recognition time using two classifiers is sum of two classifiers if the recognition is executed successively. Using parallel operation, however, the recognition time does not increase. But the cost of two OCR’s in speed or space is the negative aspect of the method.

It must be noted that the recognition rate by HMM is on the learning set. To test the ability of the HMM on the test set, another codebook is learned from the other sample set. Table 4 shows the result using the codebook. The improvement the integration is not so good, but cumulative recognition rate within best ten candidates is improved.

The reason of the poor recognition performance of HMM for the test set may be the lack of learning amount. By enhancing the learning, word recognition rate for the test set is experimentally be improved.

**Table 4.** Word recognition rate with integrating the pattern matching and HMM using patterns for the test set

recognition rate	cumulative recognition rate	
	5	10
62.9%	87.4%	94.2%

## 9 Conclusion

This paper proposes a method using multiple classifiers in cursive handwritten word recognition method cooperating with segmentation, character recognition and word recognition. An experimental system is developed using the pattern matching based on directional feature patterns and HMM as character classifiers and a series of experiments is executed on cursive handwritten words included in CEDAR databases. Setting a lexicon size 115, 175 cursive handwritten words are recognized. Though first rank recognition rate using only the pattern matching is 62.9% and that using only HMM is 57.1%, the first rank recognition rate is improved to 79.4% by integrating two classifiers. This result shows that the proposed method is effective. In the future, it is planned to improve character recognition rate using HMM and verify the effect by integrating more than two character recognition classifiers.

## Acknowledgement

We use CEDAR database made by and supplied from the Center of Excellence for Document Analysis and Recognition of SUNY at Buffalo.

We appreciate many advices by Prof. Minoru Maruyama and Dr. Hidetoshi Miyao of Information Engineering Department of Shinshu University.

## References

1. Rice, S. et al: A Report on the Accuracy of OCR Devices. ISRI Technical Report. Information Science Research Institute. University of Nevada. Las Vegas (1992) 1–6 140
2. Matsui, T. et al: State of the Art of Handwritten Numeral Recognition in Japan. Proc. 2nd Int. Conference on Document Analysis and Recognition (1993) 391–396 140
3. P. Sinha and J. Mao: Combining Multiple OCRs for Optimizing Word Recognition. Proc.14th ICPR. Brisbane, Australia (1998) 436–438 140

4. Hirobumi Yamada and Yasuaki Nakano: Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis. *IEICE Trans.INF&SYSTE*. vol.**E79-D-5** (1996) [141](#)
5. S. Mori, C. Y. Suen and K. Yamamoto: Historical Review of OCR Research and Development. *Proc. IEEE*. **80(7)** (1992) 1029–1058 [140](#), [142](#), [143](#)
6. G. E. Kopec and P. A. Chou: Document Image Decoding Using Markov Source Models. *IEEE Trans. on PAMI*. **16(6)** (1994) 602–617 [144](#)
7. Lawrence R. Rabiner: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proc.IEEE*. **77(2)** (1989) 257–286 [140](#), [142](#), [144](#), [145](#), [146](#)
8. J. C. Simon: Off-line cursive word recognition. *Proc. IEEE* **80(7)** (1992) 1150–1161
9. R. M. Bozinovic and S. N. Srihari: Off-line cursive footnote word recognition. *IEEE Trans. on PAMI*. **11(1)** (1989) 68–83
10. A. Kundu and P. Bahl: Recognition of Handwritten Script : A Hidden Markov Model Based Approach. *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, New York City, USA. (1998) 928–931
11. H. -S. Park and S. -W. Lee: Off-line Recognition of Large-set Handwritten Characters with Multiple Hidden Markov Models. *Pattern Recognition*. **29(2)** (1996) 231–244
12. H. -S. Park and S. -W. Lee: A truly 2-D hidden Markov model for off-line handwritten character recognition. *Pattern Recognition*. **31(12)** (1998) 1849–1864 [144](#)