

Improving the Structuring Search Space Method for Accelerating Large Set Character Recognition

Yiping Yang, Masaki Nakagawa

Graduate School of Technology, Tokyo University of Agriculture and Technology Affiliation
2-24-16 Naka-cho Koganei-shi, Tokyo, 184-8588, Japan
yangyiping@tuat.ei.tuat.ac.jp, nakagawa@cc.tuat.ac.jp

Abstract

This paper proposes enhancement of the “structuring search space” (SSS) method attempted in [1] to further accelerate the recognition speed. It consists of structuring the search space into two layers, improving the candidate selection algorithm and selecting candidates depending on the top candidate. For two-layered search space, we divide all of the prototypes into smaller clusters and derive the centroid of each cluster as a pivot, then again cluster all of the pivots and derive the centroid of each cluster (super cluster) as a super pivot. An input pattern is compared with all the super pivots and several super clusters are selected whose super pivots are close to the input pattern. Then, the input pattern is compared with pivots in the selected super clusters, close pivots are selected and prototypes within the clusters of the selected pivots are treated as candidates for fine classification. Thus, the number of prototypes compared with the input pattern is greatly reduced. Moreover, we employ a synthetic candidate selection algorithm and a top candidate dependent candidate selection method. Since the top candidate suggests where the input pattern is mapped in the feature space, it can provide the information on how candidates should be selected in coarse classification. Thus, this information is specified in each prototype for the case when it is selected as the top candidate and specified values are employed for selecting a variable number of candidates. These improvements have been incorporated into a practical off-line Japanese character recognizer consisting of coarse classification and fine classification with the result that the coarse classification time is reduced to 28.6% and the whole recognition time is reduced to 31.3% from the original time while sacrificing a very limited recognition rate (98.1% to 97.7%).

1. Introduction

Japanese, Chinese or Korean have a large character set with several thousands different categories, so that their character recognition takes more time than western alphabet or numeral recognition. Although the hardware development for personal computers is remarkable, the

faster the character recognition is the more capacity can be provided to run multiple recognizers, to combine them with context post-processing, to incorporate them in handwritten text search and so on. Moreover, there is high demand for character recognition on a small device like a mobile phone or PDA. Therefore, to accelerate large character set recognition has been studied with practical importance.

The two-stage architecture of coarse classification (pre-classification or candidate selection) and fine classification has been employed in many practical systems [2]-[7]. Coarse classification should be significantly faster than the fine classification and select a limited number of candidates robustly with the effect that prototype patterns for the fine classification to match with an input pattern is reduced from several thousands to several hundreds or even less. Consequently, the whole recognition process is accelerated. This type of candidate selection is made during recognition so we call it as dynamic process.

On the other hand, we introduced a different strategy by a static process which structures search space before recognition so that the search for candidates can be made only to some portion in the search space [1]. We termed this as a “structuring search space (SSS)” method.

There have been some methods to structure the search space and make the search faster. The simplest is the ordered space. Another is the tree structure. They can be applied when prototypes are ordered in some sense or classified into a tree structure. On the other hand, the SSS method does not assume such structures and we make a structure by clustering.

This method was applied to our practical off-line handwritten Japanese character recognizer consisting of coarse classification and fine classification, with the result that it shortened the coarse classification time to about 61% of the original, and the whole recognition time to 70% while keeping the same recognition rate.

In this paper, we attempt to enhance the method in three components. The first is to structure the search space into multiple layers in order to further reduce the number of prototypes compared with an input pattern. We confirmed the effect of two-layered search space but not in three-layered space. The second is to make a synthetic

candidate selection algorithm based on the two algorithms employed in the initial SSS study. The third is to select candidates depending on the top candidate since the top candidate provides the information on where the input pattern is mapped in the feature space.

Section 2 of this paper describes the off-line recognizer based on which the proposed method is evaluated. Section 3 presents the basic design of enhancement and Section 4 presents the detailed design and implementation. Section 5 describes preparation for evaluation and Section 6 presents experiments and considerations. Section 7 concludes the paper.

2. Off-line Character Recognizer

The off-line character recognizer used for this research represents each character as a 256-dimensional feature vector. It scales every input pattern to a 64x64 grid by non-linear normalization [8]. Then, it decomposes the normalized image into 4 contour sub-patterns representing directional feature of the 4 main orientations. Finally, it extracts a 64-dimensional feature vector for each contour pattern from the convolution with a blurring mask (Gaussian filter).

The coarse classification step precedes the actual fine classification. The original form of the coarse classification selects 40 candidates with the shortest Euclidian distances between the categories' mean vectors and an input pattern.

The fine classification employs a modified quadratic discriminant function (MQDF2) [9]. It was trained by training set composed of three databases, namely Nakayosi database [10, 11] (with 1,695,689 patterns of 4,438 categories), ETL9b database (607,200 patterns of 3,036 categories), and HP-JEITA database (1,917,480 patterns of 3,214 categories). All together these databases consist of 4,443 different categories, including digits, western characters, symbols, katakana, hiragana and Japanese Kanji characters.

While off-line databases ETL9b and HP-JEITA can be immediately used for training the off-line recognizer, the on-line database Nakayosi must be transformed to off-line format (bitmap images) first. We have employed a unique method for generating realistic Kanji character images from on-line patterns [12]. This method combines on-line patterns with a calligraphic stroke shape library, which contains genuine off-line patterns written with different writing tools. Since the artificially generated off-line images are combinations of on-line and actual off-line patterns, they look very natural and realistic.

3. Basic Design

In this section, we present the basic design for enhancement, which is composed of three components.

3.1 Two-layered search space

In the initial SSS method, we divided all the prototypes into smaller clusters and derived the centroid of each cluster as a pivot. Then, an input pattern was compared with all the pivots and only a limited number of clusters whose pivots had higher similarities (or smaller distance) to the input pattern were searched for. This time, we introduce two layers in the search space. A structuring method of the lower layer's search space is the same as the initial SSS method: it takes the entire prototypes and divide them into a group of small clusters and regards the centroid of each cluster as a pivot. For the upper layer's search space, we cluster all of the pivots and derive the centroid of each cluster as a super pivot. Thus, the two-layered search space is structured.

Given an input pattern, it is compared with all the super pivots in the upper layer of the search space and several super clusters are selected whose super pivots are close to the input pattern.

Then, the input pattern is compared with pivots in the selected super clusters, close pivots are selected and prototypes within the clusters of the selected pivots are treated as candidates for fine classification. Thus, the number of prototypes compared with the input pattern is further reduced. This is based on the assumption that the search space is a distance space.

Fig. 1 is a conceptual figure of the feature space drawn in two-dimensional space (although typical feature space for large character set recognition takes 256 or 512 dimensions). Note that each cluster is made up of different character categories rather than multiple prototypes of a single category.

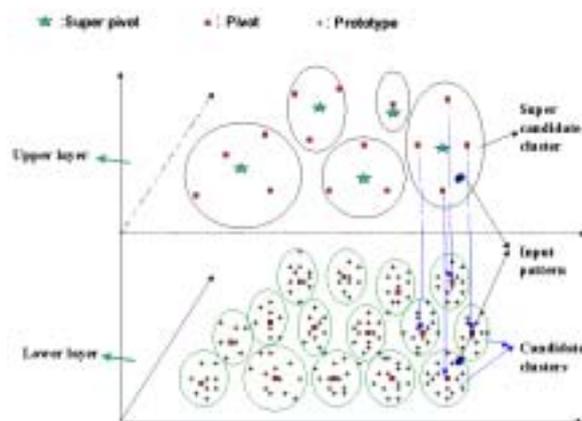


Fig. 1 Conceptual figure of two-layered search space.

3.2 Synthetic algorithm for candidate selection

In the initial SSS method, we tested two algorithms to select candidate clusters:

(1) We set a constant for the number of candidate clusters, and select l clusters whose centroids have the shortest Euclidian distance up to the l -th shortest Euclidian distances to the input pattern (Algorithm-I).

(2) We find the nearest centroid with the distance d_{min} , and then we set a multiplying coefficient m ($m > 1$) so that all clusters within the distance $m * d_{min}$ become candidate clusters (Algorithm-II).

Although the two algorithms produced almost the same results, detailed examination revealed different characteristics between them.

When the Algorithm-I is employed, the final prototype candidates in the candidate clusters can be limited to a relative small and stable range, but this algorithm must spend much time to sort all the candidates (they can be pivots in the upper layer or prototypes in the lower layer) by their scores obtained through comparing them with an input pattern. On the other hand, Algorithm-II need not to sort the candidates, but the number of candidates has a wide variable range, for example, from several tens for some input patterns to nearly 2,000 for other input patterns.

Here, we propose a synthetic algorithm to utilize the advantages of both the algorithms. At first, we adopt Algorithm-II to utilize high speed and reduce the number of candidates and then adopt Algorithm-I to further reduce their number to a relative small and stably range. Of course, for some categories of input patterns, if the output of Algorithm-II is less than 40, the number of candidates for the fine classification, Algorithm-I is skipped.

3.3 Candidate selection specified by the top candidate

Simple methods for candidate selection select a fixed number of candidates by a coarse classification. Better methods select a variable number of candidates using confidence ratio, accumulated confidence or whatever. We extend the latter by selecting a variable number of candidates depending on the top candidate selected.

The top candidate suggests where in the feature space an input pattern is mapped so that it can specify how dense the feature space is around it and how much candidates should be selected for fine classification. The number of candidates and the multiplying coefficient for candidate selection (hereafter, we call them parameters for candidate selection) in the case that it is selected as the top candidate can be specified in each prototype and they can be robustly obtained from a large amount of learning samples.

This strategy can be applied to both the upper layer and lower layer of search space in our method and even to any candidate selection methods.

4. Detail Design and Implementation

In this section, we present the detailed design and implementation.

4.1 Forming two-layered search space

There remains still several problems here, which are similar to the initial SSS: the number of clusters in each layer, the algorithm for clustering and so on. They can influence the efficiency and accuracy. The experiments and results of the initial SSS can guide us to determine some of them, but some still need experiments to solve.

4.1.1 Optimal number of clusters

The structure of the lower layer of search space is the same as the SSS method, so that the optimal number of clusters can be obtained from about 200 to 500. Since a relatively large number of pivots is more beneficial for the upper layer of search space, we just choose 500 as the number of clusters for structuring the lower layer.

For the upper layer, as was in the initial SSS, we have to make experiments to decide it.

4.1.2 Details of clustering

We employ the LBG algorithm [13] for clustering. It is one of the simplest and effective methods, and its effect has been proved in SSS.

Although the result of the LBG clustering depends on the initial selection of centroids, our previous study has revealed little difference among three methods of selecting the initial centroids. Thus, in this study, we simply partition a sequence of prototypes in the order of their character codes into the equal size of clusters and select the centroid of each cluster as the initial centroid.

4.1.3 Management of clusters

After all the prototypes are divided into clusters and all pivots are divided into super clusters, we manage them so as to speed up the search.

For the lower layer, we adopt the same method as SSS. We store in a file the total number of clusters, the number of prototypes and the centroid in each cluster, indexes to all the prototypes in each cluster and the dimension of feature vectors.

For the upper layer, we also adopt almost the same method, i.e., we store the total number of super clusters, the number of pivots in each super cluster, its super pivot as its centroid, the code of the super pivot, indexes to all the pivots in each super cluster. When the search process is carried out, all the pre-calculated information is loaded from the file.

4.2 Parameters for synthetic algorithm

The synthetic algorithm is easy to realize and is expected to have high effect. We only have to determine the multiplying coefficient m for Algorithm-II and the constant l for the number of candidates for Algorithm-I. In order to determine these parameters, however, we need to make experiments, which will be described in Section 6.

4.3 Top candidate dependent candidate selection

This strategy can be applied to both the upper layer and lower layer of search space. Each pivot or super pivot specifies the parameters for candidate selection in the case that it is selected as the top candidate. We adopt training to fill them using a large amount of learning patterns.

The strategy of this training is similar to that for a character recognizer to extract features from learning patterns. It consists of the following three steps:

The first step is to make an experiment with the original constant number of candidates to know the recognizer's original recognition rate for the learning patterns. Then, learning patterns that cannot be recognized correctly are removed from the learning set, and the new learning set with 100% recognition rate is formed, it will be used for later steps of training.

The second step is to set the parameters to the smallest values, i.e., l as 1 and m as 1.0 for each pivot/super pivot.

The third step is to recognize every learning pattern and select the closest pivot/super pivot, and select candidates according to the parameters set for this pivot/super pivot. If the correct answer is not included in the candidates while it is in the original constant number of candidates in the first step, we increase the parameters by a reasonable amount of interval until the correct answer is included.

After this step, parameters are set between the smallest values to the largest values in the first step.

5. Preparation for experiments

This section describes the environment and preparation for experiments.

5.1 Testing set and environment

We have created a testing set from the Nakayosi database, which is an established benchmark for Japanese handwritten characters. The set has 4,443 testing patterns with one randomly selected from each category.

We made experiments on a PC with an Intel Pentium4 CPU of 2.4GHz and 512M RAM employing Microsoft Windows 2000 professional.

5.2 Original state of the recognizer

Before SSS and three components to enhance it were not introduced, the coarse classification of the original character recognizer produced 40 candidates with the coarse classification rate (the rate that the correct answer is within the candidates) as 99.6% in 17.3 milliseconds. And the whole recognition produced 98.3% recognition rate in 27 milliseconds.

6. Experiments

Since we have presented three components to enhance SSS, we separate the experiments into three parts in order to clarify the degree of contribution by each component.

6.1 Two-layered search space.

For the lower layer's search space, we simply adopted the best scheme presented in SSS. Here, the main problem is at the upper layer. We need to solve the several key problems like the number of super pivots (super clusters), the number of candidate super clusters in the upper layer.

Table 2 shows the number of super clusters, the number of super candidate clusters and under each condition the corresponding coarse classification time and whole recognition time.

Fig.1 shows coarse classification time and whole recognition time with respect to the range of 25 to 300 super clusters. In the whole range, the coarse classification time and whole recognition time are almost stable and reduced up to about 9.5 and 15.5 millisecond, i.e., 54.5% from the original 17.3 millisecond and 59.4% from the original 27 milliseconds, respectively.

Here, the effective number of super clusters has a relatively wide range; the change of recognition time with the increase or decrease in the number of super clusters is not obviously as SSS. We think there are two reasons. One is when the number of super clusters is increased, the number of candidate super clusters also is increased but the its ratio to the total number of clusters can be decreased so that candidate clusters (pivots) for the lower layer is decreased. Thus, the time for selecting candidate super clusters become relative long, but the time can be saved to select candidate clusters (pivots) in the lower layer. When the number of super clusters is decreased, the situation is vice versa. The other reason is that the acceleration effect of the upper layer is not so strong as SSS (or the lower layer), and the whole recognition time is not decreased as the number of super clusters is changed. For this reason, we do not consider another layer of search space to add.

Table 1. Performance when coarse classification rate is around 99.8% and whole recognition rate is 98.2%.

No. of super clusters	25	35	40	50	70	85	100	150	200	250	300
coarse classification time (msec.)	9.7	9.5	9.6	9.5	9.4	9.4	9.5	9.5	9.7	9.8	10
whole recognition time (msec.)	15.6	15.6	15.5	15.6	15.5	15.3	15.3	15.6	15.4	15.6	16.3
No. of candidate super clusters	20	26	29	35	41	50	61	92	115	130	150

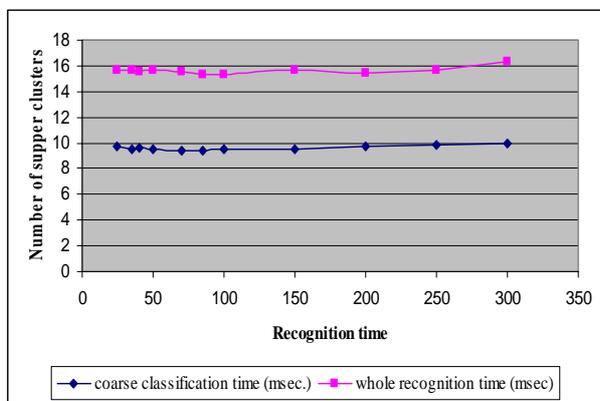


Fig. 2. Coarse classification and whole recognition time with respect to the number of super clusters.

6.2 Synthetic algorithm

We applied the synthetic algorithm to the two-layered search space and the original coarse classification. We sought out the minimum values for the number of candidates l and the multiplying coefficient m with the result that they were 30 and 1.7 for the upper layer of search space, 105 and 1.8 for the lower layer and the original template candidate number 40 and 1.8, respectively for the coarse classification without losing recognition rate. The recognizer with this synthetic algorithm incorporated and these key values assigned reduced the coarse classification time to 46.9% and the whole recognition time to 41.7% from the original processing time.

6.3 Top candidate dependent candidate selection

This method can be applied to the upper layer and lower layer in the two-layered search space and the original coarse classification.

We employed the first 100 (0-99) person's character patterns in Nakayosi database to derive the parameters for candidate selection, which are l and m for each layer. The average m and l in the upper layer were reduced from 30 and 1.7 to 21 and 1.34 while those in the lower layer were reduced from 100 and 1.8 to 28 and 1.31, respectively.

The final result derived through testing the other 50 (100-149) person's character patterns in Nakayosi database shows that the coarse classification time is reduced to 34.8% and the whole recognition time is reduced to 40.1% from the original processing time while sacrificing a very limited recognition rate (98.1% to 97.7%).

We have not attempted this method to the original candidate selection yet since the original coarse classification must consider 4,443 different prototypes and we must apply a huge amount of learning patterns to train the parameters for candidate selection.

6.4 Three components altogether

Finally we assembled all the components into a single system and obtained the result that the coarse classification was reduced to 28.6% and the whole recognition time is reduced to 31.3% from the original time while sacrificing a very limited recognition rate (98.1% to 97.7%).

6.5 Analysis on experiments

There are two results that must be considered. One is that the total effect of the three improvements does not reach to the simple sum of each effect. The other is that the three components contribute to the acceleration of the coarse classification and even more to that of the whole recognition although we did not change any part for the fine recognition.

We think the reason of the first result is that all of the three components try to reduce the number of candidates and the effect of each component overlaps to one another. The second result is due to the decreased number of candidates passed to the fine classification.

7. Main text

This paper presented three components for enhancing "structuring search space" method to further accelerate large character set recognition. They are structuring two-layered search space, applying a synthetic candidate selection algorithm and employing top candidate dependent candidate selection. The latter two were applied to the two-layered search space. Each of the three components accelerated our off-line character recognizer further. When these three components were incorporated

into a single system, however, the total effect did not reach to the simple sum of each effect. Nevertheless, we still obtained the best effect in this case and the coarse classification time was reduced to 28.6% and the whole recognition time is reduced to 31.3% from the original speed while sacrificing a very limited recognition rate (98.1% to 97.7%). Although we showed the three components together in this paper, each can be applied separately.

Acknowledgement

Thanks are due to Professor Yamamoto and other people for providing us with the database ETL-9 and HP-JEITA.

8. References

- [1] Y. Yang, O. Velek, M. Nakagawa: Accelerating Large Character Set Recognition Using Pivots, *Proc. 7th ICDAR*, Vol. 4C, 262-267, 2003.
- [2] S. Mori, K. Yamamoto, M. Yasuda: Research on machine recognition of handprinted characters, *IEEE PAMI*, Vol.6, No.4, 386-405, 1984.
- [3] T. Kumamoto, et al: On speeding candidate selection in handprinted Chinese character recognition, *Pattern Recognition*, Vol.24, No. XXX, 793-799, 1991.
- [4] T. H. Hildebrandt, W. Liu: Optical recognition of handwritten Chinese characters: advances since 1980, *Pattern Recognition*, Vol.26, No.2, 205-225, 1993.
- [5] C.-H. Tung, H.-J. Lee, J.-Y. Tsai: Multi-stage pre-candidate selection in handwritten Chinese character recognition systems, *Pattern Recognition*, Vol.27, No.8, 1093-1102, 1994.
- [6] Y.-H. Tseng, C.-C. Kuo, H.-J. Lee: Speeding up Chinese character recognition in an automatic document reading system, *Pattern Recognition*, Vol.31, No.11, 1601-1612, 1998.
- [7] C.-L. Liu and M. Nakagawa: Precise candidate selection for large character set recognition by confidence evaluation, *IEEE PAMI*, Vol. 22, No. 6, 636-642, 2000.
- [8] J. Tsukumo, H. Tanaka: Classification of handprinted Chinese characters using non-linear normalization and correlation methods, *Proc. 9th ICPR*, Roma, Italy, 168-171, 1988.
- [9] F. Kimura: Modified quadratic discriminant function and the application to Chinese characters, *IEEE PAMI* Vol.9, No.1, 149-153, 1987.
- [10] M. Nakagawa, et al.: On-line character pattern database sampled in a sequence of sentences without any writing instructions, *Proc. 4th ICDAR*, 376-380, 1997.
- [11] K. Matsumoto, T. Fukushima and M. Nakagawa: Collection and analysis of on-line handwritten Japanese character patterns, *Proc. 6th ICDAR*, Seattle, 496-500, 2001.
- [12] O. Velek, M. Nakagawa, C.-L. Liu: Vector-to-image transformation of character patterns for on-line and off-line recognition, *International Journal of Computer Processing of Oriental Languages*, Vol.15, No2, 187-209, 2002.
- [13] Y. Linde, A. Buzo, and R. M. Gray: An algorithm for vector quantization design, *IEEE Trans. on Communications*, Vol. COM-28, 84-95, 1980.
- [14] S. Kaneko, T. Mae and A. Miyamoto: Using Constraint Inequality on Estimated Correlation for Rapid Image Search, *Proc. 16th ICPR*, XX-YY, 2002.