End-to-End Scene Text Recognition with Character Centroid Prediction

Wei Zhao and Jinwen Ma(✉)

Department of Information Science, School of Mathematical Sciences And LMAM, Peking University, Beijing 100871, China
jwma@math.pku.edu.cn

Abstract. Scene text recognition tries to extract text information from natural images, being widely applied in computer vision and intelligent information processing. In this paper, we propose a novel end-to-end approach to scene text recognition with a specially trained fully convolutional network for predicting the centroid and pixel cluster of each character. With the help of this new information, we can solve the character instance segmentation problem effectively and then combine the recognized characters into words to accomplish the text recognition task. It is demonstrated by the experimental results on ICDAR2013 dataset that our proposed method with character centroid prediction can get a promising result on scene text recognition.

Keywords: Scene text recognition · Character centroid prediction · Fully convolutional networks · Character instance segmentation

1 Introduction

Text recognition has been investigated and applied for many years. Actually, Optical Character Recognition (OCR) is considered to be a powerful character recognition tool for the scanned images of papers or articles, but it is still rather difficult to detect and recognize text in natural scenes due to the complex environment, low image quality and other uncontrollable factors [1]. In recent years, deep convolutional neural networks have shown great capability on solving the computer vision problems and they also work well for scene text recognition. When using a CNN to undertake a text recognition task, it is usual to divide the whole problem into two subproblems: text localization and cropped word recognition, and we then train two different models to solve them separately. For the text localization problem, some specific characteristics of text are utilized to detect the fields of text in the image [3–5]. Apart from those methods, some general object detection techniques like the Faster-RCNN [2] can work as well. For the cropped word recognition problem, most recent methods are based on CRNN [6] that utilizes a CNN to extract the features and a RNN to deal with the sequence learning. Under this localization-and-recognition framework, we can get some good results in certain cases. However, it has certain innate drawbacks

D. Liu et al. (Eds.): ICONIP 2017, Part III, LNCS 10636, pp. 291–299, 2017.
https://doi.org/10.1007/978-3-319-70090-8_30

to train the two models separately. First, since each model only learns from a part of the training data, there will be certain loss in the final result. Second, since the both models use a DCNN to extract the features from an image, a lot of sharable computation are repeated, which leads to low efficiency.

In order to resolve these issues, we propose a novel end-to-end method for scene text recognition. Our main idea is to firstly make the instance segmentation for all the characters in an image and then combine the recognized characters into words. We consider the character instance segmentation as a clustering analysis problem where each character is corresponding to a cluster and our object is to divide the related pixels into a number of clusters which are corresponding to the characters, respectively. In order to do so, we train a fully convolutional neural network to predict the centroid and pixel cluster of each character. It is demonstrated by the experiments on ICDAR2013 dataset that our proposed method leads to a promising result on scene text recognition.

2 Related Works

2.1 Character Instance Segmentation

As shown in Fig. 1, character instance segmentation can be roughly considered as a combination of object detection and semantic segmentation. In fact, most general instance segmentation methods are either detection-based or segmentation-based methods. The detection-based methods [9] first detect all the possible instances in an image and then predicts the mask of each instance, while the segmentation-based methods [8] first put a label on each pixel and then group the pixels into some instances. Here, we adopt the segmentation-based method to make the character instance segmentation since the current object detection methods are not so good with small objects which is also the reason why we do not take text recognition with direct character detection.

2.2 Fully Convolutional Network

Fully convolutional network (FCN) only contains a number of convolutional layers. In comparison with the other deep learning neural network architectures, it has neither pooling layer nor fully connected layer. As a result, it can take an arbitrary-sized image as input and predict what we need for each pixel which makes them good at solving semantic segmentation problem. In a typical convolutional neural network, feature map will be down-sampled several times while it goes through the convolutional layers. As it goes deeper, more semantic information are extracted but a lot spatial information are abandoned. To deal with this effect, one common technique is to up-sample deep layer and concatenate it with shallow layer and do the final prediction base on this concatenated feature map, as proposed in [10]. With this modification, FCN can predict pixel label accurately. Of course, character instance segmentation cannot be done with only pixel label so that we train an FCN to predict character centroid as well as pixel label.

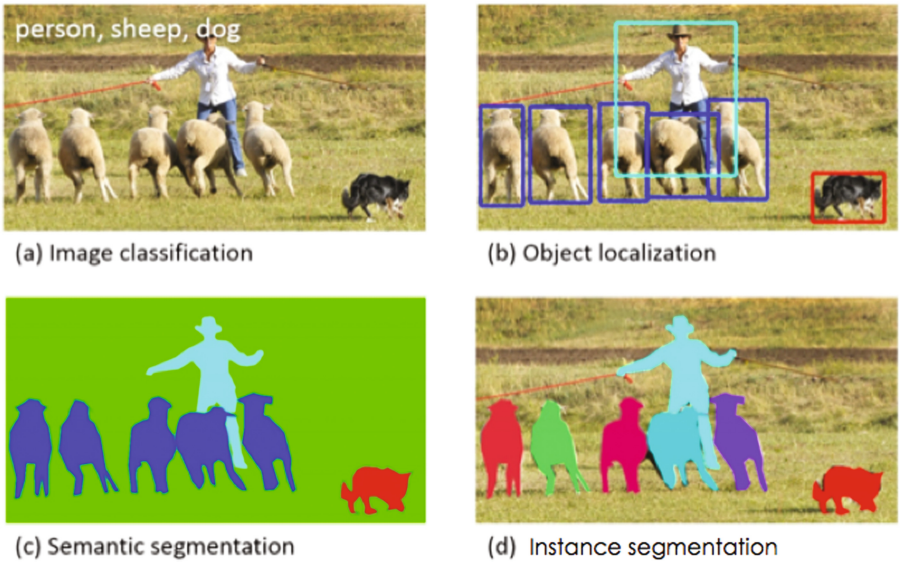

Fig. 1. An example of image classification, object localization (also known as object detection), semantic segmentation and instance segmentation from [7]. (Color figure online)

2.3 Character Centroid Prediction

In fact, character centroid prediction is important for text recognition. Last year, Zhang et al. [5] utilized this idea to make text localization via a FCN-based model. In particular, a common FCN was trained to make the two-class semantic segmentation, with the segmentation result shown in Fig. 2. Actually, character centroid prediction acted as an auxiliary helper to select text line candidate so they didn't put too much effort on it. As we can see, their centroid prediction result is not very good, especially in some cases centroid regions even overlap with each other. There is one unnatural setting in their model that may be the reason to harm their result: they consider the whole problem as a classification

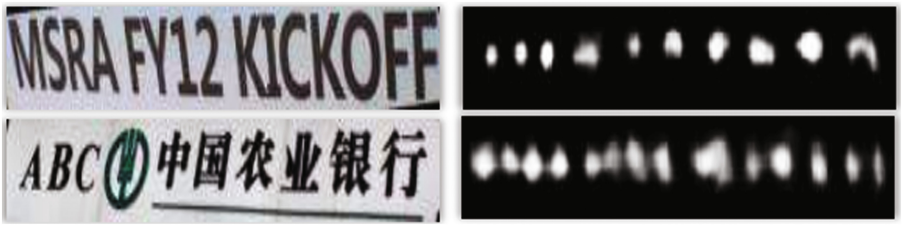

Fig. 2. The centroid prediction result given in [5].

problem and the label of a pixel is determined by setting a threshold on the distance between a pixel and the real centroid.

Now, we consider the character centroid prediction as a regression problem. Instead of predicting whether a pixel is near the real centroid, we train our FCN to learn pixel's relative position to the real centroid. As shown in Fig. 5, our FCN learns this target very well and we can easily accomplish character instance segmentation with our FCN's prediction.

3 Proposed Network with Character Centroid Prediction

3.1 Network Structure

As mentioned in the previous sections, our method's main component is an FCN. Figure 3 shows the network structure. We design our network based on two principles: 1. The size of feature map shouldn't shrink too much, in order to get precise character mask; 2. The size of receptive field must be large enough so that context information can be used. We adopt the residual learning structure proposed in [11] and use the consecutive downsample-upsample to enlarge receptive field while keeping the size of feature map. For each pixel, our FCN gives two predictions. The first one is an n_c-d vector representing pixel label probability. The other one is an $6n_c$-d vector, representing the pixel's relative position to its corresponding character centroid and neighbor character centroid given the pixel's label. Figure 4 shows an illustration of our network's prediction.

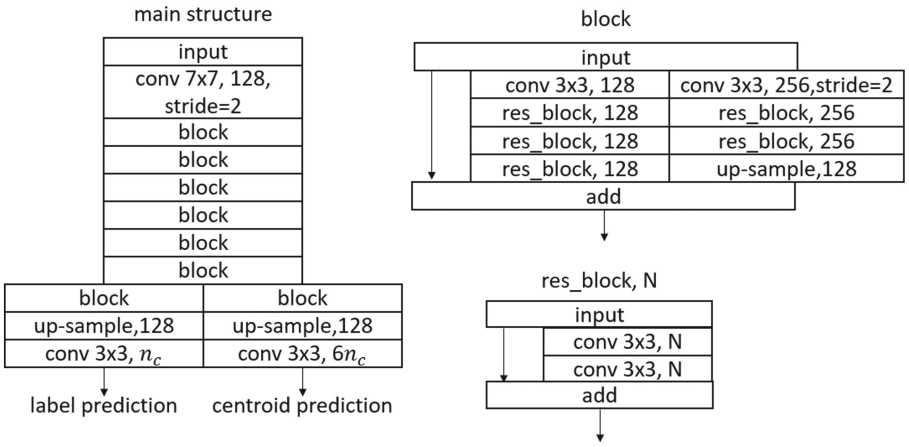

Fig. 3. Our network structure. Convolutional layer's kernel size and channels are showed in the figure and stride is 1 if not specified. Up-sampling is done by deconvolution (transposed convolution) which enlarges previous feature map 2 times and keeps the channel number unchanged. n_c is the number of classes which equals to 37 in our case (10 digits, 26 letters and background).

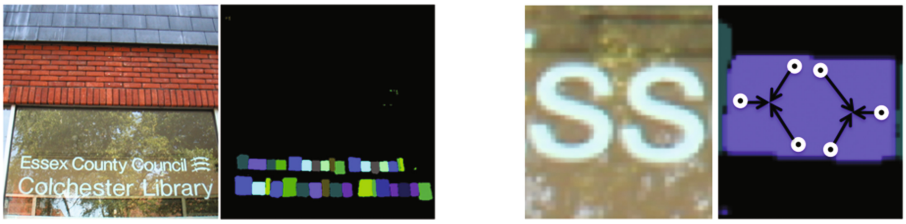

Fig. 4. Illustration of our network's prediction. The left two images show the label prediction result on test image. As we can see, the prediction is very accurate. However, without other information, we can't group pixels into characters, especially for those consecutive characters. The right two images show our goal of centroid prediction. With every pixel pointing to corresponding character's centroid, we can easily accomplish instance segmentation for character. We also let our network to predict neighbor character's centroid in order to combine characters into words. (Color figure online)

3.2 Training Process

During the training, we optimize the following two-part loss function:

$$Loss = \lambda \sum_{i,j=0}^{n-1} \mathbf{1}_{c_{ij} \neq 0} L1_{smooth}(\mathbf{pos}_{ijc_{ij}}^{pred}, \mathbf{pos}_{ij}^{truth}) - \sum_{i,j=0}^{n-1} log(p_{ij}(c_{ij})), \quad (1)$$

where

$$L1_{smooth}(f, g) = \begin{cases} 0.5(f-g)^2, & \text{if } |f-g| < 1; \\ |f-g| - 0.5, & \text{otherwise.} \end{cases} \quad (2)$$

In fact, the first part is just the smoothed L1 loss which guides relative centroid position regression, where n is the size of input image, $\mathbf{1}_{c_{ij} \neq 0}$ denotes if pixel $I(i,j)$ belongs to a character, $\mathbf{pos}_{ijc_{ij}}^{pred}$ denotes predicted relative position for pixel $I(i,j)$ if $I(i,j)$'s true label is c_{ij}, $\mathbf{pos}_{ij}^{truth}$ denotes the true relative position for pixel $I(i,j)$. Both $\mathbf{pos}_{ijc_{ij}}^{pred}$ and $\mathbf{pos}_{ij}^{truth}$ are 6-d vectors. If a character is the first or the last in a word, we let its left/right neighbor to be itself. The second part is the cross-entropy loss for pixel label prediction where c_{ij} is the true label of pixel $I(i,j)$ and $p_{ij}(c_{ij})$ is predicted probability that $I(i,j)$'s label is c_{ij}. We use a hyper-parameter λ to balance these two parts of loss and we set λ to 10^{-6} during in practice.

We train our network on synthetic data made by [12]. Our loss function is very simple and can be optimized by any gradient-based method. We use SGD with momentum to train our model. Our learning rate starts at 0.01, and cuts into half after [9600, 19200, 48000, 96000, 192000, 384000, 768000, 1152000] times parameter update.

3.3 Inference Principle

For a test image, we first use the trained FCN to predict the pixel's label and character's centroid. We can't just group together pixels which have same predicted centroid because there are small error in centroid prediction. We use non-maximum-suppression to solve this problem. In detail, for a non-background pixel $I(i,j)$, the sequence $I(i,j), PC(I(i,j)), PC(PC(I(i,j))), ...$ "converges" very quickly in most cases, where $PC(I(i,j))$ denotes $I(i,j)$'s predicted centroid's nearest pixel, and we use $PC^*(I(i,j))$ to denote the "limitation" of this sequence. We say $I(i,j)$ is $I(i',j')$'s "supporter" if $PC^*(I(i,j)) = I(i',j')$. We let pixel $I(i',j')$ to be a centroid candidate if it has enough supporters. For each pair of two centroid candidates $I(i_1,j_1)$ and $I(i_2,j_2)$, we change $I(i_1,j_1)$'s supporters' final predicted centroid to $I(i_2,j_2)$ and remove $I(i_1,j_1)$ from centroid candidates if the distance between these two candidates is too small and $I(i_1,j_1)$ has less supporters than $I(i_2,j_2)$. At last, for each character there is only one centroid candidate left, which we use as the final predicted character centroid and its supporters make up character's mask. For the probability of character's label, we let it be the average of its pixels' label probability. At this point, character instance segmentation is done and we now combine them into words. As described above, our network not only predicts pixel's corresponding character's

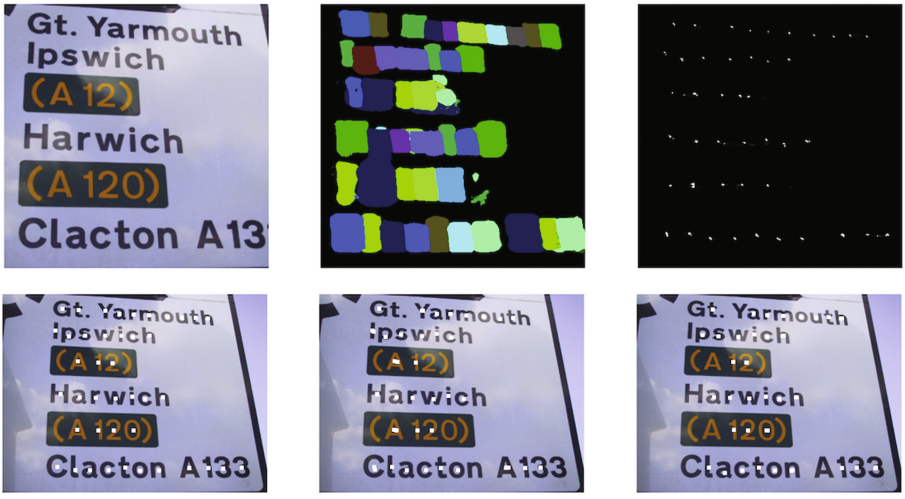

Fig. 5. Top row: test image, pixel label prediction, and heat map of pixel's supporter count(clipped to an appropriate interval). As we can see, the centroid prediction is very compact and consistent so we can easily get the final centroid prediction result. Bottom row: final centroid prediction for characters, characters' left neighbors, and characters' right neighbors(if a character is the start/end of a word, we set its left/right neighbor to itself). Our network not only predict character centroid accurately, the neighbor centroid prediction result is also good enough for us to combine characters into words. (Color figure online)

centroid, but also neighbor character's centroid. For a character, we use its final centroid candidate's prediction as neighbor character prediction result. Given a pair of characters, if one's centroid prediction is close enough to the other one's neighbor centroid prediction, then we determine that they belong to the same word. Figure 5 visualizes our network's prediction result and we can see that our method is valid.

4 Experimental Results

We test our proposed method on ICDAR2013 dataset [15]. Since this dataset contains images of many different scales, we just adopt the image pyramid technique to make multi-scale text recognition on task4. Our end-to-end recognition experimental results are listed in Table 1, in comparison with some state-of-the-art methods. We also test our proposed method on text localization task and give the experimental results in Table 2.

Table 1. The experimental results of text recognition on ICDAR2013 task 4.

Method	End-to-end			Word spotting		
	Recall	Precision	HMean	Recall	Precision	HMean
hust_mclab [6,16]	87.68%	95.83%	91.57%	90.77%	97.25%	93.90%
vggmaxbbnet [17,18]	82.12%	91.05%	86.35%	86.68%	94.64%	90.49%
Deep2Text II+[17,19]	72.08%	94.56%	81.81%	75.82%	96.29%	84.84%
Proposed method	84.62%	90.76%	87.58%	91.82%	93.24%	92.53%

In fact, ICDAR2013 task 4 was strongly contextualized. It can be seen from Table 1 that our proposed method can get a rather good result. Although it does not surpass the result of some state-of-the-art methods on the web, we just use the single-model end-to-end method while the other three methods all make text localization and cropped word recognition in two steps, respectively. So, our proposed method is more promising since our network can be further optimized

Table 2. The experimental results of text location on ICDAR2013 task1.

Method	Recall	Precision	HMean
CTPN [3]	82.98%	92.98%	87.69%
TextConv+WordGraph [4]	81.02%	93.38%	86.76%
MCLAB_FCN [5]	79.65%	88.40%	83.80%
IWRR2014 [13]	78.65%	85.89%	82.11%
HUST_MCLAB [14]	76.05%	87.96%	81.58%
Proposed method	87.16%	88.82%	87.98%

globally. Moreover, it can be seen from Table 2 that our proposed method is remarkably better than the other existing methods according to deteval criterion. It should be noted that our recall rate surpasses the other methods by a large margin.

5 Conclusion

We have proposed an end-to-end text recognition method with character centroid prediction. It is based on a specially trained fully convolutional network to predict the centroid and pixel cluster of each character so that the character instance segmentation problem can be solved effectively and then the recognized characters can be combined into words to accomplish the text recognition task. It is demonstrated by the experimental results on ICDAR2013 dataset that our proposed method can get a promising result on scene text recognition.

Acknowledgments. This work was supported by the Natural Science Foundation of China under Grant 61171138.

References

1. Ye, Q., Doermann, D.: Text detection and recognition in imagery: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **37**(7), 1480–1500 (2015)
2. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28, pp. 91–99 (2015)
3. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 56–72. Springer, Cham (2016). doi:10.1007/978-3-319-46484-8_4
4. Zhu, S., Zanibbi, R.: A text detection system for natural scenes with convolutional feature learning and cascaded classification. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 625–632 (2015)
5. Zhang, Z., Zhang, C., Shen, W., et al.: Multi-oriented text detection with fully convolutional networks. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 4159–4167 (2016)
6. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1 (2016)
7. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). doi:10.1007/978-3-319-10602-1_48
8. Zhang, Z., Fidler, S., Urtasun, R.: Instance-level segmentation for autonomous driving with deep densely connected mrfs. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 669–677 (2016)

9. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 3150–3158 (2016)
10. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 3431–3440 (2015)
11. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of 2016 IEEE conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 770–778 (2016)
12. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), pp. 2315–2324 (2016)
13. Zamberletti, A., Noce, L., Gallo, I.: Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In: Jawahar, C.V., Shan, S. (eds.) ACCV 2014. LNCS, vol. 9009, pp. 91–105. Springer, Cham (2015). doi:10.1007/978-3-319-16631-5_7
14. Zhang, Z., Shen, W., Yao, C., et al.: Symmetry-based text line detection in natural scenes. In: Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), pp. 2558–2567 (2015)
15. Karatzas, D., Shafait, F., Uchida, S., et al.: ICDAR 2013 robust reading competition. In: Proceedings of 2013 International Conference on Document Analysis and Recognition, pp. 1484–1493 (2013)
16. Liao, M., Shi, B., Bai, X., et al.: TextBoxes: a fast text detector with a single deep neural network. In: Prooceedings of AAAI 2017, pp. 4161–4167 (2017)
17. Jaderberg, M., Simonyan, K., Vedaldi, A., et al.: Reading text in the wild with convolutional neural networks. arXiv preprint arXiv:1412.1842 (2014)
18. Jaderberg, M., Simonyan, K., Vedaldi, A., et al.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014)
19. Yin, X.C., Yin, X., Huang, K., et al.: Robust text detection in natural scene images. IEEE Trans. Pattern Anal. Mach. Intell. **36**(5), 970–983 (2014)