CNN-Based Chinese Character Recognition with Skeleton Feature

Wei Tang[1,2,3], Yijun Su[1,2,3], Xiang Li[1,2,3], Daren Zha[3], Weiyu Jiang[3(✉)], Neng Gao[3], and Ji Xiang[3]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China
[3] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{tangwei,suyijun,lixiang9015,zhadaren,jiangweiyu,gaoneng, xiangji}@iie.ac.cn

Abstract. Recently, the convolutional neural networks (CNNs) show the great power in dealing with various image classification tasks. However, in the task of Chinese character recognition, there is a significant problem in CNN-based classifiers: insufficient generalization ability to recognize the Chinese characters with unfamiliar font styles. We call this problem the Style Overfitting. In the process of a human recognizing Chinese characters with various font styles, the internal skeletons of these characters are important indicators. This paper proposes a novel tool named Skeleton Kernel to capture skeleton features of Chinese characters. And we use it to assist CNN-based classifiers to prevent the Style Overfitting problem. Experimental results prove that our method firmly enhances the generalization ability of CNN-based classifiers. And compared to previous works, our method requires a small training set to achieve relatively better performance.

Keywords: Chinese character recognition · Convolutional neural networks · Style Overfitting · Skeleton feature

1 Introduction

Chinese characters have been widely used (modified or extended) in many Asian countries such as China, Japan, Korea, and so on [22]. There are more than tens of thousands of different Chinese characters with variable font styles. Most of them can be well recognized by most people. However, in the field of artificial intelligence, Chinese character automatic recognition is considered as an extremely difficult task due to the very large number of categories, complicated structures, similarity between characters and the variability of font styles [3]. Because of its unique technical challenges and great social needs, during the last five decades there are intensive research in this field and a rapid increase

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11305, pp. 461–472, 2018.
https://doi.org/10.1007/978-3-030-04221-9_41

of successful applications [8,10,15,17]. However, higher recognition performance is continuously needed to improve the existing application and to exploit new applications.

Recently, the convolutional neural networks (CNNs) show their great power in dealing with multifarious image classification tasks [5,6,11,14,16,21]. CNN-based classifiers break the bottleneck of Chinese character recognition and achieve excellent performance even better than human on ICDAR'13 Chinese Character Recognition Competition [1,4,18,20,24]. But there is a significant problem in CNN-based classifiers: insufficient generalization ability to recognize Chinese characters with unfamiliar font styles (e.g. it perform poorly when test on the characters with the font style that the trained model have never seen). We call this problem the *Style Overfitting*. However, most people are able to deal with this problem easily. In the process of a human recognizing Chinese characters with unfamiliar font style, the internal skeletons of these characters are significant indicators. If a Chinese character printed in two or more different font styles, the inherent skeletons of them are usually the same. Figure 1 shows three pairs of same Chinese characters with different font styles that challenge CNN-based classifiers.

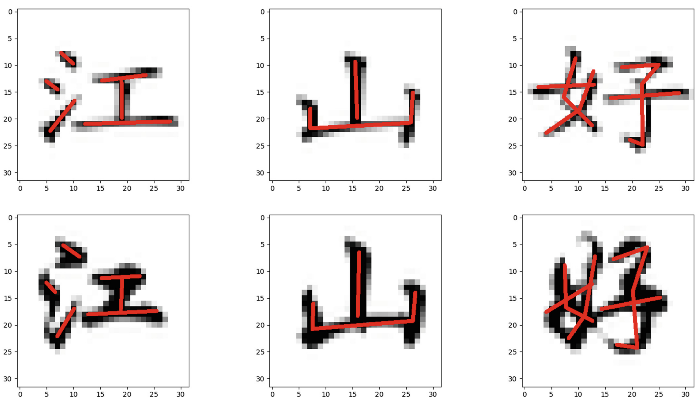

Fig. 1. Three pairs of same Chinese characters with different font styles that challenge CNN-based classifiers. It could be seen that the inherent skeleton (highlighted by the red lines) of the above character and the bottom character are essentially the same. (Color figure online)

In this paper, we propose a novel tool named *Skeleton Kernel* to capture skeleton features of Chinese characters to prevent the Style Overfitting problem. A Skeleton Kernel is designed as a long narrow rectangle window sliding along the horizontal or vertical axis of the input image. It calculates the cumulative distributions of pixel values to capture skeleton features of input images. Then the same pattern of the same Chinese character printed in different fount styles could be easily recognized as the same one. Figure 2 illustrates the same pattern extracted by Skeleton Kernel of the same Chinese character printed in different font styles.

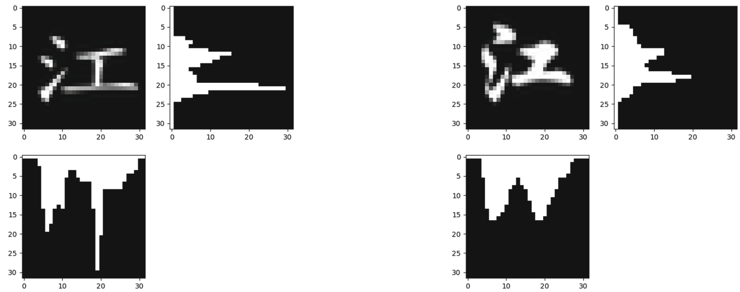

Fig. 2. The right histograms of each character depict the cumulative distributions of pixel values calculated by a long narrow rectangle window sliding along the vertical axis. And the bottom histograms do the same along the cross direction. The same inherent skeletons of each character are represented by the same patterns in these histograms.

Overall, our contributions are as follows:

1. We introduce the Style Overfitting problem of CNN-based classifiers in dealing with Chinese character recognition: insufficient generalization ability to recognize Chinese characters with unfamiliar font styles.
2. We propose a novel tool named Skeleton Kernel to extract skeleton features, which are important indicators to identify Chinese characters. And we use it to assist CNN-based classifiers to prevent aforementioned Style Overfitting problem.
3. We have done a series of experiments to prove that the Style Overfitting problem indeed exists, and our method alleviates this problem by firmly enhancing the generalization performance of CNN-based classifiers.

The rest of this paper is organized as follows. Section 2 summarizes the related works. Section 3 discusses the Style Overfitting problem from the practical view. Section 4 introduces the proposed Skeleton Kernel in details. Section 5 presents the experimental results. Finally in Sect. 6, we conclude our work and discuss the future work.

2 Related Work

Convolution neural networks has greatly promoted the development of image recognition technology. Lecun [7] first introduces the convolutional neural network specifically designed to deal with the variability of 2D shapes. Krizhevsky et al. [6] create a large, deep convolutional neural network (named Alex Net) that was used to win the ILSVRC 2012 (ImageNet Large-Scale Visual Recognition Challenge). Zeiler et al. [21] explain a lot of the intuition behind CNNs and showing how to visualize the filters and weights correctly, and their ZF Net won the ILSVRC 2013. GoogLeNet [16] is a 22 layer CNN and was the winner of ILSVRC 2014, and it is one of the first CNN architectures that really strayed from the general approach of simply stacking convolution and pooling layers on top of each other in a sequential structure. Simonyan et al. [14] created a 19 layer CNN (name VGG Net) that strictly used 3×3 filters with stride and pad of 1, along with 2×2 maxpooling layers with stride 2. VGG Net is one of the most influential CNN because it reinforced the notion that convolutional neural networks have to have a deep network of layers in order for this hierarchical representation of visual data to work.

CNN-based Chinese character recognition has achieved unprecedented success. But all these successful CNN-based Chinese character classifiers are test on the character set with the same font styles as the training set. Wu et al. [18] propose a handwriting Chinese character recognition method based on relaxation convolutional neural network, and took the 1st place in ICDAR'13 Chinese Handwriting Character Recognition Competition [20]. Meier et al. [4] create a multi-column deep neural network achieving first human-competitive performance on the famous MNIST handwritten digit recognition task. And this CNN-based classifier classifies the 3755 classes of handwritten Chinese characters in ICDAR'13 with almost human performance. Zhong et al. [24] design a streamlined version of GoogLeNet for handwritten Chinese character recognition outperforming previous best result with significant gap. Chen et al. [1] propose a CNN-based character recognition framework employ random distortion [13] and multi-model voting [12]. This classifier performed even better than human on MNIST and ICDAR'13.

Few works are devoted to reducing overfitting in CNN-based Chinese character recognition. Xu et al. [19] propose a artificial neural network architecture called cooperative block neural networks to address the variation in the shape of Chinese characters by considering only three different fonts. Lv [9] successfully applied the stochastic diagonal Levenberg-Marquardt method to a convolutional neural network to recognize a small set of multi-font characters used in Baidu CAPTCHA, which consists of the Arabic numerals and English letters without the Chinese characters. Zhong et al. [23] propose a CNN-based multi-font Chinese character recognizer using multi-pooling and data augmentation achieving acceptable result. But they use 240 fonts for training and 40 fonts for test. The size of the training set is more than 500% of the test set, and the huge training set containing a great deal of font styles especially reducing the overfitting. Different from training on a predetermined large training set, we use a relatively small training set with its size being 10% of the test set. And we gradually increase the number of font styles contained in the training set to dynamically verify the effectiveness of our method.

3 Style Overfitting

The font styles of Chinese characters are multifarious. In practice, to train a CNN-based Chinese character classifier, it is hard to provide a perfect training set with all font styles. And it is absolutely expensive to use such massive amount of images to train a deep neural network. In view of this situation, a well trained classifier should obtain excellent generalization ability to recognize characters with unfamiliar font styles.

However, CNN-based classifiers could not satisfy this demand. We have done a lot of experiments showing that the CNN-based classifiers perform poorly when it test on Chinese character sets with unfamiliar font styles. Even adding more amount of convolution layers, this problem still exists. We call this problem the Style Overfitting. It is a significant problem seriously restricting the generalization ability of CNN-based Chinese character classifiers.

We believe that the main cause of this problem is that CNNs excessively focus on the texture features that strongly indicate the font styles, and relatively ignore the inherent skeleton features. Inherent skeletons are important indicators to classify Chinese characters. One Chinese character could be printed in diverse font styles, and the inherent skeleton of each Chinese character are commonly fixed. Therefore, to solve Style Overfitting problem, the influence of skeleton features on CNN-based classifiers must be strengthened.

4 Method

To prevent aforementioned Style Overfitting problem, we propose a novel tool named Skeleton Kernel to extract skeleton features of Chinese characters. And we use Skeleton Kernel to enhance the generalization performance of CNN-based classifiers.

4.1 Skeleton Kernel

A Skeleton Kernel is a window with a specific shape sliding along a specific path (like the Convolution Kernel [6]). Considering the deformation and scaling of strokes in Chinese characters, we design the Skeleton Kernel as a long narrow rectangle window sliding along the horizontal or vertical axis of the input image. There are 3 main differences between Skeleton Kernel and Convolution Kernel: (1) the Skeleton Kernel appear long and narrow, while Convolution Kernel is always a relatively small square; (2) the weights in Skeleton Kernel are predetermined, which in Convolution Kernel are learning from training set; (3) the output of the Skeleton Kernel is a vector, and the Convolution Kernel produce a matrix. Figure 3 briefly illustrates how Skeleton Kernels work.

The Skeleton Kernel has two important hyperparameters: size and stride. It should be noted that the excessively thin or fat window is bad for capture skeleton features. It is because a excessively thin window has insufficient resilience to tolerate the deformation and scaling of a stroke (the main difference between diverse font styles), and a excessively fat window could be confused by too many strokes in it.

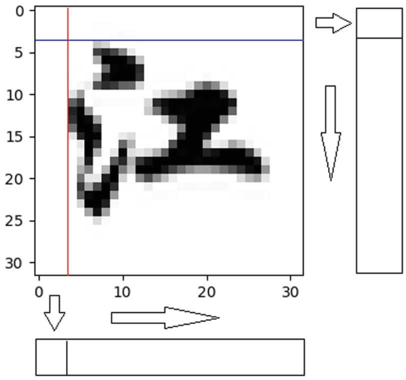

Fig. 3. A pair of Skeleton Kernels slide along the coordinate axis of the input image and output two skeleton feature vectors. The right and bottom vectors are separately produced by the blue and red window. These two vectors contain the skeleton information by calculate the cumulative distributions of pixel values in the input image. (Color figure online)

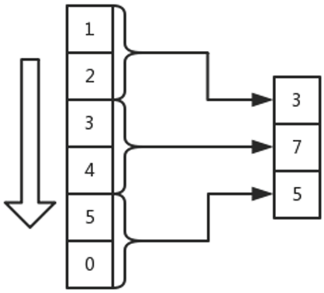

Fig. 4. This figure briefly show how Chunk-Sum Pooling works. Specifically, in this figure, the hyperparameters (size, stride) of the sliding window are (2, 2). The input vector is cut into many chunks by this sliding window. We add 0 to the end of the input vector if the sliding window exceeds the bound of it. CSP calculates the sum of all the elements in each chunk and then produce a new vector.

4.2 Chunk-Sum Pooling

To avoid the aforementioned problem, we could designed more than one Skeleton Kernel with different sizes and strides. More effectively, we use only one Skeleton Kernel with its size to be $1 \times$ the height of input image (or the width of input image $\times 1$) and stride to be 1. Then, we employ Chunk-Sum Pooling (noted as CSP) to process the output vector to achieve the same effect as using a lot of Skeleton Kernels. CSP is a variant of Chunk-Max Pooling [2]. It cuts a vector into a lot of chunks and calculates the sum of all the elements in each chunk to produce a new vector, just like a sliding window with certain hyperparameters: size and stride. Figure 4 shows how CSP works.

4.3 CNN-Based Framework with Skeleton Kernel

In order to prevent the Style Overfitting problem of CNN-based classifiers, we propose a new CNN-based framework employing Skeleton Kernel. The main difference between our new framework and the original framework is that the new framework contain a extra bypass consisting of several Skeleton Kernels and parallel CSP modules. A pair of Skeleton Kernels are used to extract skeleton features, and these CSP modules are used to process the feature vectors. The diagram of the new framework is illustrated in Fig. 5.

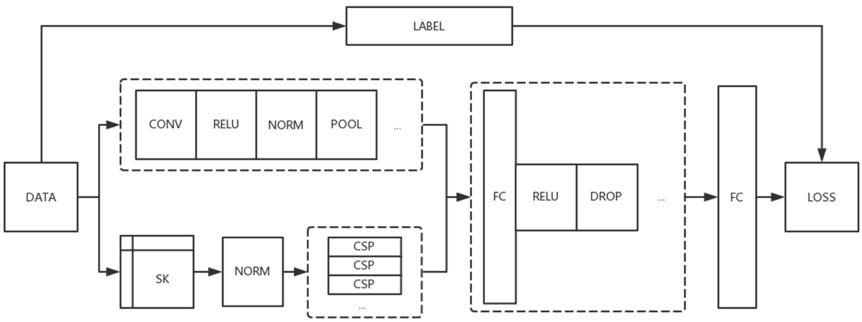

Fig. 5. Diagram of the new CNN-based Chinese character recognition framework employing Skeleton Kernel. The CONV–RELU–NORM–POOL and FC–RELU–DROP modules constitute the main structures of popular convolutional networks (e.g. Alex nets [6]). Skeleton Kernels (SK) in bypass produced several vectors that represent skeleton features of input image. There are several parallel CSP modules with different sizes and strides to process these skeleton features.

Using Skeleton Kernel does not affect the convergence and complexity of the original convolutional neural network. It is because our method is equivalent to the addition of extra feature extraction kernels, and each of these extra kernels obtain fixed weights and biases.

5 Experiments

We have done a series of experiments to evaluate the generalization ability of CNN-based framework and our new framework to recognize Chinese characters with unfamiliar font styles. Results prove that the Style Overfitting problem does exist, and our method indeed alleviate this problem by firmly enhancing the generalization performance of CNN-based classifiers.

5.1 Data

The data are extracted from True Type font (TTF) files. TTF is a font file format jointly launched by Apple and Microsoft. With the popularity of Microsoft

Windows operating systems, it has become the most commonly used format of font file. We extract 130 candidate sets from 130 TTF files with widely varying font styles. Each of them contains 3755 frequently-used Chinese characters (level-l set of GB2312-80). Each Chinese character is presented by a 32 × 32 PNG image. Figure 6 shows the example of Chinese character 'JIANG' in 130 different fonts.

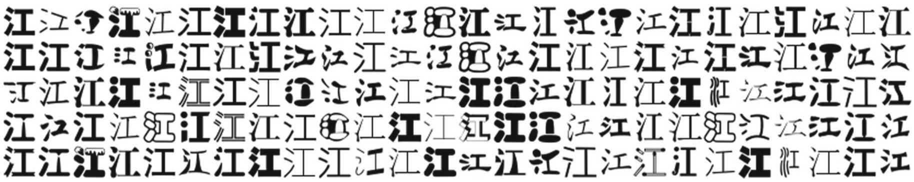

Fig. 6. Example of Chinese character 'JIANG' in 130 different fonts.

5.2 Existence of Style Overfitting

We evaluate the generalization ability of the popular deep convolutional neural network VGG [14] to recognize Chinese characters with unfamiliar font styles. We chose two typically VGG nets: VGG-11 and VGG-19. VGG-11 is a 11 layers CNN consisting of 8 convolution layers and 3 full connection layers. VGG-19 is a 19 layers CNN consisting of 16 convolution layers and 3 full connection layers. These two CNNs are used to assess the impact of network depth on the Style Overfitting problem.

Tables 1 and 2 show the experimental results of VGG-11 and VGG-19. In this experiment, there are 10 character sets randomly selected from the 130 candidate sets. The 10 randomly selected character sets are tagged with the ID 1 to 10. We use 1 of the 10 selected sets for training and other 9 sets for test. The first rows of these tables list the IDs of the training sets (noted as tra-ID, e.g. tra-1); the first columns of these tables list the IDs of the test sets (noted as test-ID, e.g. test-1).

It could be seen that the train accuracy are very high (highlighted with bold font), but the test accuracy on other character sets are extremely low. It prove that CNN-based classifiers lack the generalization ability to recognize Chinese characters with unfamiliar font styles. Even adding more amount of convolution layers, the Style Overfitting problem still exists.

5.3 Performance Comparison

In order to prove that our method indeed alleviate the Style Overfitting problem and to compare the generalization performance between our method and CNN-based classifier using data augmentation and multi-pooling, we set up 3 controlled groups for the experiments: (1) VGG-19 [14]; (2) VGG-19 using data

Table 1. Results of VGG-11 on 10 randomly selected Chinese character sets.

IDs	tra-1	tra-2	tra-3	tra-4	tra-5	tra-6	tra-7	tra-8	tra-9	tra-10
test-1	**0.9827**	0.0749	0.1209	0.0817	0.1368	0.1118	0.1172	0.1007	0.0945	0.1119
test-2	0.0818	**0.9752**	0.1356	0.1393	0.0970	0.1241	0.1265	0.0821	0.1127	0.0868
test-3	0.1100	0.1268	**0.9776**	0.1460	0.1030	0.1369	0.0937	0.1358	0.1239	0.1063
test-4	0.0852	0.1286	0.1449	**0.9550**	0.1281	0.0919	0.1403	0.1382	0.1155	0.1368
test-5	0.1427	0.0842	0.1017	0.1310	**0.9830**	0.1334	0.1076	0.0804	0.0943	0.1446
test-6	0.1233	0.1358	0.1414	0.0823	0.1414	**0.9744**	0.1419	0.1142	0.1334	0.0999
test-7	0.1305	0.1313	0.0860	0.1443	0.1193	0.1443	**0.9795**	0.0964	0.1382	0.1132
test-8	0.1079	0.0951	0.1465	0.1385	0.0820	0.1198	0.1007	**0.9664**	0.1145	0.0765
test-9	0.0897	0.1244	0.1356	0.1025	0.0908	0.1206	0.1278	0.1089	**0.9664**	0.1395
test-10	0.1119	0.0815	0.0948	0.1387	0.1387	0.1039	0.1196	0.0810	0.1342	**0.9704**

Table 2. Results of VGG-19 on 10 randomly selected Chinese character sets.

IDs	tra-1	tra-2	tra-3	tra-4	tra-5	tra-6	tra-7	tra-8	tra-9	tra-10
test-1	**0.9947**	0.0895	0.1156	0.0905	0.1491	0.1366	0.1412	0.1167	0.0998	0.1178
test-2	0.0824	**0.9861**	0.1396	0.1390	0.0919	0.1411	0.1409	0.1012	0.1356	0.0874
test-3	0.1262	0.1433	**0.9859**	0.1566	0.1076	0.1505	0.0919	0.1532	0.1460	0.1063
test-4	0.0894	0.1497	0.1588	**0.9606**	0.1393	0.0924	0.1512	0.1505	0.1126	0.152
test-5	0.1493	0.1079	0.1089	0.1401	**0.9902**	0.1502	0.1323	0.0897	0.1004	0.1451
test-6	0.1233	0.1350	0.1494	0.1044	0.1446	**0.9869**	0.1552	0.1297	0.1318	0.1154
test-7	0.1289	0.1382	0.1049	0.1462	0.1169	0.1518	**0.9870**	0.1127	0.1342	0.1316
test-8	0.1119	0.0888	0.1441	0.1449	0.0932	0.1243	0.1057	**0.9752**	0.1148	0.0927
test-9	0.1009	0.1196	0.1298	0.1278	0.1010	0.1411	0.1446	0.1262	**0.9752**	0.1472
test-10	0.1247	0.0983	0.1132	0.1488	0.1523	0.1114	0.1255	0.0840	0.1504	**0.9832**

augmentation and multi-pooling (noted as VGG-19 with MP) [23]; (3) VGG-19 with Skeleton Kernel (our method, noted as VGG-19 with SK).

According to experience, the integrity of training data will affect the generalization ability of the model. Therefore, the controlled variable is the number of font styles in training set. We first randomly choose 1 candidate set for training and randomly choose other 10 candidate sets for test, and then we add the number of font styles for training until there are 9 font styles for training and other 90 for test. Each experiment is carried out 10 times by randomly change the training set and test set, and the size of the training set is constantly 10% of the test set. Finally, we calculate the mean accuracy. Figure 7 shows the results of these experiments.

In the experiments of VGG nets with Skeleton Kernel, there is a pair of Skeleton Kernels to extract skeleton features of the input image. The size and stride of the window sliding along the horizontal axis of the input image are 32×1 and 1; the size and stride of the window sliding along the vertical axis of the input image are 1×32 and 1. And the weights and bias of all Skeleton Kernels are

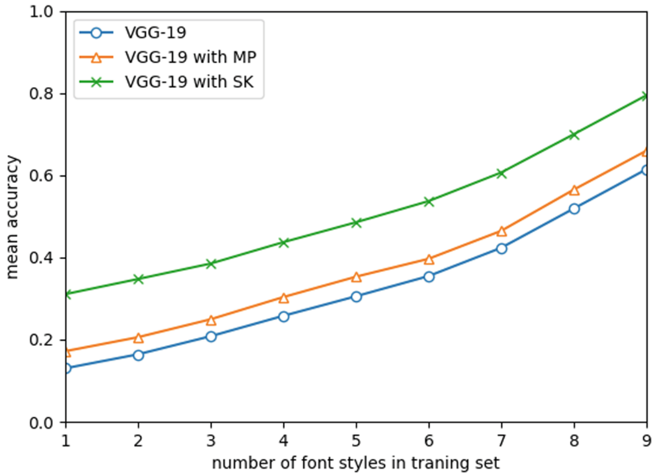

Fig. 7. Experimental results of VGG-19, VGG-19 with MP, and VGG-19 with SK.

predetermined to be constant 1 and 0. Each feature vector produce by Skeleton Kernels is normalized by the norm of it. We employ 5 CSPs to process these skeleton feature vectors. The hyperparameters (size, stride) of these CSPs are (1, 1), (2, 1), (2, 2), (4, 2) and (4, 4).

The experimental results show that our method remarkably enhances the generalization ability of CNN-based classifier to recognize Chinese characters with unfamiliar font styles. And compared to previous works, our method only requires a small training set to achieve relatively better results on widely varying test sets. It prove that our method is effective for alleviating the Style Overfitting problem.

6 Conclusion and Future Work

CNN-based classifiers lack generalization ability to recognize Chinese characters with unfamiliar font styles (Style Overfitting problem). To solve this problem, we propose Skeleton Kernel to extract skeleton features of Chinese characters to assists CNN-based classifiers. Experimental results prove that our method alleviate the Style Overfitting problem by firmly enhancing the generalization performance of CNN-based Chinese character recognizers.

This paper design the Skeleton Kernel as a long narrow window with fixed weights and bias sliding along the axis of input images. In the future, it could be designed a train-able Skeleton Kernel with more kinds of shape and sliding path to achieve better performance.

Acknowledgment. We thank Lei Wang, Yujin zhou, Zeyi Liu, Jiahui Sheng and Yuanye He for their useful advice. This work is partially supported by National Key Research and Development Program of China.

References

1. Chen, L., Wang, S., Fan, W., Sun, J.: Beyond human recognition: a CNN-based framework for handwritten character recognition. In: IAPR Asian Conference on Pattern Recognition, pp. 695–699 (2015)
2. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: Event extraction via dynamic multi-pooling convolutional neural networks. In: The Meeting of the Association for Computational Linguistics (2015)
3. Dai, R., Liu, C., Xiao, B.: Chinese character recognition: history, status and prospects. Front. Comput. Sci. China **1**(2), 126–136 (2007)
4. Dan, C., Meier, U.: Multi-column deep neural networks for offline handwritten Chinese character classification. In: International Joint Conference on Neural Networks, pp. 1–6 (2015)
5. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Computer Science, pp. 580–587 (2013)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105. Curran Associates Inc, Red Hook (2012)
7. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
8. Long, T., Jin, L.: Building compact MQDF classifier for large character set recognition by subspace distribution sharing. Pattern Recogn. **41**(9), 2916–2925 (2008)
9. Lv, G.: Recognition of multi-fontstyle characters based on convolutional neural network. In: Fourth International Symposium on Computational Intelligence and Design, pp. 223–225 (2011)
10. Mori, S., Yamamoto, K., Yasuda, M.: Research on machine recognition of hand-printed characters. IEEE Trans. Pattern Anal. Mach. Intell. **6**(4), 386–405 (1984)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: International Conference on Neural Information Processing Systems, pp. 91–99 (2015)
12. Schmidhuber, J., Meier, U., Ciresan, D.: Multi-column deep neural networks for image classification. In: Computer Vision and Pattern Recognition, pp. 3642–3649 (2012)
13. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: International Conference on Document Analysis and Recognition, p. 958 (2003)
14. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Computer Science (2014)
15. Stallings, W.: Approaches to Chinese character recognition. Pattern Recogn. **8**(2), 87–98 (1976)
16. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
17. Umeda, M.: Advances in recognition methods for handwritten Kanji characters. IEICE Trans. Inf. Syst **79**(5), 401–410 (1996)
18. Wu, C., Fan, W., He, Y., Sun, J., Naoi, S.: Handwritten character recognition by alternately trained relaxation convolutional neural network. In: International Conference on Frontiers in Handwriting Recognition, pp. 291–296 (2014)

19. Xu, N., Ding, X.: Printed Chinese character recognition via the cooperative block neural networks. In: IEEE International Symposium on Industrial Electronics, vol. 1, pp. 231–235 (1992)
20. Yin, F., Wang, Q.F., Zhang, X.Y., Liu, C.L.: ICDAR 2013 Chinese handwriting recognition competition (ICDAR), pp. 1464–1469 (2013)
21. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
22. Zhang, X.Y., Yin, F., Zhang, Y.M., Liu, C.L., Bengio, Y.: Drawing and recognizing Chinese characters with recurrent neural network. IEEE Trans. Pattern Anal. Mach. Intell. (2016)
23. Zhong, Z., Jin, L., Feng, Z.: Multi-font printed Chinese character recognition using multi-pooling convolutional neural network. In: International Conference on Document Analysis and Recognition, pp. 96–100 (2015)
24. Zhong, Z., Jin, L., Xie, Z.: High performance offline handwritten Chinese character recognition using GoogleNet and directional feature maps. In: International Conference on Document Analysis and Recognition, pp. 846–850 (2015)