Character Recognition via a Compact Convolutional Neural Network

Haifeng Zhao*, Yong Hu* and Jinxia Zhang†
*School of Software Engineering, Jinling Institute of Technology,
99 Hongjing Avenue, Nanjing 211169, China
Email: {zhf,huyong}@jit.edu.cn
†Key Laboratory of Measurement and Control of CSE, Ministry of Education,
School of Automation, Southeast University, Nanjing 211189, China
Email: jinxiazhang@seu.edu.cn

***Abstract*—Optical Character Recognition (OCR) in the scanned documents has been a well-studied problem in the past. However, when these characters come from the natural scenes, it becomes a much more challenging problem, as there exist many difficulties in these images, e.g., illumination variance, cluttered backgrounds, geometry distortion. In this paper, we propose to use a deep learning method that based on the convolutional neural networks to recognize this kind of characters and words in the scene images. Based on the original VGG-Net, we focus on how to make a compact architecture on this net, and get both the character and word recognition results under the same framework. We conducted several experiments on the benchmark datasets of the natural scene images. The experiments has shown that our method can achieve the state-of-art performance and at the same time has a more compact representation.**

I. Introduction

The characters were invented by the ancient human thousands of years ago to represent symbols and objects in daily life. They have been used as the base of the language to communicate between people. Different communities of people have created different characters and formed different languages, so that there are many kinds of languages and characters in the world. As characters are the base of words, phrases, and sentences, it is important to recognize them to get a more understanding of the semantic information. Characters appear in various kinds of object surfaces, e.g., documents, advertisement boards, road signs and product boxes. All these characters help people to communicate with each other well. Figure 1 shows some examples of characters in the natural scenes[1]. Note that, there are different characters in different languages and font styles.

Fig. 1. Some examples of the characters in the natural scenes, including road signs, product boxes, and advertisement boards, etc.

Although the Optical Character Recognition (OCR) problem has been intensively studied in recent years, the character recognition in natural scenes is still challenging because of several reasons. Firstly, as the smart phones are becoming more and more popular, the character images are usually captured by cameras, which can introduce geometry distortion, motion blur, etc. Also, the light conditions can affect the quality of the produced images. Secondly, the background of the images are much more complicated or cluttered compared to the well scanned document images. Thirdly, there exist different font styles, colors, aspects of text lines in natural scene images. These characteristics make the scene character reading a more difficult and different problem compared OCR. In recent years, as many computer vision tasks involve reading text in natural images, more and more researchers focus on solving this problem, making it a hot topic in the computer vision community.

To solve these problems, many researchers have proposed different kinds of methods. In 2003, the ICDAR 2003 [1] held a competition on text reading in the wild which gained a lot of attention to address this problem. After that, the succeeding ICDARs also held contest on this topic [2]–[4], which advanced the progress on the scene text reading a lot. Sheshadri *et al.* [5] proposed to use an exemplar based support vector machine (SVM) based multi-layered approach to classify the Indic script language Kannada characters making use of the characteristics of the exemplars in the dataset. Fraz *et al.* [6] presented to involve color information for text region detection, which segments the text very well. The bilateral regression combined with the off-the-shelf feature extraction and classification for the recognition of characters in the presence of noise.

The existing approaches in the above have used the specifically designed features to represent characters in the scenes. They rely on the discriminative features and classifiers to recognize characters. They also involve steps of preprocessing and post-processing to finalize the whole procedure, e.g., choosing the best word from the pool of the character candidates. These methods which contain complex processing

[1] These images are downloaded from Internet by search engine.

978-1-5386-2839-3/17/$31.00 ©2017 IEEE

procedures are less suitable to generalize to more real world applications. Also, there are much room to improve the performance of these methods.

To address these problems, many researchers [7]–[10] employed the deep learning [11] method for the scene character recognition. Deep learning is a promising approach which learns the features and classifiers in a multiple layered manner [12]. The method has also been successfully used in the image classification [12], [13], speech recognition [14], machine translation [15], and face recogntion [16], etc. The main difference between conventional neural networks is that it introduces a more deeper network architecture, which can capture more discriminative information to infer the optimal solution. There exist two types of deep neural networks, the convolutional neural network (CNN) [17], and the recurrent neural network (RNN) [18].

Wang *et al.* [7] brought into the generic object recognition methods for character recognition, and proposed an end-to-end approach using the Bayesian inference framework to improve the accuracy. Wang *et al.* [19] employed CNN as the base character detector and recognizer to recognize the whole words in the street view images. They built an end-to-end framework. Jaderberg *et al.* [8] uses a deep learning method to the word recognition with a dictionary. After that, the same authors [9] proposed a new method which combined the character recognition and word recognition together to achieve better classification results. Gupta *et al.* [20] provides an elegant framework on the Jaderberg's method [10] for the text recognition.

In this paper, we propose to use the deep learning method on the scene character and word recognition. Based on the idea of VGG-Net [21], we analyze the original architecture of the network, and make use of the initialization and data augmentation strategy to get a more compact network. Different from our previous work [37], which uses a simplified version of VGG to recognize characters by tuning parameters, we focus more on the input and output of the network. In this way, we are able to recognize the characters and words in the same framework.

The contributions of this paper are as follows. Firstly, we exploit the initialization potential of the network by initializing using different strategies. Secondly, the use of data augmentation provides more data to fed into the network, which makes the model more discriminative. Thirdly, by designing a reasonable size of input and output, we can get the character and word recognition in the same network framework. We conduct experiments on three benchmark datasets in the literature, and show the effectiveness of our proposed method.

The rest of paper is organized as follows. In Section II, we review the related work on the scene character recognition. In Section III, we introduce our method, and the derivation of it in details. Experiments are conducted in Section IV, and the experimental results are shown on benchmark datasets. We conclude in Section V.

II. Related Work

In this section, we review the work that on the scene character reading. In general, there are two steps to recognize characters in the scenes. The first is the character detection, which localizes the position of characters and gets the bounding boxes or regions of all the character lines. The other is the character recognition, which recognize characters based on the regions detected in the previous step. An end-to-end framework involves both the detection and recognition steps so that one can directly input the images and get the final result of the all the characters, which are actually words and text lines.

A. Character Detection

Zhang *et al* [22] proposed a novel approach for text detection. They use both local and global cues for localizing text lines in a coarse-to-fine manner. Full CNN is trained to predict the salient map of text regions, and then text line hypotheses are estimated by combining the salient map and character components. Another Full convolutional network is trained for predicting centroid of each character. This framework can deal with multiple orientations, languages, and different fonts of characters.

Liao *et al* [23] proposed an end-to-end trainable fast scene text detector, named TextBoxes, which detects scene text with both high accuracy and efficiency in a single network. The detection uses a standard non-maximum suppression, and very fast.

Shi *et al* [24] introduced the Segment Linking(SegLink) method to detect text in the natural images. The segment is represented by the bounding box of the partial character lines, and the link connects them together. The authors used the multi-scales CNN to detect all the segments and links in the images, and combined them together to get the final detection results. This method can not only deal with English characters, but also the Chinese characters.

Yao *et al* [25] proposed to use a unified framework for both detection and recognition. They can deal with the texts not only on the horizontal lines but also the varying orientations. Meanwhile, the authors used the dictionary search method to correct the recognition errors.

Inspired by the Canny detector, Cho *et al.* [26] brought the double threshold and hysteresis tracking into the text detection, and proposed the Canny Text Detector. It makes use of the edges and structural information of the text lines, and can be used in the scene character detection.

B. Character Recognition

Bai *et al* [27] proposed a multi-scale representation of the characters, called strokelets, which can capture the sub structural information of characters. This representation can be automatically learned from data, and insensitive to interferences. Based on the strokelets, the recognition of characters can achieve the state-of-art results on the standard benchmark datasets.

He *et al* [28] developed a Deep text recurrent network (DTRN) for text reading. By viewing it as a sequence labelling problem, they firstly use the CNN to generate an ordered high-level sequence from the whole word image. Then, the LSTM [11] is used to recognize the CNN sequences. Highly ambiguous words can be recognized using this method, and it is also robust to the image distortions. Additionally, the approach is independent of the word lexicons, so that it can be used to predict unseen words in natural images.

As the CNN is widely used in the character recognition, as the RNN is used for sequence data inference. Shi *et al.* [29] proposed a method called CRNN which combines the CNN and RNN together to achieve a better result on recognition. They used the CNN as the feature extractor and bidirectional LSTM [11] and Connectionist Temporal Classification (CTC) [30] as the sequence recognizer. In this way, the CRNN can predict words and character sequences with arbitrary length. This makes the method more applicable to the real world applications.

As the words and characters are basically parts of language. It is useful to employ the language models in the recognition process. Poznanski and Wolf [31] proposed a CNN based method making use of the n-gram model. They estimated the n-gram frequency profiles in the handwritten words in a large dictionary. In a single network, the authors get success of the several benchmark datasets. The method is also applied in printed words, e.g., the natural scene images.

III. The Proposed Method

In this section, we will introduce our proposed method in detail. Before that, we first give a brief review of the VGG-Net [21], which is the base of our work. Then we describe our deep model to show how an better solution to robust character recognition can be derived.

A. Brief Review of the VGG-Net

The VGG-Net [21] was introduced by the VGG team for the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [13] in 2014. The main idea of the VGG-Net is in that it makes use of very small convolution filters in each of the layers to achieve a very deep network architecture, e.g., VGG-19, which contains 19 weight layers. With this architecture, the performance is significantly improved.

The main components of the VGG-Net are a stack of convolutional layers followed by three Fully-Conntected (FC) layers. In each of the convolutional layer, the receptive field of filters is 3×3 with convolution stride fixed to one. The small size of filters makes feasible to extend the depth of the network. Meanwhile, a stack of the 3×3 convolutional layers can have same effect on larger receptive field layers, but with more discriminative. In the VGG-Net, two or three convolutional layers are stack together, with a single 2×2 max pooling layer. The width of the convolutional layers is starting from 64, and increasing by 2 until 512. The activation function in each convolutional layer is ReLU [12] non-linearity. For FC layers, the first two have 4096 channels and the last one has 1000 channels corresponding to the 1000 classes in ImageNet. The final output layer is softmax.

TABLE I
The comparison of VGG-Net and our model.

Layer	VGG-Net [21]	Our Model
Input	224×224 RGB image	32×32 gray image
Convolution	conv3-64 conv3-64	conv3-64 -
Pooling	2×2 maxpool	2×2 maxpool
Convolution	conv3-128 conv3-128	conv3-128 conv3-128
Pooling	2×2 maxpool	2×2 maxpool
Convolution	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 -
Pooling	2×2 maxpool	2×2 maxpool
Convolution	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 -
Convolution	conv3-512 conv3-512 conv3-512	- - -
Pooling	2×2 maxpool	2×2 maxpool
FC	FC-4096 FC-4096 FC-1000	FC-2048 FC-2048 FC-62
Output	softmax	softmax

B. Our Deep Model

Our deep network model is based on the VGG-Net. Similar to the VGG-Net, our network also has several successive small size convolutional layers (conv) and FC layers. However, there are three differences in the network settings. The first one is the network input. We use 32×32 gray images as the input. As characters usually are much smaller than objects in the scenes, using small images can get the tradeoff between data size and training time. In the second, we use less convolutional layers than the VGG-Net. The reduction of layers is based on the empirical analysis on the dataset. Here, we conduct experiments on the validation set of Chars74K *Hnd* [32]. Each time, we reduce a convolutional layer and re-run the test on validation set. If the test result is almost the same as the original network, we remove this layer. After a number of attempts, we get our final network architecture. Then, throughout the paper, the architecture of the network is fixed.

Table I shows the comparison of VGG-Net and our network architecture. Note that, in our model, we have eliminated one conv-256, and four conv-512 layers. For FC layers, we reduce the channels from 4096 to 2048. In following out this line, we successfully reduce the depth of the whole network, and makes the training faster without decreasing the overall performance.

Other configurations of our network are as follows. For the input layer, all images are converted into gray images. Then, we subtract the mean of the training set. We use the stochastic gradient descent (SGD) to carry out the optimization process. The momentum is set to 0.9. The learning rate is set to 0.001, and keeps the same in the whole training process. We also employ the batch normalization and dropout as regularization. The dropout ratio is set to 0.5. As the channel number of the third FC layer has to set to the number of classes in the dataset. This value is set to 62 for character recognition, and set to 657 for the script recognition.

For word recognition, we use the image size of 32×100 for the input. The output are lexicon labels. As each image only contains a single word, the lexicon can be used as labels. In this way, we get a classification problem where the data is word images, and the labels are word content, i.e., the lexicon. We use the above mentioned deep network to do training and test.

The initialization of the network plays an important role for the resulting performance. To this end, we have used the He method [33]. As most of CNNs are initialized using random weights drawn from Gaussian distributions. Under this setting, it is difficult to a fast convergence for very deep networks. He method employed a different model, which takes the ReLU activation function and nonlinearities into account. In this way, this method can make the deep model to converge. We also use the Xavier [34] initialization method. In the experiments, we use both methods. The reported results are the one using He's method, as the performance of the He's method is better than Xavier's.

As the deep neural network is a data hungry model, we apply some data augmentation to manually increase the size of training set. This includes some random transformations on the original input images. The augmentation step is as follows. For each image in the training set, we first to shift to the left or right in random pixels. After that, we zoom out or zoom in the result image. And then apply rotation and shear transformation. Finally, we invert the color of the image in some of the pixels randomly. The resulting image is a combination of the above transformations. We do data augmentation several times so that we can get more data for the purpose of training.

We implemented our deep model using the Caffe [35] framework. In the training, we first apply data augmentation for all the images in the training set, and then resize all images into the same size. The test images are also resized. We convert all the RGB images into gray images. All image data are converted into LMDB format and randomly shuffled to feed into the Caffe framework.

IV. Experiments

In this section, we perform experiments on three benchmark datasets: Chars74K [32], ICDAR2003 [1], and Street View Text (SVT) [7], [36]. We firstly conduct the English alphanumeric character recognition on the Chars74K and ICDAR2003 dataset. Then, experiments on a script language Kannada dataset are conducted to show the generalization of the proposed method. Finally, We do the experiments on character sequence (word) recognition on ICDAR2003 and SVT datasets. We also compare our method with others in the literature.

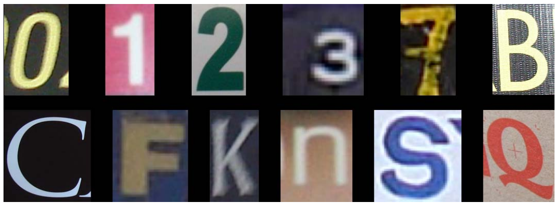

Fig. 2. Some examples of the Chars74K English dataset

TABLE II
The comparison on the recognition accuracy (in Percent) for the Chars74K-15 and ICDAR-CH datasets.

Method	Chars74K-15	ICDAR-CH
Our Method	**70.26**	**81.05**
Simplified VGG [37]	69.99	80.14
HOG+ESVM+AFF [5]	69.66	70.53
Global HOG+SVM [4]	62	76
HOG+NN [36]	57.5	51.5
NATIVE+FERNS [7]	54	64
MKL [32]	55.26	-

A. Character Recognition

The Chars74K English dataset [32] [2] contains 0-9, A-Z, and a-z alphanumeric English characters, totally 62 classes. The whole dataset is divided into three subsets, namely *Img*, *Hnd* and *Fnt*, to represent images collected from the street scenes, hand written characters and computer fonts. The number of characters in each subset is 7705, 3410 and 62992 respectively. The characters from natural scenes are of excessive occlusion, low resolution and noise, making it a very challenging dataset. Following [32], we use Chars74K-15 from *Img* subset as our benchmark dataset. That is, we use 15 images in each class for training and the rest for testing purpose. Figure 2 shows some examples of the Chars74K English dataset.

The ICDAR2003 dataset [1] is also comprised of three subsets for the purpose of robust text locating and recognition. The dataset is a challenging benchmark for the scene text recognition problem. Following [5], we use the ICDAR-CH for character recognition in the experiments. This dataset is one of the subset of ICDAR2003 for robust character recognition. It contains 6185 characters for training and 5430 characters for test.

In Table II are the recognition accuracy for the Chars74K-15 and ICDAR-CH datasets, which are compared with other methods in the literature. We also present our previous work [37] for comparison. The bold fonts indicate the best results. On both of the sets, our method outperforms all the other methods. As ICDAR-CH has more samples than Chars74K-15, this implies that more data can make our model to get better results.

B. Script Recognition

In the Chars74K dataset, there is another component, called Kannada dataset. The Kannada is an Indic script language,

[2] http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

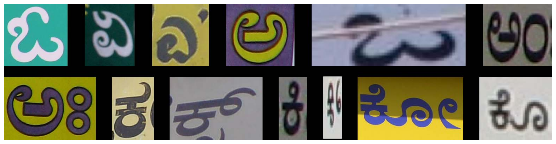

Fig. 3. Some examples of the Chars74K Kannada dataset

TABLE III
THE COMPARISON ON THE RECOGNITION ACCURACY (IN PERCENT) FOR THE CHARS74K KANNADA DATASET.

Method	Test on *Hnd*	Test on *Img*
Our Method	**61.03**	2.35
Simplified VGG [37]	60.99	2.37
HOG+ESVM+AFF [5]	54.13	1.76
SC [32]	29.88	**3.49**

which contains almost 1000 symbols. Compared to the English dataset, which has only 62 classes, the Kannada dataset has much more classes. This makes it more difficult than the English dataset. The Kannada dataset is also divided into subsets called *Hnd* and *Img*. Following [32] and [5], we conducted two sets of experiments. In the first, we use only *Hnd* subset. The subset has 657 classes. The subset is split into 12 samples for training and 13 samples for test per class. In the second, we train with all the *Hnd* samples, and test with all samples with same classes in *Img*. Some of examples in Kannada dataset are shown in Figure 3.

Table III shows the script recognition results on the Kannada dataset. Other methods [5], [32] in the literature and our previous work [37] are also presented for the purpose of comparison. Note that our method achieves the best result on the *Hnd* test set. However, for the *Img* test set, our result is not as good as other methods. One possible reason is that the *Hnd* training set is highly structural, and with various kinds of background, the information in these images were not well encoded in our trained model, which makes the performance lower than others.

C. Character Sequence Recognition

In this section, we conduct experiments directly on the character sequence recognition, that is, the word recognition. We make use of two benchmark datasets, the ICDAR2003 [1] dataset and the Street View Text (SVT) [7], [36] dataset.

As mentioned above, the ICDAR2003 [1] dataset contains three subsets. Besides the robust character recognition dataset, there is a robust word recognition dataset. The words in this dataset are extracted from the natural scenes. Each image contains a single word. Some of examples of the images are shown in Figure 4. There are totally 1156 images in the training set, and 1110 images in the test set. The image sizes vary according to the original images. In the experiment, all the images are scaled into the same size without keeping the aspect ratio.

The Street View Text (SVT) [7], [36] dataset was collected from the Google Street View. It contains 647 words and 3796 characters in 249 images. Compared to the ICDAR2003 [1], the SVT dataset is more challenging because of the various character types, distortions and illumination. Figure 5 shows some examples of the SVT dataset. We follow [7], [36] on the training set and test set. The lexicon is used as all the words appearing in the dataset.

TABLE IV
THE COMPARISON ON THE WORD RECOGNITION ACCURACY (IN PERCENT) FOR THE ICDAR 2003 AND SVT DATASET.

Method	ICDAR 2003	SVT
Our Method	**62.12**	55.30
SYNTH+PLEX [7]	62	**57**
ICDAR+PLEX [7]	57	56
ABBYY [7]	55	35

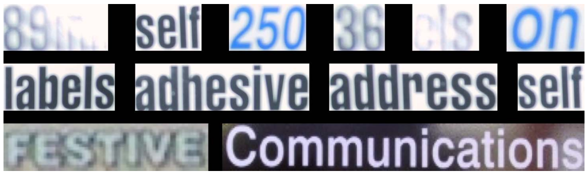

Fig. 4. Some examples of the ICDAR2003 Word dataset

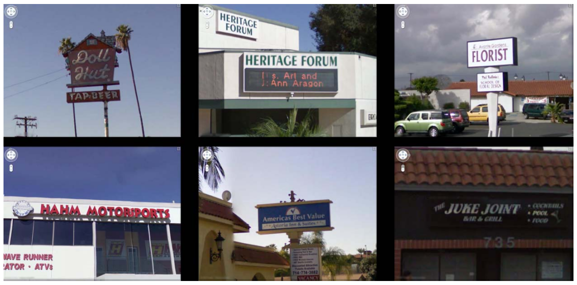

Fig. 5. Some examples of the SVT dataset

Table IV shows the word recognition results on the ICDAR 2003 [1] and SVT [7], [36] dataset. We compare our results with Wang *et al.* [7] and the ABBYY [7] methods. The SYNTH+PLEX was trained on a synthetic data. The ICDAR+PLEX was trained on the ICDAR-CH data. The ABBYY is a generic system which does not need a lexicon. From the Table IV, we can note that our method get the best results on the ICDAR 2003 dataset. Meanwhile, we get comparable results on the SVT dataset. This indicates that the SVT is a more difficult dataset, as the words are cluttered in different backgrounds and has large variance.

V. CONCLUSION

In this paper, we proposed to use the deep learning method directly on the scene character and word recognition without involving man-craft post-processing steps. Regarding the lexicon as labels, we convert the word recognition to be the same problem as characters. In this way, we solve the two problems in the same framework. Our approach uses less convolutional processing units and different initialization strategies to achieve better results. We conduct experiments on three benchmark datasets, and show the effectiveness of our methods in the literature. In the future, we will extend our model and exploit the detection and recognition of characters in a single deep framework.

Acknowledgement

This work is supported by the Major Program of University Natural Science Research of Jiangsu Province (No. 16KJA520003), the Research Foundation for Advanced Talents of Jingling Institute of Technology (No. jit-b-201717), the National Natural Science Foundation of China (NSFC) (No. 61401227 and 61703100), and the Natural Science Foundation of Jiangsu (No. BK20170692).

References

[1] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "Icdar 2003 robust reading competitions," in *IAPR International Conference on Document Analysis and Recognition*, 2003.

[2] S. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J. Jolion, L. Todoran, M. Worring, and X. Lin, "Icdar 2003 robust reading competitions: Entries, results, and future directions," *International Journal on Document Analysis and Recognition*, 2005.

[3] A. Shahab, F. Shafait, and A. Dengel, "Icdar 2011 robust reading competition challenge 2: Reading text in scene images," in *IAPR International Confernce on Document Analysis and Recognition*, 2011.

[4] C. Yi, X. Yang, and Y. Tian, "Feature representations for scene text character recognition: A comparative study," in *International Conference on Document Analysis and Recognition*, 2013.

[5] K. Sheshadri and S. K. Divvala, "Exemplar driven character recognition in the wild," in *British Machine Vision Conference*, 2012.

[6] M. Fraz, M. S. Sarfraz, and E. A. Edirisinghe, "Exploiting colour information for better scene text recognition," in *British Machine Vision Conference*, 2014.

[7] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *International Conference on Computer Vision*, 2011.

[8] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *European Conference on Computer Vision*, 2014.

[9] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *NIPS Deep Learning Workshop*, 2015.

[10] ——, "Reading text in the wild with convolutional neural networks," *International Journal on Computer Vision*, vol. 116, pp. 1–20, 2016.

[11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Annual Conference on Neural Information Processing Systems*, 2012.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *International Conference on Learning Representations*, 2015.

[16] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1891–1898.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[18] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[19] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3304–3308.

[20] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *International Conference on Computer Vision and Pattern Recognition*, 2016.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[22] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *AAAI*, 2017.

[24] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[25] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4737–4749, 2014.

[26] H. Cho, M. Sung, and B. Jun, "Canny text detector: Fast and robust scene text localization algorithm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3566–3573.

[27] X. Bai, C. Yao, and W. Liu, "Strokelets: A learned multi-scale mid-level representation for scene text recognition," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2789–2802, 2016.

[28] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Detecting oriented text in natural images by linking segments," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

[29] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[30] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[31] A. Poznanski and L. Wolf, "Cnn-n-gram for handwriting word recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2305–2314.

[32] T. E. de Campos and B. R. Babu, "Character recognition in natural images," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, Lisbon, Portugal, February 2009.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *arXiv preprint arXiv:1502.01852*, 2015.

[34] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics (AISTAT)*, 2010, pp. 249–256.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[36] K. Wang and S. Belongie, "Word spotting in the wild," in *European Conference on Computer Vision*, 2010.

[37] H. Zhao, Y. Hu, and J. Zhang, "Reading text in natural scene images via deep neural networks," in *to appear in Proceedings of the 4th Asian Conference on Pattern Recognition*. IEEE, 2017.