

Effects of Line Densities on Nonlinear Normalization for Online Handwritten Japanese Character Recognition

Truyen Van Phan, JinFeng Gao, Bilan Zhu and Masaki Nakagawa

Department of Computer and Information Science

Tokyo University of Agriculture and Technology

Tokyo, Japan

E-mail: truyenvanphan@gmail.com

Abstract— In offline handwritten character recognition, the nonlinear normalization (NLN) method based on line density equalization has been proven very effective. This paper shows the effects on online handwritten Japanese character recognition. We apply the nonlinear normalization based on line density equalization to online character patterns. Since the curve-fitting-based normalization methods and their pseudo 2D extensions yields superior performance on offline patterns, we also combine these methods with the way using line density projection. We have compared the methods using trajectory-based projection with ones using line density projection. As a result, line density-based methods yield superior accuracy and a competitive time-complexity.

Keywords—Online character recognition; Line Density Equalization; Nonlinear Normalization; Pseudo 2D normalization.

I. INTRODUCTION

Online handwriting recognition generally exploits the time sequence information so that it is more robust against stroke catenation, running strokes and deforming of character patterns, but it is sensitive to stroke order variations. As it is well-known, the strengths of offline method can compensate for online method's weakness. Offline method-combined online recognition has been proposed and shows higher recognition accuracies [1][2]. Since the research of applying offline method to online character recognition has started.

As discussed, for offline Japanese/Chinese handwritten characters that contain multiples strokes, nonlinear normalization based on the line density equalization has been invented [3][4] and many variations have been proposed. The method by Tsukumo et al. [3] is still widely adopted. Despite its popularity, however, the normalized shapes are not smooth due to the local transformation nature. On the contrary, curve-fitting-based global transformation methods estimate a few parameters efficiently from global shape features and generate smoothly normalized shapes. Some recent new methods such as Moment (MN), Bi-moment (BMN), centroid-boundary alignment (CBA) and its modification (MCBA) also perform comparably with an improved version of Tsukumo et al.'s method (NLN-T) [5-8].

Most of nonlinear normalization in offline recognition is applied on a 2D image. Recently, Liu et al. has successfully applied curve-fitting-based nonlinear normalization and pseudo 2D normalization to online character recognition and obtained high performance [9]. However, line density-based normalization which is evidently superior to others in offline character recognition has not been tried yet. Therefore, we propose using line density in nonlinear normalization strategy. In all methods, we convert the online pattern to 2D image, and calculate projections of line density. While NLN-T directly uses line density projections for coordinate mapping, others use them to calculate centroid and moments.

This paper describes an effectiveness of nonlinear normalization using line density in online Japanese character recognition. To evaluate the recognition performance of the normalization methods, we use directional feature extraction and the modified quadratic discriminant function for classification [10]. We have experimented on the TUAT HANDS online handwritten Japanese character databases Kuchibue and Nakayosi [11] and confirmed that the accuracies of proposed methods are comparable to or higher than that of existing methods.

The remainder of this paper is organized as follows. Section 2 presents an overview of the recognition system. Section 3 describes the line density-based nonlinear normalization, curve-fitting-based nonlinear normalization, pseudo 2D normalization methods. Section 4 describes direction feature extraction. Section 5 presents the results of experiments and finally, Section 6 draws conclusion.

II. SYSTEM OVERVIEW

The recognition system for online handwritten Japanese characters consists of three main steps: 1) pre-processing, 2) feature extraction and 3) classification

An input pattern trajectory consists of a sequence of strokes, and each stroke consists of coordinates of sampled pen-points from pen-down to pen-up. The stroke smoothing is operated in the pre-processing step. Smoothing can reduce stroke shape variation in a small local region. In each stroke, except the start point and end point, we replace the coordinate of every point by the weighted average of that of its two neighbors and itself. Next, in normalization, the shape of a smoothed pattern is

transformed to a size-standardized and shape-regulated pattern. In this step, we apply line density-based normalization [3], curve-fitting-based normalization [5][6], and pseudo 2D normalization [7].

In feature extraction, the direction of normalized line segment and normalized gradient vector are used to extract 8-directional features in each of 8 x 8 regions, thus obtaining 512 features. Then, we apply the Fisher linear discriminant analysis (FLDA) [12] to reduce 512 dimensions to 160.

At last, we apply a two-stage classifier composed of coarse and fine classifiers. In coarse classification, we select 100 candidate classes according to Euclidean distance to class means. And in fine classification, the modified quadratic discriminant function (MQDF2) of Kimura et al. [10] is used to select the output class from candidate classes.

III. NONLINEAR NORMALIZATION

Normalization is used to reduce the shape variation between patterns of the same class. An original pattern $f(x,y)$ with width W_1 and height H_1 is transformed to a normalized pattern $f'(x',y')$ of standard size with width W_2 and height H_2 . While in offline, $f(x,y)$ is a pixel sampled from a continuous image plane, in online, this is a sampled pen-point of a line segment. Normalization is implemented by coordinate mapping of pixels or pen-points as follows:

$$\begin{cases} x' = u(x,y), \\ y' = v(x,y). \end{cases} \quad (1)$$

The shape of a normalized pattern depends on the coordinate mapping function $u(x,y)$ and $v(x,y)$. These functions are various in different methods.

A. Line Density-based Normalization

Linear normalization is not sufficient to absorb the shape variation of divergent writing styles. Consider that in Japanese/Chinese characters, the shape variation mainly lies in the ununiformity of stroke distribution; Tsukumo et al. [3] and Yamada et al. [4] proposed the nonlinear normalization using line density equalization. The NLN is implemented by mapping:

$$\begin{cases} x' = W_2 \sum_{i=0}^x h_x(i), \\ y' = H_2 \sum_{j=0}^y h_y(j). \end{cases} \quad (2)$$

where $h_x(i)$ and $h_y(j)$ are the normalized line density histograms of x axis and y axis, respectively, which are obtained by normalizing the projections of local line densities as follows:

$$\begin{cases} h_x(x) = \frac{p_x(x)}{\sum_x p_x(x)} = \frac{\sum_y d_x(x,y)}{\sum_x \sum_y d_x(x,y)}, \\ h_y(y) = \frac{p_y(y)}{\sum_y p_y(y)} = \frac{\sum_x d_y(x,y)}{\sum_y \sum_x d_x(x,y)}. \end{cases} \quad (3)$$

where $p_x(x)$ and $p_y(y)$ are line density projections onto x axis and y axis, respectively; and $d_x(x,y)$ and $d_y(x,y)$ are local line density functions. Tsukumo et al., take these functions as the reciprocal of horizontal/vertical run-length in background area, or a small constant in stroke area. On the other hand, Yamada et al., consider both background run-length and stroke run-length, and unify them to render $d_x(x,y) = d_y(x,y)$. The two methods give comparable performance but the method of Tsukumo et al. is more efficient in computation [13]. Moreover, an improved version of Tsukumo's method, which adjusts the density functions of marginal and stroke areas empirically, achieves better performance [7]. We use this version for calculating line density projection and compare with other methods.

For online patterns, to apply this method we convert an online pattern to a 2D image. We imagine that the character trajectory is laid on a grid. Each stroke (pen movement) of the trajectory is viewed as a sequence of line segments in an imaginary image (each pixel corresponding to a unit cell in the grid), each defined by a pair of consecutive sampled points. On the grid, each pixel crossed by a line segment is assigned black as Fig. 1.

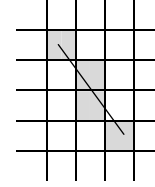


Figure 1. Pixels of a line segment

B. Curve-fitting-based Normalization

Three curve-fitting-based normalization methods using quadratic coordinate functions, called bi-moment normalization (BMN) [5], centroid-boundary alignment (CBA), modified CBA (MCBA), and line density projection fitting (LDPF) [6], have been proposed for offline character recognition. BMN, CBA and MCBA, except LDPF, are successfully applied to online character recognition [9]. The projections $f_x(x)$ and $f_y(y)$ are calculated directly from trajectory without converting to a 2D image. From projections the centroid and moment can be computed and used in these methods.

The BMN method is an extension of the moment normalization (MN) method [14], which aligns the centroid of an input pattern (x_c, y_c) to the center of a normalized pattern $(x'_c, y'_c) = (W_2/2, H_2/2)$ and rescales the pattern according to second order moments. In BMN, the rescaled width and height are treated asymmetric with respect to the centroid. Then the reset bounds and the centroid are used to estimate the quadratic functions for coordinate mapping.

The quadratic functions are also used to align the boundaries and centroid $(0, x_c, W_1)$ and $(0, y_c, H_1)$ with $(0, 0.5, 1)$ in the CBA method. While in the MCBA method,

to adjust the stroke density in the central area, a sine function is combined with the quadratic function. The MCBA method can be also used to fit the projections of line densities of NLN, called LDPF.

For these curve-fitting-based normalization methods, only MCBA has been combined with the density-based normalization (called as LDPF) [6]. In this paper, we combine several curve-fitting-based normalization methods with the density-based normalization and investigate the combination effects.

C. Pseudo 2D Normalization

In pseudo 2D normalization [7], the coordinate mapping functions, $u(x, y)$ and $v(x, y)$, are obtained by combining three 1D coordinate functions $u^{(i)}(x)$ with their weight functions $w^{(i)}(y)$, $i=1,2,3$. These functions are computed from three soft strips of the original pattern. The 1D functions are combined by:

$$u(x, y) = \begin{cases} w^{(1)}(y)u^{(1)}(x) + w^{(2)}(y)u^{(2)}(x), & y < y_c, \\ w^{(3)}(y)u^{(3)}(x) + w^{(2)}(y)u^{(2)}(x), & y \geq y_c. \end{cases} \quad (4)$$

Similarly, the 2D function $v(x, y)$ can be obtained.

Corresponding to the method of calculating 1D coordinate mapping functions, we can extend to pseudo 2D. As a result, we have 5 pseudo 2D normalization methods: P2DMN, P2DBMN, P2DCBA, P2DMCBA, and P2DLDPF. The weight ω_0 in (9) is set to 0.75. Note that Liu et al. did not extend the CBA method to pseudo 2D, and instead referred to the pseudo 2D extension of MCBA as P2DCBA [7][8].

Except for LDPF and P2DLDPF, because other methods of curve-fitting-based normalization and pseudo 2D normalization use projections calculated directly from trajectories for an online character pattern or an image for offline character pattern, we call those trajectory-based projection methods for online patterns. Fig. 2 shows 4 online character patterns and their 8 normalized patterns by these methods.

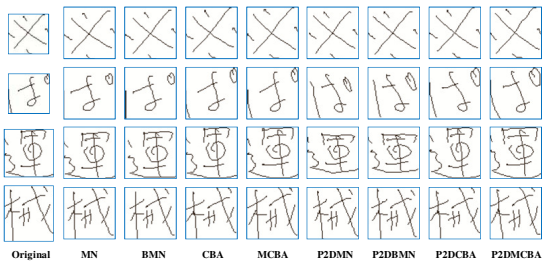


Figure 2. Original pattern and normalized patterns by methods using trajectory-based projection.

D. Line Density-combined Normalization

The basic concept of this method is to use line density projection in every method. MN, BMN, CBA, MCBA are using pixel intensity (for offline image) or histogram of trajectory (for online pattern), $f_x(x)$ and $f_y(y)$. Now we try

to use line density projection $p_x(x)$ and $p_y(y)$ to calculate a centroid and moments in these methods. We call them line density-based projection methods. Obviously, computational costs of such methods are more expensive than original ones.

Then we can call LDPF is the extension of MCBA in this way. Extensions of others are called MN-T, BMN-T, CBA-T, and P2DMN-T, P2DBMN-T, P2DCBA-T for pseudo 2D methods, respectively. We also implement NLN-T as a 1D method and line density projection interpolation (LDPI) as a pseudo 2D method that combines NLN-T with the pseudo 2D normalization [7]. The normalized patterns by them are shown in Fig. 3.

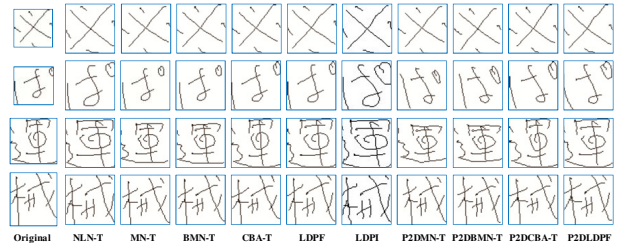


Figure 3. Original pattern and normalized patterns by methods using line density-based projection.

IV. DIRECTION FEATURE EXTRACTION

Direction feature extraction is accomplished in three steps: directional decomposition, blurring and sampling. The local stroke segments are assigned to a number of direction planes, then each plane is blurred (low-pass filtered) and the sampled values are taken as feature values.

A. Direction decomposition

After normalization, we get a normalized pattern. We implement two methods to extract feature form this one. While gradient feature extraction (GRD-FE) needs a pattern to be converted to a 2D image, directional line segment feature extraction (SEG-FE) can directly extract feature from pattern trajectory.

We use eight direction planes, corresponding to eight chain code directions as shown (Fig. 4(a)). Following the method by Kawamura et al. [15], each line segment or gradient vector is decomposed into two components in two neighboring chain code directions (Fig. 4(b)).

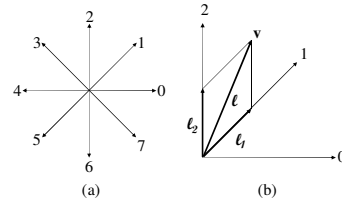


Figure 4. Eight chain code directions (a) and directional decomposition of a line segment/gradient vector

For SEG-FE, each line segment in the normalized pattern trajectory $\mathbf{v} = ((x_1, y_1), (x_2, y_2))$ is decomposed into two components as in Fig. 2(b), with lengths ℓ_1

(direction 1) and ℓ_2 (direction 2). The corresponding two direction planes are given weights $w_1 = \ell_1/\ell$ and $w_2 = \ell_2/\ell$ (ℓ is the length of \mathbf{v}). Then in the plane of direction 1 and 2, each pixel overlapping with \mathbf{v} is given a value of overlapping length times w_1 and w_2 , respectively.

In GRD-FE, by using the Sobel operator, the gradient vector, computed on the normalized image converted from normalized pattern, is decomposed into components in eight chain code directions. The gradient vector $\mathbf{g}(x, y) = [g_x, g_y]^T$ at a pixel (x, y) in a normalized image is computed by:

$$\begin{aligned} g_x(x, y) &= f(x+1, y-1) + 2f(x+1, y) + f(x+1, y+1) \\ &\quad - f(x-1, y-1) - 2f(x-1, y) - f(x-1, y+1) \\ g_y(x, y) &= f(x-1, y+1) + 2f(x, y+1) + f(x+1, y+1) \\ &\quad - f(x-1, y-1) - 2f(x, y-1) - f(x+1, y-1) \end{aligned} \quad (5)$$

The gradient strength and direction can be computed from the vector $[g_x, g_y]^T$. The length of the vector on each component as shown in Fig. 2(b) is assigned to the corresponding direction plane.

B. Blurring and sampling

On directional decomposition, each direction plane is blurred using a low-pass Gaussian filter. The pixel values are sampled uniformly, and according to the Sampling Theorem, the variant parameter σ_x is related to the sampling interval between blurring masks t_x [16]:

$$\sigma_x = \frac{\sqrt{2}t_x}{\pi} \quad (6)$$

We set the size of normalized plane and direction plane to 64x64 pixels, the sampling interval to 8. As a result, we obtain 64 feature values from each direction plan and totally 512 feature values. The extracted feature values are causal variables. Power transformation is used to improve the Gaussianity of feature distribution [12]. We set the power of variable transformation to 0.5.

V. PERFORMANCE EVALUATION

To compare the performance between methods on normalization and feature extraction, we have experimented on the TUAT HANDS - online Japanese character databases, Kuchibue and Nakayosi [11]. As many previous work did [9][17][18], we also experimented with 3,345 classes of JIS level-1 Kanji character (2965 classes) and kana, alpha numerals, symbols and so on (380 classes). We used the samples of Nakayosi for training and Kuchibue for testing. In Nakayosi, there are 9,309 patterns for the JIS level-1 Kanji character and symbols, while in Kuchibue, the number is 11,951. Similarly, we also experimented with 2,965 JIS level-1 Kanji characters only.

For reducing the classifier complexity and improving classification accuracy, the 512-dimension of the feature vector is transformed to 160-dimension by Fisher linear discriminant analysis (FLDA) [12].

For classification, we use two classifiers: the Euclidean distant to the class mean (minimum distance classifier) and the MQDF2 [10]. On the reduced vector, we select 100 candidate classes according to the Euclidean distance. The MQDF2 is then computed on the candidate classes only. We use 50 principal eigenvectors for each class, make the minor eigenvalue class-independent and optimize it via holdout cross-validation on the training dataset.

The experiment compares the accuracy and time complexity of the above methods.

A. Accuracies

The normalization and feature extraction methods are evaluated on 3,345 classes of Kanji, kana, alpha numerals, symbols and so on, and also on 2,965 classes of Kanji only. The normalization methods are divided into two groups, the one using trajectory-based projection and the other using line density-based projection. We evaluate these in two methods of feature extraction, SEG-FE and GRD-FE. TABLE I shows the test accuracies for 3,345 classes and TABLE II shows the results for 2,965 classes.

TABLE I. ACCURACIES (%) FOR 3,345-CLASS KANJI AND SYMBOL

3,345 classes		SEG-FE		GRD-FE	
1D	P2D	1D	P2D	1D	P2D
Trajectory-based Projection					
MN	P2DMN	91.01	91.07	90.93	90.77
BMN	P2DBMN	91.02	91.02	90.80	90.72
CBA	P2DCBA	91.08	91.49	91.15	91.45
MCBA	P2DMCBA	91.34	91.65	91.47	91.55
Line Density-based Projection					
MN-T	P2DMN-T	91.54	90.99	91.43	90.77
BMN-T	P2DBMN-T	91.34	91.01	91.12	90.71
CBA-T	P2DCBA-T	91.10	91.49	91.19	91.46
LDPF	P2DLDPF	91.48	91.67	91.65	91.60
NLN-T	LDPI	91.59	92.08	91.82	92.10

TABLE II. ACCURACIES (%) FOR 2,965-CLASS KANJI

2,965 classes		SEG-FE		GRD-FE	
1D	P2D	1D	P2D	1D	P2D
Trajectory-based Projection					
MN	P2DMN	97.66	98.38	97.66	97.92
BMN	P2DBMN	97.70	98.40	96.97	97.97
CBA	P2DCBA	97.97	98.10	97.34	97.52
MCBA	P2DMCBA	98.16	98.33	97.57	97.85
Line Density-based Projection					
MN-T	P2DMN-T	98.21	98.39	97.68	97.92
BMN-T	P2DBMN-T	98.21	98.41	97.70	97.95
CBA-T	P2DCBA-T	97.97	98.09	97.33	97.51
LDPF	P2DLDPF	98.15	98.34	97.61	97.84
NLN-T	LDPI	98.26	98.45	97.72	98.00

From the results, we can firstly see that for most of the methods, using line density-based projection has higher accuracy than using trajectory-based projection. Second, In our implementation, while the accuracies of CBA and MCBA are higher than MN and BMN in trajectory-based projection, the opposite results are obtained in line density-based projection. Third, NLN-T and LDPI are the best in all cases. Forth, about the feature extraction, SEG-FE is effective with Kanji, but it is not so effective as GRD-FE in whole.

B. Time complexity

Although the effect of line density is undoubted, it is more time-consuming. To evaluate the computational complexity of normalization methods, we profile the processing time in two sub-tasks of preprocessing: smoothing and normalization. From an input pattern, the CPU time is counted until the normalized coordinates are computed. It does not cover either the normalized image generation (for GRD-FE) or the feature extraction procedure. We implemented the experiments on Intel® Core™2 Quad CPU Q9400 2.66GHz. The averaged CPU time is calculated over the samples from 3,345 classes of both Kuchibue and Nakayosi (1,517,367 + 1,434,120 = 2,951,487 samples).

TABLE III. AVERAGE CPU TIME (MS) FOR NORMALIZATIONS

	Trajectory-based Projection		Line Density-based Projection		
	1D	P2D		1D	P2D
MN	0.007	0.020	MN-T	0.038	0.052
BMN	0.008	0.024	BMN-T	0.041	0.052
CBA	0.008	0.022	CBA-T	0.038	0.052
MCBA	0.017	0.038	LDPF	0.048	0.069
			NLN-T	0.039	0.047

The CPU time is shown in TABLE III. Normalization methods using line density computation are more expensive than those using trajectory-based computation. MN, BMN, CBA seem to have the same speed. Especially, the LDPI method which yields the best accuracy is as expensive as these methods in line density-based projection group. Moreover, P2DMCBA which has the best rate among the trajectory-based projection group has the same time cost with NLN-T. Overall, we can use NLN-T or even LDPI in place of P2DMCBA and others in the line density-based projection group.

VI. CONCLUSION

We implemented normalization methods with or without using line density projection to evaluate the effect of line density to online character recognition. The comparison of normalization methods shows that NLN-T and most of curve-fitting-based methods which use line density projection have higher accuracies than ones which simply use trajectory-based projection. Not only for offline character patterns but also for online character patterns, the line density-based methods yield high accuracy. Although they are more time consuming, the computational cost is acceptable due to improved performance. In this work, with line density projection, we reach a superior rate of over 92.10%. We intend to modify the feature extraction for better performance in future work.

ACKNOWLEDGMENT

This work is being supported by the R&D fund for "Development of pen & paper based user interaction" under Japan Science and Technology Agency. The authors thank Prof. Cheng-Lin Liu and Dr. Xiang-Dong Zhou for their helpful discussions.

REFERENCES

- [1] H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama, M. Nakagawa, "Hybrid Pen-input Character Recognition System based on Integration of On-line and Off-line Recognition," Proc. 5th ICDAR, Bangalore, India, 1999, pp.209-212.
- [2] H. Oda, B. Zhu, J. Tokuno, M. Onuma, A. Kitadai, M. Nakagawa, "A compact on-line and off-line combined recognizer," Proc. 10th IWFHR, La Baule, France, 2006, pp. 133-138.
- [3] J. Tsukumo, H. Tanaka, "Classification of handprinted Chinese characters using non-linear normalization and correlation methods," Proc. Ninth International Conference on Pattern Recognition, Roma, Italy, 1988, pp. 168-171.
- [4] H. Yamada, K. Yamamoto, T. Saito, "A nonlinear normalization method for hanprinted Kanji character recognition line density equalization," Pattern Recognition 23(9) (1990) 1023-1029.
- [5] C.-L. Liu, H. Sako, H. Fujisawa, "Handwritten Chinese character recognition: alternatives to nonlinear normalization," Proc. 7th ICDAR, Edinburgh, Scotland, 2003, pp.524-528.
- [6] C.-L. Liu, K. Marukawa, H. Sako, H. Fujisawa, "Global shape normalization for handwritten Chinese character recognition: a new method," Proc. 9th IWFHR, Tokyo, 2004, pp.300-305.
- [7] C.-L. Liu, K. Marukawa, "Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition," Pattern Recognition, 38(12): 2242-2255, 2005.
- [8] C.-L. Liu, "Handwritten Chinese character recognition: Effects of shape normalization and feature extraction," In: Arabic and Chinese Handwriting Recognition, S. Jaeger and D. Doermann (Eds.), LNCS Vol.4768, Springer, 2008, pp.104-128.
- [9] C.-L. Liu, X.-D. Zhou, "Online Japanese character recognition using trajectory-based normalization and direction feature extraction," Proc. 10th IWFHR, France, 2006.
- [10] F. Kimura, K. Takashina, S. Tsuruoka, Y. Miyake, "Modified quadratic discriminant functions and the application to Chinese character recognition," IEEE Trans. PAMI, vol. 9, no 1, pp.149-153, 1987.
- [11] M. Nakagawa, K. Matsumoto, "Collection of on-line handwritten Japanese character pattern databases and their analysis," Int. J. Document Anal. Recognit. 7(1), 69-81, 2004.
- [12] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd edition, Academic Press, 1990.
- [13] S.-W. Lee, J.-S. Park, "Nonlinear shape normalization methods for the recognition of large-set handwritten characters," Pattern Recognition 27 (7) (1994) 895-902.
- [14] R.G. Casey, "Moment normalization of handprinted character", IBM J. Res. Develop. 14 (1970) 548-557.
- [15] A. Kawamura, et al., "On-line recognition of freely handwritten Japanese characters using directional feature densities," Proc. 11th ICPR, The Hague, 1992, Vol.2, pp.183-186.
- [16] C.-L. Liu, Y.-J. Liu, R.-W. Dai, "Preprocessing and statistical/structural feature extraction for handwritten numeral recognition," in Progress of Handwriting Recognition, A.C. Downton and S. Impedovo (Eds.), World Scientific, 1997, pp.161-168.
- [17] C.-L. Liu, S. Jaeger, M. Nakagawa, "Online recognition of Chinese characters: the state-of-the-art," IEEE Trans. Pattern Anal. Mach. Intell., 26(2): 198-213, 2004.
- [18] A. Kitadai, M. Nakagawa, "A learning algorithm for structural character pattern representation used in on-line recognition of handwritten Japanese characters," Proc. 8th IWFHR, Ontario, Canada, 2002, pp.163-168.