Received February 8, 2018, accepted March 15, 2018, date of publication March 19, 2018, date of current version April 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2817342

Deep Contextual Stroke Pooling for Scene Character Recognition

ZHONG ZHANG[1,2], (Member, IEEE), HONG WANG[1,2], SHUANG LIU[1,2], AND BAIHUA XIAO[3]
[1]Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin 300387, China
[2]College of Electronic and Communication Engineering, Tianjin Normal University, Tianjin 300387, China
[3]State Key Laboratory of Management and Intelligent Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Zhong Zhang (zhong.zhang8848@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61501327 and Grant 61711530240, in part by the Natural Science Foundation of Tianjin under Grant 17JCZDJC30600 and Grant 15JCQNJC01700, in part by the Open Projects Program of the National Laboratory of Pattern Recognition under Grant 201700001 and Grant 201800002, and in part by the China Scholarship Council under Grant 201708120039 and Grant 201708120040.

ABSTRACT Characters, as a kind of symbols carrying rich semantic information, are composed of strokes arranged in a certain structure and are of great significance in our daily life. In this paper, we are concerned with the problem of scene character recognition, and study the problem from the perspective of feature representation. We propose a novel pooling method termed deep contextual stroke pooling (DCSP) for scene character recognition. The proposed DCSP discovers the most prominent stroke information by using stroke detectors and captures the spatial context of discriminative strokes by learning contextual factor. Specifically, we first utilize the convolutional summing map in one convolutional layer to select discriminative strokes and use the convolutional activation features of discriminative strokes to train stroke detectors. Then, we propose the contextual factor to represent the co-occurrence probability of the stroke and its location. Finally, in the response regions, we incorporate the contextual factor into the detector scores and obtain the deep contextual confidence vectors of scene characters. Extensive experiments are conducted on three databases, i.e., ICDAR2003, Chars74k, and SVHN, and the experimental results demonstrate that our method achieves higher accuracies than the state-of-the-art methods.

INDEX TERMS Scene character recognition, deep contextual stroke pooling, contextual factor.

I. INTRODUCTION

Automatically understanding text information contained in scene images is a fundamental issue for a number of vision applications, such as image and video retrieval [1]–[3], human-computer interaction [4]–[6], web content analysis [7], [8] and so on. Scene text understanding system usually contains two stages: text detection and text recognition. Text detection stage is established to segment the text components from the input scene images. As the prerequisite and basis for successful text recognition, many detection approaches [9]–[11] have been proposed and achieved promising results in recent years. Even though the text components are completely and accurately detected, recognizing them is still an open problem due to the interferences of complex background, non-uniform illumination, noise, blur, various fonts, etc. Hence, more and more researchers focus on the text recognition stage. For example, Shi *et al.* [12] built part-based tree-structures and utilized a conditional random field model for scene text recognition. Wang *et al.* [13] utilized convolutional neural networks (CNNs) to recognize scene texts and achieved impressive accuracy using both the original training images and those synthetic ones. Yao *et al.* [14] proposed strokelets to capture the underlying substructures of characters and constructed a scene text recognition algorithm based on strokelets. Typically, scene text recognition employs a two-stage pipeline, i.e., scene character recognition and scene text recognition by combining language models [12], [15], [16]. Since the scene character recognition is the primary determinant to the following scene text recognition, we focus on the scene character recognition in this paper.

The existing scene character recognition methods can be divided into two main categories: the optical character recognition (OCR) based methods and the object recognition based methods. The OCR based methods [17], [18] first perform scene text binarization and then rely on the highly developed OCR engines to recognize the scene characters. However,

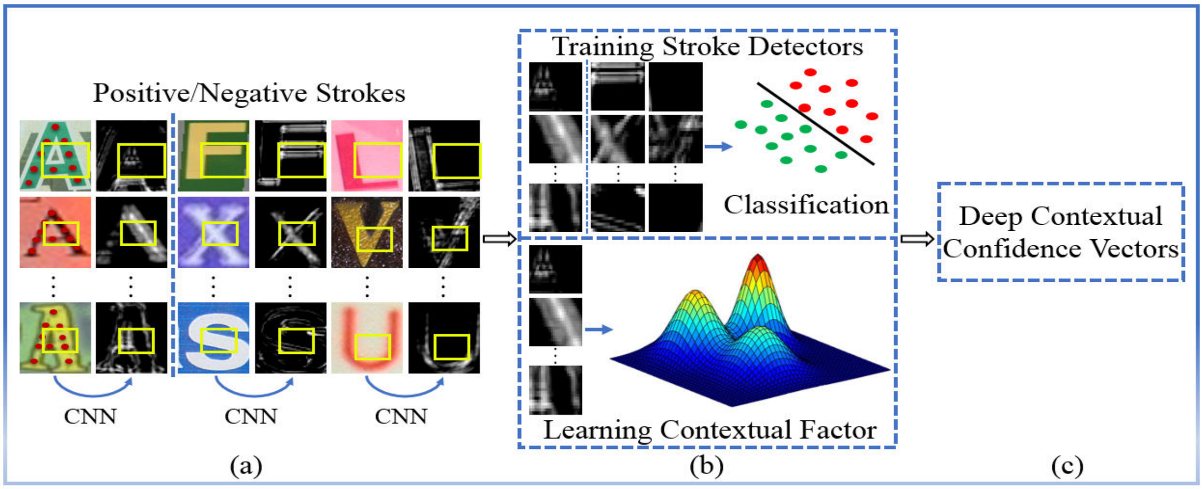

Positive/Negative Strokes
CNN
CNN
CNN
(a)
Training Stroke Detectors
Classification
Learning Contextual Factor
(b)
Deep Contextual Confidence Vectors
(c)

FIGURE 1. **The flowchart of the proposed DCSP method.**

binarizing scene text block is a challenging task due to different lighting conditions, heavy occlusion and low resolutions. The object recognition based methods [12], [13] skip the binarization step and treat each kind of character as a special object. Therefore, the scene character recognition problem is translated into a multi-class object classification task. The challenges of accurately recognizing scene characters lie in arbitrary fonts, noises, deformations, complex backgrounds and so on. Therefore, a powerful and effective feature representation strategy is indispensable for scene character recognition. Mishra *et al.* [16] utilized the histograms of oriented gradients (HOG) features to describe scene character images. Zhang *et al.* [19] utilized the histograms of sparse codes (HSC) features to express the scene characters. Weinman *et al.* [20] tied together several important information sources, i.e., similarity, language properties and lexical decision, in the stage of scene character recognition. Although these methods achieve promising results, the spatial context among local regions, which plays a profound role in scene character representation, is largely ignored.

In order to overcome the above-mentioned limitation, Gao *et al.* [22] proposed a spatiality embedded dictionary (SED) to model co-occurrence of several local strokes which introduces precise spatial information during feature representation stage. Lee *et al.* [23] incorporated the pixel-wise low-level image features and automatically mined discriminative features in subregions, so that the resulting features are able to effectively model distinctive spatial structures of each individual character. Gao *et al.* [24] proposed stroke banks to consider the spatial context in the stage of scene character representation. Shi *et al.* [25] extended [24] to discriminative multi-scale stroke detector-based representation (DMSDR) which utilizes strokes of various scales for recognizing scene characters. Tian *et al.* [26] proposed the co-occurrence histogram of oriented gradient (Co-HOG) features and convolutional Co-HOG (ConvCo-HOG) features for accurate representing scene characters. Compared with traditional HOG features which count orientation frequency of each single pixel, the Co-HOG features could learn more spatial contextual information by capturing spatial distribution of neighboring orientation pairs. Additionally, the ConvCo-HOG features exhaustively extract Co-HOG features from every possible patch regions within a character image for more spatial information.

In recent years, many researchers resort to extracting features from convolutional neural network (CNN). Wang *et al.* [27] regarded the output of the last fully-connected layer of CNN as the final image representations. Jaderberg *et al.* [28] trained a CNN model for scene character recognition and reported impressive accuracy by using the fully-connected layer based features. Wang *et al.* [30] proposed to encode the multi-order co-occurrence activations (MCA) on the convolutional map, and then combined MCA with Fisher vector.

In this paper, we propose a novel pooling method named deep contextual stroke pooling (DCSP) for scene character recognition. The flowchart is shown in Figure 1. We first train a CNN for scene character recognition and then select discriminative strokes from convolutional summing map (CSM) in one convolutional layer. The CSM in the convolutional layer could describe the important features and spatial structural information which is especially meaningful for character recognition. In addition, we extract convolutional activation features from discriminative strokes and utilize the obtained convolutional activation features to train stroke detectors. Then, we learn the contextual factor to represent the co-occurrence probability of the stroke and its location. Finally, in the response regions, we incorporate the contextual factor into the detector scores and obtain the deep contextual confidence vectors of scene characters. Experimental results demonstrate the effectiveness of the proposed methods. The major advantages of the proposed method lie in: (1) we

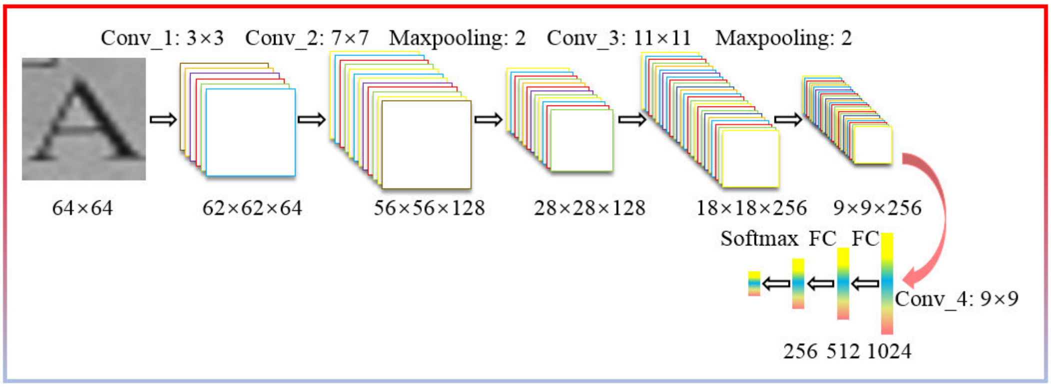


FIGURE 2. The architecture of the CNN model for scene character recognition.

select discriminative strokes from the CSM and utilize convolutional activation features to train the stroke detectors; (2) we propose the contextual factor, which are learned from a Gaussian mixture model (GMM) for each stroke, to explicitly consider the spatial context of strokes.

The rest of this paper is organized as follows. We introduce the proposed deep contextual stroke pooling (DCSP) method for scene character recognition in Section 2. Section 3 provides comprehensive analysis of the proposed DCSP method on three databases, i.e., ICDAR2003, Chars74k and SVHN, and the experimental results demonstrate that our method outperforms the other state-of-the-art methods. Finally, we conclude the paper in Section 4.

II. APPROACH

In this section, we first introduce the convolutional summing map (CSM) of the CNN which is utilized to select discriminative strokes, train stroke detectors and learn contextual factors. Afterwards, we detail the procedure of deep contextual stroke pooling for scene character images.

A. CONVOLUTIONAL SUMMING MAP

In the convolutional layer of a CNN, the filters traverse the image in a sliding-window manner to generate convolutional maps. The convolutional maps can be regarded as a tensor with the size of $W \times H \times M$, which possesses M convolutional maps with width W and height H. Typically, the top-left (bottom-right) activation response in a convolutional map is generated by the top-left (bottom-right) part of the input image. Each activation response in a convolutional map describes a local part of the input image and the high responses indicate the salient parts. Hence, we utilize the convolutional maps for selecting discriminative strokes, training stroke detectors and learning contextual factors.

In this paper, we train a CNN for scene character recognition and the network architecture is shown in Figure 2. In the CNN, the size of the input image is 64 × 64. First, the input image is convolved with 64 filters of size 3 × 3, resulting in a convolutional map of size 62 × 62 and 64 convolutional maps. The 64 convolutional maps are convolved with 128 filters of size 7 × 7. Then, max pooling is implemented by a 2 × 2 pixel window, with stride 2. The 128 convolutional maps are further convolved with 256 filters of size 11× 11. The sequence continues by performing max pooling and convolving with 1024 filters of size 9 × 9. Subsequently, the two fully-connected (FC) layers are 512 and 256 dimensional vectors, respectively. The last FC layer is followed by a softmax unit which converts the activations into character probabilities. To learn the parameters (weights and bias), we train the network using the backpropagation gradient-descent procedure [29] with a mini-batch size of 48. The procedure is terminated at 80 epochs. For the first 60 epochs, the learning rate is set to 0.001. While for the remaining 20 epochs, the learning rate is set to 0.0001.

The convolutional maps in the convolutional layer could describe the spatial structural information of characters. To further capture the completed spatial structural information, we add all the convolutional maps of one convolutional layer to obtain the convolutional summing map (CSM). Let $C_l(i,j)$ denote the activation response of CSM at position (i,j) in the l-th convolutional layer:

$$C_l(i,j) = \sum_{m=1}^{M} c_l^m(i,j), \tag{1}$$

where $c_l^m(i,j)$ denotes the activation response of the m-th convolutional map at position (i,j) in the l-th convolutional layer and M is the number of the convolutional maps in the l-th convolutional layer. The shallow convolutional layers usually contain structural and textural local features which are the key cues for scene character recognition. Hence, we utilize the CSM in the 2-th convolutional layer for selecting discriminative strokes, training stroke detectors and learning contextual factors.

B. SELECTING DISCRIMINATIVE STROKES

Each category of scene characters possess different stroke structures. To capture the main structural information of characters, we label key points for all the character images in the

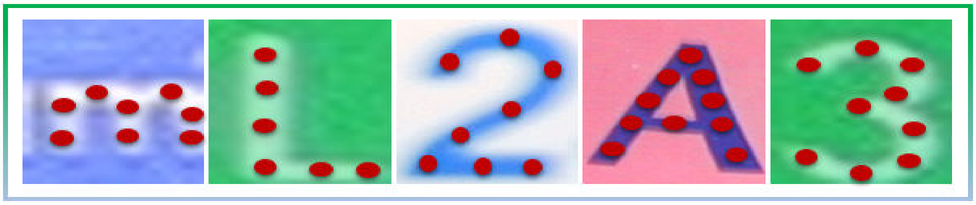


FIGURE 3. Some samples that labelled key points.

TABLE 1. The number of the labelled key points for 62 character categories.

Category	‘a’	‘b’	‘c’	‘d’	‘e’	‘f’	‘g’
Number	9	8	7	8	8	8	8
Category	‘h’	‘i’	‘j’	‘k’	‘l’	‘m’	‘n’
Number	7	4	6	9	5	7	8
Category	‘o’	‘p’	‘q’	‘r’	‘s’	‘t’	‘u’
Number	8	8	6	7	9	9	8
Category	‘v’	‘w’	‘x’	‘y’	‘z’	‘A’	‘B’
Number	7	9	9	8	9	10	10
Category	‘C’	‘D’	‘E’	‘F’	‘G’	‘H’	‘I’
Number	7	7	9	9	9	11	9
Category	‘J’	‘K’	‘L’	‘M’	‘N’	‘O’	‘P’
Number	6	9	6	9	7	8	8
Category	‘Q’	‘R’	‘S’	‘T’	‘U’	‘V’	‘W’
Number	10	10	9	8	7	7	9
Category	‘X’	‘Y’	‘Z’	‘0’	‘1’	‘2’	‘3’
Number	9	7	9	8	6	8	9
Category	‘4’	‘5’	‘6’	‘7’	‘8’	‘9’	-
Number	8	10	8	7	7	8	-

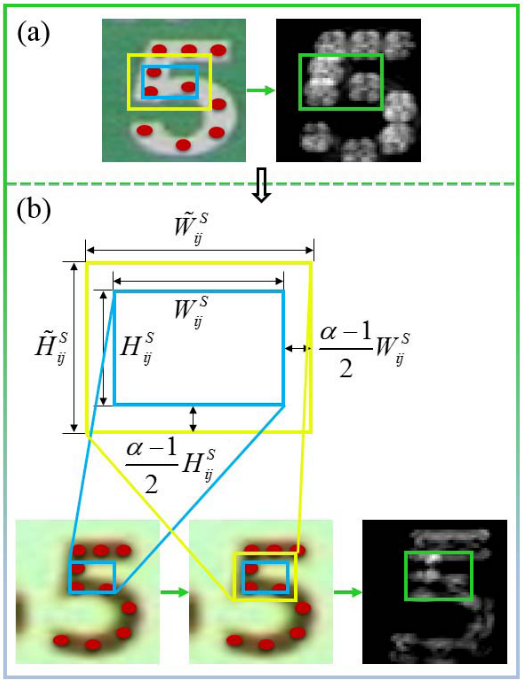


FIGURE 4. The procedure of discriminative stroke selection.

training set. We first normalize character images into size $64\times$ 64, where 64 and 64 are the height and width of an image, respectively. Then, key points for a-z, A-Z, 0-9 are labelled manually, and some examples are shown in Figure 3. The number of labelled key points for 62 character categories are listed in Table 1.

To discover the most discriminative strokes, we utilize the CSM to build the relationship between the original image and the corresponding convolutional activations. Based on the labelled key points of category i ($i \in \{1, 2, \cdots, n\}$, and n is the number of character categories) and the CSM in the 2-th convolutional layer, the discriminative strokes that only belong to category i can be selected. When a desired discriminative stroke of one training image is selected manually, the corresponding discriminative strokes can be automatically extracted from all the training images in category i. The above procedure is listed as follows:

(1) We arbitrarily choose one training image in category i. As shown in Figure 4 (a), we first delimit the blue rectangle manually which contains the main structural information of the character, and there are at least two key points in the blue rectangle. Then, we extend the blue rectangle into the yellow rectangle by using α, where α is a positive number to adjust the scale of extension. Finally, the yellow rectangle is mapped to the green rectangle in the CSM to obtain one discriminative stroke. The discriminative stroke is denoted as $Stroke_{ij}$ which represents the j-th stroke in category i.

(2) For the remaining training images in category i, we first calculate the minimal rectangle $H^S_{ij} \times W^S_{ij}$ (the blue rectangle in Figure 4(b)) which contains the same key points as $Stroke_{ij}$ and extend it into the size of $\tilde{H}^S_{ij} \times \tilde{W}^S_{ij}$ (the yellow rectangle in Figure 4(b)). Here, $\tilde{H}^S_{ij} = \alpha H^S_{ij}$ and $\tilde{W}^S_{ij} = \alpha W^S_{ij}$. Then, the yellow rectangle is mapped to the green rectangle in Figure 4(b) to obtain the corresponding discriminative stroke. The stroke number in each category is determined by the unique structural information of the characters.

C. TRAINING STROKE DETECTORS

To discover the important feature information of each stroke, we train a stroke detector and find the optimal separating hyperplane. The stroke detector is trained by the discriminative strokes in the CSM. More formally, let S_{ij} represent the j-th stroke detector in category i. The procedure is listed as follows:

(1) Collecting positive and negative stroke regions. We first locate the minimal rectangle according to the key points in one image from the i-th category and utilize the extension value α to obtain a patch region. The patch region is mapped to the CSM to obtain corresponding discriminative stroke which is treated as the positive stroke region for training the stroke detector. We crop the negative stroke regions, with the same size and position as the positive one, from the random training images in other character categories. Repeating the above steps, we can collect the positive and negative stroke regions for training the stroke detector S_{ij}.

(2) Extracting convolutional activation features. The procedure is shown in Figure 5. First, the response values in the same position of all the convolutional maps are concatenated into a feature vector. Then, we aggregate all feature vectors in the stroke region using average pooling, resulting in the convolutional activation feature to represent the stroke region.

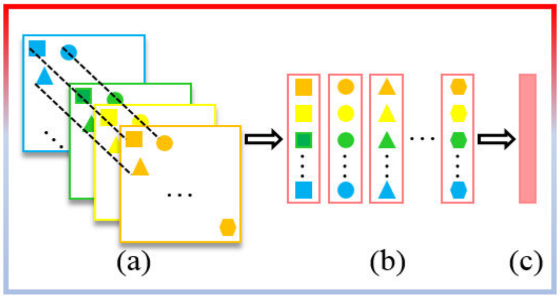


FIGURE 5. The procedure of extracting convolutional activation features for stroke regions. It's composed of (a) concatenating the response values in the same position of the convolutional maps, (b) performing average pooling to obtain (c) the convolutional activation feature.

(3) Training stroke detectors. We feed the convolutional activation features of stroke regions into SVM classifier with the linear kernel to train the stroke detector S_{ij}. Let S represent the stroke detector set for all the character categories and it can be formulated as:

$$S = (S_{11}, S_{12}, \cdots, S_{1j}, \cdots, S_{21}, S_{22}, \cdots, S_{2j}, \cdots, S_{n1}, S_{n2}, \cdots, S_{nj}, \cdots). \quad (2)$$

The traditional methods [24], [25] utilize the trained stroke detectors to generate the detector scores by employing the sliding window strategy in the response region. Then, they aggregate these scores using max pooling. However, the max pooling does not explicitly consider the spatial context, which is insufficient to generate rich and robust feature representations. Moreover, they only select the maximum score in the response region while ignores the other important feature information.

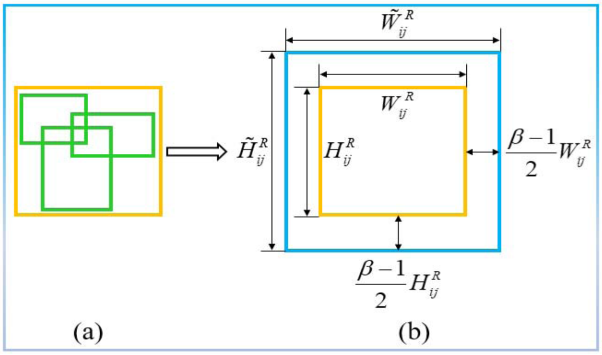


FIGURE 6. The procedure of delimiting a response region in the CSM.

D. DEEP CONTEXTUAL STROKE POOLING

To overcome the above-mentioned limitations, we propose the DCSP for scene character recognition. Firstly, we propose the contextual factor, which learns a GMM for each stroke, to explicitly consider the spatial context of strokes. Specifically, we extract the position information l_{ij} of the positive stroke regions in the CSM for $Stroke_{ij}$, where l_{ij} is the set of upper left coordinates for the j-th stroke in category i. Then, we utilize the position information l_{ij} to learn a GMM for $Stroke_{ij}$. The GMM is composed of several Gaussian functions and reflects a statistical relationship between $Stroke_{ij}$ and the position l_{ij}. The bigger the probability of $Stroke_{ij}$ appearing in the position l_{ij} is, the greater the relevance between them is. Hence, the relationship between the stroke and its position is formulated as:

$$P(l_{ij}|Stroke_{ij}) = \sum_{k=1}^{K} B_{ijk}\eta_{ijk}(l_{ij}, \mu_{ijk}, \Sigma_{ijk}), \quad (3)$$

where $P(l_{ij}|Stroke_{ij})$ represents the probability of $Stroke_{ij}$ appearing in the position l_{ij}, K is the number of Gaussian components, and B_{ijk} is the weight of the k-th Gaussian. Here, η_{ijk} is a Gaussian probability density function:

$$\eta_{ijk}(l_{ij}, \mu_{ijk}, \Sigma_{ijk}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma_{ijk}|^{\frac{1}{2}}} \times exp[-\frac{1}{2}(l_{ij} - \mu_{ijk})^T {\Sigma_{ijk}}^{-1}(l_{ij} - \mu_{ijk})], \quad (4)$$

where d is the dimensionality of l_{ij}, and μ_{ijk} and Σ_{ijk} are the mean vector and the covariance matrix of the k-th Gaussian component, respectively. Since $P(l_{ij}|Stroke_{ij})$ reflects the spatial co-occurrence relationship between l_{ij} and $Stroke_{ij}$, we define it as the contextual factor γ_{ij}, i.e., $\gamma_{ij} = P(l_{ij}|Stroke_{ij})$.

In the pooling stage, we incorporate the contextual factor into the detector scores and obtain the deep contextual confidence vectors. The pipeline of representing the deep contextual confidence vectors consists of two main components.

(1) Delimitating a response region in the CSM for $Stroke_{ij}$. To adjust the stroke location changes for various environments, we first obtain the union set (the orange rectangle) of all positive stroke regions (the green rectangles) for $Stroke_{ij}$ as shown in Figure 6(a). Then we extend it by a extension

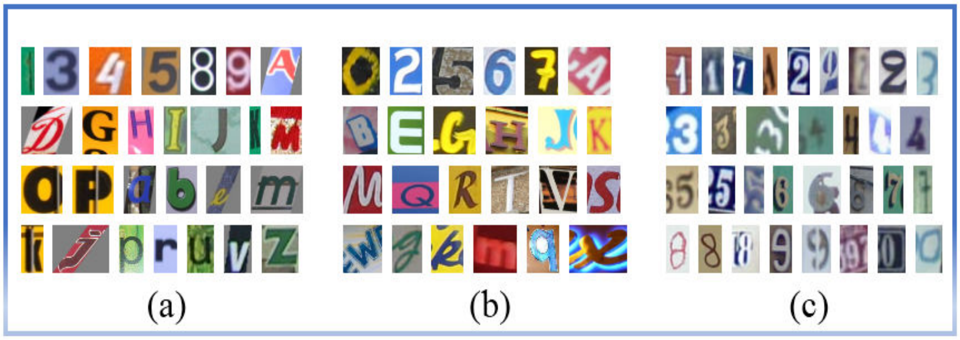


FIGURE 7. Some examples from (a) ICDAR2003, (b) Chars74k and (c) SVHN databases.

value β as shown in Figure 6(b). Finally, the response region is delimited as $\tilde{H}_{ij}^R \times \tilde{W}_{ij}^R$, where $\tilde{H}_{ij}^R$ and $\tilde{W}_{ij}^R$ are the height and width of the response region, respectively. Here, $\tilde{H}_{ij}^R = \beta H_{ij}$ and $\tilde{W}_{ij}^R = \beta W_{ij}$.

(2) Incorporating the contextual factor. In the response region, we use a sub-window to densely sample the local activation response region a ($a \in \{1, 2, \cdots, h\}$, and h is the number of activation response regions). Then, the stroke detector S_{ij} is applied to classify all the h activation response regions, and the detector scores are:

$$O_{ij} = (O_{ij1}, O_{ij2}, \cdots, O_{ijh}), \tag{5}$$

where O_{ij} is a set of detector scores, and O_{ijh} is the detector score for the h-th activation response region. To obtain the most prominent stroke information, we retain the top $[T \cdot h]$ detector scores and set the remaining ones to 0. Here, T ($0 \leq T \leq 1$) is the ratio of preserved detector scores, and $[\cdot]$ represents the rounding operation. To consider the spatial context, we calculate the contextual factor for each activation response region:

$$\gamma_{ij} = (\gamma_{ij1}, \gamma_{ij2}, \cdots, \gamma_{ijh}), \tag{6}$$

where γ_{ijh} represents the contextual factor for the h-th activation response region. It indicates the probability that $Stroke_{ij}$ appears in l_{ijh}. We also preserve the top $[T \cdot h]$ contextual factors and set the remaining ones to 0, so as to obtain the most prominent spatial context. The final deep contextual confidence vector C is obtained for each scene character image:

$$\begin{aligned} C = (&C_{11}, C_{12}, \cdots, C_{1j}, \cdots, C_{21}, C_{22}, \cdots, \\ &C_{2j}, \cdots, C_{n1}, C_{n2}, \cdots, C_{nj}, \cdots), \end{aligned} \tag{7}$$

where $C_{ij} = \gamma_{ij} \cdot O_{ij} = \sum_{a=1}^{h} \gamma_{ija} O_{ija}$, and C_{ij} is the contextual confidence score for $Stroke_{ij}$.

Given a scene character image, we first transmit the scene character image into the CNN and calculate the deep contextual confidence vector using Equation (7). Then, the deep contextual confidence vectors of all training images are used to train multi-class SVMs. In the test stage, we use the same response region of each stroke as the training stage, and then employ Equation (7) to calculate the deep contextual confidence vectors. Finally, those deep contextual confidence vectors of all test images are fed into the multi-class SVMs to obtain class labels.

III. EXPERIMENTAL RESULTS

A. DATABASE AND EXPERIMENTAL SETTINGS

We evaluate the effectiveness of the proposed DCSP on three public databases, i.e., ICDAR2003 [31], Chars74k [32] and SVHN [33] databases. The ICDAR2003 database is a typical scene character recognition database. It contains 6,185 training and 5,430 test images distributed in 62 classes of characters, i.e., lower English letters a-z, upper English letters A-Z and digits 0–9. These images undergo extensive variances such as nonuniform illumination, distortions and complex backgrounds. Figure 7(a) shows some images from the ICDAR2003 database. The Chars74k database contains 62 character classes (a-z, A-Z, 0-9). The characters in this database are cropped from the images of advertisement signs, products from stores, and vary in color, size, font, background, etc. Some images from the Chars74k database are shown in Figure 7(b). In the experiment, we randomly select 30 images in each class, among which 15 images are used for training and the remaining are used for test as in [25] and [32]. The SVHN database is a collection of street view images with text of various deformations, distortions, lighting conditions and complex backgrounds. The database consists of over 600,000 labelled characters comprising full numbers and cropped digits. The training database contains 73,257 digits and 531,131 additional less difficult training images. The test database consists of 26,032 digits. Some characters from the SVHN database are listed in Figure 7(c). Similar to [25], we only report results by using 700 training images per class and ignoring the 531,131 additional less difficult training images in the experiments.

All of the images are scaled to 64 × 64. The extension values α and β are both empirically set to 1.5. To verify the effectiveness, the proposed method are compared with other leading methods as well as the baseline algorithms,

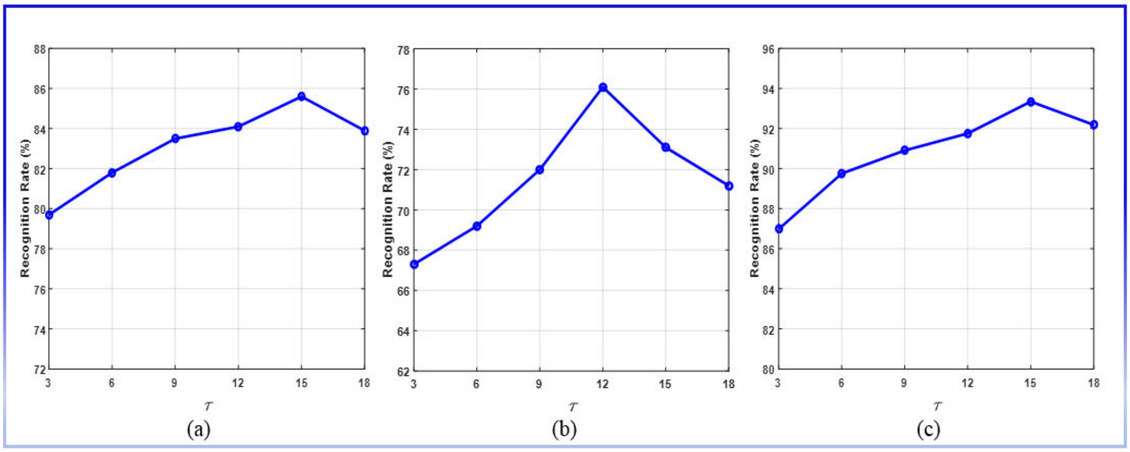


FIGURE 8. Performance under different number of discriminative strokes per character. (a), (b) and (c) are for the ICDAR2003, Chars74k and SVHN database, respectively.

i.e., average pooling (AP), max pooling (MP) and contextual stroke pooling (CSP). The AP (MP) means that we employ the CSM in the convolutional layer for selecting discriminative strokes, training stroke detectors and learning contextual factors, and utilize the average (max) pooling strategy to obtain the final representation. While the CSP means that we select discriminative strokes, train stroke detectors and learn contextual factors on the original character images, and utilize the proposed pooling strategy to obtain the final representation.

B. INFLUENCE OF PARAMETER VARIANCES

Before presenting the results, we first evaluate the performance of the proposed DCSP with respect to the index of convolutional layers l, the discriminative stroke number τ in each character class, the number of Gaussian components K in each GMM, and the ratio T of the preserved detector scores and contextual factors in each response region.

(1) In a CNN, the shallow convolutional layers usually contain structural and textural local features while the deep convolutional layers usually contain high-level semantic information. In the scene character recognition task, to discover the most prominent feature and stroke information, we select the CSM in the shallow convolutional layers for selecting discriminative strokes, training stroke detectors and learning contextual factors. We evaluate the performance of the proposed method when using CSM in different shallow convolutional layers. The results are listed in Table 2. From Table 2, on the three databases, the experimental results indicate that the proposed method achieves the highest results when utilizing the CSM in the 2-th convolutional layer.

TABLE 2. Performance of the CSM in different convolutional layers on the ICDAR2003, Chars74K and SVHN databases.

Layer	ICDAR2003	Chars74K	SVHN
Conv_1	77.9	70.2	86.0
Conv_2	**85.6**	**76.1**	**93.4**

(2) Discriminative stroke number τ in each character class is an importance parameter as it determines the number of stroke detectors and effects the dimensionality of the final deep contextual confidence vector. We study the impact of the discriminative stroke number τ in each character class on the ICDAR2003, Chars74K and SVHN databases. As shown in Figure 8, we find that the recognition rate improves with the increasing number of discriminative strokes in a range, and then the improvement stops when coming to a certain point. The results mean that when τ is too small, there is few discriminant information in the final representation, yet when τ is too large, the dimensionality of the final representation is too high which may result in the dimensionality curse. Therefore, the proper τ can not only preserve the discriminative information, but also make the final deep contextual confidence vectors possess the appropriate dimensionality. As for the ICDAR2003, Chars74K and SVHN databases, τ is ultimately set to 15, 12 and 15, respectively, where the highest accuracy is achieved.

(3) To build the optimal GMM for the spatial context of strokes, we study the effect of the number of Gaussian components K in each GMM. The paper mainly reports the results on the ICDAR2003 database, and our experiments show that the conclusions can be generalized to the Chars74k and SVHN databases as well. Figure 9 shows the performance when $K = 1, 2, 3, 4$. As can be seen, large number of Gaussian components in each GMM leads to better performance. However, when K goes larger than 3, the performance starts to drop. Hence, K is set to 3.

(4) To capture the most prominent information of discriminative strokes, we discuss the impact of the ratio T which is the preserved detector scores and contextual factors in each response region, on one baseline method (AP) and the proposed DCSP. As shown in Figure 10, as for the ICDAR2003 database, both the AP and the proposed DCSP

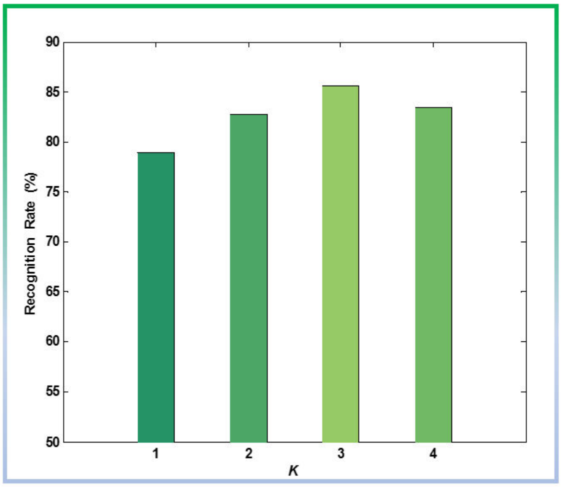


FIGURE 9. Influence of the number of Gaussian components *K* in each GMM on the ICDAR2003 database.

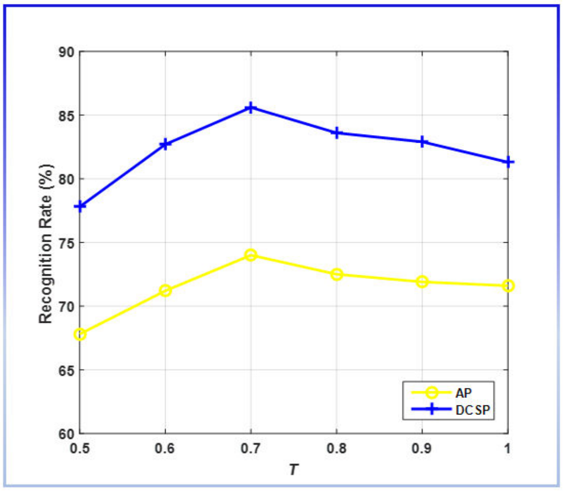


FIGURE 10. Performance under different ratio *T* of the preserved detector scores and contextual factors in each response region on the ICDAR2003 database.

method achieve the highest accuracy when T is equal to 0.7. The experimental result can be generalized to the Chars74k and SVHN databases.

C. ICDAR2003 dATABASE

We compare the proposed DCSP with other competing methods and the baseline algorithms on the ICDAR2003 database. The recognition results are listed in Table 3. With the optimal parameters, i.e., $\tau = 15$, $K = 3$ and $T = 0.7$, the proposed achieves the highest accuracy of 85.6%. Furthermore,the following three points can be drawn through analyzing the experimental results.

First, the DCSP reduces to the AP when all contextual factors are set to $1/h$, where h refers to the same meaning as in Equation (6). The accuracy of DCSP is about 11% higher than that of AP. Intrinsically, the DCSP learns the pooling weights using contextual factors, while the pooling weights of AP are equal.

TABLE 3. Recognition results of different algorithms on the ICDAR2003 database.

Algorithms	Accuracy (%)
HOG+SVM [25]	77
Co-HOG [26]	80.5
Stroke Bank [24]	79.8
DMSDR [25]	81.7
SED [22]	82.0
DSEDR [25]	82.6
MCA-FV [30]	83.4
CNN_Softmax [34]	81.5
AP	74.0
MP	79.4
CSP	84.1
DCSP	**85.6**

Second, when ignoring contextual factors and only preserving the maximum detector scores in the response region, the proposed DCSP degenerates to the traditional MP method. Since the proposed DCSP explicitly considers the spatial context of strokes using contextual factors, it outperforms MP by 6.2%.

Third, comparing the results of DCSP and CSP, the former is 1.5% higher than the latter one. It is because the proposed DCSP utilizes the CSM in the convolutional layer which could capture more most prominent stroke and feature information and remove the less important ones, while the CSP select discriminative strokes, train stroke detectors and learn contextual factors on original character images.

D. CHARS74K DATABASE

We evaluate the proposed DCSP on the Chars74K. From Table 4, the experimental results indicate that when utilizing the optimal parameters, the proposed DCSP achieves the highest accuracy of 76.1%. The proposed DCSP outperforms CNN_Softmax by more than 2%. It is because the proposed DCSP utilizes the CSM, stroke detectors and contextual factors to integrate the most prominent feature and stroke information into the deep contextual confidence vector. Compared with AP and MP, our method achieves superior performance due to preserving the spatial context of strokes and the most prominent stroke information in the pooling stage. The proposed DCSP obtains higher results than CSP because the proposed DCSP utilizes the CSM in the convolutional layer which could capture more meaningful information and remove the less important ones, while the CSP selects discriminative strokes, trains stroke detectors and learns contextual factors on original character images.

E. SVHN DATABASE

We compare the proposed DCSP with other competing methods and the baseline algorithms on the SVHN database. Table 5 lists the performance of different algorithms. From

TABLE 4. **Recognition results of different algorithms on the Chars74K database.**

Algorithms	Accuracy (%)
HOG+SVM [25]	62
Stroke Bank [24]	65.7
DMSDR [25]	66.1
SED [22]	67.1
DSEDR [25]	71.8
CNN_Softmax [34]	73.5
AP	60.8
MP	66.5
CSP	72.3
DCSP	**76.1**

TABLE 5. **Recognition results of different algorithms on the SVHN database.**

Algorithms	Accuracy (%)
HOG+SVM [25]	80.0
K-means [33]	90.6
ConvNet/Smaller training [35]	91.6
DMSDR [25]	91.6
DSEDR [25]	89.2
AP	82.0
MP	89.0
CSP	92.8
DCSP	**93.4**

the Table 5, we can see that the proposed DCSP obtains the best recognition accuracy when using the optimal parameters. With 73,257 training samples, ConvNet/Smaller training [35] utilizes CNN to classify digits of real-world house numbers and achieves a recognition accuracy of 91.6%. With 700 training samples per class, i.e., 7,000 training images in total, the proposed DCSP could correctly recognize 93.4% of the test images. The superiorities of our method lie in: (1) the convolutional activation features extracted from the second convolutional layer possess stronger discriminative ability; (2) the proposed DCSP can capture more most prominent stroke and feature information and remove the less important ones by using contextual factors. Once again, we prove the effectiveness of our method on this challenging database.

IV. CONCLUSION

In this paper, we have introduced a novel feature pooling method named DCSP for recognizing characters in natural scenes. The proposed DCSP utilizes the discriminative strokes selected from CSM to train stroke detectors, and uses the contextual factor to reflect the spatial context information of discriminative strokes. Based on detector scores and the contextual factor, the most representative convolutional activation features can be selected from the response regions, which could improve the discrimination and robustness of the final deep contextual confidence vectors. The proposed DCSP has been validated on three well-known databases, i.e., ICDAR2003, Chars74k and SVHN, and the experimental results outperform other previous methods in scene character recognition.

REFERENCES

[1] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H.-J. Zhang, "Image classification for content-based indexing," *IEEE Trans. Image Process.*, vol. 10, no. 1, pp. 117–130, Jan. 2001.

[2] D. C. G. Pedronette, R. T. Calumby, and R. D. S. Torres, "A semi-supervised learning algorithm for relevance feedback and collaborative image retrieval," *EURASIP J. Image Video Process.*, vol. 2015, no. 1, p. 27, 2015.

[3] S. Li, S. Purushotham, C. Chen, Y. Ren, and C.-C. J. Kuo, "Measuring and predicting tag importance for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2423–2436, Dec. 2017.

[4] V. I. Pavlovic, R. Sharma, and T. S. Huang, "Visual interpretation of hand gestures for human-computer interaction: A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 677–695, Jul. 1997.

[5] N. Dawar and N. Kehtarnavaz, "Real-time continuous detection and recognition of subject-specific smart tv gestures via fusion of depth and inertial sensing," *IEEE Access*, vol. 6, pp. 7019–7028, 2018.

[6] M. Chen, F. Herrera, and K. Hwang, "Cognitive computing: Human-centered computing with cognitive intelligence on clouds," *IEEE Access*, to be published, doi: 10.1109/ACCESS.2018.2791469.

[7] R. Kosala and H. Blockeel, "Web mining research: A survey," *ACM SIGKDD Explorations Newslett.*, vol. 2, no. 1, pp. 1–15, 2000.

[8] S. C. Herring, "Web content analysis: Expanding the paradigm," in *International Handbook of Internet Research*. Dordrecht, The Netherlands: Springer, 2009, pp. 233–249.

[9] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A Laplacian approach to multi-oriented text detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 412–419, Feb. 2011.

[10] Y.-F. Pan, X. Hou, and C.-L. Liu, "A hybrid approach to detect and localize texts in natural scene images," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 800–813, Mar. 2011.

[11] H. Zhang, K. Zhao, Y.-Z. Song, and J. Guo, "Text extraction from natural scene image: A survey," *Neurocomputing*, vol. 122, pp. 310–323, Dec. 2013.

[12] C. Shi, C. Wang, B. Xiao, Y. Zhang, S. Gao, and Z. Zhang, "Scene text recognition using part-based tree-structured character detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2961–2968.

[13] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[14] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4042–4049.

[15] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[16] A. Mishra, K. Alahari, and C. V. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2687–2694.

[17] X. Chen and A. L. Yuille, "Detecting and reading text in natural scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, pp. 366–373.

[18] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3538–3545.

[19] D. Zhang, D.-H. Wang, and H. Wang, "Scene text recognition using sparse coding based features," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 1066–1070.

[20] J. J. Weinman, E. Learned-Miller, and A. R. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1733–1746, Oct. 2009.

[21] C.-Z. Shi, C.-H. Wang, B.-H. Xiao, S. Gao, and J.-H. Hu, "Scene text recognition using structure-guided character detection and linguistic knowledge," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 7, pp. 1235–1250, Jul. 2014.

[22] S. Gao, C. Wang, B. Xiao, C. Shi, W. Zhou, and Z. Zhang, "Learning Co-occurrence strokes for scene character recognition based on spatiality embedded dictionary," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5956–5960.

[23] C.-Y. Lee, A. Bhardwaj, W. Di, V. Jagadeesh, and R. Piramuthu, "Region-based discriminative feature pooling for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4050–4057.

[24] S. Gao, C. Wang, B. Xiao, C. Shi, and Z. Zhang, "Stroke bank: A high-level representation for scene character recognition," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2014, pp. 2909–2913.

[25] C.-Z. Shi, S. Gao, M.-T. Liu, C.-Z. Qi, C.-H. Wang, and B.-H. Xiao, "Stroke detector and structure based models for character recognition: A comparative study," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4952–4964, Dec. 2015.

[26] S. Tian *et al.*, "Multilingual scene character recognition with co-occurrence of histogram of oriented gradients," *Pattern Recognit.*, vol. 51, pp. 125–134, Mar. 2016.

[27] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 3304–3308.

[28] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 512–528.

[29] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. Int. Conf. Comput. Statist.*, 2010, pp. 177–186.

[30] Y. Wang, C. Shi, C. Wang, B. Xiao, and C. Qi, "Multi-order co-occurrence activations encoded with Fisher Vector for scene character recognition," *Pattern Recognit. Lett.*, vol. 97, pp. 69–76, Oct. 2017.

[31] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. Int. Conf. Document Anal. Recognit.*, Aug. 2003, pp. 682–687.

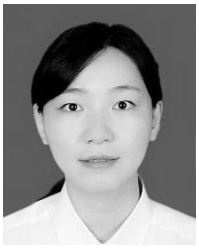
[32] T. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in *Proc. Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2009, pp. 273–280.

[33] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, p. 5.

[34] C. Shi, Y. Wang, F. Jia, K. He, C. Wang, and B. Xiao, "Fisher vector for scene character recognition: A comprehensive evaluation," *Pattern Recognit.*, vol. 72, pp. 1–14, Dec. 2017.

[35] P. Sermanet, S. Chintala, and Y. LeCun, "Convolutional neural networks applied to house numbers digit classification," in *Proc. Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 3288–3291.

ZHONG ZHANG (M'14) received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently an Associate Professor with Tianjin Normal University, Tianjin, China. He has authored or co-authored about 70 papers in international journals and conferences, such as the *Pattern Recognition*, the IEEE Transactions on Circuits Systems Video Technology, the IEEE Transactions on Information Forensics and Security, *Signal Processing* (Elsevier), CVPR, ICPR, and ICIP.



HONG WANG is currently pursuing the master's degree with Tianjin Normal University, Tianjin, China. Her research interests include scene character recognition and machine learning.

SHUANG LIU received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China. She is currently an Associate Professor with Tianjin Normal University, Tianjin, China.

BAIHUA XIAO received the B.S. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 1995, and the Ph.D. degree in pattern recognition and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2000. Since 2005, he has been a Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include pattern recognition, computer vision, image processing, and machine learning.