Reading Text in Natural Scene Images via Deep Neural Networks

Haifeng Zhao* Yong Hu* Jinxia Zhang†
*School of Software Engineering, Jinling Institute of Technology,
99 Hongjing Avenue, Nanjing 211169, China
†Key Laboratory of Measurement and Control of CSE, Ministry of Education
School of Automation, Southeast University, Nanjing 210096, China

Abstract

Text recognition in the natural scenes have gained much attention in recent years. Although optical character recognition (OCR) was well studied in the past, many effects including different backgrounds, font styles, illumination and noises make it a challenge problem. In this paper, we propose a novel method based on the deep neural networks to recognize characters and words in the natural scene images. We use a convolutional neural network (CNN) model based on the VGG-Net, emphasizing more information on the network architecture. We conducted experiments on the state-of-art benchmark datasets in the literature, and demonstrated the effectiveness of our method.

1. Introduction

Text is the representation of civilization which can be dated from thousands of years ago the inscriptions on bones or tortoise shells of the Shang Dynasty. Text is used widely in our daily life, not only in documents, but also in boxes, desks, store captions, road signs, etc. All these texts help people to communicate with each other well. With the development of computing and Internet technology, automatically recognizing objects to leverage labor is becoming more and more important. Text as a kind of special objects has gained more and more attention in the literature. With the text recognition, one can help blind people to navigate on the road and in the supermarket[28], give more attention on the automatic car driving[10], search on the Internet videos[12], and do automatic annotation of images[12].

Although OCR is well studied in the past decades, and has gotten much more satisfying results. They mainly focused on the text on the document paper. In this environment, the texts are mainly in a well organized form. The variants are of limited. Whereas, the text in the natural scenes has much more variants including fonts, backgrounds, illumination, noises and so on. This makes the text recognition a very challenging problem.

The text recognition has gained a lot of attention since ICDAR 2003[18], where a competition was held to address this problem. Researchers employed different recognition methods [17, 20, 29] to solve this problem. Wang *et al.*[27] brought in the generic object recognition methods for character recognition, and proposed an end-to-end approach using the Bayesian inference framework to improve the accuracy. Sheshadri *et al.*[24] resorted to the variants in the classifiers. They used an exemplar based SVM to classify the Kannada characters making use of the characteristics of the exemplars in the dataset. Fraz *et al.* [3] presented to involve color information for text region detection, and used bilateral regression for classification in the presence of noise. Though effective, these methods have to employ man-craft features or classifier variants to get the a relative good results. The preprocessing and post-processing steps make them hard to generalize to more complex applications, such as the texts with cluttered backgrounds.

Recently, deep learning[6] has become an exciting approach since the success on the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [19], where Krizhevsky *et al.*[15] trained a large deep convolutional neural network (CNN)[16] to do object recognition on millions of images. Deep learning has also been successfully applied to speech recognition[9], machine translation[1], and face recogntion[26], etc. The deep learning methods employ the traditional neural networks, and make the network much deeper than the traditional one by the new activation function and optimization method. The optimization is one of the foundation of the machine learning. It can be used in the saliency detection[30] and hashing[22, 21, 23]. In deep learning, one can directly input the raw data and get the final output from the netowrk output values. This end-to-end feature makes deep learning a good framework for the scene text recognition.

Deep learning has been paid attention by many researchers, and been applied to the text recognition. For text recognition in the natural scenes, Jaderberg *et al.*[13]

2327-0985/17 $31.00 © 2017 IEEE
DOI 10.1109/ACPR.2017.25

uses a deep learning method to the word recognition with a dictionary. After that, the same authors[11] proposed a new method which combined the character recognition and word recognition together to achieve better classification results. Gupta *et al.*[7] provides an elegant framework on the Jaderberg's method[12] for the text recognition. Frinken *et al.*[4] proposed to use Recurrent Neural Network (RNN)[6] to do the word recognition making use of the relationship between words and characters by the language model.

The use of deep learning methods advances the scene text recognition, and achieves better results than before. However, due to the large variance and inefficient data, the above mentioned approaches still use carefully designed post-processing step to deal with these problems, and do not make full use of the deep learning strength. In this paper, we propose an end-to-end approach for the scene text recognition.

To solve this problem, we propose to use a CNN[16]-based model. Based on the idea of VGG-Net[25], we employ different initialization and data augmentation strategy to get less training errors. For the seek of the training speed, we slightly adjust the architecture of the deep networks. The strength has two fold. Firstly, the different initialization method can employ different initial network parameters, which has effects on the final model. Secondly, the use of data augmentation can increase much more data samples than the original dataset. And at the same time, it can keep the distribution of the samples. We conduct experiments on two benchmark datasets in the literature. The proposed method can do better training and still get better results.

The rest of paper is organized as follows. In Section 2, we put our method in details. In Section 3, we do the experiments on benchmark datasets and analyze the results. We conclude in Section 4.

2. The Proposed Method

In this section, we will elaborate our proposed method. Before that, we first give a brief review of the VGG-Net[25]. Then we introduce the details of our deep model to show how a better solution to robust character recognition can be derived.

2.1. The Original VGG-Net

The VGG-Net[25] was introduced by the VGG team for the ILSVRC [19]. The insight 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 group of convolutional layers followed by three Fully-Connected (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[15] 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.

2.2. Our Deep Network

Our deep network model is based on the VGG-Net. Similar to the VGG-Net, we use several successive small size convolutional layers and FC layers. The main difference is the network configuration. Instead of using a convolutional layer with 64 channels, we directly resize images into 64×64 at the input layer. The main motivation of this idea is that the many times of convolution operations are more time consuming compared to the resizing operation. Using the resized images directly can still get comparable results. Besides, instead of using the colorful images, we use the gray images as input. As most text are written in a consistent manner, the color information plays not as important role as the other features, e.g., shape and texture.

By using the resizing operation, we can reduce one convolutional layer, and make the convolutional layer starting from 128 channels. For other convolutional layers, we use the same settings as VGG-Net at first, and conduct experiments on the validation set of Chars74K *Hnd*[2]. 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.

Table 1 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. As the character recognition, we do not subtract the mean of the training set. We use the min-batch gradient descent to carry out the optimization process. The batch size is set to 64, and 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 weight decay and dropout as regularization. The weight decay is set to 0.001 and dropout ratio is set to 0.5. As the channel number of the third FC layer has to set

Table 1. The comparison of VGG-Net and our model.

Layer	VGG-Net[25]	Our Model
Input	224×224 RGB image	64×64 gray image
Convolution	conv3-64	-
	conv3-64	-
Pooling	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-2048
	FC-4096	FC-2048
	FC-1000	FC-62
Output	softmax	softmax

to the number of classes in the dataset. This value is set to 62 for alphanumeric character recognition, and set to 657 for the script character recognition.

The initialization of the network plays an important role for the resulting performance. To this end, we have used He's method[8]. 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's 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 Xavier[5] 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[14] framework. In the training, we first apply data augmentation for all the images in the training set, and then resize all images into 64×64. In the test, we directly resize images into 64×64. 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.

3. Experiments

In this section, we perform experiments on the two benchmark datasets: Chars74K[2] and ICDAR2003[18]. We show the learning process that how the loss changes as the iteration increases. Besides the English alphanumeric character recognition, we also conduct experiments on a script language Kannada to show the generalization of our method. We compare our model with other methods in the literature on both datasets. The training on both datasets took several hours, however, the test time is in seconds for each sample.

3.1. Alphanumeric Character Recognition

Figure 1. Some examples of the Chars74K English dataset

The Chars74K English dataset [2] [1] 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 challenging dataset. Following [2]'s settings, we use Chars74K-15 from *Img* subset as our benchmark dataset. Here, we use 15 images for training and the rest for test in each class. Figure 1 shows some examples of the Chars74K English dataset.

The ICDAR2003 dataset [18] is also comprised of three subsets for the purpose of robust text locating and recognition. The dataset is a challenging benchmark for the text recognition problem. Following [24], we use the ICDAR-CH 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.

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

Table 2. The comparison on the recognition accuracy (in Percent) for the Chars74K-15 and ICDAR-CH datasets.

Method	Chars74K-15	ICDAR-CH
Our Method	**69.99**	**80.14**
HOG+ESVM+AFF[24]	69.66	70.53
Global HOG+SVM[29]	62	76
HOG+NN [28]	57.5	51.5
NATIVE+FERNS [27]	54	64
MKL[2]	55.26	-

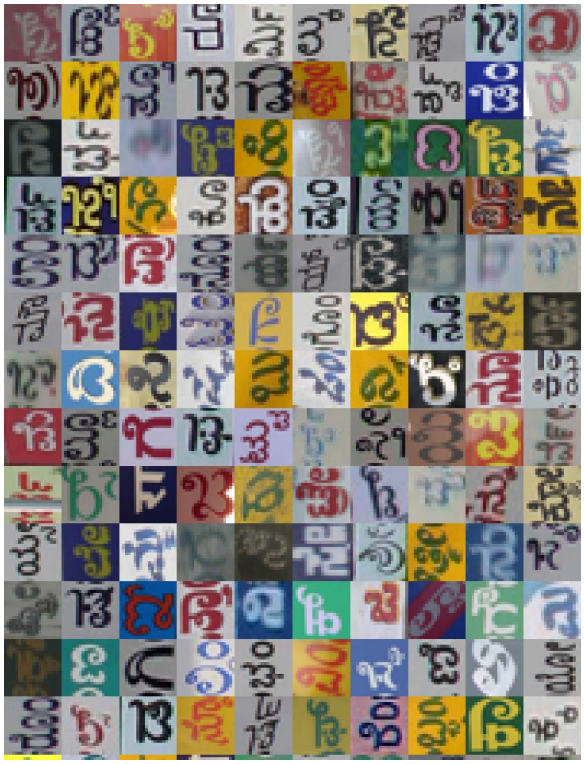


Figure 2. Some examples of the Chars74K Kannada dataset

We present the recognition accuracy in Table 2 for the Chars74K-15 and ICDAR-CH datasets, and compared with other methods in the literature. The bold fonts indicate the best results. On both of the sets, our method outperforms all the other methods. In ICDAR-CH, our method even improves by about 4%, compared to the 76% in [29]. As ICDAR-CH has more samples than Chars74K-15, this implies that more data can make our model to get better results.

3.2. Script Character Recognition

The Chars74K has another component, called Kannada dataset. The Kannada is an Indic script language, 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 [2] and [24], 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 2.

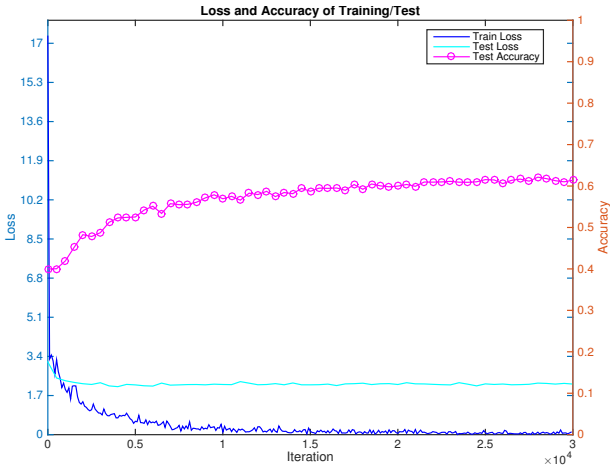
Table 3. The comparison on the recognition accuracy (in Percent) for the Chars74K Kannada dataset.

Method	Test on *Hnd*	Test on *Img*
Our Method	**60.99**	2.37
HOG+ESVM+AFF[24]	54.13	1.76
SC[2]	29.88	**3.49**

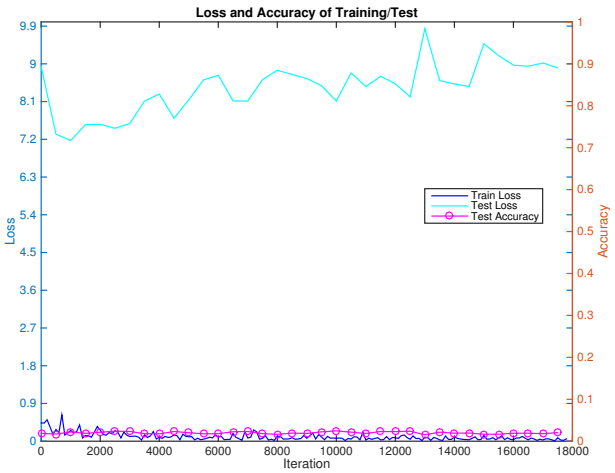
Table 3 shows the recognition results on the Kannada dataset. As [29], [28] and [27] did not report the results on the Kannada dataset, we only compare the results with [24] and [2]. Our method achieves the best result on the *Hnd* test set. With regard to the *Img* test set, we outperform the [24] method, but not better than [2]. One possible reason is that *Hnd* training set is highly structural, our method does not capture as much structural information as the shape contexts (SC)[2]. Meanwhile, the *Img* test set is cluttered in different backgrounds and has large variance compared to the *Hnd* training set, which makes it even harder to recognize. Figure 3 shows the training and test loss as a function of the iterations on the two subsets, the *Hnd* and *Img*. Note that, in Figure 3(a), the training loss is decreasing as the iteration increases. The test loss is also monotonically decreasing. After 15000 iterations, the training converges on the *Hnd* subset. Whereas, in Figure 3(b) the training is obviously overfit on the *Hnd* subset, as the training loss is very small on the *Hnd* and the test loss is still large on the *Img*. This verifies our hypothesis on the large variance of the two subsets.

4. Conclusion

In this paper, we proposed to use the deep learning method directly on the scene text recognition without involving man-craft post-processing steps. The approach is based on the VGG-Net[25], and uses less convolutional processing units and different initialization strategies to achieve better results. We conducted experiments on two benchmark datasets, the Chars74K[2] and ICDAR2003[18], and show that we can outperform the methods in the literature. In the future, we will extend our model and exploit more deep characteristic on the end-to-end multilingual text recognition.



(a) Test on Hnd



(b) Test on Img

Figure 3. Training and test loss/accuracy as a function of iterations on the Chars74K Kannada dataset

Acknowledgement

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

References

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*, 2015. 1

[2] 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. 2, 3, 4

[3] M. Fraz, M. S. Sarfraz, and E. A. Edirisinghe. Exploiting colour information for better scene text recognition. In *British Machine Vision Conference*, 2014. 1

[4] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke. A novel word spotting method based on recurrent neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):211–224, 2012. 2

[5] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 249–256, 2010. 3

[6] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. 1, 2

[7] 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. 2

[8] 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. 3

[9] 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*, 29(6):82–97, 2012. 1

[10] M. Jaderberg. *Deep Learning for Text Spotting*. PhD thesis, University of Oxford, 2015. 1

[11] 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. 2

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal on Computer Vision*, 116:1–20, 2016. 1, 2

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *European Conference on Computer Vision*, 2014. 1

[14] 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. 3

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems*, 2012. 1, 2

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. 1, 2

[17] 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. 1

[18] 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. 1, 3, 4

[19] 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)*, 115(3):211–252, 2015. 1, 2

[20] 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. 1

[21] F. Shen, C. Shen, W. Liu, and H. T. Shen. Supervised discrete hashing. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 37–45. IEEE, 2015. 1

[22] F. Shen, C. Shen, Q. Shi, A. Van Den Hengel, and Z. Tang. Inductive hashing on manifolds. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1562–1569. IEEE, 2013. 1

[23] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, and H. T. Shen. Hashing on nonlinear manifolds. *IEEE Transactions on Image Processing*, 24(6):1839–1851, 2015. 1

[24] K. Sheshadri and S. K. Divvala. Exemplar driven character recognition in the wild. In *British Machine Vision Conference*, 2012. 1, 3, 4

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 2, 3, 4

[26] 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*, pages 1891–1898, 2014. 1

[27] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *International Conference on Computer Vision*, 2011. 1, 4

[28] K. Wang and S. Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, 2010. 1, 4

[29] 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. 1, 4

[30] J. Zhang, K. A. Ehinger, H. Wei, K. Zhang, and J. Yang. A novel graph-based optimization framework for salient object detection. *Pattern Recognition*, 64:39–50, 2017. 1