

# Layered Search Spaces for Accelerating Large Set Character Recognition

Yiping Yang, Masaki Nakagawa

Graduate School of Technology, Tokyo University of Agriculture and Technology  
2-24-16 Naka-cho Koganei-shi, Tokyo, 184-8588, Japan

E-mail: [yangyiping@hands.ei.tuat.ac.jp](mailto:yangyiping@hands.ei.tuat.ac.jp), [nakagawa@cc.tuat.ac.jp](mailto:nakagawa@cc.tuat.ac.jp)

## Abstract

*This paper describes “layered search spaces” (LSS) to accelerate recognition of a large category set. The basic concept is to employ pivots into a search space of character pattern prototypes. Given an input pattern, it is compared only with the pivots and those close to it are selected. Then, it matched with prototypes close to the selected pivots. This paper introduces multiple layers. An input pattern is compared with the top-layer pivots and those close to it are selected. Then, it is compared with the 2nd-top-layer pivots close to the selected top-layer pivots. This comparison is repeated until in the base-layer and a small set of candidate prototypes are selected. We applied this method to a handwritten Japanese character recognizer with the result that the coarse classification time was reduced to 47.1% and the whole recognition time was reduced to 46.2% while keeping classification and recognition rates as the original.*

## 1. Introduction

Large character set recognition is problematic not only in recognition rate but also in recognition speed. Chinese, Japanese or Korean have thousands of different categories, so that recognition takes more time than Latin alphabet or number recognition.

A general approach to improve the recognition speed is to perform coarse classification, pre-classification or candidate selection before the fine classification [1-4]. The common characteristic of these methods is that candidates are selected during recognition process, so that we call them as a dynamic approach.

The alternative approach is to structure the search space [5, 6]. The feature space of categories is divided into smaller clusters and the centroid of each cluster is derived as a pivot. Since the structuring is made before the recognition process, we distinguish it from the dynamic approach mentioned above. We call this a “structuring search space (SSS)” method and we have applied it to our recognizer of handwritten Japanese characters with notable effect [5, 6]. There have been some other methods to structure the search space. 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.

Tseng et al. proposed a static approach by clustering prototypes employing a small number of simple features for printed character recognition [8]. Fujimoto et al. proposed another where coarse classification is made in a sub-space much smaller than that for fine classification. Several different small sub-spaces of the same structure were adopted to accelerate the recognizer [9]. The above two methods, however, adds the additional problem of finding a feature space for coarse classification. On the other hand, our method works in the original feature space for fine classification. We only have to assume distance space for this single space. We tried a similar approach as the above with a smaller set of features for handwritten Japanese character recognition but we could not speed up recognition without sacrificing recognition rate.

This paper is an updated version of [7] and presents a systematic extension of SSS to multiply layered search spaces. We call it as “layered search spaces” (LSS).

In this paper, Section 2 describes the handwritten Japanese character recognizer for which the proposed method has been evaluated. Section 3 presents the LSS model. Section 4 describes candidate selection method. Section 5 presents experiments. Section 6 presents the analysis on the results.

## 2. Japanese character recognizer

The recognizer of handwritten Japanese characters used for this research represents each scanned image of a character pattern as a 256-dimensional feature vector. It scales every input pattern to a 64x64 grid by non-linear normalization [9]. Then, it decomposes the normalized image into 4 contour sub-patterns representing directional features 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 selects 40 candidates with the shortest Euclidian distances between the categories’ prototype and an input pattern. The fine classification employs the modified quadratic discriminant function (MQDF2) [10].

### 3. Model of layered search space

Fig. 1 is a conceptual figure of the feature space drawn in two-dimensional space of two-layered search spaces. First, we cluster all the prototypes and calculate the centroid of each cluster as a pivot to create the base-layer search space. Then, we follow the same procedure for all the obtained pivots and prepare the

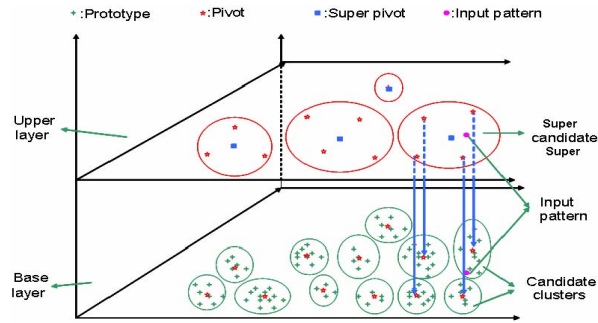


Fig. 1. Conceptual figure of the feature space.

2nd-layer search space. A repetition of this procedure creates multiple layered search spaces.

When an input pattern is recognized, it is compared with all the top-layer pivots and those close to it are selected. Then, it is compared with the 2nd-top-layer pivots belonging to the clusters of the selected top-layer pivots. This comparison is repeated until in the base-layer and a finite set of candidate prototypes are selected. Thus, the number of prototypes compared with the input pattern is reduced.

Since the number of clusters in each layer and number of layers affect the performance of acceleration, the optimal numbers must be determined. We make experiments to determine them.

### 4. Candidate selection

We select pivots or prototypes close to the input pattern using the following hybrid candidate selection method (CS-H, in short), which is the same as for SSS. It is composed of the following two algorithms:

**Candidate selection algorithm-I** (CS-I) sets a constant  $l$  for the number of candidate pivots or prototypes, and then selects only  $l$  pivots or prototypes as candidates, which have less than or equal to the  $l$ -th shortest Euclidian distance to the input pattern.

**Candidate selection algorithm-II** (CS-II) finds the nearest pivot or prototype with the shortest distance  $d_{min}$  to the input pattern, and then sets a multiplying coefficient  $m$  ( $m > 1$ ), thus all the pivots or prototypes within the distance  $m * d_{min}$  become candidates.

**Hybrid candidate selection method** (CS-H) first applies CS-II for its high speed to reduce the number of

Table 1. Performance of the original recognizer.

Set	Measure		Coarse classification rate (%)		Coarse classification time (msec.)		Whole recognition rate (%)		Whole recognition time (msec.)	
	$Tr$	$Te$	$Tr$	$Te$	$Tr$	$Te$	$Tr$	$Te$	$Tr$	$Te$
$Tr-1$	$Te-1$	99.0	99.3	13.1	13.1	98.2	97.1	20.0	20.0	
$Tr-2$	$Te-2$	99.1	99.3	13.1	13.1	98.3	96.4	20.0	20.1	
$Tr-3$	$Te-3$	99.1	98.8	13.1	13.0	98.4	96.0	20.1	20.1	
$Tr-4$	$Te-4$	99.1	98.8	13.1	13.1	98.4	96.0	20.0	20.0	
$Tr-5$	$Te-5$	99.1	98.9	13.0	13.1	98.3	96.1	20.1	20.1	
$Tr-Av$	$Te-Av$	99.1	99.0	13.1	13.1	98.3	96.3	20.0	20.0	

candidates and then applies CS-I to further reduce their number. Note that in some cases when the output of CS-II is less than the constant  $l$ , CS-I is skipped.

## 5. Experiments

We first prepare training and testing sets of handwritten Japanese character patterns. Secondly, we find the best parameters to speed up recognition without damaging recognition accuracy for the base-layer search space. Thirdly, we find the best parameters for the 2nd-layer search space.

### 5.1 Preparation for experiments

#### (1) Training set, testing set and environment

We use HP-JEITA database. It includes 580 persons' handwriting for the 3,214 categories of digits, Latin alphabet, symbols, hiragana (a set of phonetic Japanese characters), katakana (another set of phonetic Japanese characters) and Kanji characters of Chinese origin. We choose 543 persons' patterns by removing 37 persons' imperfect collections.

We split the database into group 1 to group 5 of 100 persons' patterns each and the group 6 of 43 persons' patterns. We select one group from the group 1 to 5 as the testing set  $i$  and merge all the remaining groups ( $100 \times 4 + 43$  persons' patterns) as the training set  $i$ .

We perform experiments on a PC with an Intel Pentium4 CPU of 2.4GHz and 512M RAM employing Microsoft Windows XP Professional.

#### (2) How to use training sets and testing sets

For each testing set  $i$ , we use the training set  $i$  to train the original recognizer described in 2 for obtaining the prototype of each character category in the feature space. We also use the training set  $i$  to obtain the best values for the parameters  $m$  and  $l$  for each layer and then evaluate the performance on the testing set  $i$ .

#### (3) Original state of the recognizer

Table 1 shows the original performance of the recognizer to each training set  $Tr-i$  by which prototypes have been trained and to each testing set  $Te-i$ .  $Tr-Av$  and  $Te-Av$  denote the average for the training sets and that for testing sets, respectively.

### 5.2 Construction of the base-layer search space

Construction of the base-layer search space for LSS is completely the same as that of the single layer for SSS. Referring to [6], we know that acceleration effect is obtained on a wide range where the search space is split into from about 100 to 400. According to the algorithm of LSS, we consider that the larger the number is, the more the search is accelerated, so that we select 400 to build the base-layer search space.

When the search space is divided into 400 clusters, the optimal values of parameters  $m$  and  $l$  are 130 and 1.75, respectively. These values were experimentally obtained in our previous research on SSS, the single-layered LSS [6]. Since we employ five pairs of training sets and testing sets to make experiments, we should validate those values to be also suitable for these five pairs of pattern sets.

We made experiments by shifting  $m$  from 120 to 140 and  $l$  from 1.65 to 1.85 for these 5 pairs of pattern sets. The experiment results show that, for all the sets, including even testing sets, the recognition rates can be kept as the same as the original rate up to the third significant digit in the range around the  $l=130\pm 5$  and  $m=1.775\pm 0.025$ , and the accelerate effect is the same as SSS. Table 2 summarizes the results for a pair of a training set and a testing set into a single row.

**Table 2. Performance with 400 clusters and CS-H.**

Measure Set	Coarse classification		Base-layer		Coarse classification rate (%)		Coarse classification time (msec.)		Whole recognition rate (%)		Whole recognition time (msec.)		
	$l$	$m$	$l$	$m$									
Tr-1	Te-1	40	1.8	131	1.775	99.0	99.3	7.30	7.30	98.2	97.1	10.7	10.5
Tr-2	Te-2			132	1.775	99.1	99.3	7.30	7.31	98.3	96.4	10.7	10.8
Tr-3	Te-3			135	1.800	99.1	98.8	7.35	7.32	98.4	96.0	11.1	11.2
Tr-4	Te-4			127	1.775	99.1	98.8	7.28	7.28	98.4	96.0	10.5	10.4
Tr-5	Te-5			125	1.75	99.1	98.9	7.22	7.21	98.3	96.1	10.1	10.0
Tr-Av	Te-Av			130	1.775	99.1	99.0	7.29	7.28	98.3	96.3	10.6	10.6

### 5.3 Construction of the 2nd-layer search space

In order to find the optimal number of super clusters in the 2nd-layer, we must try possible numbers of super clusters through experiments as we did in [6].

For obtaining the best acceleration effect for each number of super clusters, we must still search for the optimal parameters  $l$  for CS-I and  $m$  for CS-II with respect to the number of super clusters. Then, we employ the best  $l$  and  $m$  for CS-H.

For deriving the optimal parameters  $m$  and  $l$ , we search for the best parameter values for a training set, then we validate these derived parameter values for a testing set. We repeat this for the five pairs of training sets and testing sets.

#### 5.3.1 Candidate selection by CS-I for the 2nd-layer

First, we find out the optimal value of  $n$  (number of super clusters) and the optimal value of  $l$

corresponding to  $n$  on the training set ( $Tr-1\sim Tr-5$ ).

For confirming these optimal values of the parameters we adopt these values to recognize the testing set ( $Tr-1\sim Tr-5$ ).

The results of experiments show that the best performance is obtained when  $n$  is around 40. Table 3 shows the optimal value of  $n$  and that of  $l$  for each training set and testing set when the original coarse classification rate and whole recognition rate are kept. From Table 1 and Table 3, we can confirm that the coarse classification time is reduced from 13 to 6.30 msec. (48.5% of the original) and the whole recognition time is reduced from 20.1 to 9.48 msec. (47.2% of the original) by setting  $n$  and  $l$  as around 40 and 29, respectively, while keeping the original coarse classification rate and whole recognition rate up to the third significant digit.

**Table 3. Performance on training/testing sets with the 2nd-layer divided into around 40 super clusters and SC-I used for candidate selection.**

Pattern set	$n$ : No. of super clusters	$l$ : No. of candidate super clusters	Coarse classification rate (%)		Coarse classification time(msec.)		Whole recognition rate (%)		Whole recognition time(msec.)	
			$Tr$	$Te$	$Tr$	$Te$	$Tr$	$Te$	$Tr$	$Te$
1	40	29	99.0	99.3	6.29	6.31	98.2	97.1	9.48	9.48
2	40	29	99.1	99.3	6.30	6.31	98.3	96.4	9.50	9.49
3	41	30	99.1	98.8	6.38	6.38	98.4	96.0	9.56	9.59
4	40	29	99.1	98.8	6.32	6.30	98.4	96.0	9.48	9.47
5	40	28	99.1	98.9	6.25	6.24	98.3	96.1	9.42	9.41
Av	40	29	99.1	99.0	6.30	6.30	98.3	96.3	9.49	9.49

#### 5.3.2 Candidate selection by CS-II for the 2nd-layer

Secondly, we test CS-II. We should find out the optimal parameter  $m$  corresponding to the optimal number of super clusters. We follow the same strategy as the above to find the best value of  $m$ .

Table 4 shows the optimal value of  $n$  and that of  $m$  for each training set and testing set when the original coarse classification rate and whole recognition rate are kept. From Table 1 and Table 4, we know that the best values of the parameters  $n$  and  $m$  are located around 40 and  $1.9\pm 0.025$ .

For every pair of training set and testing set, the coarse classification time is reduced to 47.9% of the original and the whole recognition time is reduced to 46.8% of the original, while the coarse classification rates and the whole recognition rates are kept to the third significant digit of their originals.

#### 5.3.3 Candidate selection by CS-H for the 2nd-layer

We apply CS-H for the 2nd-layer as well as the 1st-layer and the coarse classification. The necessary parameters for CS-H have been obtained from the above experiments. These are:

$l$  of CS-I is  $29\pm 1$ ,  $m$  of CS-II is  $1.9\pm 0.25$  and the

number of super cluster  $n$  is around 40 in the 2nd-layer;

**Table 4. Performance on training/testing sets with the 2nd-layer divided into around 40 super clusters and CS-II used for candidate selection.**

Pattern set	n: No. of super clusters	m: Multiplying coefficient	Coarse classification rate (%)		Coarse classification time(msec.)		Whole recognition rate (%)		Whole recognition time(msec.)	
			Tr	Te	Trl	Te	Tr	Te	Trl	Te
1	40	1.900	99.0	99.3	6.25	6.26	98.2	97.1	9.44	9.44
2	40	1.925	99.1	99.3	6.23	6.25	98.3	96.4	9.40	9.39
3	41	1.900	99.1	98.8	6.24	6.22	98.4	96.0	9.42	9.40
4	40	1.875	99.1	98.8	6.20	6.20	98.4	96.0	9.38	9.37
5	40	1.875	99.1	98.9	6.21	6.19	98.3	96.1	9.40	9.38
Av	40	1.895	99.1	99.0	6.23	6.22	98.3	96.3	9.41	9.40

$l$  is  $130 \pm 5$ ,  $m$  is  $1.775 \pm 0.025$  and  $n$  is 400 in the base-layer;  $l$  is 40 and  $m$  is 1.8 in the coarse classification without losing recognition rate.

Table 5 shows the results on the training sets and testing sets, respectively. The coarse classification is reduced from original 13 msec. to 6.13 msec. (47.1% of the original) and the whole recognition time is reduced from original 20.1 msec. to 9.28 msec. (46.2% of the original) while the coarse classification time and the whole recognition rate is kept to the third significant digit of the original.

## 6. Analysis on experiments

Fig. 3 shows the coarse classification time and the whole recognition time of the original recognizer and the recognizer used along with single-layered LSS (SSS) and that with two-layered LSS when CS-H is adopted to select candidates. It shows that the single-layered LSS (SSS) accelerates them significantly and that two-layered LSS accelerates them to some extent without degrading recognition performance. However, the acceleration effect of the 2nd-layer is not as high as the base-layer. The reason is that the distances between the pivots are much longer than that between the prototypes and the number of pivots is much less than that of prototypes. When we cluster the pivots into super clusters, the optimal number of super clusters becomes as small as 40 and the diameter of each super cluster become very large. Therefore, we must keep a high portion (around 29) of super clusters in order to obtain the necessary pivot, and the cluster represented by this pivot must include the right prototype. For the input pattern located at the boundary

**Table 5. Performance on testing sets with the 2nd-layer divided into around 40 super clusters and CS-H used for candidate selection.**

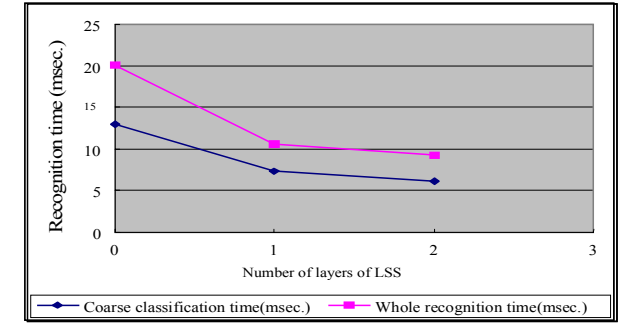
Pattern set	Coarse classification		Base layer		2nd layer			Coarse classification rate (%)		Coarse classification time (msec.)		Whole recognition rate (%)		Whole recognition time (msec.)		
	l	m	n	l	m	n	l	m	Tr	Te	Tr	Te	Tr	Te	Tr	Te
1	40	1.8	400	131	1.775	40	29	1.900	99.0	99.3	6.13	6.15	98.2	97.1	9.28	9.29
2				132	1.775	40	29	1.925	99.1	99.3	6.13	6.13	98.3	96.4	9.27	9.29
3				135	1.800	41	30	1.900	99.1	98.8	6.15	6.14	98.4	96.0	9.30	9.30
4				127	1.775	40	29	1.875	99.1	98.8	6.10	6.12	98.4	96.0	9.27	9.26
5				125	1.750	40	28	1.875	99.1	98.9	6.09	6.11	98.3	96.1	9.25	9.24
Av				130	1.775	40	29	1.895	99.1	99.0	6.12	6.13	98.3	96.3	9.27	9.28

of super clusters, a high portion of super clusters must be considered. For this reason, we do not consider to add another layer to the search space.

## Acknowledgement

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

## References



**Fig. 3. Performance of LSS with one and two layers.**

- [1] T. H. Hildebrandt, W. Liu: Optical recognition of handwritten Chinese characters: advances since 1980, *Pattern Recognition*, 26 (2), 205-225, 1993.
- [2] C. -L. Liu and M. Nakagawa: Precise candidate selection for large character set recognition by confidence evaluation, *IEEE PAMI*, 22 (6), 636-642, 2000.
- [3] T. Kumamoto, et al.: On speeding candidate selection in handprinted Chinese character recognition, *Pattern Recognition*, 24 (8), 793-799, 1991.
- [4] C. -H. Tung, et al.: Multi-stage pre-candidate selection in handwritten Chinese character recognition systems, *Pattern Recognition*, 27 (8), 1093-1102, 1994.
- [5] Y. Yang, et al.: Accelerating large character set recognition using pivots, *Proc. 7th ICDAR*, Edinburgh, 262-267, 2003.
- [6] Y. Yang, et al.: Structuring search space for accelerating large set character recognition, *IEICE*, E88-D (8), 1799-1806, 2005.
- [7] Y. Yang and M. Nakagawa: Improving the structuring search space method for accelerating large set character recognition, *Proc. 9th IWFHR*, Tokyo, 251-256, 2004.
- [8] Y. -H. Tseng et al.: Speeding up Chinese character recognition in an automatic document reading system, *Pattern Recognition*, 31(11), 1601-1612, 1998.
- [9] F. Fujimoto, et al.: Fast, precise pre-classification method using projections of feature regions (in Japanese), *PRMU*, 97-220, 1998-02.
- [10] J. Tsukumo and H. Tanaka: Classification of handprinted Chinese characters using non-linear normalization and correlation methods, *Proc. 9th ICPR*, Roma, 168-171, 1988.
- [11] F. Kimura: Modified quadratic discriminant function and the application to Chinese characters, *IEEE PAMI* 9 (1), 149-153, 1987.