

Sensing Analysis of Feature Extraction Types for Handwritten Character Recognition

Jin-feng Gao,^{1,2} Ru-xian Yao,^{2*} Yu Zhang,¹
Han Lai,² Jun-ming Zhang,¹ and Ting-Cheng Chang^{2**}

¹School of Information Engineering, Huanghuai University, Zhumadian, Henan 463000, China

²Henan Key Laboratory of Smart Lighting, Zhumadian, Henan 463000, China

(Received September 27, 2019; accepted March 12, 2020)

Keywords: handwritten character recognition, line density direction feature, gradient direction feature, convNet-based feature extraction

The line density direction (LDD) feature, gradient direction (GD) feature, and deep convolution neural network-based (convNet-based) feature widely employed in handwritten character sensing recognition have acceptable accuracies. The convNet-based method determines feature expression not only from a raw pattern image but also from domain-specific LDD and GD knowledge. These methods are named convNet-based-Raw, convNet-based-LDD, and convNet-based-GD, respectively. In this paper, we present an independent sensing analysis of the five features under identical working conditions considering the preprocessing and algorithm implementation of two handwritten character databases: CASIA-HWDB1.0 (Chinese) and TUAT HANDS (Japanese). The experimental results demonstrate that convNet-based feature extraction is more robust and discriminating than LDD and GD, two traditional methods for both handwritten Chinese character recognition (HCCR) and handwritten Japanese character recognition (HJCR). Furthermore, the convNet-based-GD has the highest accuracy for both HCCR and HJCR among the three convNet-based feature extraction methods. Compared with the traditional methods, LDD and GD, the best accuracies when using convNet-based-GD are improved by 3.04 and 2.31% for HCCR, and 3.15 and 1.54% for HJCR, respectively. Similarly, compared with the two other convNet-based methods, convNet-based-Raw and convNet-based-LDD, the best accuracies are improved by 0.44 and 0.25% for HCCR, and 0.65 and 0.08% for HJCR, respectively. Experimental comparisons of sensing analysis results are acceptable and valuable.

1. Introduction

With the extensive use of electronic pen input devices, personal digital assistants (PDAs), smartphones, and handheld devices, human-like freehand input is currently being strongly demanded by users. The touch sensor of handheld devices detects the moving path of the finger or pen-based input device to obtain the coordinate sequence of a handwritten input character. Then, the handwritten character pattern is recognized as the corresponding character by a

*Corresponding author: e-mail: yaoruxian@126.com

**Corresponding author: 18250922163@163.com

<https://doi.org/10.18494/SAM.2020.2688>

trained character classifier. Feature extraction from the handwritten character pattern during character recognition not only improves the character recognition accuracy but also reduces the complexity of character recognition. Nowadays, the methods of extracting the line direction density (LDD) feature, gradient direction (GD) feature, and convolution neural network-based (convNet-based) feature are the leading techniques for handwritten character recognition. To obtain robust feature expression by using the advantage of each feature extraction, it is necessary to study the sensing sensitivity of each feature extraction type for handwritten character recognition. In this paper, we focus on the sensing analysis of feature extraction for handwritten Chinese and Japanese character recognition.

Handwritten Chinese character recognition (HCCR) has been studied for more than fifty years. It is a challenge to deal with the large number of character classes, confusion between similar characters, and distinct handwriting styles across individuals.⁽¹⁾ It is identical to handwritten Japanese character recognition (HJCR). On the basis of the type of input data, handwriting recognition can be divided into online and offline. Offline HCCR and HJCR have found many applications, such as mail sorting and bank check reading. Online HCCR and HJCR have been widely applied to, for example, pen input devices, PDAs, smartphones, and computer-aided education. Moreover, HCCR and HJCR respectively are also a critical and integral part of handwritten Chinese and Japanese text recognition, in which segmentation and recognition are considered simultaneously. High character recognition accuracy is essential for the success of handwritten text/string recognition.⁽²⁾

The online method involves an online input handwritten character pattern, which is a time sequence of pen-tip coordinates, and although it can be easily made robust to stroke connection and deformation, the online method is sensitive to stroke order variation. On the other hand, in the offline method, the handwritten character is compared with an offline pattern, which is a character pattern image, and although it is insensitive to stroke order variation or duplicated strokes, it is not very robust to stroke connection and deformation. The offline recognition method is combined with the online process to form a combined recognizer for handwritten character recognition.⁽³⁾

Although the accuracy of handwritten character recognition depends on the combined contributions of pattern normalization, feature extraction, and recognizer training, a stronger discriminating feature expression (feature extraction) is crucial to handwritten character classification.⁽⁴⁾ In online handwritten character recognition, it is more effective and efficient to use extracted feature points instead of all the pen-tip points to express character patterns. The direction feature, inspired by the stroke segments of Chinese characters, can be approximated into four orientations (horizontal, vertical, left-diagonal, and right-diagonal), and was further employed to significantly improve the accuracy of online handwritten character recognition. In offline handwritten character recognition, wavelet transforms and a Gabor filter are successively applied to offline-written English, Indian, and Chinese character recognition.⁽⁵⁾ Similar to the online method, the trajectory-based LDD feature and GD feature methods also showed superiority in offline handwritten character recognition, especially in HCCR and HJCR.⁽⁶⁾ The LDD and GD features are extracted from online written character patterns for online handwritten character recognition and from handwritten character pattern images for offline handwritten character recognition, respectively.

As for online recognition, to achieve elastic matching between feature points, dynamic programming (DP) matching was introduced.⁽⁷⁾ Thus, its greedy variation, named linear-time flexible matching, is combined with structured character representation, which is used with the effect of memory reduction and consistency against style variations. The hidden Markov model (HMM) and Markov random field (MRF) after DP matching are successively applied in online handwritten character recognition.⁽⁸⁾ Historically, the handwritten character recognizer, which employs the nearest-neighbor method, was first employed for optical character recognition (OCR) in the offline process, but was replaced by the modified quadratic discriminant function (MQDF). Discriminative classifiers such as a neural network (NN) and a support vector machine (SVM) are also candidates, but statistical classifiers can generalize better than discriminative ones when the number of sample patterns is limited.⁽⁹⁾ Both Chinese and Japanese have thousands of character categories, thus statistical MQDF, rather than discriminative classifiers such as SVM, is the optimal selection.

In recent years, the deep convolution neural network (convNet) has been successfully used in HCCR and HJCR. Since it considers the feature extraction and classifier training as an entire system to simultaneously optimize the parameters, its recognition performance is further improved. Although the deep convNet can directly learn discriminative representations from a raw pattern image,⁽¹⁰⁾ the well-studied domain-specific knowledge shown is helpful also for improving the performance of HCCR.⁽¹¹⁾ A direction feature, such as LDD or GD, is the most critical domain-specific knowledge.⁽¹²⁾

These previous results are applied to a different framework, using only one evaluation database without considering the characteristics of the handwritten character database. Moreover, although combining domain-specific knowledge with deep convNet creates a new benchmark in HCCR, similar research has not been carried out in HJCR. The previous works mainly reported on the progress in the accuracy of the handwritten character recognition system, but the comparisons of the performances of different feature extractions are seldom made. Therefore, more comprehensive comparative experiments of different feature extractions, i.e., LDD, GD, and convNet-based methods, and their variations for different evaluation databases under the same framework, are necessary and valuable.

The rest of this paper is organized as follows. In Sect. 2, we depict the evaluation databases. In Sects. 3 and 4, we describe the procedures of the three types of feature extraction and feature reduction. Then, two types of handwritten character recognizer are introduced in Sect. 5. In Sects. 6 and 7, we show some experimental results with further analytical experiments. Finally, some conclusions are drawn in Sect. 8.

2. Overall System and Evaluation Database

The handwritten character recognition system consists of two phases: classifier training and testing. The process of classifier training involves four steps, as shown in Fig. 1. The left side of Fig. 1 shows the dataflow in the training of the classifier, whereas the right side shows the corresponding modules. In the first step, the input pattern is normalized by nonlinear modified centroid-boundary alignment (NLNMCBA).⁽⁵⁾ After that, the five types of feature vector

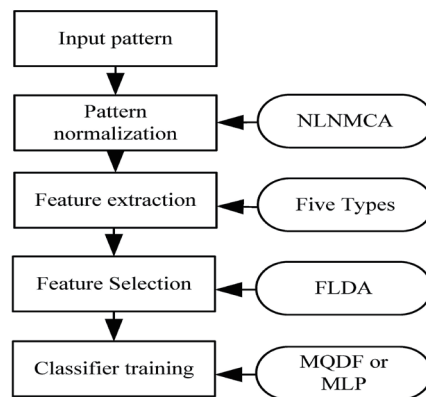


Fig. 1. Dataflow of handwritten character classifier training.

are extracted from the normalized pattern. In the third step, the extracted feature vector is compressed by Fisher linear discriminant analysis (FLDA).⁽¹³⁾ Finally, the compressed feature vector is employed to train MQDF. After being trained, the classifier is used to assign the input handwritten character pattern to the corresponding character class.

In HCCR, the CASIA-OLHWDB1 database is employed to evaluate the performance of different feature expressions.⁽¹⁴⁾ This online database of handwritten Chinese characters and texts, collected by Anoto pen on paper, contains unconstrained handwritten characters of 4037 categories (3866 Chinese characters and 171 symbols) written by 420 people, and there are 1694741 handwritten samples in total.

In HJCR, the TUAT HANDS database, which consists of Kuchibue and Nakayosi databases of online handwritten Japanese characters, is employed as training and testing databases to evaluate the performance of different feature extractions.⁽¹⁵⁾ The Kuchibue database contains online handwritten character patterns of 120 writers, i.e., 11962 patterns per writer covering 3356 character classes. The 11962 patterns per writer include two parts: the sentential part consisting of 10154 characters, often used for text recognition testing, and 1808 single characters that appear less often. The Nakayosi database contains the samples of 163 writers, 10403 patterns covering 4438 character classes per writer. Therefore, Kuchibue in total contains a total of $11962 \times 120 = 1435440$ character patterns and Nakayosi contains $10403 \times 163 = 1695689$ patterns. For Kuchibue and Nakayosi, each character pattern was written in individual writing boxes.

3. Feature Extractions

The LDD and GD feature extractions are quite similar and contain three steps: character pattern or character image normalization, directional decomposition, and feature sampling. The slight variation is in the directional elements of directional decomposition in the second step, i.e., LDD employs trajectory-based elements, whereas GD employs gradient-based elements. The convNet-based feature extraction directly uses the deep convNet. The feature extraction is described in detail below.

3.1 LDD feature extraction

In the LDD feature extraction, we assume that the online character C is composed of n strokes, noted as $C = \{S_1, S_2, \dots, S_n\}$. P_{ij} ($j = 1, \dots, m$) is the j -th point on the stroke S_i ($i = 1, \dots, n$) and the trajectory vector A_{ij} is the vector from P_{ij} to P_{ij+1} , i.e., $A_{ij} = P_{ij+1} - P_{ij}$. Then all the trajectory vectors A_{ij} ($j = 1, \dots, m$) of the online pattern are employed to extract the trajectory-based feature vector. The decomposition of the trajectory vector to major directions is necessary to obtain a robust feature vector. Let e_1, e_2 , up to e_8 be the elementary vector of each of the 8 directions, as shown in Fig. 2. Then, the angle range is divided into 8 subranges, Ω^1, Ω^2 up to Ω^8 . Each elementary vector e_i ($i = 1, \dots, 8$) corresponds to one direction plane \mathcal{O}_i ($i = 1, \dots, 8$), which stores the decomposed vector along the elementary vector. After the decomposition of the trajectory vector along the elementary vectors, the vector A_{ij} is noted as $A_{ij} = (|a_{ij}^1|, |a_{ij}^2|, |a_{ij}^3|, |a_{ij}^4|, |a_{ij}^5|, |a_{ij}^6|, |a_{ij}^7|, |a_{ij}^8|)$, and the subvectors can be determined using Eq. (1). The decomposed vectors $a_{ij}^1, \dots, a_{ij}^8$ are stored into the corresponding directional planes \mathcal{O}_i ($i = 1, \dots, 8$). More effectively, the trajectory vector is directly decomposed into two components in standard directions between which a trajectory vector lies.⁽¹⁶⁾ The length of each component is assigned to the corresponding direction plane at the corresponding pixel.

$$A_{ij} = \begin{cases} a_{ij}^1 e^1 + a_{ij}^2 e^2; a_{ij}^3 = a_{ij}^4 = \dots = a_{ij}^8 = 0 & \text{if } A_{ij} \in \Omega^1 \\ a_{ij}^2 e^2 + a_{ij}^3 e^3; a_{ij}^1, a_{ij}^4, \dots, a_{ij}^8 = 0 & \text{if } A_{ij} \in \Omega^2 \\ a_{ij}^3 e^3 + a_{ij}^4 e^4; a_{ij}^1, a_{ij}^2, a_{ij}^5, \dots, a_{ij}^8 = 0 & \text{if } A_{ij} \in \Omega^3 \\ a_{ij}^4 e^4 + a_{ij}^5 e^5; a_{ij}^1, \dots, a_{ij}^3, a_{ij}^6, \dots, a_{ij}^8 = 0 & \text{if } A_{ij} \in \Omega^4 \\ a_{ij}^5 e^5 + a_{ij}^6 e^6; a_{ij}^1, \dots, a_{ij}^4, a_{ij}^7, a_{ij}^8 = 0 & \text{if } A_{ij} \in \Omega^5 \\ a_{ij}^6 e^6 + a_{ij}^7 e^7; a_{ij}^1, \dots, a_{ij}^5, a_{ij}^8 = 0 & \text{if } A_{ij} \in \Omega^6 \\ a_{ij}^7 e^7 + a_{ij}^8 e^8; a_{ij}^1, \dots, a_{ij}^6 = 0 & \text{if } A_{ij} \in \Omega^7 \\ a_{ij}^8 e^8 + a_{ij}^1 e^1; a_{ij}^2, \dots, a_{ij}^7 = 0 & \text{if } A_{ij} \in \Omega^8 \end{cases} \quad (1)$$

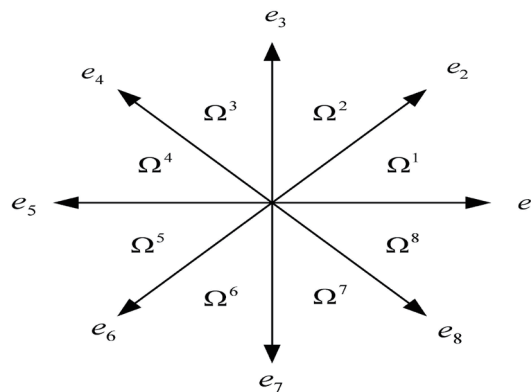


Fig. 2. Elementary directions and subranges.

To extract feature vectors of moderate dimensionality, each direction plane must to be blurred and resampled; then, the total or average value of each zone is taken as the feature value. For the implementation of blurring, the convolution of the Gaussian filter is given in Eq. (2), and the directional plane is operated using Eq. (4) to obtain the feature value of this filter window. The variance parameter is related to the sampling frequency computed using Eq. (3), where t_x is the sampling interval.

$$h(x, y) = \frac{1}{2\pi\sigma_x^2} \exp\left(-\frac{x^2 + y^2}{2\sigma_x^2}\right) \quad (2)$$

$$\sigma_x = \frac{\sqrt{2}t_x}{\pi} \quad (3)$$

$$F(x_0, y_0) = \sum_x \sum_y f(x, y) h(x - x_0, y - y_0) \quad (4)$$

From the above descriptions, the feature extraction process of LDD is shown in Fig. 3. We apply NLNMCBA to the character patterns. The local stroke direction is decomposed into eight direction planes, and then 64 feature elements from each directional plane are extracted using

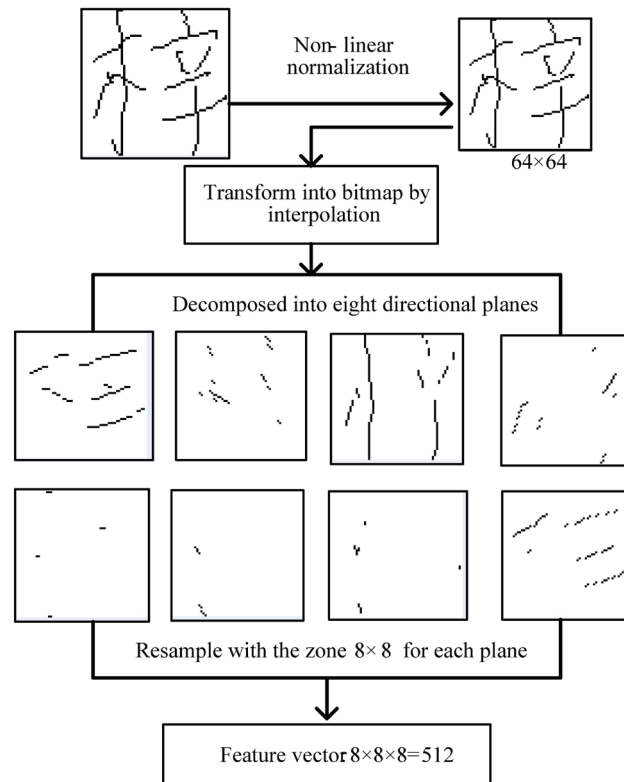


Fig. 3. LDD feature extraction.

the Gaussian filter with a filter window of 8×8 pixels; thus, the dimensionality of feature vectors is 512. To improve the Gaussianity of feature distribution, each value of the 512 features is transformed by the Box–Cox transformation.⁽¹⁷⁾

3.2 Gradient feature extraction

Since the GD feature cannot be directly extracted from an online pattern, a pattern image converted from the corresponding online pattern is needed. The gradient vector $g(x, y) = [g_x, g_y]^T$ at pixel (x, y) in a normalized pattern image is computed using Eq. (5). The gradient strength and direction can be computed from the gradient vector $[g_x, g_y]^T$.⁽¹⁸⁾ Then the remaining steps, i.e., directional decomposition and feature sampling, are the same as in the feature extraction of LDD.

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

3.3 Deep convNet-based feature extraction

Generally, deep convNet-based feature extraction directly learns discriminative representation from a raw pattern image. However, the well-studied domain-specific knowledge, i.e., the directional feature map, is shown to still be helpful for further improving the performance of handwritten character recognition. In this research, the domain-specific knowledge focuses on the LDD and GD features. Therefore, the convNet-based feature extraction method extracts three types of feature vector: raw pattern image as input with a deep convNet-based feature vector and LDD and GD feature maps as input integrated with the convNet-based feature extraction, denoted as convNet-based-Raw, convNet-based-LDD, and convNet-based-GD features, respectively.

The architecture of deep convNet is employed to extract the three convNet-based features, as shown in Fig. 4. It comprises six convolution layers and three fully connected layers. The character pattern is first transformed into a binary image and normalized to 64×64 pixels, then the normalized pattern image for convNet-based-Raw, LDD feature maps for convNet-based-LDD and GD feature maps for convNet-based-GD acting as input layers are passed through a stack of convolution layers separately to capture the notions of left/right, up/down, and center.⁽¹⁹⁾ The number of convolution strides is fixed to one and the number of feature maps is gradually increased from 60 (layer-1) to 360 (layer-6). In our architecture, spatial pooling is implemented after every two convolution layers to increase the depth of the network by the max-pooling of the 2×2 window with stride 2.

After the stacking of 6 convolution layers and 3 max-pool layers, the feature maps are flattened and concatenated into a vector with a dimensionality of 9000. Two fully connected



Fig. 4. (Color online) Architecture of deep convNet for convNet-based feature extraction.

(FC) layers (with 1500 and 200 hidden units) are then added. Finally, the softMax layer is used to perform the N -way classification. It is necessary to specify that the number N differs between HCCR and HJCR, that is, 4037 for HCCR and 4438 for HJCR. The prediction is made by taking the class with the maximal probability for the given test pattern. The network's input is 64×64 with the dimensionality of 9216 for convNet-based-Raw, whereas the $d \times n \times n$ dimensionality is for convNet-based-LDD and convNet-based-GD, where d is the number of quantized directions, which is eight, and $n = 64$ is the size of the map with the dimensionality of 32768. The numbers of neurons in the network's remaining layers for all three methods are the same, as given by 216000-403680-100920-131220-150000-40560-36300-29160-9000- N .

4. FLDA

The FLDA is well known and common in pattern recognition for feature reduction. We assume that there are c known character classes w_1, w_2, \dots, w_c , and the i -th class has N_i training samples among the total of N training samples. $x = \{x_j^i\}$ ($j = 1, 2, \dots, c, i = 1, 2, \dots, N_j$) is a set of samples with n -dimensions, where N_j is the number of samples in the j -th class. The mean vector of each class and of all classes can be expressed as $\bar{x}_j = (1/N_j) \sum_{i=1}^{N_j} x_j^i$ and $\bar{x} = (1/N) \sum_{j=1}^c \sum_{i=1}^{N_j} x_j^i$, respectively. Let the interclass matrix and intraclass scatter matrix respectively be defined as

$$S_b = \sum_{j=1}^c N_j (\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^T, \quad (6)$$

$$S_w = \sum_{j=1}^c \sum_{i=1}^{N_j} (x_j^i - \bar{x}_j)(x_j^i - \bar{x}_j)^T. \quad (7)$$

The process to obtain the transformation matrix is the maximization of the following Eq. (8), called the Fisher discriminant criterion,⁽²⁰⁾ where $\{w_i | i = 1, 2, \dots, m\}$ are m n -dimensional

eigenvectors of $S_w^{-1}S_b$ corresponding to the m largest eigenvalues. W_{opt} is an $m \times n$ matrix composed of m n -dimensional eigenvectors. Using the transformation matrix W_{opt} , we can reduce the feature dimensions from n -dimensions to m -dimensions.

$$W_{opt} = \arg \max_w \left| \frac{w^T S_b w}{w^T S_w w} \right| = [w_1 w_2 \dots w_m] \quad (8)$$

5. Handwritten Character Classifier

To comprehensively compare the feature extraction methods, we employ the statistical classifier MQDF and the discriminative classifier, the multilayer perceptron (MLP), for HCCR and HJCR.

5.1 MQDF

We employ the MQDF since it has been most often employed for HCCR and HJCR. It is superior to the quadratic discriminant function (QDF) in terms of recognition rate, speed, and memory size. Kimura *et al.* proposed two types of MQDF: MQDF1 and MQDF2.⁽²¹⁾ MQDF1 employs a type of Bayesian estimate of the covariance matrix instead of the maximum likelihood estimate, whereas MQDF2 is a version of the QDF smoothed by replacing the minor eigenvalues with a larger constant. In the system, MQDF2 is used for character classification. Given the d -dimensional feature vector denoted as x , MQDF2 of class ω_i is computed using Eq. (9), where μ_i is the mean vector of class ω_i , λ_{ij} and ϕ_{ij} ($j = 1, \dots, d$) are the eigenvalues and eigenvectors of the covariance matrix of class ω_i , respectively, κ denotes the number of principal axes, and the minor eigenvalues are replaced with a constant δ_i . The parameter δ_i is set as a class-independent constant computed using Eq. (10), and $tr(\Sigma_i)$ denotes the trace of covariance.

$$g_2(x, \omega_i) = \sum_{j=1}^{\kappa} \frac{1}{\lambda_{ij}} \left[(x - \mu_i)^T \phi_{ij} \right]^2 + \frac{1}{\delta_i} \left\{ \|x - \mu_i\|^2 - \sum_{j=1}^{\kappa} \left[(x - \mu_i)^T \phi_{ij} \right]^2 \right\} + \sum_{j=1}^{\kappa} \log \lambda_{ij} + (d - \kappa) \log \delta_i \quad (9)$$

$$\delta_i = \frac{tr(\Sigma_i) - \sum_{j=1}^{\kappa} \lambda_{ij}}{n - \kappa} = \frac{1}{n - \kappa} \sum_{j=\kappa+1}^n \lambda_{ij} \quad (10)$$

5.2 MLP

For M -class classification, the MLP has M output units. On a d -dimensional feature vector $x = [x_1, \dots, x_d]^T$, the total input of all the hidden layers x_j^{h+1} received by neuron j in layer $h + 1$ is computed using Eq. (11), where y_i^h is the state of the i -th neuron in the preceding h -th layer, w_{ji}^h is the weight of the connection from the i -th neuron in layer h to the j -th neuron in layer $h + 1$,

and σ is the sigmoid activation function. The output of the MLP for class k is computed using Eq. (12), where $o - 1$ denotes the last hidden layer and w_k is the bias of class k .

$$x_j^{h+1} = \sigma \left(\sum_i y_i^h w_{ji}^h \right) \quad (11)$$

$$x_k^o = \sigma \left(\sum_i y_i^{o-1} w_{ji}^{o-1} + w_k \right) \quad (12)$$

On a training set of N samples (x^n, c^n) , $n = 1, \dots, N$ (c^n is the class label of x^n), the connecting weights of MLP are adjusted to minimize the regularized squared error given by Eq. (13), where β is a coefficient of weight decay to exclude the biases; t_k^n is the target value of class k and is 1 for the genuine class and 0 otherwise. The weights and biases are initialized to small random values, and by stochastic gradient descent (SGD), they are iteratively updated on the training samples until the squared error becomes minimum.⁽²²⁾

$$E = \frac{1}{2N} \left\{ \sum_{n=1}^N \sum_{k=1}^M [x_k^o - t_k^n]^2 + \beta \sum_{w \in W} w^2 \right\} \quad (13)$$

6. Experiments and Analysis

To evaluate the performance of feature extraction for HJCR, we use the Nakayosi database as the training data and the Kuchibue database as the testing data. However, for the evaluation of each feature extraction method for HCCR, we employ the 5-fold cross-validation on the CASIA-OLHWDB1 database and only report the mean of the 5-fold cross-validation. Considering that the statistical classifier can generalize better than the discriminative classifier in a large category classification task, the MQDF is used in this section; however, MLP is only used in Sect. 7 for comparison. The evaluation is implemented on a computer with an Intel Core 2 Duo E8500 CPU of 3.16 GHz and 8 GB of RAM.

6.1 LDD feature extraction

As for the LDD feature extraction, we set the size of the normalized pattern and direction plane to 64×64 pixels and the sampling interval to 8. As a result, we obtained 64 feature values from each direction plane and 512 feature values in total. We set the power of variable transformation to 0.5 to improve the Gaussianity of feature distribution.

To reduce the complexity of the MQDF classifier and improve the classification accuracy, the extracted feature vector is compressed to a lower dimensionality by FLDA. We set the reduced dimensionality d in the range from 100 to 160 in steps of 20. The number of principal axes of the classifier MQDF k takes values ranging from 10 to 60 increased in steps of 10. Tables 1 and 2 show the accuracies of character recognition based on the feature extraction of LDD for

Table 1
Accuracies of offline recognizers with LDD feature for HCCR and HJCR (%).

Accuracies of different recognizers for HCCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	90.72	91.42	91.60	91.67	91.55	91.35
120	90.98	91.61	91.79	91.85	91.80	91.64
140	91.11	91.71	91.86	91.93	91.88	91.75
160	91.21	91.78	91.89	91.92	91.87	91.72
Accuracies of different recognizers for HJCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	90.64	91.23	91.23	91.43	91.37	91.14
120	90.93	91.46	91.63	91.66	91.55	91.38
140	91.11	91.61	91.74	91.77	91.73	91.50
160	91.23	91.68	91.79	91.82	91.75	91.56

Table 2
Accuracies of offline recognizers with GD feature for HCCR and HJCR (%).

(a) Accuracies of different recognizers for HCCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	91.09	91.70	91.85	91.83	91.63	91.23
120	91.56	92.14	92.29	92.33	92.26	92.03
140	91.79	92.33	92.47	92.66	92.62	92.35
160	91.97	92.44	92.57	92.52	92.47	92.46
(b) Accuracies of different recognizers for HJCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	92.37	92.93	93.09	93.15	93.07	92.89
120	92.62	93.1	93.28	93.33	93.28	93.11
140	92.77	93.2	93.37	93.37	93.43	93.22
160	92.83	93.27	93.38	93.35	93.37	93.25

HCCR and HJCR, respectively. The accuracies of Chinese character recognition range from 90.72 to 91.93%, as shown in Table 1(a), and the accuracies of Japanese character recognition range from 90.64 to 91.82%, as shown in Table 1(b). Therefore, the accuracy variations of HCCR and HJCR based on the feature extraction of LDD fall into similar ranges.

6.2 GD feature extraction

Other than the gradient vector at each pixel of the normalized pattern image, instead of the trajectory vector of the online character pattern, being decomposed into each directional plane, the feature extraction of GD is almost the same as that of LDD. To enable a fair comparison, the size of the normalized pattern image, the number of directional planes, and the sampling interval are set to the same values with the feature extraction of LDD described in Sect. 6.1. Similarly, to train the character classifier MQDF, the 512-dimensional gradient feature vector is reduced to the same lower dimensionality *d* ranging from 100 to 160 in steps of 20. For each dimension, we take a different number of principal components from 10 to 60 in steps of 10, so that we produce 24 different offline recognizers. Table 2 shows the accuracies of

24 different offline classifiers for HCCR and HJCR. The accuracies of 24 offline classifiers for HCCR range from 91.09 to 92.66%, as shown in Table 2(a), and those for HJCR vary from 92.37 to 93.43%, as shown in Table 2(b). The best accuracy based on the feature extraction of LDD for HCCR, 91.93%, is slightly higher than that for HJCR, 91.82%, as depicted in Sect. 6.1. However, the best accuracy based on GD feature extraction for HJCR, 93.43%, surpassed that for HCCR, 92.66%, by 0.77%.

6.3 Deep convNet-based feature extraction

As for the three feature extraction methods, deep ConvNet-based-Raw, convNet-based-LDD, and convNet-based-GD for HCCR and HJCR, the same learning strategy and parameter settings are used. We employed minimum cross-entropy on all the training data sets to optimize the deep convNet using the SGD with the momentum of 0.9. The mini-batch size is 512 and the base learning rate is initialized for all trainable parameters at 0.005 for the entire deep convNet. The networks were trained for 5000 epochs, and the results obtained for every 100 training epochs are presented in Fig. 5. The best accuracies of the three convNet-based methods for HCCR and HJCR after 5000 epochs are given in Table 3. From Table 3 and Fig. 5, the deep

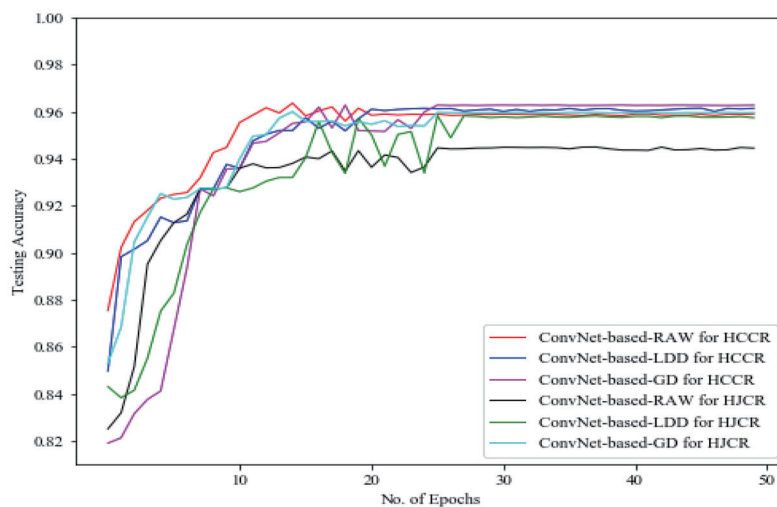


Fig. 5. (Color online) Training process of deep convNet for HCCR and HJCR.

Table 3

Best accuracies of three convNet-based methods for HCCR and HJCR (%).

Feature extraction method/Language	Chinese	Japanese
ConvNet-based-RAW	95.87	95.45
ConvNet-based-LDD	96.12	95.83
ConvNet-based-GD	96.22	95.96

convNet of the three methods converge around the best recognition accuracy; those of convNet-based-Raw are 95.87 and 95.45%, those of convNet-based-LDD are 96.12 and 95.83%, and those of convNet-based-GD are 96.22 and 95.83% for HCCR and HJCR, respectively.

Deep convNet is an integration of feature extraction with recognition. For further evaluation, the feature extraction part of the optimized deep convNet, i.e., the former six convolution layers, is employed to extract a 9000-dimensional feature vector to the three convNet-based feature extractions for HCCR and HJCR. For each convNet-based feature extraction, similar to the steps described in Sect. 6.1, after extraction, the 9000-dimensional feature vector is reduced to lower dimensionalities, and each dimension takes a different number of principal components to generate 24 different offline MQDF recognizers in total. Tables 4–6 show the accuracies of HCCR and HJCR when using convNet-based-Raw, convNet-based-LDD, and convNet-based-GD feature extractions, respectively.

Tables 4(a), 5(a), and 6(a) show that the convNet-based-GD method for HCCR obtains the best accuracy of 94.97%, which is higher than those of convNet-based-Raw (94.53%) and convNet-based-LDD (94.73%) by 0.43 and 0.24 percentage points, respectively. Similarly, the convNet-based-GD method for HJCR also gains the best accuracy of 95.11%, outperforming by 0.79 and 0.22 percentage points, those of the convNet-based-Raw and convNet-based-LDD, respectively, as illustrated in Tables 4(b), 5(b), and 6(b).

Table 4
Accuracies when using ConvNet-based-Raw for HCCR and HJCR (%).

(a) Accuracies of different recognizers for HCCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	93.41	93.92	94.12	94.15	94.06	93.88
120	93.62	94.15	94.33	94.36	94.13	94.02
140	93.77	94.24	94.38	94.53	94.41	94.25
160	93.85	94.3	94.43	94.45	94.45	94.31
(b) Accuracies of different recognizers for HJCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	93.24	93.79	93.97	94	93.92	93.74
120	93.42	93.96	94.14	94.18	94.17	94.02
140	93.55	94.07	94.22	94.28	94.32	94.11
160	93.61	94.08	94.24	94.26	94.25	94.13

Table 5
Accuracies when using ConvNet-based-LDD for HCCR and HJCR (%).

(a) Accuracies of different recognizers for HCCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	93.52	94.22	94.4	94.47	94.35	94.15
120	93.78	94.41	94.59	94.65	94.6	94.44
140	93.91	94.51	94.66	94.73	94.68	94.55
160	94.01	94.58	94.69	94.72	94.67	94.52
(b) Accuracies of different recognizers for HJCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	93.71	94.30	94.30	94.50	94.44	94.21
120	94.00	94.53	94.70	94.73	94.62	94.45
140	94.18	94.68	94.81	94.84	94.80	94.57
160	94.30	94.75	94.86	94.89	94.82	94.63

Table 6
Accuracies when using ConvNet-based-GD for HCCR and HJCR (%).

(a) Accuracies of different recognizers for HCCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	93.4	94.01	94.16	94.14	93.94	93.54
120	93.87	94.45	94.6	94.64	94.57	94.34
140	94.1	94.64	94.78	94.97	94.93	94.66
160	94.28	94.75	94.88	94.83	94.78	94.77
(b) Accuracies of different recognizers for HJCR (%)						
<i>d/n</i>	10	20	30	40	50	60
100	94.05	94.61	94.77	94.83	94.75	94.57
120	94.3	94.78	94.96	95.01	94.96	94.79
140	94.45	94.88	95.05	95.05	95.11	94.9
160	94.51	94.95	95.06	95.03	95.05	94.93

7. Results and Discussion

In this section, we first analyze the impact of the characteristics of the database and different recognizers on the recognition accuracy. Then, we compare the results with those of related works.

7.1 Impact of characteristics of database on recognition accuracy

From Tables 1(a), 2(a), 4(a), 5(a), and 6(a), LDD, GD, and three other convNet-based feature extraction methods for HCCR with the CASIA-OLHWDB1 database achieved the best accuracies of 91.93, 92.66, 94.53, 94.73, and 94.97%, respectively. Similarly, the best accuracies of using the above five feature extraction methods for HJCR are 91.82, 93.43, 94.32, 94.89, and 95.11%, as given in Tables 1(b), 2(b), 4(b), 5(b), and 6(b), respectively. Comparisons of the best recognition accuracies among the five feature extraction methods for HCCR and HJCR are shown in Fig. 6. It is revealed that the convNet-based-GD is the most robust for handwritten character recognition among these five feature types. While the difference in the best recognition accuracy in the feature extraction of GD between HCCR and HJCR is 0.77%, those of using LDD and the other three convNet-based feature extractions are 0.11, 0.21, 0.16, and 0.14%, which indicate that the differences in accuracy are increased by 0.66, 0.56, 0.61, and 0.63%, respectively, compared with those of the latter four methods.

To further determine why the difference in accuracy between HCCR and HJCR based only on GD is apparently enlarged, we investigate the character patterns in the handwritten Chinese database (CASIA-HWDB1.0) and the Japanese handwritten database (TUAT HANDS). It is found that the average length between two neighboring points of all Chinese testing patterns is 18.5 pixels and that for the Japanese testing pattern is 10.3 pixels. Then, on the basis of the average interval length between two neighboring points of each handwritten pattern, which is denoted as *len*, the testing character patterns of both Chinese and Japanese are divided into four groups: $len < 5$, $5 < len < 10$, $10 < len < 15$, and $len > 15$. We selected the best accuracy of HCCR and HJCR on the basis of the GD feature (92.66% for HCCR and 93.43% for HJCR)

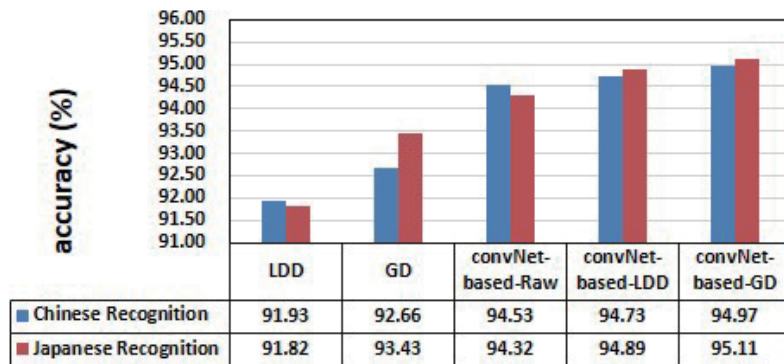


Fig. 6. (Color online) Comparisons of five feature extractions in terms of best accuracy for HCCR and HJCR.

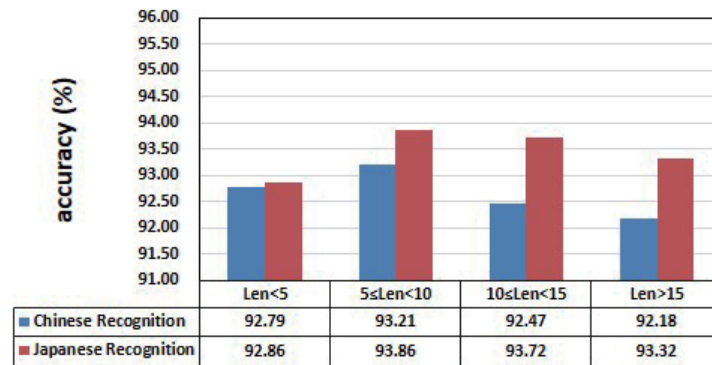


Fig. 7. (Color online) Comparisons of GD in terms of best accuracy for four pattern groups of HCCR and HJCR.

to determine the recognition accuracy of each group, and the results are shown in Fig. 7. It is revealed that the larger the average interval length between two neighbors, the more seriously the accuracy is degraded, which indicates that the GD feature is sensitive to a sampled interval of handwritten patterns.

7.2 MLP

The MLP is an optimal candidate of MQDF in handwritten character recognition. Considering that the recognition part of the deep convNet given in Fig. 2 is composed of two FC layers (with 1500 and 200 hidden units) and a softMax layer, we construct the MLP of four layers, which is the same as the recognition part of the deep convNet. The number of neurons of each layer for the MLP is given by $d-1500-200-N$, where d is the dimensionality of a feature vector and N is the number of classifications, i.e., 4037 for HCCR and 4438 for HJCR.

To enable a fair comparison with the MQDF, the same dimensionality of the feature vector as that of LDD, GD, convNet-based-Raw, convNet-based-LDD, and convNet-based-GD used for MQDF is employed. The 512-dimensional feature vectors of the LDD and GD features and

the 9000-dimensional feature vector of the three convNet-based methods are reduced to lower dimensionality from 80 to 160 with steps of 20 by FLDA. The accuracy trends of the MLP for HCCR and HJCR are shown in Figs. 8 and 9, respectively. For each feature extraction method, the highest accuracy when using the MLP is slightly lower than that of using the MQDF for both HCCR and HJCR. Similarly to using the MQDF as a recognizer, the feature extraction method convNet-based-GD again shows the highest robustness and discrimination among the five feature extraction methods in the MLP used as a classifier for HCCR and HJCR.

7.3 Comparison with related works

To enable a comparison with related works, the above experiments are repeated employing the test data set used in the ICDAR 2010 and 2013 competitions for HCCR.⁽²³⁾ In this paper, we report the results of HCCR using the test data set, targeting 3,755 Chinese characters of the GB2312-80 first-level set. Note that our evaluation for HCCR in Sect. 6 included not only these 3755 Chinese character classes but also numbers and symbols, for a total of 4038 categories in total. The comparison with related works on HCCR is given in Table 7. The first to third methods used multiple strategies to improve their performance, such as a multiple supervised training strategy and deep convNet. However, our method of using a deep convNet integrated with domain-specific knowledge still outperforms them. Although domain-specific knowledge is already used in the fourth method, our convNet-based-GD method can still outperform them. The comparison with related works on HJCR is given in Table 8. Similarly, our convNet-based-GD method outperforms them also in HJCR.

Therefore, our proposed method of considering the LDD and GD features as domain-specific knowledge and then combining them with convNet outputs traditional feature expressions for handwritten character recognition and further improves the handwritten recognition accuracy in both HCCR and HJCR.

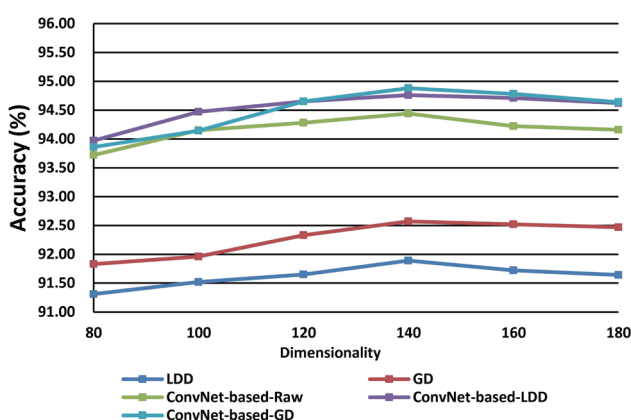


Fig. 8. (Color online) Accuracy trend of using MLP for HCCR.

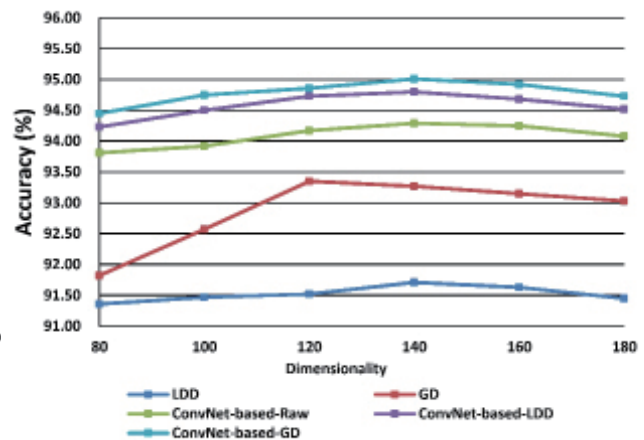


Fig. 9. (Color online) Accuracy trend of using MLP for HJCR.

Table 7
Comparisons with related works for HCCR.

Method	Ref.	Accuracy (%)
DFE+DLQDF	24	92.72
ICDAR-2010 Winner: HKU	23	89.99
ICDAR-2013 Winner: Fujitsu	25	94.77
DirectMap+ConvNet	26	96.95
ConvNet-based-Raw	Ours	95.67
ConvNet-based-LDD	Ours	96.85
ConvNet-based-GD	Ours	97.09

Table 8
Comparisons with related works for HJCR.

Method	Ref.	Accuracy (%)
SEG-FE	6	92.08
GRD-FE	6	92.10
ConvNet-based-Raw	Ours	94.32
ConvNet-based-LDD	Ours	94.89
ConvNet-based-GD	Ours	95.11

8. Conclusions

For HCCR, the three convNet-based methods obtained the highest accuracies of 94.53, 94.73, and 94.97%, but the accuracies of the other two types, LDD and GD, were 91.93 and 92.66%, respectively. Similarly, for HJCR, the accuracies of the three convNet-based methods were 94.32, 94.89, and 95.11%, but those of LDD and GD were 91.82 and 93.43%, respectively. Therefore, the convNet-based feature extraction is the most robust and discriminating compared with the two traditional methods, LDD and GD, for both HCCR and HJCR. The two convNet-based methods with domain-specific knowledge, convNet-based-LDD and convNet-based-GD, obtained the best accuracies of 94.73 and 94.97% for HCCR and HJCR, respectively. However, the accuracies of convNet-based-Raw without domain-specific knowledge were 94.53 and 94.32% for HCCR and HJCR, respectively. The results demonstrate that domain-specific knowledge makes the convNet-based method more efficient and effective for HCCR and HJCR.

Acknowledgments

We would like to thank the Henan Natural Science Foundation for funding this research under grant number 162300410195.

References

- 1 L.-J. Wen, Z.-Z. Yao, and Y. Zhang: *Acta Autom. Sin.* **42** (2016) 8. <https://doi.org/10.16383/j.aas.2016.c150725>
- 2 J.-F. Gao, B.-L. Zhu, and N. Masaki: *Pattern Recogn. Let.* **35** (2014) 6. <https://doi.org/10.1016/j.patrec.2013.08.014>
- 3 H. Tanaka, K. Nakajima, K. Ishigaki, K. Akiyama, and N. Masaki: *Proc. 5th Int. Conf. Document Analysis and Recognition (IEEE, 1999)*. <https://doi.org/10.1109/ICDAR.1999.791761>
- 4 C.-L. Liu: *Summit on the Arabic and Chinese Handwriting Recognition (IEEE, 2006)*. https://doi.org/10.1007/978-3-540-78199-8_7
- 5 Y. Xiong, T. Lu, and Y. Jiang: *Proc. SPIE Pattern Recognition and Computer Vision (IEEE, 2018)*. <https://doi.org/10.1117/12.2288374>
- 6 T. V. Phan, J.-F. Gao, and B.-L. Zhu: *Proc. Int. Conf. Document Analysis and Recognition (IEEE, 2011)*. <https://doi.org/10.1109/ICDAR.2011.171>
- 7 W. Yi, M. R. Morelande, and L. J. Kong: *Proc. Radar Conf. (IEEE, 2012)*. <https://doi.org/10.1109/RADAR.2012.6212190>
- 8 B.-L. Zhu and M. Nakagawa: *Proc. Int. Conf. Document Analysis and Recognition (IEEE, 2011)*. <https://doi.org/10.1109/ICDAR.2011.127>

- 9 E. Mofteh and A. H. Ayoub: *Sensors* **18** (2019) 9. <https://doi.org/10.3390/s18092786>
- 10 S. Roy, N. Das, M. Kundu, and M. Nasipuri: *Pattern Recognit. Lett.* **90** (2017) 25. <https://doi.org/10.1016/j.patrec.2017.03.004>
- 11 Y. Bengio, A. Courville, and P. Vincent: *IEEE Trans. Pattern Anal. Mach. Intell.* **35** (2013) 8. <https://doi.org/10.1109/TPAMI.2013.50>
- 12 Z. Zhong, L. Jin, and Z. Xie: *Proc. Int. Conf. Document Analysis and Recognition (IEEE, 2015)*. <https://doi.org/10.1109/ICDAR.2015.7333881>
- 13 M. Loog, R. P. W. Duin, and R. Haeb-Umbach: *IEEE Trans. Pattern Anal. Mach. Intell.* **23** (2001) 7. <https://doi.org/10.1109/34.935849>
- 14 D.-H. Wang, C.-L. Liu, J.-L. Yu, and X.-D. Zhou: *Proc. 10th Int. Conf. Document Analysis and Recognition (IEEE, 2009)*. <https://doi.org/10.1109/ICDAR.2009.163>
- 15 J.-F. Gao, B.-L. Zhu, and M. Nakagawa: *IEICE Trans. Inf. Syst.* **96** (2013) 10. <https://doi.org/10.1587/transinf.E96.D.927>
- 16 C.-L. Liu and X.-D. Zhou: *Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition (IEEE, 2006)*. <https://doi.org/10.1109/ICDAR.1993.395748>
- 17 R.V.D. Heiden, F. C. A. Gren: *Pattern Recognit.* **30** (1997) 9. [https://doi.org/10.1016/s0031-3203\(96\)00077-5](https://doi.org/10.1016/s0031-3203(96)00077-5)
- 18 L. Huang, A. Shimizu, Y. Hagihara, and H. Kobatake: *Pattern Recognit.* **36** (2003) 12. [https://doi.org/10.1016/s0031-3203\(03\)00130-4](https://doi.org/10.1016/s0031-3203(03)00130-4)
- 19 K. Simonyan and A. Zisserman: *Proc. Int. Conf. Learning Representations (IEEE, 2015)*.
- 20 P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman: *IEEE Trans. Pattern Anal. Mach. Intell.* **19** (2002) 7. <https://doi.org/10.1007/BFb0015522>
- 21 F. Kimura, K. Takashina, S. Tsuruoka, and Y. Miyake: *IEEE Trans. Pattern Anal. Mach. Intell.* **9** (1987) 4. <https://doi.org/10.1109/TPAMI.1987.4767881>
- 22 H. Amakdouf, M. El Mallahi, and A. Zouhri: *Procedia Computer Science* **127** (2018) 7. <https://doi.org/10.1016/j.procs.2018.01.118>
- 23 C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang: *Proc. 2010 Conf. Pattern Recognition (IEEE, 2010)*. <https://doi.org/10.1109/CCPR.2010.5659229>
- 24 C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang: *Pattern Recognit.* **46** (2013) 1. <https://doi.org/10.1016/j.patcog.2012.06.021>
- 25 F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu: *Proc. Int. Conf. Document Analysis and Recognition (IEEE, 2013)*. <https://doi.org/10.1109/ICDAR.2011.291>
- 26 X.-Y. Zhang, Y. Bengio, and C.-L. Liu: *Pattern Recognit.* **61** (2017) 8. <https://doi.org/10.1016/j.patcog.2016.08.005>

About the Authors



Jin-feng Gao received his M.E. degree from Guizhou University, Guiyang City, China, and his Ph.D. degree from Tokyo University of Agriculture, Tokyo, Japan, in 2009 and 2013, respectively. He is currently an associate professor at the College of Information Engineering, Huanghuai University, Henan, China. His research interests include handwriting recognition, natural language processes, and pattern recognition. (gaojinfeng0213@126.com)



Ru-xian Yao received his B.S. and M.E. degrees from Jinan University and Tongji University in 1997 and 2009, respectively. He is currently a professor at the College of Information Engineering, Huanghuai University, Henan, China. His current research interests include machine learning, data mining, and pattern recognition. (yaoruxian@126.com)



Yu Zhang received his B.S. degree from Henan University of Technology, China, in 2011 and his Ph.D. degree from Jilin University, China, in 2018. He is currently an associate professor at the College of Information Engineering, Huanghuai University, Henan, China. His research interests include semantic web and ontology reasoning. (zhangyujlu@163.com)



Han Lai received his M.Sc. and Ph.D. degrees from Northumbria University, UK, in 2005 and 2012, respectively. From 2007 to 2012, he was a researcher and lecturer at the same university. Since 2017, he has been a lecturer at Huanghuai University, China. His research interests include information and knowledge sharing and seeking, machine learning, and pattern recognition. (han.lai@hotmail.com)



Jun-ming Zhang received his M.S. in control engineering from Changchun University of Technology, Changchun, China, in 2009. He is currently an associate professor in the College of Information Engineering, Huanghuai University, Henan, China. His research interests include machine learning, biomedical signal processing, and automatic sleep staging. (271937125@qq.com)



Ting-Cheng Chang received his M.S. and Ph.D. degrees in Process Control and Mechanical Engineering in 1992 and 1996 from University of Houston and University of Texas at Arlington, Texas, USA, respectively. He is now a professor of computer and information engineering, Ningde Normal University, Fujian, China. His research interests are in the fields of Internet of Things, data mining, big data, and optimal theory. (18250922163@163.com)