IJDAR (2013) 16:273–283
DOI 10.1007/s10032-012-0191-y

ORIGINAL PAPER

An evaluation of statistical methods in handwritten hangul recognition

Gyu-Ro Park · In-Jung Kim · Cheng-Lin Liu

Received: 13 March 2012 / Revised: 11 July 2012 / Accepted: 17 August 2012 / Published online: 13 September 2012

Abstract Although structural approaches have shown better performance than statistical ones in handwritten Hangul recognition (HHR), they have not been widely used in practical applications because of their vulnerability to image degradation and high computational complexity. Statistical approaches have not received high attention in HHR because their early trials were not promising enough. The past decade has seen significant improvements in statistical recognition in handwritten character recognition, including handwritten Chinese character recognition. Nevertheless, without a systematic evaluation on the effects of statistical methods in HHR, they cannot draw enough attention because of their discouraging experience. In this study, we comprehensively evaluate state-of-the-art statistical methods in HHR. Specifically, we implemented fifteen character normalization methods, five feature extraction methods, and four classification methods and evaluated their performances on two public handwritten Hangul databases. On the SERI database, statistical methods achieved the best performance of 93.71 % accuracy, which is higher than the best result achieved by structural recognizers. On the PE92 database, which has small number of samples per class, statistical methods gave slightly lower performance than the best structural recognizer.

Keywords Handwritten hangul recognition · Statistical methods · Character normalization · Feature extraction · Classification

Abbreviations

HHR	Handwritten hangul recognition
HCCR	Handwritten Chinese character recognition
LN	Linear normalization
LDE/PDE	Line/pixel density equalization
LDPF/PDPF	Line/pixel density projection fitting
MN	Moment normalization
BMN	Bi-moment normalization
CBA	Centroid-boundary alignment
MCBA	Modified CBA
LDPI/PDPI	Line/pixel density projection interpolation
NBFE	Normalization-based feature extraction
NCFE	Normalization-cooperated feature extraction
MDC	Minimum distance classifier
QDF	Quadratic discrimination function
MQDF	Modified QDF
DLQDF	Discriminative learning QDF

1 Introduction

Character recognition technology has been applied in many fields. For alpha-numeric and Chinese characters, recognition methods have matured enough to achieve high accuracy on not only printed but also handwritten characters. However, handwritten Hangul recognizers still cannot provide sufficient performance for practical applications. The major difficulty of handwritten Hangul recognition (HHR) comes from a multitude of confusing characters and excessive cursiveness in writing.

G.-R. Park · I.-J. Kim (✉)
School of CSEE, Handong Global University,
Heunghae-eup, Bukgu, Pohang, Gyeongbuk 791-708,
Republic of Korea
e-mail: ijkim@handong.edu

G.-R. Park
e-mail: tachyon.park@gmail.com

C.-L. Liu
National Laboratory of Pattern Recognition, Institute of Automation
of Chinese Academy of Sciences, 95 Zhongguancun East Road,
Beijing 100190, People's Republic of China
e-mail: liucl@nlpr.ia.ac.cn

Researchers have developed various methods to recognize handwritten Hangul characters. Among them, structural methods have reported the best results [1–3]. However, they have not been widely used in real fields because of several practical limitations: They are vulnerable to image degradation and require heavy computation in structural matching. The learning of structural models also remains open. On the other hand, statistical recognizers have been widely applied in handwritten Chinese character recognition (HCCR) and have shown high performance [4–6]. Nevertheless, they have not received high attention in HHR, because their early trials showed much poorer performance than those of structural recognizers [7].

The past decade has seen significant improvements in statistical recognition methods, especially in HCCR. Particularly, wisely designed character normalization and feature extraction algorithms as well as discriminative classifier learning algorithms were found effective to alleviate shape variations and to improve discrimination ability [5,6,8–10]. We suggest that most of the recent improvements in statistical recognition methods are also applicable to HHR. However, without a systematic evaluation, they cannot draw enough attention in HHR because of their discouraging experience in the past.

In this study, we comprehensively evaluate the effects of state-of-the-art statistical methods in HHR. Specifically, we implemented fifteen character normalization methods, five feature extraction methods, and four classification methods known effective to HCCR and evaluated their performances on two public Hangul databases. We compare the best performance achieved by statistical methods with that of the best structural recognizers reported so far. The experimental results show that in addition to their computational efficiency, statistical recognition methods can perform competitively as structural recognizers in HHR. This has not been reported previously.

The rest of this paper is organized as follows: Sect. 2 briefly reviews previous works on HHR and related statistical recognition methods. Section 3 describes the recognition system used in our evaluation study. Sections 4–6 respectively explain the character normalization methods, feature extraction methods, and classification methods evaluated in this study. Section 7 presents our experimental results, and finally, concluding remarks are provided in Sect. 8.

2 Related works

Character recognition methods can be grouped into two categories: structural and statistical. Structural methods describe the input character as strokes or contour segments and identify the class by matching with the structural models of candidate classes. The structural models are usually built by hand because off-the-shelf algorithms for automatic structural learning are not available. Also, the structural matching, as a combinatorial optimization problem, is computationally expensive. On the other hand, statistical methods represent the character image as a feature vector and classify the feature vector using statistical classifiers (in general sense, including all classifiers that work on feature vectors). There are many algorithms for statistical classifier learning, and the classification of vectors is computationally efficient. Although statistical methods are known better in recognition of many other languages, structural methods outperformed statistical methods in HHR.

Kim and Kim proposed a structural method based on hierarchical random graph representation for HHR [1]. Given a character image, they extract strokes and represented into an attributed graph, which is matched with character models using a bottom-up matching algorithm. Kang and Kim proposed an improved method by modeling between-strokes relationship [2]. Jang proposed a post-processing method for the methods of [1] and [2] to improve discrimination ability [3]. The post-processor consists of a set of pair-wise discriminators, each specialized for a pair of graphemes with similar shapes.

Some researchers tried statistical methods to recognize handwritten Hangul. Bae et al. proposed an HHR method based on neural networks [11]. They first classify the input image into one of six predefined types by a neural network. Then, on extracting features from the input image using dynamic bars pursuant to the character type, the feature vector is classified using a secondary neural network specialized to the character type. Kim et al. developed a recognizer using a hierarchical interactive neural network [12]. Jeong developed a handwritten Hangul recognizer using a clustering algorithm and a set of neural network classifiers [7]. The method of [11] and [12] reported recognition rates 85.8 and 95 %, respectively, on different datasets of small sizes. The study of [7] reported recognition performance evaluated on a well-known public Hangul database PE92, which is significantly lower than those of the structural recognizers mentioned above even though it considered a smaller number of classes than the other works. On the public database PE92, the best performance was 87.7 % reported in [2]. On another public Hangul database SERI, also known as KU-1, the best performance was 93.4 % reported in [3].

Despite the superior performance achieved by structural methods in HHR, statistical methods are popularly used for the recognition of other scripts, including handwritten Chinese character recognition (HCCR) [6]. In recent years, there have been significant improvements in statistical recognition methods. Particularly, classification algorithms based on the quadratic discriminant function (QDF) and the modified QDF (MQDF, proposed by Kimural et al. [14]) have reported superior performance. Liu et al. proposed an improved

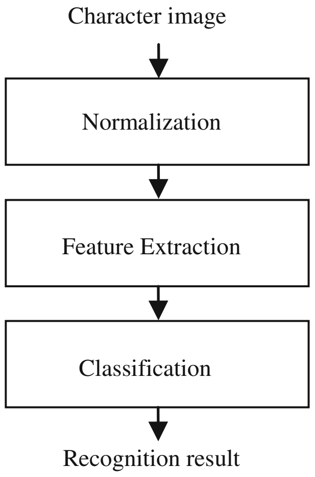

Fig. 1 Block diagram of recognition system

version of MQDF called discriminative learning QDF (DLQDF) [8]. As well, there were significant advances in the methods of character normalization [9,10,15–19] and feature extraction [4,20], which improves the recognition performance via reshaping the distributions of classes in the feature space and improving separability.

3 Statistical handwritten hangul recognition system

In order to evaluate the performance of various statistical recognition methods, we built an experimental recognition system as shown in Fig. 1. It consists of three main steps: character normalization, feature extraction, and classification. Normalization is to regulate the size and alleviate the shape variation of the character image. Feature extraction is to represent the normalized image as a feature vector reflecting the characteristics of the character shape. Classification is to select a class label as the recognition result of the input character by analyzing the feature vector. Each step has multiple options of implemented methods. In this study, we implemented fifteen normalization methods, five feature extraction methods, and four classification methods and evaluated their performances on two public handwritten Hangul databases. The implemented algorithms are briefly explained in the following Sections.

4 Normalization methods

Normalization is a transformation from an input character image to another image with standard size and reduced shape variation. Denoting the input and the output images by $f(x, y)$ and $g(x', y')$, respectively, a normalization algorithm is implemented by a coordinate mapping from a coordinate (x, y) on $f(\cdot)$ to its counterpart (x', y') on $g(\cdot)$ as

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

For easing the computation of coordinate mapping and alleviating the shape distortion in normalization, many early normalization algorithms used 1D mapping functions

$$\begin{cases} x' = x'(x), \\ y' = y'(y). \end{cases} \tag{2}$$

In the following, we briefly describe popular 1D normalization algorithms, and then their 2D extensions.

4.1 1D normalization methods

Linear normalization (LN) is the simplest normalization algorithm, which regulates the size and aspect ratio of character image. Imagine that both the input character image and the normalized image are enclosed by bounding boxes. Denote the width and height of the input image as W_1 and H_1, and those of the normalized image as W_2 and H_2; the coordinate mapping functions of LN are

$$\begin{cases} x' = \frac{W_2}{W_1} x, \\ y' = \frac{H_2}{H_1} y. \end{cases} \tag{3}$$

Linear normalization does not change the relative position and the density of strokes, and therefore, is limited in regulating the character shape. On the other hand, nonlinear normalization algorithms regulate the character shape as well as the size. The nonlinear normalization algorithm based on line density equalization [15,16] has been shown very effective and has been widely used in HCCR. Its coordinate mapping functions can be represented as

$$\begin{cases} x' = W_2 \sum\limits_{u=0}^{x} h_x(u), \\ y' = H_2 \sum\limits_{v=0}^{y} h_y(v), \end{cases} \tag{4}$$

where $h_x(x)$ and $h_y(y)$ are normalized line (or pixel) density histograms along the x-axis and y-axis, respectively. Denoting by $d_x(x, y)$ and $d_y(x, y)$ as the horizontal and vertical local line (or pixel) densities, and $p_x(x)$ and $p_y(y)$ as their projections onto x and y axes, respectively, the normalized density histograms are obtained by

$$\begin{cases} h_x(x) = \frac{p_x(x)}{\sum_u p_x(u)}, \\ h_y(y) = \frac{p_y(y)}{\sum_v p_y(v)}, \end{cases} \tag{5}$$

where $p_x(x) = \sum_v d_x(x, v) + \alpha$ and $p_y(y) = \sum_u d_y(u, y) + \beta$ are projections; α and β are used to remedy the rows or columns of zero density projection. They usually take zero for line density and nonzero (2 in our experiments) for pixel density.

The definitions of the local density functions $d_x(x, y)$ and $d_y(x, y)$ are variable. In the pixel density equalization (PDE), either $d_x(x, y)$ or $d_y(x, y)$ is simply one for foreground pixels and zero for background pixels. In the line density equalization (LDE), the local line density functions can be obtained in several ways. Among them, Tsukumo and Tanaka's method showed a good performance at a reasonable computation cost in a previous study [17]. It computes the horizontal/vertical line densities $d_x(x, y)$ and $d_y(x, y)$ by the reciprocal of horizontal/vertical run-length in the background area or takes a small constant in the foreground area.

The line density projection fitting (LDPF) method is an alternative of the line density equalization [9]. With the density projections $h_x(x)$ and $h_y(y)$, it fits the accumulated densities $\sum_{u=0}^{x} h_x(u)$ and $\sum_{v=0}^{y} h_y(v)$ with a pair of quadratic functions. Then, the quadratic functions substitute the accumulated density functions in (4). The resulting coordinate mapping functions are smoother than those of the density equalization, and therefore, the normalized image has smoother stroke shapes.

The moment normalization (MN) aligns the centroid (x_c, y_c) of the input image to the geometric center of the normalized image $(x'_c, y'_c) = (W_2/2, H_2/2)$ and re-bounds the input image according to the second-order 1D moments [13]. Denoting the second-order central moments as μ_{20} and μ_{02}, and letting $\delta_x = 4\sqrt{\mu_{20}}$ and $\delta_y = 4\sqrt{\mu_{02}}$ as the re-set character width and height, the coordinate mapping functions are

$$\begin{cases} x' = \frac{W_2}{\delta_x}(x - x_c) + x'_c, \\ y' = \frac{H_2}{\delta_y}(y - y_c) + y'_c. \end{cases} \tag{6}$$

The moment normalization (MN) is actually a linear transformation. Its difference from the simple linear normalization (LN) lies in the centroid alignment and character re-bounding. The alignment of centroid is particularly effective to reduce the within-class shape variation.

The bi-moment normalization (BMN) [9] is a nonlinear extension of the MN. It also aligns the centroid of the input image, but the width and height are treated asymmetrically with respect to the centroid. In BMN, the second-order moments are split into two parts at the centroid: $\mu_x^-, \mu_x^+, \mu_y^-$, and μ_y^+. The boundaries of the input image are re-set to $\left[x_c - 2\sqrt{\mu_x^-},\ x_c + 2\sqrt{\mu_x^+}\right]$ and $\left[y_c - 2\sqrt{\mu_y^-},\ y_c + 2\sqrt{\mu_y^+}\right]$. The x-coordinate mapping function is defined using a quadratic function $u(x) = ax^2 + bx + c$ that aligns three points $\left(x_c - 2\sqrt{\mu_x^-}, x_c, x_c + 2\sqrt{\mu_x^+}\right)$ to the normalized coordinates (0, 0.5, 1), respectively. Similarly, the coordinate mapping is defined using a quadratic function $v(y)$ that works for the y-axis. With $u(x)$ and $v(y)$, the coordinate mapping functions of BMN are

$$\begin{cases} x' = W_2 u(x), \\ y' = H_2 v(y). \end{cases} \tag{7}$$

The centroid-boundary alignment (CBA) algorithm [9] aligns the physical boundaries (spread limits of stroke pixels) and centroid, that is, maps $(0, x_c, W_1)$ and $(0, y_c, H_1)$ to (0,0.5,1), using a pair of quadratic functions. A modified version of CBA (MCBA) [10] further adjusts the stroke density in the central area by combining sine functions with the quadratic functions as

$$\begin{cases} x' = W_2[u(x) + \eta_x \sin(2\pi u(x))], \\ y' = H_2[v(y) + \eta_y \sin(2\pi v(y))], \end{cases} \tag{8}$$

where the amplitudes of the sine waves, η_x and η_y, are estimated from the extent of the central area, defined by the centroid of the partial images divided by the global centroid.

In the above 1D normalization methods, the LN and MN are linear transformation methods, while the line/pixel density equalization methods (LDE, PDE), LDPF, BMN, CBA and MCBA methods are nonlinear ones. The 1D coordinate mapping functions using these methods can be extended to 2D functions using the pseudo normalization strategy introduced below.

4.2 Pseudo 2D normalization methods

Although 1D normalization algorithms are simple and fast, their shape restoration capacity is limited because the pixels on the same row/column on the input image are mapped to the same row/column on the normalized image. Pseudo 2D normalization algorithms overcome this limitation while controlling the excessive shape distortion of character images by smoothing the 2D coordinate mapping functions.

Horiuchi et al. proposed a 2D extension of nonlinear normalization based online density equalization [18]. Instead of 1D line density projection, they equalized horizontal/vertical local line densities of each row/column. And to avoid excessive shape distortion, they smoothed the local line densities with a Gaussian filter. This pseudo 2D LDE method results in improved recognition performance but is computationally expensive.

The pseudo 2D normalization method based on line density projection interpolation (LDPI) was shown to yield comparable recognition performance with the above 2D extension by Gaussian smoothing at much lower computation cost [19]. The LDPI method gives 2D coordinate mapping function by combining three 1D mapping functions with a parameterized weighting function. For x-coordinate mapping, the input image is vertically divided into three overlapping soft horizontal strips. Given the local horizontal density function $d_x(x, y)$ of the input image, the local density function of each strip is obtained by

$$d_x^{(i)}(x, y) = w^{(i)}(y) d_x(x, y), \quad i = 1, 2, 3, \tag{9}$$

where $w^{(i)}(y)$, $i = 1, 2, 3$, are piecewise linear weight functions for the strips. A pre-defined constant $w_0 > 0$ is used

Table 1 Summary of normalization methods

Type	1D	Pseudo 2D
Linear normalization	Linear normalization (LN)	–
Density equalization	Line/pixel density equalization (LDE/PDE)	Pseudo 2D LDE/PDE (P2DLDE/P2DPDE)
		Line density projection interpolation (LDPI)
	Line density projection fitting (LDPF)	Pseudo 2D LDPF (P2DLDPF)
Moment-based normalization	Moment normalization (MN)	Pseudo 2D MN (P2DMN)
	Bi-moment normalization (BMN)	Pseudo 2D BMN (P2DBMN)
Centroid-boundary alignment	Centroid-boundary alignment (CBA)	–
	Modified CBA (MCBA)	Pseudo 2D MCBA (P2DCBA)

in the weighted functions to control the flexibility of shape transformation (details in [19]). The horizontal density functions of the three strips are then projected onto the x-axis as

$$p_x^i(x) = \sum_v d_x^i(x, v), \quad i = 1, 2, 3. \tag{10}$$

From the density projection of each strip, a 1D coordinate mapping function $x'^{(i)}(x), i = 1, 2, 3$, is obtained using an 1D normalization method (one introduced in Sect. 4.1). Finally, the three 1D coordinate functions are combined into 2D mapping function by interpolation as

$$x'(x, y) = \begin{cases} w^{(1)}(y)x'^{(1)}(x) + w^{(2)}(y)x'^{(2)}(x), & y < y_c, \\ w^{(3)}(y)x'^{(3)}(x) + w^{(2)}(y)x'^{(2)}(x), & y \geq y_c. \end{cases} \tag{11}$$

The 2D coordinate mapping function $y'(x, y)$ for the y-axis is obtained similarly by dividing the input image into soft vertical strips and combine three 1D coordinate mapping functions $y'^{(i)}(y)$ using weight functions $w^{(i)}(x), i = 1, 2, 3$.

By projection interpolation, only three 1D coordinate mapping functions are computed and smoothed for either the x-coordinate or y-coordinate. Hence, its computation cost is significantly lower than the 2D extension by Gaussian smoothing, which computes and smoothes the 1D coordinate mapping functions of each row and each column. Moreover, the projection interpolation strategy can be flexibly combined with any 1D normalization method, which is used to compute the 1D coordinate mapping function for each strip. The extension of 1D normalization methods to pseudo 2D methods is summarized in Table 1. The pseudo 2D extension of the LN and CBA is not implemented since they are not among the top performing ones. It is noteworthy that the P2DLDE (line density equalization) and P2DPDE (pixel density equalization) are based on 2D extension by Gaussian smoothing (Horiuchi et al. [18]), while the other pseudo 2D algorithms are based on projection interpolation.

Figure 2 shows the normalized images of an input character image using the 1D normalization methods and pseudo 2D methods. It can be seen that pseudo 2D normalization methods better equalize stroke densities than 1D methods but sometimes they yield excessive shape distortion.

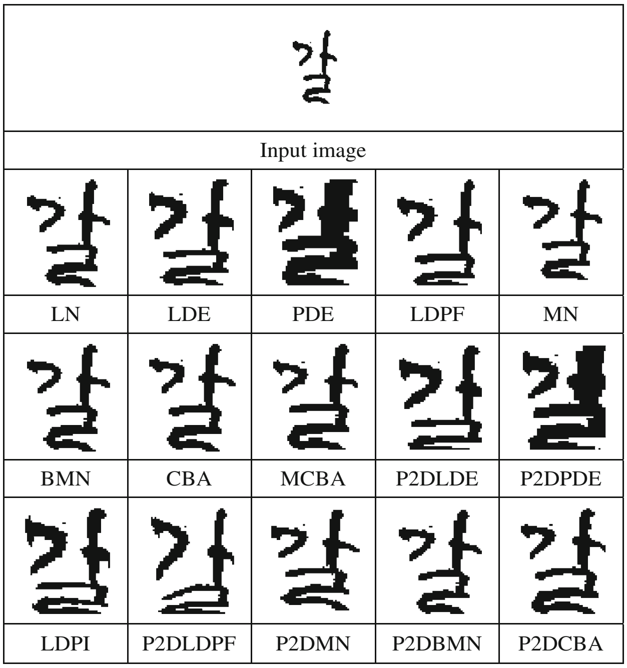

Fig. 2 Normalized images using 1D and pseudo 2D normalization methods

5 Feature extraction methods

Although numerous types of features have been proposed for character recognition, the orientation/direction histograms of contour chaincode or gradient is dominant and among the best-performing ones [6]. The feature extraction process usually consists of two stages: orientation/direction decomposition and feature blurring/sampling. In the first stage, the contour or edge pixels of the character image are assigned to a number of orientation/direction planes. Decomposition into 4 or 8 directions is popularly adopted. In the second stage,

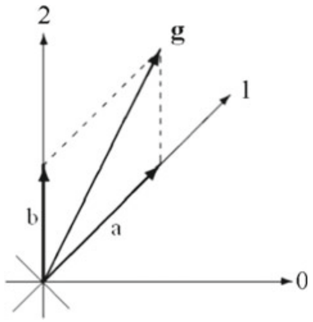

Fig. 3 Decomposition of gradient vector

each plane is convolved with a Gaussian blurring mask (low-pass filter) to extract feature values.

The chaincode direction is determined through contour tracing but can be equivalently done by raster scanning. The gradient direction feature is more robust against noise because the gradient is computed from a neighborhood, often using the Sobel operator. The decomposition of gradient into 8 standard directions (corresponding to the 8 chaincode directions) is briefly outlined hereon. At a pixel (x, y), its gradient vector $\mathbf{g} = (g_x, g_y)$ computed by the Sobel operator is decomposed into its two neighboring standard directions using the parallelogram rule as shown in Fig. 3. The amplitudes (corresponding to the lengths) of the two sub-vectors (a and b in Fig. 3) are added to the corresponding direction plane at the pixels of the same location (x, y). For obtaining 4 orientation planes, every two direction planes of opposite directions (e.g., left and right) are merged to one.

From an orientation/direction plane, the feature values are the sampled pixel values after Gaussian filtering. This is equivalent to convolve the plane with a Gaussian blurring mask (impulse response function) centered at the locations of sampling points. The variance parameter of the Gaussian filter can be empirically estimated from the sampling interval [4]. At each sampling point, the feature values of multiple orientations/directions can be viewed as the elements of a local histogram.

Conventionally, feature extraction is performed after character normalization, that is, features are extracted from the normalized image. This procedure is called normalization-based feature extraction (NBFE). For orientation/direction histogram feature extraction, chaincode/gradient direction decomposition can be performed directly on the input image. In this case, the contour/edge direction of original image is assigned to direction planes. The normalized image is not generated, but the coordinate mapping functions are used in direction decomposition: to assign the direction amplitude of pixel (x, y) in input image to pixel (x', y') of direction planes. This strategy is called normalization-cooperated feature extraction (NCFE) [20]. It has two advantages: saves computation of normalization and overcomes direction distortion caused by normalization. NCFE was initially proposed for contour direction feature, and Liu proposed an NCFE method for gradient direction feature, called normalization-cooperated gradient feature extraction (NCGFE) [21]. The NCGFE has an alternative that extracts the normalized gradient direction of input image (according to coordinate mapping functions, again not need to generate normalized image). This is called as normalized direction NCGFE (nNCGFE).

In this study, we implemented both chaincode and gradient direction features using either NBFE or NCFE. The variations of features are summarized in Table 2.

6 Classification methods

For the classification of handwritten Hangul recognition, we evaluated some statistical classifiers that have demonstrated superior in HCCR. Particularly, the MQDF proposed by Kimura et al. [14] is dominantly used in HCCR and is among the best performing one. It is based on Bayesian decision by assuming multivariate Gaussian density for each class. To modify the quadratic discriminant function (QDF) resulted from Gaussian density, the smallest eigenvalues of the covariance matrix of each class are replaced by a constant. Denoting the d-dimensional feature vector by $\mathbf{x}$, the MQDF of class $\omega_i (i = 1, 2, \ldots, M)$ is

$$
\begin{aligned}
g_2(\mathbf{x}, \omega_i) &= \sum_{j=1}^{k} \frac{1}{\lambda_{ij}} \left[(x - \mu_i)^T \phi_{ij} \right]^2 \\
&\quad + \frac{1}{\delta_i} \left\{ ||x - \mu_i||^2 - \sum_{j=1}^{k} \left[(x - \mu_i)^T \phi_{ij} \right]^2 \right\} \\
&\quad + \sum_{j=1}^{k} \log \lambda_{ij} + (d - k) \log \delta_i, \qquad (12)
\end{aligned}
$$

where μ_i is the mean vector of class ω_i, λ_{ij}, and ϕ_{ij}, $j = 1, 2, \ldots, d$, are the eigenvalues (sorted in nonascending

Table 2 Feature extraction methods

Feature type	NBFE	NCFE
Chaincode	Normalization-based chaincode feature extraction (NBCFE)	Normalization-cooperated chain code feature extraction (NCCFE)
Gradient	Normalization-based gradient feature extraction (NBGFE)	Normalization-cooperated gradient feature extraction (NCGFE)
		Normalized direction NCGFE (nNCGFE)

order) and their corresponding eigenvectors of the covariance matrix of class ω_i. It is seen that by replacing the smallest eigenvalues with a constant δ_i, the corresponding eigenvectors are not necessarily stored and computed in the discriminant function. The regulation of smallest eigenvalues also helps alleviate the curse of dimensionality, and the shortage of training sample data, and consequently, improves the generalized classification performance.

Two artificial parameters of MQDF are the number k of retained principal eigenvectors per class and the constant δ_i substituting smallest eigenvalues. The former parameter is determined empirically: try several values and select the one that gives nearly optimal performance. For determining the constant eigenvalue δ_i, a strategy is to hypothesize multiple values of class-independent constant and select one by cross-validation on the training dataset [6]. We used five-fold holdout partitioning of the training data for saving the computation of cross-validation on large dataset.

The MQDF is a generative model with parameters estimated by maximum likelihood, which does not consider the boundary between classes. The discriminative learning QDF (DLQDF) [8] is an improved version of MQDF by discriminative optimization of the parameters under a classification-oriented objective such as the minimum classification error (MCE) criterion [22]. More details can be found in [8].

Besides the powerful MQDF, the nearest prototype classifier is also frequently used for its low computation cost. This class of classifier includes the nearest class means, multiple prototypes estimated by clustering, and supervised prototype learning by learning vector quantization [23]. We use a recently proposed prototype learning algorithm called log-likelihood of margin (LOGM) [24].

In classification, we also reduce the dimensionality of feature vectors by subspace projection, with the subspace parameters learned by the Fisher linear discriminant analysis (FDA). Dimensionality reduction helps reduce the computation cost of classifier learning and classification and often improves the classification performance.

7 Experiments

We evaluated the normalization, feature extraction, and classification methods on two public handwritten Hangul datasets: SERI and PE92 [25]. The SERI database, also known as KU-1, consists of 520 most frequently used classes, and each class has about 1,000 samples. The PE92 database contains 2,350 classes, and each class has about 100 samples. For each database, 90 % of samples per class were used for training, and the other 10 % of samples were used for testing. Table 3 shows the numbers of classes and samples used in our experiments, and Fig. 4 shows some samples of the two databases. The experiments were performed on a PC that has an Intel Q6600 CPU (2.4 GHz) and 4 GB memory.

Table 3 Specification soft wo public databases

Database	# Classes	# Samples	# Training	# Test
SERI	520	517,535	465,598	51,937
PE92	2,350	221,207	197,887	23,320

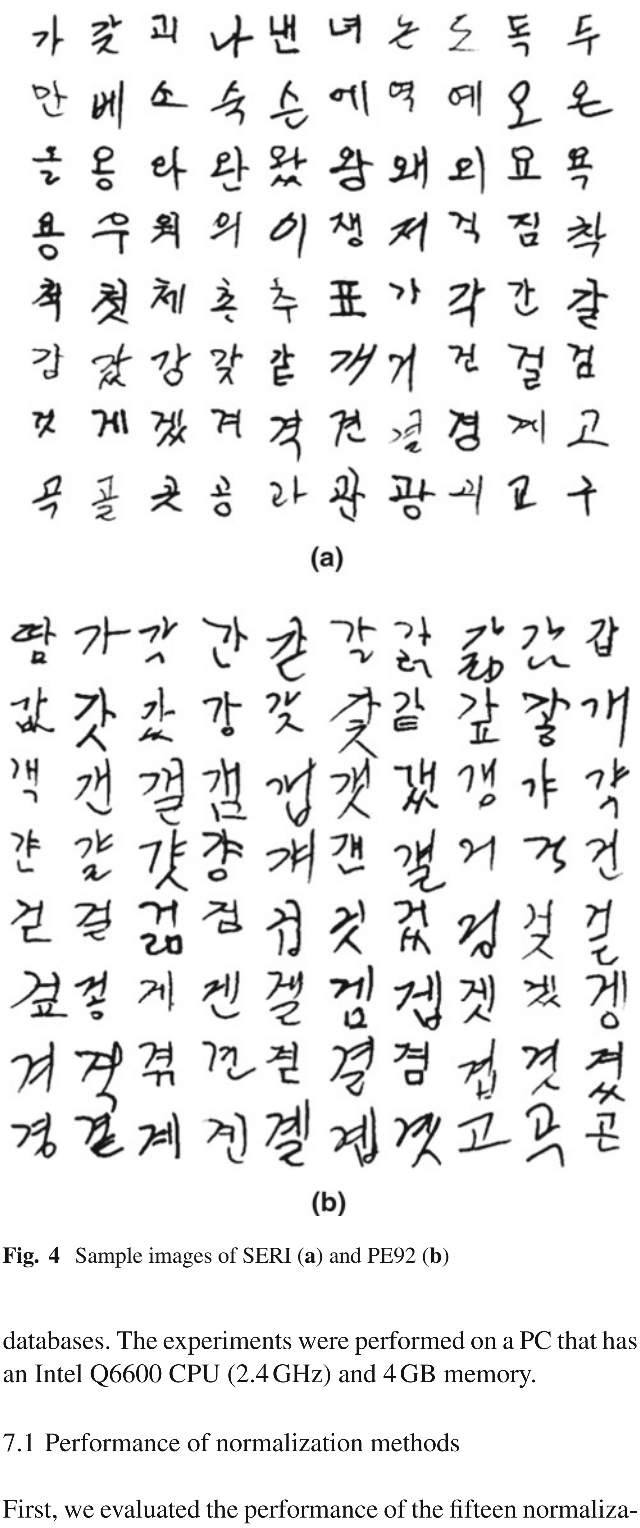

Fig. 4 Sample images of SERI (**a**) and PE92 (**b**)

7.1 Performance of normalization methods

First, we evaluated the performance of the fifteen normalization methods described in Sect. 4 on the SERI database using a standard feature extraction method (NCGFE) and classifier (MQDF). The NCGFE was shown to perform best in HCCR [21]. We set the size of normalized image (direction planes) as 64 × 64 pixels; from each plane, 8 × 8 feature values are extraction by Gaussian blurring. Thus, the dimensionality of feature vector is 512. The feature vectors are reduced to 160D subspace by FDA (as often done in HCCR [6]).The MQDF uses $k = 60$ principal eigenvectors per class. We use the SERI database for evaluation because it has a larger number of samples than the PE92 database, and thus, the recognition result is more confident.

Table 4 shows the test accuracies on the SERI database using different normalization methods, the second column shows the results of 1D normalization methods, and the fourth column shows the results of pseudo 2D methods. It is evident that the pseudo 2D methods all outperform their 1D counterparts. The best performance, test accuracy 93.01 %, was given by the pseudo 2D methods LDPI and P2DBMN.The P2DLDE performs comparatively well, giving test accuracy 92.95 %.

Table 5 shows the average computation time for coordinate mapping of the normalization methods. We only show the coordinate mapping time because the normalized image is not necessarily generated for normalization-cooperated feature extraction (NCFE). Generally, the pseudo 2D normalization methods are slower than their corresponding 1D counterparts. Especially, the P2DLDE and P2DPDE algorithms (2D extensions by Gaussian smoothing) are very computationally expensive compared with the other algorithms. The pseudo 2D algorithms based on projection interpolation are only slightly more costly than the 1D nonlinear normalization method LDE. When the normalized image is to be generalized, the 1D normalization methods cost 0.012 ms, while the pseudo 2D normalization methods cost 0.473 ms.

Table 4 Test accuracies of normalization methods on SERI database

Method	Accuracy (%)	Method	Accuracy (%)
LN	88.32		
LDE	91.65	P2DLDE	92.95
PDE	91.09	P2DPDE	92.25
LDPF	91.20	**LDPI**	**93.01**
MN	91.76	P2DLDPF	92.66
BMN	91.90	P2DMN	92.83
CBA	91.30	**P2DBMN**	**93.01**
MCBA	91.53	P2DCBA	92.84

The boldface names and numbers indicate the best performance and the method that provided it

Table 5 Average CPU times for coordinate mapping of normalization methods

Method	Time (ms)	Method	Time (ms)
LN	0.007		
LDE	0.061	P2DLDE	1.880
PDE	0.021	P2DPDE	1.961
LDPF	0.073	LDPI	0.098
MN	0.020	P2DLDPF	0.136
BMN	0.021	P2DMN	0.064
CBA	0.016	P2DBMN	0.065
MCBA	0.036	P2DCBA	0.096

7.2 Performance of feature extraction methods

Using the best normalization method P2DBMN and the MQDF classifier ($k = 60$), we then evaluated the five feature extraction methods described in Sect. 5. In all cases of feature extraction, the normalized plane size remains 64 × 64, and 8 × 8 feature values are extracted from each of 8 direction planes, resulting 512D feature vector. And the feature vectors are reduced to 160D subspace by FDA. The test accuracies on the SERI database are shown in Table 6. We can see that the NCFE methods (NCCFE, NCGFE, nNCGFE) outperform the NBFE methods (NBCFE, NBGFE), and the gradient feature NCGFE gives the best performance of 93.01 % test accuracy. The comparison of feature extraction methods is again similar to the results reported in HCCR in [21]. We did not compare computational costs of the feature extraction methods, because they were already compared very well in [21] and the computational cost is independent of the database or character set.

7.3 Performance of classification methods

On selecting the best normalization (P2DBMN) and feature extraction (NCGFE) methods, we evaluated three classification algorithms: MQDF, DLQDF, and nearest prototype classifier (NPC) under the Euclidean distance metric. Both the SERI and PE92 databases were evaluated in this case.

Before we compare the classification methods, we measured the performance of MQDF with variable FDA subspace dimensionality and principal eigenvector number k. For the

Table 6 Test accuracies of feature extraction methods on SERI database

Feature extraction	Accuracy (%)
NBCFE	91.52
NBGFE	92.19
NCCFE	92.64
NCGFE	**93.01**
nNCGFE	92.64

The boldface names and numbers indicate the best performance and the method that provided it

SERI database, the subspace dimensionality varies from 100 to 200 by step 20, and k from 40 to 80 by step 10. For the PE92 database, because each class has less than 100 training samples, we use MQDF classifier with smaller k (from 20 to 60) and lower dimensional subspaces (from 80 to 180). The test accuracies of two databases are shown in Tables 7 and 8, respectively. We can see that on the SERI database (with large number of samples per class), the best performance was obtained on 120D subspace, and the classifier MQDF with larger k gives higher performance. While on the PE92 database (with small number of samples per class), the best performance was obtained on 80D subspace, and the classifier MQDF with smaller k gives higher performance, with best performance given by $k = 30$.

For evaluating the performance of NPC and DLQDF, we chose the best-performing subspace dimensionality 120 for the SERI database and 80 for the PE92 database. Each class was learned 1, 2, 3, 4, 5 prototypes by k-means clustering and by supervised learning algorithm LOGM [2]. The test accuracies on two databases are shown in Table 9. We can see that supervised prototype learning LOGM yields significantly higher accuracies than k-means clustering. LOGM yielded the highest accuracy 91.58 % on the SERI database and 82.50 % on the PE92 database. In comparison, the accuracies of the nearest mean classifier (one prototype by k-means) are 85.88 % on the SERI database and 80.85 % on the PE92 database.

Table 7 Test accuracies of SERI database by MQDF with varying k and subspace dimensionality d

SERI	$k = 40$	$k = 50$	$k = 60$	$k = 70$	$k = 80$
$d = 100$	92.91	93.03	93.03	92.95	92.95
$d = 120$	93.05	93.11	93.14	**93.15**	**93.15**
$d = 140$	92.94	93.05	93.10	93.09	93.09
$d = 160$	92.89	93.00	93.01	93.04	93.08
$d = 180$	92.72	92.83	92.92	92.99	93.04
$d = 200$	92.59	92.75	92.86	92.92	92.95

The boldface names and numbers indicate the best performance and the method that provided it

Table 8 Test accuracies of PE92 database by MQDF with varying k and subspace dimensionality d

PE92	$k = 20$	$k = 30$	$k = 40$	$k = 50$	$k = 60$
$d = 80$	85.41	**85.45**	84.28	84.28	84.28
$d = 100$	85.13	85.37	84.90	84.90	84.90
$d = 120$	84.77	85.16	85.10	84.94	84.94
$d = 140$	84.13	84.45	84.55	84.37	84.37
$d = 160$	83.67	83.82	83.89	83.92	83.92
$d = 180$	83.20	83.06	83.24	83.30	83.30

The boldface names and numbers indicate the best performance and the method that provided it

Table 9 Test accuracies using NPC with variable prototype number per class

	# Prototype/class	1	2	3	4	5
SERI	k-means	85.88	87.92	88.71	89.04	**89.47**
	LOGM	89.91	91.07	91.45	**91.58**	91.55
PE92	k-means	80.85	81.58	**81.60**	81.22	80.70
	LOGM	**82.50**	82.19	81.70	81.77	81.28

The boldface names and numbers indicate the best performance and the method that provided it

Table 10 Highest text accuracies (%) of three classifiers on two databases

Classifier	SERI	PE92
NPC	91.58	82.50
MQDF	93.15	85.45
DLQDF	**93.71**	**85.99**

The boldface names and numbers indicate the best performance and the method that provided it

The DLQDF used the parameters of MQDF ($d = 120, k = 60$ for SERI and $d = 80, k = 30$ for PE92) as initial values, which are updated discriminatively on the training dataset. As result, the test accuracies of DLQDF are 93.71 % on SERI and 85.99 on PE92; both are higher than the performance of MQDF. The highest accuracies of three classifiers are collected in Table 10. Computational costs of the classification methods are presented in Table 11. NPC classifiers were much faster than QDF-based classifiers.

Finally, we compare the best results of our methods (best combination of P2DBMN, NCGFE and DLQDF classification) on the two databases with those reported in previous literatures. The compared accuracies are listed in Table 12. On the SERI database, the performance achieved in this study is slightly better than the best structural recognizer [3]. On the PE92 database, the accuracy of our approach is higher than that of the structural recognizer in [1] but is lower than that of another structural recognizer in [2]. The proposed statistical approach could not achieve higher accuracy on the PE92 because the training dataset is small (less than 100 samples per class). Higher accuracies can be expected if increasing the training sample size with either real samples or synthesized samples.

8 Conclusion

In this paper, we comprehensively analyzed the effects of recently emerged statistical recognition methods in handwritten Hangul recognition. We evaluated fifteen normalization methods, five feature extraction methods, and three classification methods on two well-known public handwritten

Table 11 Computational costs of classification methods

	Parameters	Training (s)	Test (ms)	Remarks
(a) SERI database, 120D subspace				
NPC *k*-means	5 prototypes	242.1	0.859	
NPC LOGM	4 prototypes	3,451.7	0.655	
MQDF	60 eigen vectors	212.9	2.115	100 candidates selected by Euclidian distance for acceleration
DLQDF	60 eigen vectors	22,000	2.219	
(b) PE92 database, 80D subspace				
NPC *k*-means	3 prototypes	84.7	0.964	
NPC LOGM	1 prototype	974	0.473	
MQDF	30 eigen vectors	107.6	1.156	100 candidates selected by Euclidian distance for acceleration
DLQDF	30 eigen vectors	6,497	1.162	

Table 12 Recognition accuracies (%) compared with previous results

Classifier	DB	
	SERI	PE92
Kim and Kim [1]	86.30	82.20
Kang and Kim [2]	90.30	**87.70**
Jang [3]	93.40	N/A
Proposed	**93.71**	85.99

The boldface names and numbers indicate the best performance and the method that provided it

Hangul databases. The highest accuracies were achieved by combining P2DBMN, NCGFE, and DLQDF classifier. The highest test accuracy on the SERI database achieved by the proposed statistical approach is 93.71 %, which is higher than the best result in the literature. The highest accuracy on the PE92 database is 85.99 %, which is slightly lower than the best previous result. These results demonstrate that the state-of-the-art statistical methods can be as competent as structural methods in HHR, which was not confirmed in previous works. We expect that large training dataset and more advanced classification/learning algorithms can yield even higher recognition accuracies in handwritten Hangul recognition.

Acknowledgments The work of Cheng-Lin Liu was supported by the National Natural Science Foundation of China (NSFC) Grants 60825301 and 60933010. The work of In-Jung Kim and Gyu-Ro Park was financially supported by the Ministry of Education, Science Technology (MEST) and National Research Foundation of Korea (NRF) through the Human Resource Training Project for Regional Innovation.

References

1. Kim, H.Y., Kim, J.H.: Hierarchical random graph representation of handwritten characters and its application to Hangul recognition. Pattern Recognit. **34**(2), 187–201 (2001)
2. Kang, K.-W., Kim, J.H.: Utilization of hierarchical, stochastic relationship modeling for Hangul character recognition. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1185–1196 (2004)
3. Jang, S.I.: Post-processing of Handwritten Hangul Recognition Using Pair-Wise Grapheme Discrimination. Master Thesis, KAIST (2002)
4. Liu, C.-L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten digit recognition: investigation of normalization and feature extraction techniques. Pattern Recognit. **37**(2), 265–279 (2004)
5. Liu, C.-L.: High accuracy handwritten Chinese character recognition using quadratic classifiers with discriminative feature extraction. In: Proceedings of the 18th ICPR, vol. 2, pp. 942–945. Hong Kong (2006)
6. Liu, C.-L.: Handwritten Chinese character recognition: effects of shape normalization and feature extraction. In: Jaeger, S., Doermann, D. (eds.) Arabic and Chinese Handwriting Recognition, LNCS, pp. 104–128. Springer, Berlin (2008)
7. Jeong, S.H.: Handwritten Hangul recognition based on character cluster segmentation. Technical Memo, Electronics and Telecommunication Research Institute, Taejon (2002)
8. Liu, C.-L., Sako, H., Fujisawa, H.: Discriminative learning quadratic discriminant function for handwriting recognition. IEEE Trans. Neural Netw. **15**(2), 430–444 (2004)
9. Liu, C.-L., Sako, H., Fujisawa, H.: Handwritten Chinese character recognition: alternatives to nonlinear normalization. In: Proceedings of the 7th ICDAR, pp. 524–528. Edinburgh, Scotland (2003)
10. Liu, C.-L., Marukawa, K.: Global shape normalization for handwritten Chinese character recognition: a new method. In: Proceeding of the 9th IWFHR, pp. 300–305. Tokyo, Japan (2004)
11. Bae, H.J., Yun, J.M., Cha, E.Y.: Neural network for hand-written character recognition using dynamic bar method. Proc. Korea Inf. Sci. Autumn Conf. **17**(2), 251–254 (1990)
12. Kim, M.W., Jang, J.S., Lim, C.D., Song, Y.S., Kim, J.H.: Improvements to a hierarchical interaction neural network for context-dependent pattern recognition and its experimentation with handwritten Korean character recognition. Technical Report, Electronics and Telecommunication Research Institute, Taejon, Korea (1992)
13. Casey, R.G.: Moment normalization of handprinted character. IBM J. Res. Dev. **14**, 548–557 (1970)
14. Kimura, F., Takashina, K., Tsuruoka, S., Miyake, Y.: Modified quadratic discriminant functions and the application to Chinese character recognition. IEEE Trans. Pattern Anal. Mach. Intell. **9**(1), 149–153 (1987)
15. Tsukumo, J., Tanaka, H.: Classification of handprinted Chinese characters using non-linear normalization and correlation

methods. In: Proceedings of the 9th ICPR, pp. 168–171. Rome, Italy (1988)

16. Yamada, H., Yamamoto, K., Saito, T.: A nonlinear normalization method for handprinted Kanji character recognition–line density equalization. Pattern Recognit. **23**(9), 1023–1029 (1990)
17. Lee, S.-W., Park, J.-S.: Nonlinear shape normalization methods for the recognition of large-set handwritten characters. Pattern Recognit. **27**(7), 895–902 (1994)
18. Horiuchi, T., Haruki, R., Yamada, H., Yamamoto, K.: Two dimensional extension of nonlinear normalization method using line density for character recognition. In: Proceedings of the 4th ICDAR, pp. 511–514. Ulm, Germany (1997)
19. Liu, C.-L., Marukawa, K.: Pseudo two-dimensional shape normalization methods for handwritten Chinese character recognition. Pattern Recognit. **38**(12), 2242–2255 (2005)
20. Hamanaka, M., Yamada, K., Tsukumo, J.: Normalization-cooperated feature extraction method for handprinted Kanji character recognition. In: Proceedings of the 3rd IWFHR, pp. 343–348. Buffalo, NY (1993)
21. Liu, C.-L.: Normalization-cooperated gradient feature extraction for handwritten character recognition. IEEE Trans. Pattern Anal. Mach. Intell. **29**(8), 1465–1469 (2007)
22. Juang, B.-H., Katagiri, S.: Discriminative learning for minimum error classification. IEEE Trans. Signal Process. **40**, 3043–3054 (1992)
23. Kohonen, T.: The self-organizing map. Proc. IEEE **78**(9), 1464–1480 (1990)
24. Jin, X.-B., Liu, C.-L., Hou, X.: Regularized margin-based conditional log-likelihood loss for prototype learning. Pattern Recognit. **43**(7), 2428–2438 (2010)
25. Kaist AI lab. Homepage. http://ai.kaist.ac.kr/Resource/dbase/Image20Database.htmHangulCharacter