2015 13th International Conference on Document Analysis and Recognition (ICDAR)

Content-Independent Font Recognition on a Single Chinese Character using Sparse Representation

Weikang Song, Zhouhui Lian*, Yingmin Tang, Jianguo Xiao
Institute of Computer Science and Technology, Peking University
No.128 Zhongguancun North Street, Beijing 100080, P.R. China
Email: {songweikang, lianzhouhui, tangyingmin, jianguoxiao}@pku.edu.cn

***Abstract*—Font recognition on a single Chinese character is a challenging task especially when the identity of the character is unknown and the number of possible font types is huge. In this paper, we propose a novel method using multi-scale sparse representation to solve the problem of large-scale font recognition on a single unknown Chinese character. Specifically, we first apply a saliency-based sampling approach, which exploits the saliency information of character contours, to segment local patches in multiple scales from salient regions. Then, corresponding local descriptors are extracted by implementing Sobel and Prewitt operators in 4 directions. After encoding the local descriptors into sparse codes, max pooling and spatial pyramid matching are employed to pool them into a sparse representation. Finally, a multi-scale sparse representation is obtained by concatenating three sparse representations which respectively correspond to three particular scales of local patches, and then the linear SVM classifier is utilized for font classification. Experiments performed on a large-scale database consisting of Chinese character images in 160 fonts show that our method achieves significantly better performance compared to the state of the art. Moreover, we also carry out experiments on a subset of the database to demonstrate the effectiveness of our saliency-based sampling approach and the proposed Sobel-Prewitt feature.**

I. Introduction

Over the last few decades, considerable advancements have been achieved in Optical Character Recognition (OCR). However, much less effort has been devoted to font recognition which actually has close relationship with OCR. Recently, with the significant improvement of character recognition accuracy, researchers started to pay more attention to the investigation of font recognition. As we know, font information is beneficial for layout analysis and understanding. For example, the distinct portions of documents, such as the title and abstract, are typically represented in specific font styles, while the diversity of font styles would markedly increase the difficulty of character recognition tasks. Obviously, if the font information of each character can be obtained by applying some font recognition methods in OCR systems, a mono-font character recognition algorithm can then be utilized to achieve better recognition accuracy. Moreover, with font information, document reconstruction systems are able to recover both the document contents and the characters' font styles. In addition, many individuals, especially graphic designers, are quite interested in fonts they see in daily life and often desire the identification of those fonts for further applications.

Up to now, most of the previous work in font recognition have been presented for alphabetic scripts. Existing methods can be roughly classified into the following two categories: methods based on typographical features and methods based on textural features. For instance, Zramdini *et al.* [1] adopted a statistical approach based on global typographic features. Jung *et al.* [2] introduced ascenders, descenders and serifs typographical attributes to accomplish the recognition of multi-fonts and multi-size characters. Joshi *et al.* [3] stated that the textural features are more efficient than the typographical features when applied in font recognition. More recently, a novel global feature extraction algorithm based on statistical analysis of edge pixels in binary images was presented in [4]. Khosravi *et al.* [5] employed Sobel and Roberts operators to extract features for font recognition, and Lutf *et al.* [6] attempted to extract rotation invariant features from diacritics. Although some of the aforementioned approaches have achieved considerable font recognition accuracy, they are not well-suited in applications of Chinese Character Font Recognition (CCFR) since those methods are specially designed for alphabetic scripts. Unlike alphabetic scripts, such as Arabic, English, and Persian, which are constructed by letters, Chinese characters are composed of strokes overlapped with each other to establish a block which represents a word. That makes the shapes of Chinese characters more complicated and the font recognition task more challenging compared to alphabetic scripts. Zhu *et al.* [7] first normalized input documents to create a number of uniform text blocks and then utilized a set of Gabor filters to extract textural features. However, their method is incompetent to recognize the font of a single Chinese character. Font Recognition on a Single Chinese Character (FRSCC) is helpful for OCR and document analysis if more than one font is adopted in a given Chinese document. Ding *et al.* [8] introduced an algorithm for content-independent font recognition on a single Chinese character. Except for content-independent font recognition, content-dependent font recognition algorithms are applied when the identity of the character of the input image is known. It is much easier to handle content-dependent font recognition compared to the content-independent task because the weak differences between fonts can be concealed by the significant varieties

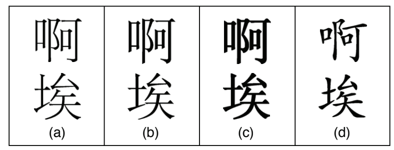

Fig. 1. Examples of Chinese characters represented in different fonts: (a) BaoSong, (b) SongTi, (c) SongTi Bold, (d) KaiTi.

*Zhouhui Lian is the corresponding author.

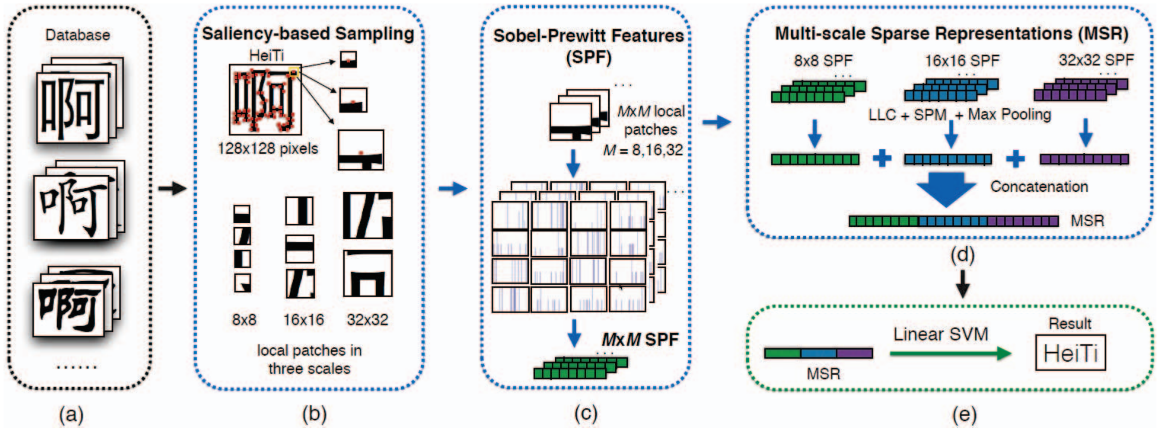

Fig. 2. Overview of our font recognition method using multi-scale sparse representation.

between contents. Take Fig.1 for an example, font styles (a) and (b) are very similar, (b) and (c) only differ in font weight, while (a), (b) and (c) are visually quite different from font (d). However, in all cases, it is easier to find a feature space where the same characters in different fonts are much similar than different characters in the same fonts. In addition, FRSCC is a large-scale recognition problem, since the number of categories of Chinese fonts might be huge. Hence, the problem of FRSCC becomes a great challenge when the identity of the character is unknown and the quantity of investigated fonts is large.

One potential way to solve the above-mentioned large-scale content-independent FRSCC problem is to adopt the latest techniques proposed in the image classification community. In the last few years, significant progresses have been made in the area of image classification. For instance, Fei-Fei *et al.* introduced the famous bag-of-features (BoF) model for the first time in [9]. However, the BoF approach disregards the spatial information of local descriptors, which results in the limitation that its performance is not satisfactory when tracking shapes or recognizing an object. To address this limitation, a typical extension of the BoF model, called Spatial Pyramid Matching (SPM), was proposed in [10]. Unfortunately, in order to get satisfactory performance, the traditional SPM method must be applied together with nonlinear Mercer kernels, e.g. the intersection kernel or the Chi-square kernel. In order to improve the scalability and performance of SPM, Yang *et al.* [11] employed sparse coding instead of vector quantization to acquire nonlinear codes, and presented the ScSPM method which achieved state-of-the-art performance. Later, Yu *et al.* [12] emphasized that under certain assumptions locality is more essential than sparity, and proposed Local Coordinate Coding (LCC) which is a modified version of ScSPM. However, both ScSPM and LCC require to solve the L1-norm optimization problem which is computationally expensive, Wang *et al.* [13] introduced a fast implementation of LCC, called Locality-constrained Linear Coding (LLC).

In this paper, we propose a novel multi-scale sparse representation for content-independent FRSCC. The main contributions of this paper are threefold. First, we employ a saliency-based sampling method based on the salient contour point detection algorithm introduced in [14], which is beneficial for capturing discriminative local patches. Second, we design a new algorithm based on Sobel and Prewitt gradients in multi-directions to extract robust local descriptors. Third, we propose a multi-scale sparse representation which is the concatenation of several sparse representations for local patches with multiple scales.

II. Method Description

In this section, we first present an overview of our method and then elaborate on the details. As depicted in Fig.2, the method consists of the following two stages: feature extraction (the blue dashed outline) and font classification (the green dashed outline).

Feature Extraction Stage: First, we normalize the resolution of each character image into 128×128. Then, we sample local patches of three scales from a given input image by the proposed *saliency-based sampling* method. In particular, resolutions of the three scales of local patches are experimentally chosen as $8{\times}8$, $16{\times}16$ and $32{\times}32$. Afterwards we extract local descriptors using *Sobel-Prewitt feature (SPF)* from each local patch. As a result, we acquire three sets of local descriptors from local patches with corresponding scales. As shown in Fig.2 (e), the three sets of local descriptors are illustrated in green, blue, and purple, respectively. For each set of local descriptors, we encode them into a sparse representation by applying LLC [13], max pooling [11] and SPM [10]. After concatenating three sparse representations that correspond to different scales, our *Multi-scale Sparse Representation (MSR)* is obtained.

Font Classification Stage: We divide all characters images into a training set and a testing set. After feature extraction stage, each input image is represented by a MSR, then the linear SVM classifier is utilized for training and classification. In this paper, we choose LIBLINEAR [15] to implement SVM classifiers.

A. Saliency-based Sampling

Our saliency-based sampling method utilizes the *salient point detection* algorithm proposed in [14] to select salient points as our interest points. The *salient point detection* algorithm first defines the *Deletion Cost* for each contour point to measure the price of deleting them. Then a contour point with the minimum *Deletion Cost* value is removed in one iteration until the global threshold is achieved. Fig.2 (b) illustrates interest points found by our saliency-based sampling method. We can observe that our method has the following advantages: (1) Compared with random sampling, our saliency-based sampling method obtains additional interest points in the severely changing parts of Chinese character contours which preserve the major divergences of different fonts. Specifically, severely changing parts of character contours include the beginning, ending, turning and overlapping regions of a stroke of Chinese characters. (2) Although the Harris corner detector [16] adopts the same idea as our method to sample salient points on the contours, the total number of points sampled by Harris corner detector are much less than our saliency-based sampling method. Actually, as we can see from experiments, our method obtains more important points and thus results in better recognition performance.

After sampling salient interest points, local patches can be obtained based on these interest points. For example, given an interest point p, we set up a bounding box whose center is p and then capture all pixels inside the bounding box as a local patch. Fig.2 (b) shows local patches with size 8×8, 16×16, and 32×32.

B. Sobel-Prewitt Feature Calculation

With salient local patches captured in the previous step, we calculate local descriptors based on image gradients in 16 directions using Sobel and Prewitt operators. Both Sobel and Prewitt operators utilize two 3×3 kernels which are convolved with the input image to calculate the approximate values of horizontal and vertical derivatives. The original SRF was proposed in [5], where Khosravi *et al.* utilized two Sobel kernels and two Reberts kernels to calculate gradients. As we know, since only two directions of kernels are used, the original SRF is not discriminative enough for local patches with complex textures. To address this problem, here we design an improved SPF method by selecting the Sobel and Prewitt kernels in $0°$, $45°$, $90°$ and $135°$ directions. Specifically, the four Sobel kernels are defined as follows

$$S_{0^\circ} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}, S_{90^\circ} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix},$$
$$S_{45^\circ} = \begin{bmatrix} +2 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, S_{135^\circ} = \begin{bmatrix} 0 & -1 & -2 \\ +1 & 0 & -1 \\ +2 & +1 & 0 \end{bmatrix}, \quad (1)$$

where S_{i° denotes the i° Sobel kernel. Similarly, for Prewitt kernels, we have

$$P_{0^\circ} = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix}, P_{90^\circ} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix},$$
$$P_{45^\circ} = \begin{bmatrix} +1 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix}, P_{135^\circ} = \begin{bmatrix} 0 & -1 & -1 \\ +1 & 0 & -1 \\ +1 & +1 & 0 \end{bmatrix}. \quad (2)$$

Generally speaking, our SPF descriptors can be extracted step by step as follows:

1) Given a local patch A, compute $G_{x,i^\circ} \in \{G_{s,i^\circ}, G_{p,i^\circ}\}$, which denote approximate values of Sobel i° and Prewitt i° derivatives, respectively, by

$$G_{x,i^\circ} = S_{i^\circ} \otimes A, \quad (3)$$

where $i \in [0, 45, 90, 135]$ and $\otimes$ denotes the 2-dimensional convolution operation.

2) At each pixel in the local patch, calculate the Sobel or Prewitt gradient magnitudes $G_{s_1}, G_{s_2}, G_{p_1}$, and G_{p_2} as

$$G_{x_i} = \sqrt{G^2_{x,d^\circ} + G^2_{x,(d+90)^\circ}}, \quad (4)$$

where $x \in \{s, p\}$, $i \in \{1, 2\}$, and $d \in \{0, 45\}$. Meanwhile, directions of gradients $\Theta_{s_1}, \Theta_{s_2}, \Theta_{p_1}$, and Θ_{p_2} can also be obtained by

$$\Theta_{x_i} = \arctan\left(G_{x,(d+90)^\circ}, G_{x,d^\circ}\right), \quad (5)$$

3) Quantize each Θ (Θ_{s_1}, Θ_{s_2}, Θ_{p_1}, and Θ_{p_2}) into 16 angles θ_i from $0°$ to $337.5°$, namely, $\theta_i \in \{0°, 22.5°, 45°, ..., 337.5°\}$. Afterwards, divide the local patch into 16 sub-blocks of the same size, then calculate 16 features corresponding to 16 angles for each sub-block as follows

$$f_i = \sum_{\Theta(x,y)=\theta_i} G(x, y), \quad (6)$$

where $i \in 1, 2, ..., 16$, (x, y) denotes the coordinates of each pixel, and f_i denotes the ith feature value corresponding to the ith angle.

4) Combine four groups of features which are f_{s_1}, f_{s_2}, f_{p_1} and f_{p_2} into one feature vector $F = [f_{s_1}, f_{s_2}, f_{p_1}, f_{p_2}]$, and then concatenate all feature vectors in 16 blocks to form the SPF descriptor: $SPF = [F_1, F_2, ..., F_{16}]$, where F_i denotes the feature vector F in ith block.

C. Multi-scale Sparse Representation

In this step, three groups of SPF descriptors are first extracted from local patches of three scales. Then, for each group of SPF descriptors, we generate a single-scale sparse representation (SSR). After normalizing three SSRs, we concatenate them to form our MSR. Each SSR can be calculated step by step as follows:

Codebook Construction: We apply the K-means clustering to generate a codebook for the SPF descriptors in each scale. More specifically, given SPF descriptors in a scale, we first randomly sample a large amount of them to form a codebook training set, then we perform K-means clustering algorithm on the codebook traning set to generate K clustering centers. At last, the K clustering centers are selected as codewords in the codebook.

Encoding: We encode each SPF descriptor x_i into a sparse code c_i based on the LLC scheme [12], which is implemented by solving the following minimization problem:

$$\begin{aligned} &\min_{C} \sum_{i=1}^{N} \|x_i - Tc_i\|^2 + \lambda \|d_i \odot c_i\|, \\ &s.t. 1^\top c_i = 1, \forall i \end{aligned} \quad (7)$$

where $\odot$ denotes the element-wise multiplication, T denotes the codebook, $d_i = \exp\left(dist(x_i, T)/\sigma\right)$, $dist(x_i, T) = [dist(x_i, t_1), ..., dist(x_i, t_K)]^T$, $dist(x_i, t_i)$ denotes the Euclidean distance between x_i and t_j, and σ is a coefficient adjusting the weighting vector d_i. We employ an approximated LLC method proposed in [13] to enhance the computational efficiency. It simply selects the K nearest basis vectors of x_i, and thus only a much smaller linear optimization problem needs to be solved to get the spare codes.

SPM and pooling: The SPM [10] method divides the original image into spatial sub-regions. For each sub-region, the encoded SPF descriptors are pooled together to construct the corresponding pooled sparse features. Afterwards, these pooled sparse features which are extracted from each sub-region are concatenated and normalized into the final image representation. Specifically, given encoded local descriptors $C = [C_1, ..., C_M]$ in a sub-region, a max pooling function [11] is employed to obtain an image-level representation SSR = $[p_1, ..., p_K]$, and the ith component of SSR is $p_i = max\{C_{1i}, ..., C_{Mi}\}$. More implementation details about SPM and max pooling can be found in [10], [11].

III. Experimental Results

The goal of this section is to evaluate the performance of our saliency-based sampling method, SPF descriptor, and MSR for the application of font recognition on a single unknown Chinese character. Our experiments are carried out on our own Chinese font databases (CFDB): CFDB-160 and CFDB-40. We implement $L2$ normalization for each SSR and the number of cluster centers is chosen as $K = 1024$. In the rest of this section, we first introduce the construction process of our database, then perform experiments on CFDB-40 to evaluate our saliency-based sampling method and SPF descriptor. Finally, we compare the performance of our method with other state-of-the-art approaches on CFDB-160.

A. Database Construction

There are two widely used approaches for generating digital document images for font recognition research: scanning and software generation. For convenience, rather than scanning images, all images in our databases were generated by rendering Chinese characters on the computer. Specifically, images in our databases were captured using our own software at the *height* of about 500 pixels and saved as binary images. Then, the size of each image was normalized into 128×128. CFDB-160 contains images of Chinese characters in 160 different kinds of fonts (40 fonts with 4 styles) which are commonly used in Chinese documents. The 4 styles are regular, italic, bold, and bold-italic. An image set of each font in CFDB-160 contains 2000 different Chinese characters randomly selected from GB2312 official character set, and 500 of them are used for training while the rest are utilized for testing. We selected 40 fonts (10 fonts with 4 styles) in CFDB-160 to build CFDB-40, which can be utilized for small-scale experiments. Similarly, an image set of each font in CFDB-40 consists of 500 images, among them 250 images are used for training and other 250 for testing. In particular, in order to increase the difficulty of font recognition, CFDB-40 contains some similar fonts, such as SongTi and BaoSong. Therefore, despite of having fewer types of fonts, CFDB-40 is similar to CFDB-160 in the capability of evaluating font recognition algorithms.

B. Comparison of Sampling Methods

We implemented and compared the following three sampling methods for the proposed multi-scale sparse representation: random contour sampling, Harris corner detector and our saliency-based sampling method. By choosing appropriate parameters, we ensure that approximately 400 local patches can be obtained from each image for the first and third sampling method. Yet, the number of interest points extracted from the second method is purely determined by the Harris corner detector algorithm. Here, we used SPF descriptors and MSR to represent local patches.

TABLE I. Sampling Method comparison

Sampling Method	Recognition Rates
Ramdon contour sampling	98.14%
Harris corner detector	97.06%
Saliency-based sampling	***99.03%***

TABLE II. Local Descriptors comparison

Local Descriptors	Recognition Rates
Pixel descriptors	98.62%
SIFT descriptors	95.94%
SPF	***99.03%***

Table I indicates that our saliency-based sampling method results in the best recognition accuracy, and the Harris corner detector performs worst among these three methods. When the number of local patches increases to hundreds, the local patches extracted by random contour sampling includes the local patches extracted by Harris corner detector, consequently the performance of random contour detector is better than Harris corner detector. Our saliency-based sampling method obtains the best performance since it captures more discriminative local patches than the other two methods.

C. Comparison of Local Descriptors

Besides sampling approaches, we would also like to know how local descriptors affect the performance of our font recognition method. Here, we designed a primitive pixel feature as a baseline method to evaluate the effectiveness of each local descriptor. The pixel-based descriptor uses the value of each pixel in local patches as an element of the feature vector. Accordingly, a 256-dimensional feature vector can be extracted from a local patch with 16×16 pixels. Moreover, we also compared our SPF descriptor with the widely used SIFT descriptor [17]. Note that, when comparing these three descriptors, the same font recognition framework and settings were adopted. Specifically, interest points are extracted by our saliency-based sampling method, and the proposed multi-scale sparse representation is employed for font classification. Table II shows that the proposed SPF descriptor clearly outperforms other methods.

D. Comparison with Existing Methods

Finally, we evaluated and compared the performance of our method with other three approaches on our CFDB-160 database. Details of these four font recognition methods are briefly described as follows:

Gabor [7]: A method that uses multi-channel Gabor filters and the weighted Euclidean distance classifier.

Wavelet [8]: A method that employs wavelet features, Box-Cox transform, linear discriminant analysis process, and

TABLE III. CLASSIFICATION RATES COMPARISON

Algorithms	Single character	Text block
Gabor [7]	21.97%	98.5%
Wavelet [8]	73.48%	94.86%
SRF [5]	28.14%	98.01%
SSR-8	91.27%	94.06%
SSR-16	96.80%	96.68%
SSR-32	95.30%	97.32%
MSR	***97.51%***	***99.36%***

MQDF. For convenience, we ignored the Box-Cox transform and replaced MQDF with quadratic distance function (QDF).

SRF [5]: A method that utilizes SRF features and the weighted Euclidean distance classifier.

Our method: The method proposed in this paper using saliency-based sampling and the SPF local descriptor. Here, we evaluated four versions of our method using three single-scale sparse representations and the multi-scale sparse representation, respectively.

The second column of Table III shows the recognition rates of above-mentioned methods in the application of font recognition on a single Chinese character. SSR-x denotes the single-scale sparse representation with local patch scale of $x \times x$, and MSR denotes the proposed multi-scale s-parse representation which is basically a linear combination of SSR-8, SSR-16 and SSR-32. As reported in [8], Zhu's method achieved 90.28% recognition rate on 28 Chinese fonts. However, Table III indicates that the performance of their method drops sharply when the number of font categories increases to 160. Zhu's [7] and Khosravi's [5] methods perform poorly for the font recognition on a single unknown Chinese character because their methods are more suitable for text blocks. Furthermore, we can see that the method concatenating three SSRs obtains better performance than those using any single SSR. In summary, the proposed method with MSR significantly outperforms the state of the art in font recognition on a single Chinese character.

Our method can also be applied in the font recognition of text blocks. Here, we conducted two types of font recognition experiments for multiple Chinese characters. The first experiment is the same as the one reported in [8], where the font type of each character was recognized first and then recognition results of all characters were fused via a majority voting strategy. They showed that when 20 characters were involved, 100% recognition accuracy can be achieved using their method [8]. Our experimental results demonstrate that 100% recognition accuracy can be obtained by our method when only 5 characters are involved. Our second experiment were carried out on the database containing text blocks instead of individual character images. Specifically, we combined M characters into a text block, where $M = 64$. As a result, the input images contain blocks with 8×8 Chinese characters. For an image set of each font, we generated 2000 text blocks, among which 1000 of them are used for training and the rest for testing. The third column of Table III demonstrates that compared to other approaches, our method (i.e., MSR) also achieves the highest classification accuracy on text blocks.

IV. CONCLUSION

This paper presented a new and powerful method for large-scale font recognition on a single unknown Chinese character. Mainly due to the utilization of a saliency-based sampling approach, an improved Sobel-Prewitt feature and the multi-scale sparse representation, the proposed method obtains excellent results when applying in the font recognition of Chinese characters. Experiments conducted on two large-scale databases demonstrated that our method significantly outperforms the state of the art in content-independent font recognition on a single Chinese character.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.: 61202230 and 61472015), National Hi-Tech Research and Development Program (863 Program) of China (Grant No.: 2014AA015102) and Beijing Natural Science Foundation (Grant No.: 4152022).

REFERENCES

[1] A. Zramdini and R. Ingold, "Optical font recognition using typographical features," *IEEE TPAMI*, vol. 20, no. 8, pp. 877–882, 1998.

[2] M.-C. Jung, Y.-C. Shin, and S. N. Srihari, "Multifont classification using typographical attributes," in *ICDAR 1999*. IEEE, 1999, pp. 353–356.

[3] G. D. Joshi, S. Garg, and J. Sivaswamy, "A generalised framework for script identification," *IJDAR*, vol. 10, no. 2, pp. 55–68, 2007.

[4] B. Bataineh, S. N. H. S. Abdullah, and K. Omar, "A novel statistical feature extraction method for textual images: Optical font recognition," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5470–5477, 2012.

[5] H. Khosravi and E. Kabir, "Farsi font recognition based on sobel–roberts features," *Pattern Recognition Letters*, vol. 31, no. 1, pp. 75–82, 2010.

[6] M. Lutf, X. You, Y.-M. Cheung, and C. Philip Chen, "Arabic font recognition based on diacritics features," *Pattern Recognition*, vol. 47, no. 2, pp. 672–684, 2014.

[7] Y. Zhu, T. Tan, and Y. Wang, "Font recognition based on global texture analysis," *IEEE TPAMI*, vol. 23, no. 10, pp. 1192–1200, 2001.

[8] X. Ding, L. Chen, and T. Wu, "Character independent font recognition on a single chinese character," *IEEE TPAMI*, vol. 29, no. 2, pp. 195–204, 2007.

[9] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *CVPR 2005*, vol. 2, 2005, pp. 524–531.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR 2006*, vol. 2, 2006, pp. 2169–2178.

[11] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR 2009*, 2009, pp. 1794–1801.

[12] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *NIPS 2009*, 2009, pp. 2223–2231.

[13] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR 2010*, 2010, pp. 3360–3367.

[14] W. Pan, Z. Lian, Y. Tang, and J. Xiao, "Skeleton-guided vectorization of chinese calligraphy images," in *MMSP 2014*, 2014, pp. 1–6.

[15] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[16] D. Tao, L. Jin, S. Zhang, Z. Yang, and Y. Wang, "Sparse discriminative information preservation for chinese character font categorization," *Neurocomputing*, vol. 129, pp. 159–167, 2014.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.