Multi-scale FCN with Cascaded Instance Aware Segmentation for Arbitrary Oriented Word Spotting In The Wild

Dafang He[1], Xiao Yang[2], Chen Liang[1], Zihan Zhou[1], Alex G Ororbia[1], Daniel Kifer[2], and C.Lee Giles[1]

[1]Information Science and Technology, The Penn State University
[2]Computer Science and Technology, The Penn State University

Abstract

Scene text detection has attracted great attention these years. Text potentially exist in a wide variety of images or videos and play an important role in understanding the scene. In this paper, we present a novel text detection algorithm which is composed of two cascaded steps: (1) a multi-scale fully convolutional neural network (FCN) is proposed to extract text block regions; (2) a novel instance (word or line) aware segmentation is designed to further remove false positives and obtain word instances. The proposed algorithm can accurately localize word or text line in arbitrary orientations, including curved text lines which cannot be handled in a lot of other frameworks. Our algorithm achieved state-of-the-art performance in ICDAR 2013 (IC13), ICDAR 2015 (IC15) and CUTE80 and Street View Text (SVT) benchmark datasets.

1. Introduction

Reading text from scene images could contribute to many applications such as image, video indexing and reading systems for visually impaired people. Thus it has received an increasing attention these years. However, reading arbitrary oriented text lines from scene images is still difficult and only partially solved. As mentioned in [41], reading multi-oriented text lines is a much harder problem than only considering horizontal text lines and the ability to read multi-oriented or even curved text [28] is important in many scenarios. However, there is still a gap between the need in practice and the performance of existing algorithms for multi-oriented text detection. This can be seen in the results in ICDAR 2015 competition [1].

Previous works in scene text detection could be mainly classified into two categories: (1) sliding window-based detection methods [5, 18, 33], (2) region proposal-based detection methods [25, 7, 13, 14, 10]. In early stages, sliding window-based methods received more attention. They typically have high computational cost because windows in different scales are used to slide in an exhaustive manner. Region proposal-based methods received more attention these years because of the fast computation in proposal generation and the high recall of the proposals.

Convolutional neural network, which is proven to be effective in extracting high level features from images, has also been incorporated in scene text detection [18, 17, 14, 41, 31, 10]. The ability of CNN in extracting high level feature representations greatly improve the accuracy. Several works, which combined region proposal and convolutional neural network, have achieved good performances in localizing text in horizontal or near horizontal orientations [14, 10]. They typically follow the same scheme as generating a set of proposals, and then classify each proposal to get potential individual characters. Then a bottom-up grouping algorithm is applied to group characters into text lines. However, such scheme has intrinsic problems: (1) It assumes individual characters could be identified. (2) Only horizontal lines could be detected. This is because when considering multi-oriented text lines, traditional grouping algorithms will easily find incorrect lines.

Recently, several works have made great breakthroughs in scene text detection. In [40], Zhang et al. proposed to use FCN in scene text detection and achieved a breakthrough in multi-oriented scene text detection. Tian [31] adapted the idea of using CNN to generate proposals as originally presented in [8] for object detection. However, it cannot handle curved text lines and text lines that are close to vertical orientation since they used heuristic proposal linking mechanism.

In this paper, we present an algorithm which adopts the idea of using FCN in scene text detection. The algorithm runs in a cascaded fashion which can handle truly arbitrary oriented text. We have two cascaded levels. In the whole image level, we use a multi-scale FCN in extracting representative features and removing most negative regions to

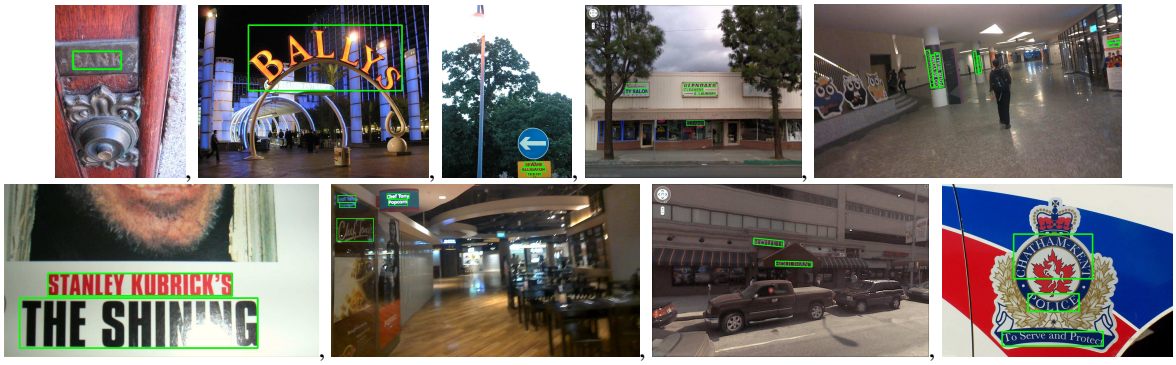


Figure 1. Scene texts that have been successfully detected by our proposed systems.

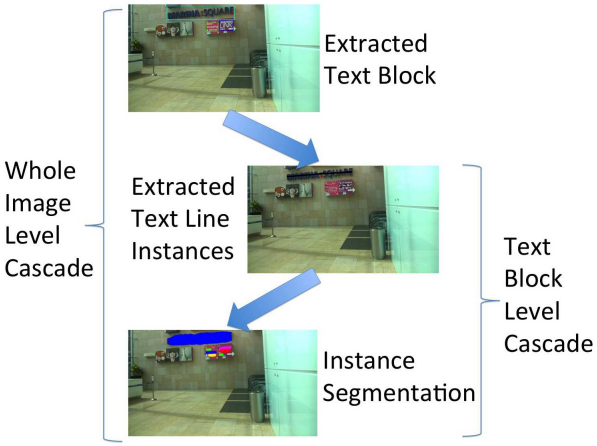


Figure 2. The pipeline of our proposed algorithms with demonstration of the two level cascades. We first extract text blocks with the multi-scale FCN. Then for each extracted text block, we predict the text center line by the proposed TL-CNN. After extracting all the text lines, we use IA-CNN to extract each instance word. Our pipeline could extract text line in arbitrary orientations.

obtain text blocks. In the text block level, we design a text instance segmentation network to obtain each text instances from the text block. We divide the challenging text instance segmentation task into two easier sub-tasks in a cascaded fashion as inspired by [6]: (1) We first manage to find text center lines by training a FCN which predicts the center line of each instance word. (2) Then we extract each single text center line from the previous output. (3) Finally, we append the extracted text line information to the text block image to further extract each text line segmentation separately. Only simple low level processing is needed in order to achieve instance level (word or text line) segmentation.

Fig. 1 shows several examples of end-to-end results. Our pipeline, which contains both the text block FCN and the word instance segmentation, is shown in Fig. 2. More results are on supplementary material.

In summation, our contribution are the followings:

1. We design a unique, instance segmentation-based model for obtaining word instance. We break the task of instance segmentation into easier tasks which achieves a good performance in separating word instances in a cascaded fashion. The text instance segmentation model has several advantages over traditional methods(including proposal-net based methods): (1)invariant to orientation (2) able to find text instance even when characters are connected (3) able to separate text lines that are close to or even touch each other. The step of obtaining text instance is also crucial for end-to-end text reading since current scene text reading methods can only read a single word or line. Our algorithm, as the first attempt of designing instance segmentation model for scene text detection, should be of great value for further researches.
2. We present a multi-scale FCN model for scene text block detection. It could help identify text block regions with large scale variances and also combine more context information for each prediction. The text block FCN aims at removing most false positive regions by extracting multi-scale feature representations. It serves as the first step of the cascaded framework.
3. We conduct thorough evaluations on several benchmark datasets, including the challenging IC15, and a curve text based CUTE80. Results show that our model achieves state-of-the-art performances.

We organize the remainder chapters in the following ways: We first briefly introduce related work in Sec. 2. In Sec. 3, our model of multi-scale, shared-net FCN is presented. In sec. 4, we describe our model of word instance segmentation. Experiments and conclusion are presented in Sec. 5, Sec. 6, respectively.

2. Related Work

Scene text detection is a challenging task. It differs from traditional object detection in several key aspects: (1) Text in images varies a lot in terms of its scales. Even really small text lines are expected to be detected. (2) Text lines or

words are expected to be extracted instead of a single character. In traditional methods, this typically means a further step of grouping characters into text lines. (3) Features of a single character is typically not enough for distinguishing it from some background noises. For example, features from a single character "I" could not be well distinguished from that of a vertical brick. Similar claims had also been made in [10]. The three properties made scene text detection a unique problem.

In early works of scene text detection [5, 18, 33], researchers mostly focused on using sliding window based methods. Such methods are typically not efficient in computation. Later, region proposal based methods, in which, extreme region (ER) or maximally extreme region (MSER) dominate, have attracted great attention [25, 7, 13, 14, 10]. They are computationally efficient, and also achieves high recall. However, region proposal based methods often fail when characters are connected or strokes are separated. In addition, classifying single letter as text or not is error-prune. Many stroke-like background noises are hard to be removed. These false positives also cause the extreme difficulty of detecting multi-oriented scene text. Multi-oriented text detection has also been proposed in several works [36, 35, 34, 41]. Most of them follow the similar scheme as traditional methods by grouping character components into text lines. They suffered from the same problems mentioned above and could not achieve high performance in challenging images.

In addition to these methods, researches in synthetic data generation [16, 9] have also achieved great progress and boost the deep learning based scene text detection models. In this paper, we also use the synthetic dataset proposed in [9], which contains 800,000 synthetic images which are fully labeled with text bounding boxes. Texts on the synthetic images follow the perspective transformation of its background and are thus much more realistic.

This work shares similar features with Zhang at el. [41] for the fact that we both use FCN and treat detection problem as a segmentation problem. However, we differ from them in several key aspects which make our algorithm more robust and general: (1) Instead of using proposal and low level line orientation estimation, we design a novel instance segmentation scheme for separating text lines. We do not make any assumption on the text orientation within each text block, and only a few low level processing steps are needed. Our model can handle arbitrary oriented text lines, including curved text lines which cannot be handled by Zhang's method. (2) We use a multi-scale, shared-net FCN to capture larger context information and text with larger variances of scales, which leads to better text block detection results.

Our work also falls into the category of cascaded methods for text detection [12, 43]. We propose two level cascaded solution for the task which is novel and robust.

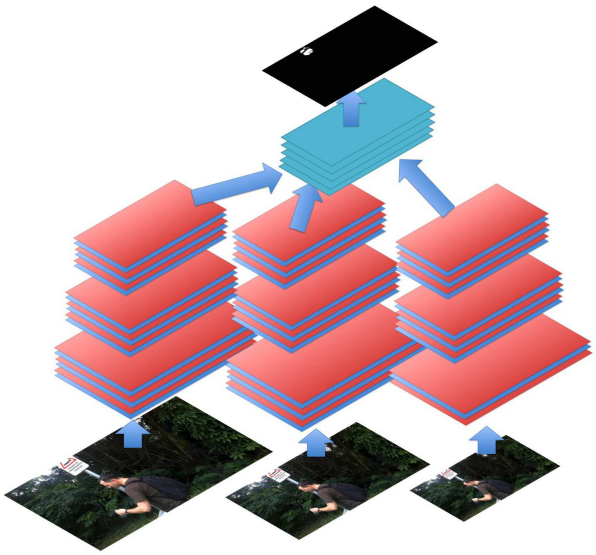


Figure 3. The architecture of our multi-scale FCN.The input is downsampled to 0.5 and 0.25 of the scale of the input image. Prediction is done by jointly considering features from three scales. The parameters of the convolutional parts are shared among all three branches.

3. Multi-Scale, Shared-Net FCN

3.1. Design Rationale

Fully convolutional neural network (FCN), which was originally presented in [23] for scene labeling, has recently been adopted by [41] in scene text detection. One problem that needs to be carefully considered is scale of the object, and several variants have been proposed to solve the problem [22, 37]. In [41], Zhang et al. tried to capture multi-scale information in an image using a single branch FCN with a skip-net model. However, it is still in doubt that text, which potentially varies a lot in terms of its relative size to the image, could be well captured without larger context information. In IC15 training sets, we calculate the relative scale of the text line, which is defined as the shorter length of its oriented bounding box divided by the height of the image. The relative scale could vary from 0.005 to 0.78. This means that a robust algorithm should be able to capture text in a wide variety of scales. If we assume that the input image in testing has a height of 500 pixels. Then the height of a text line could vary from 5 (0.01×500) to 160 (0.32×500). We argue that a single FCN is not enough to accurately capture such large variances of text.

In addition to the scale problem, background contexts could effectively improve the scene text localization performance. Similar ideas were proposed in [10, 42], which incorporated contexts in a region proposal framework. Here we have the same hypothesis and claim that it is also helpful for a FCN based scene text detection method and will improve the performance of the pixel labeling problem.

Figure 4. Comparison of the multi-scale, shared-net FCN with a single branched FCN. Left: input image. Middle: single branched FCN. right: shared-net, multi-scale FCN.

Based on the above two claims, we specifically design a multi-scale, shared-net FCN which has a larger receptive field and can capture more useful context information. It improves the performance of text block detection.

3.2. Architecture

The architecture of our model is shown in Fig 3. There are three branches with shared convolutional parameters. The encoded information, after two unpooling layers [3], is merged to produce the final results.

For each prediction in the final output, it is a joint prediction from all three branches. By joint prediction, it can capture larger context information and give more accurate prediction.

In Fig 4, we show several examples of the performance of our FCN model compared with single branched FCN. We could see that a larger context is helpful for both removing false positives and obtaining better responses around text.

3.3. Training with Per-scale Loss

We augment IC13, IC15 training sets and the synthetic dataset from [9] by random scaling and rotation. All these data are used for training our FCN model. Note that the training data contain texts in a large variance of scales. This is to mimic the situation we might encounter in testing.

Training a multi-scale, shared-net FCN is relatively harder than training a single FCN. We follow several works [21, 19, 4] that use a per-scale loss which could make the learned features from multiple scales more discriminative, and thus accelerate training and improve performance. The loss function is described in Equation 1.

$$\text{P-NLL}(\theta, D) = -\sum_{k} \log P(Y = y^k | x^k, \theta) + \sum_{i=1}^{M} -\alpha_i \times (\sum_{j} \log P(Y = y_i^j | x_i^j, \theta)) \quad (1)$$

M represents the number of scales we used. α_i represents the weight for ith scale. x^j, y^j represent the output and groundtruth, respectively. We initialize α_i to be larger in order to learn discriminative features for each scales. We then gradually decrease their values and focus on the training of the joint prediction.

Figure 5. Results of our instance segmentation model. It can capture word or text line instances in a wide variety of circumstances with arbitrary orientations. The segmented results could be directly thresholded to get the final bounding boxes.

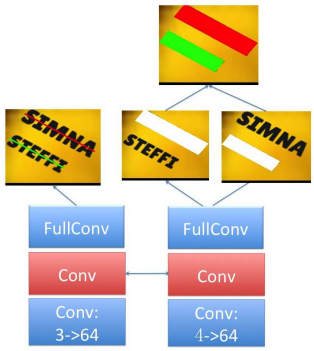
4. Cascaded Text Instance Segmentation

Given a text block, which might contain several nearby word instances, we specifically designed an instance segmentation model to segment each word instances. Instance segmentation has attracted an increasing attention in computer vision community [20, 27, 39, 6]. It is a much harder task than semantic segmentation because it has to separate out different instances of the same class. In scene text detection, we define an instance as a word or a text line which is not separable purely visually. Here, the input is the cropped text block image obtained from text block FCN, which might contain several lines or words, and we propose two networks in a cascaded fashion to solve the problem: Text Line CNN (TL-CNN) and Instance-Aware CNN (IA-CNN). The TL-CNN produces a segmentation that corresponds to the center of each text line. The IA-CNN, by taking the input as one of the text center lines, produces a segmentation mask over that text line instance. This step is crucial, not only for the evaluation of detection performance, but also for combining text recognition into an end-to-end system, since the input to recognition is typically a single word or text line. In addition to the ability of decomposing each text block, this component also serves as a further step of removing false positives. Both sub-nets are able to further remove some negative detection. Some examples of the extracted instances from text block images can be seen in Fig. 5.

4.1. Architecture

The architecture of the instance segmentation network is shown in Fig. 6. There are two branches with shared CNN parameters except for the first convolutional layer and all the fully convolutional layers.

The left branch is the TL-CNN. The network has been



Conv	VGG-16
MaxUnpool	2,2
Full-Conv	512→256,3,3
Full-Conv	256→256,3,3
MaxUnpool	2,2
Full-Conv	256→128,3,3
Full-Conv	128→128,3,3
MaxUnpool	2,2
Full-Conv	128→64,3,3
Full-Conv	64→16,3,3
Full-Conv	16→2,1,1

Figure 6. The architecture visualization (left) and details (right) of the TL-CNN and IA-CNN. The TL-CNN (left branch) is for producing the instance level words center lines. The IA-CNN (right branch) is for producing each word instance segmentation once. Their inputs have different number of channels since the IA-CNN need a 4-channel tensor. We ignored the ReLU layer and batch normalization [15] for simplicity.

trained to embed the "instance" information since it has to figure out where the center of each word is. Here we only consider word instance with more than 2 characters, since they are more distinguishable. However, instead of giving a hard threshold of removing detected text line with less than 3 components [41], we let the network learn the features that correspond to a word instance. We hypothesize that such instance-aware features are complementary to features extracted by previous text block FCN. The features extracted from text block FCN are more like traditional "textness" features. These features only capture whether a given region looks like text, but no instance information is embedded, and thus might be misled by some background noise. In Fig. 8, we show several examples that the text block FCN easily predicts as positive but are rejected by the TL-CNN.

The right branch is the instance-aware segmentation (IA-CNN) branch, whose input is a 4 channel tensor with size $4 \times h \times w$. h and w correspond to the height and width of the input text block image, respectively. For the input tensor, the first 3 channels are R,G,B channels of the text block image. The 4th channel is the text center line channel corresponding to the instance we want the network to segment. Jointly, we can produce instance level segmentation with the two network in a cascaded fashion.

4.2. Pipeline

The pipeline of the proposed instance segmentation is shown in Fig. 7. It decomposes the hard task of instance segmentation into cascaded tasks. Given a cropped text block B from the original image generated by thresholding the text block FCN output. We generate a probability map B_L by TL-CNN. Each pixel p_l in B_L represents probability of whether it belongs to the center line of one text instance. Once obtaining the output, we simply threshold the output

Figure 8. Examples of images that text block FCN might fail to remove. Instead, our TL-CNN can remove them. We only show the output map from TL-CNN of the first image to save space. The output of others are similar (completely black). We cannot extract text lines from them and thus can remove them.

with T, which is chosen based on the evaluation that will be discussed later, and do a morphological closing operation followed by a connected component analysis on it. We extract each component C and do the following operations: (1) discard those components whose height and width are less than 0.1 of the height and width of the text block, respectively, (2) discard the text block from which the overall coverage of the extracted text lines to the oriented text block bounding box is less than 0.6.

The obtained text line connected components Cs are separated, and we obtain a series of text line image I_ls from it. Each image I_l has -128 value background and 128 value on the corresponding text line. This channel together with the original text block is padded into a 4-channel tensor. Then this tensor will be used as an input to the IA-CNN branch. The output is the single text line instance segmentation I_i which corresponds to the input text center line image. Note that, when two extracted instances have high overlap, we need to keep the union of them as a post processing step. Even seldom, this might happen when the TL-CNN outputs two disconnected lines for one instance. In such case, our pipeline will extract two text center lines and produce two instances. However, the proposed IA-CNN segmentation will know that they are actually pointing to the same instance and the outputs will have high overlap. This can be seen as a way of error handling in the IA-CNN part, which will be discussed later. After we finish merging such instances, we simply threshold each instance segmentation probability map I_i with 0.5 to obtain the bounding boxes.

Fig. 5 shows more results. In testing, for each cropped image patch, we resize its larger dimension to be in range of 100 to 150 pixel while keeping the aspect ratio unchanged. The range is chosen for two reasons: (1) Too small size of image will cause text line instances to be unclear and hard to separate. (2) Our FCN, which is initialized with VGG-16, has a receptive field larger than 200. So for each pixel in the output, it has all the context needed to decide "where

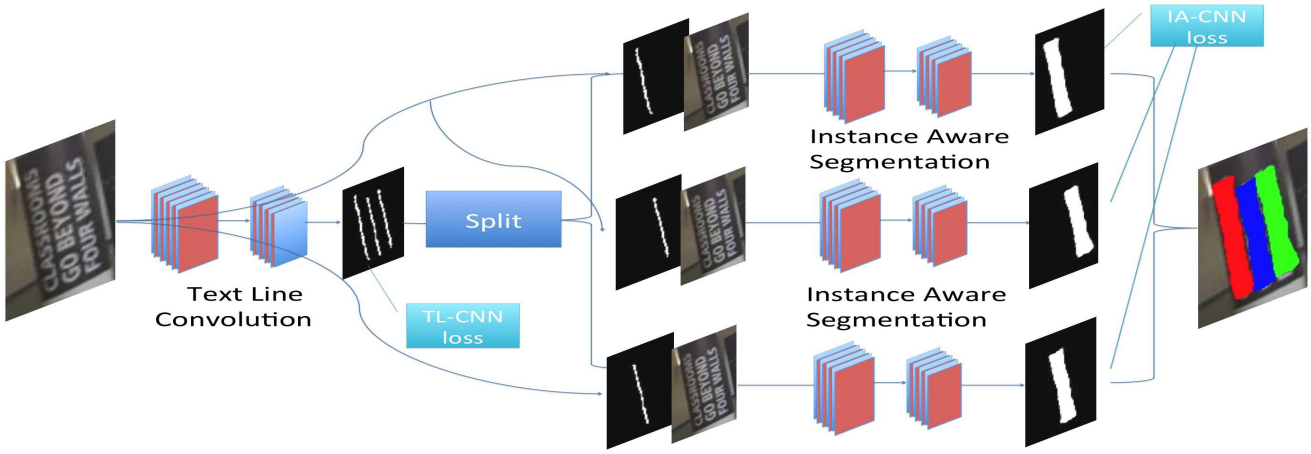


Figure 7. Our method of instance segmentation. It tries to decompose the hard task of instance segmentation into easier sub-tasks. First we use TL-CNN to generate text center line labeling. Then we simply decompose from the probability map a set of text lines. Each of these text lines will generate an input to the IA-CNN together with the original input image. Then we can produce an accurate instance segmentation of each text lines with arbitrary orientation.

one instance is". This is crucial for the TL-CNN to find each text line and give a good prediction.

4.3. Optimization

In training, we use a simple iterative scheme that iterate between training a TL-CNN and training a IA-CNN. For TL-CNN, all the convolutional layers are initialized with VGG-16 model [3]. For IA-CNN, the shared CNN parts are also initialized with VGG-16 model. All other layers are initialized with zero mean and standard deviation 0.1, Gaussian distribution. Inputs to both branches are normalized to have zero mean. Values of input range from -128 to 127. In optimization, we iteratively optimize the loss function 2.

$$\begin{aligned}\mathrm{NLL}(\theta_1,\theta_2,D) = -\alpha \times \sum_j \log P(Y=y^j|x^j,\theta_1) - \\ (1-\alpha)\times \sum_i \log P(Y=y^i|x^i,\theta_2)\end{aligned} \tag{2}$$

In the equation, α controls which branch of the network is in training. It can only equal to 0 or 1. x^i, y^i represent the prediction and groundtruth, respectively. Note that part of θ_1 and θ_2 are shared. When the loss of two branches become stable, we start to finetune the shared convolutional parts with smaller learning rate in the same iterative manner.

4.4. Error Handling in Instance-Aware Segmentation

Error handling is an important part of gaining robust performance for the instance-aware segmentation network. This is because in prediction the extracted text line could not be perfect. There might be variances of the line width, and the predicted line might not be centered well. The two end points of the line might have small offset from the center of the two sides. In order to make the model robust, we randomly add noise to the training data. Specifically, we randomly change the line width, and the location of the two end points of each line. Under certain constraints, we could obtain noisy training samples, and make the model more robust in testing. Fig. 9 illustrates how we create noisy training samples.

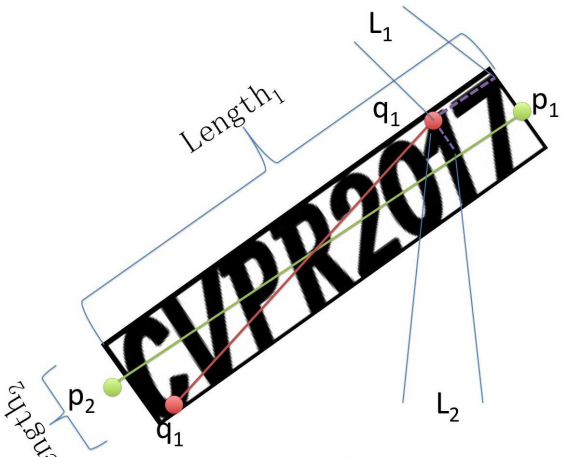
In the illustration image, L_2 represents the offset from p_1 to q_1 along the shorter side of the oriented word box, L_1 represents the offset from p_1 to q_1 along the longer side. The length of L_1 and L_2 and the width of text center line W is defined in equations below it.

The offset from p_2 to q_2 is processed in the same way as p_1 to q_1. By doing such randomization on the training set, we can train a more robust model. Several training examples are shown in Fig 10. Note that we also randomly sample negative text center lines that are on background region of a text block. For these text center lines, the corresponding ground truth are masks of all background label. The training set is from synthetic text blcok dataset and also from [9]. The line information could also be seen as a hint which tells the network where to find the instance, and thus it doesn't need to be perfect.

5. Experiments

5.1. Curved Text

Curved text typically causes a lot of troubles in scene text detection [41], and many works do not consider curved text since they have the assumption that text lines are straight.



$$L_1 = \alpha \times Length_1$$
$$L_2 = \beta \times Length_2$$
$$W = floor(\gamma)$$
$$\alpha \sim uniform(-0.5, 0.5)$$
$$\beta \sim uniform(0.0, 0.25)$$
$$\gamma \sim uniform(1, 4)$$

Figure 9. Illustration of creating noisy training set for IA-CNN network. The bounding box of the word is for illustration purposes. $Length_1$ and $Length_2$ are the length of longer side and shorter side or the word patch, respectively. p_1, p_2 and the corresponding green line represent the ground truth center line. q_1, q_2 and the corresponding red line represent the shifted noisy line. L_1, and L_2 are the offsets from p_1 to q_1 along the longer side and shorter side, respectively.

Figure 10. On the top two rows, we show some examples of the noisy training data for instance-aware segmentation. The input line image is shown in the image with a black line crossing the word. Here we only show one instance line per image for illustration purposes. On the last two rows, we show examples demonstrating the effectiveness of augmented the noisy training data. From left to right: Input text block image, instance results with model trained on good quality data, instance results with model trained on noisy data. More results about such error handling are in supplementary material.

However, many texts in signs or logos are curved and the ability to read curved text is important and will help many applications.

Our model can effectively capture the curved text by the joint power of TL-CNN and IA-CNN. In Fig 11, we show some curved text testing results on CUTE80 dataset [26].

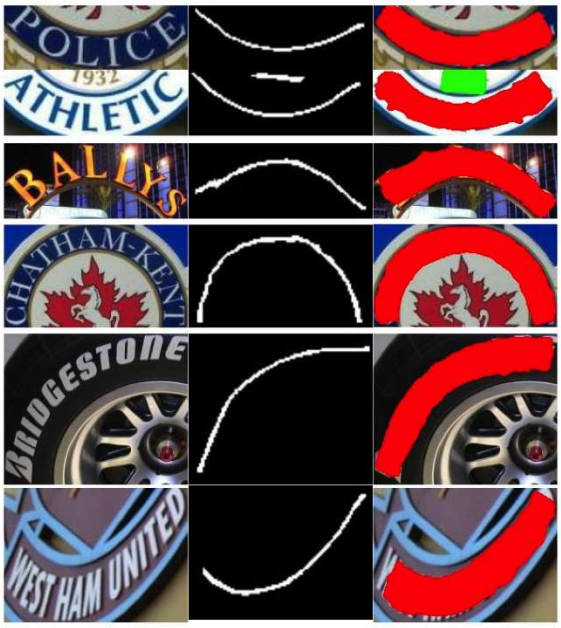


Figure 11. Results of our instance segmentation model on several curved text blocks. We can accurately capture curved text from a lot different scenes. From left to right: (1) The input image. (2) Text line captured by our text line model (3) Instance segmentation results on these curved data.

We could see that even with extreme curvature, our model can successfully estimate the text center line and further infer the instance mask for each text line.

Another surprising fact is that, we don't have any curved training data in our training set for both TL-CNN and IA-CNN. We hypothesize that this is because the model learned the intrinsic representation of an instance line which does not rely on whether it is straight or not.

5.2. Evaluation on Instance Segmentation

In order to evaluate the performance of the TL-CNN and the following IA-CNN module, we collect 1500 cropped images from IC13, and IC15 training set. They contain text blocks with 1-5 lines in each image.

Note that this evaluation is meaningful since the training data for the two networks are synthetic images, so these public training sets are used as validation purposes. Fig. 12 shows the precision and recall curve.

We can see that the choice of T only has little effect on the performance. The evaluation is based on the metric in [24]. Note that this evaluation framework artificially lowers recall. We have found that the relatively lower recall is generally caused by the fact our model usually predicts a line as one instance when there is less visual cue to separate out each word. This has little effect on end-to-end performance because current state-of-the-art recognition model [11] can directly read a line and thus will not hurt end-to-end scene

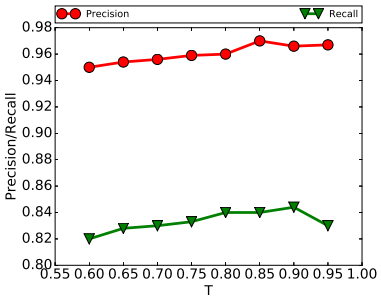


Figure 12. The precision (red) and recall (green) curve with respect to the choice of T on 1300 randomly cropped text block image from IC13, IC15 training set.

method	precision	recall	F-measure
Neumann[25]	73	65	69
Shi[29]	83	63	72
Bai et[2]	79	68	73
Zamberletti[38]	86	70	77
Tian[30]	85	76	80
Zhang[40]	88	74	80
Zhang[41]	89	78	83
tian[31]	**93**	**83**	**88**
Our model	**93**	79	85

Table 1. Localization performances (%) on ICDAR 2013 data sets. Bold number outperforms other methods.

text reading. We use 0.85 as our final choice of T in further evaluations.

We thoroughly evaluate our algorithm on four widely used benchmark datasets: IC13-focused text detection, IC15-scene text detection, CUTE80 and SVT. We choose them based on two criteria: they are widely used for evaluation and comparison or interesting for practical applications. We give a brief description and performance comparison of each dataset separately.

5.3. ICDAR 2013

ICDAR 2013 dataset is probably the most widely used dataset. It contains 251 test images with a wide variety of diversity except that all the text lines are horizontal or near horizontal.

We evaluated them by submitting our results into the ICDAR system. The evaluation protocol is based on [24]. Results are shown in Table 1.

5.4. ICDAR 2015

ICDAR 2015 dataset is a relatively newly released dataset which contains 500 testing images. They were taken with cell phone, so motion blur is common in the dataset. The texts in it are with arbitrary orientation and it poses a great challenge to scene text detection algorithms. Note that, in order to remove detected Chinese, we further ran a binary English text nontext classifier on the extracted word patches to remove some false positives.

We also evaluated our algorithm in ICDAR system, and the results are shown in Table 2. Note that some results are from ICDAR website, so there is no reference for them yet.

method	precision	recall	F-measure
HUST	44	38	41
StradVision1	53	46	50
StradVision2	77	37	50
Zhang[41]	71	43	54
tian[31]	74	52	61
Our model	**76**	**54**	**63**

Table 2. Localization performances(%) on ICDAR 2015 data sets. Bold number outperforms other methods. Some methods do not have references.

5.5. Street View Text and CUTE80

SVT dataset [32] contains images taken from street view, and CUTE80 dataset[26] contains texts that are in curved shape. They represent interesting aspects of scene text detection, and are also highly application oriented. Both datasets have the problem that they are not fully-annotated. So here we only evaluate the recall of our method in the two datasets. The results are shown in Table 3.

method	recall
Jaderberg [17]	71
He [10]	75
Our model	**78**

method	recall
Tian [31]	60
He [10]	56
Risnumawan [26]	68
Our model	**73**

Table 3. Text Localization evaluation (%) on SVT (left) and CUTE80 (right) dataset. We only evaluate recall on these datasets because they are only partially annotated.

5.6. Limitation

The proposed algorithm can handle text detection in a lot of different and challenging scenes. However, for some certain cases our current framework will fail. Fig. 13 shows some failing results. Extremely low contrast, too blurry text or text lines with scattered characters will cause problem in our framework.

Figure 13. Example images that our algorithm fail to detect correctly. Blue rectangles mean that we fail to localize the texts.

6. Conclusion

In this paper, we present a novel algorithm for scene text detection. We combined a mutli-scale FCN with a novel, cascade-style instance segmentation for end-to-end scene text detection, and achieved state-of-the-art results in benchmark datasets. We demonstrated that instance segmentation, which is gaining an increasing attention in computer vision community, is also helpful for end-to-end text reading systems.

7. Acknowledgement

This work was supported by NSF grant CCF 1317560 and a hardware grant from NVIDIA.

References

[1] ICDAR Robust Reading Competition. http://rrc.cvc.uab.es/.

[2] B. Bai, F. Yin, and C. L. Liu. Scene text localization using gradient local correlation. In *Document Analysis and Recognition (ICDAR), International Conference on*, pages 1380–1384. IEEE, 2013.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014.

[4] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, June 2016.

[5] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, volume 2, pages II–366. IEEE, 2004.

[6] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, June 2016.

[7] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, pages 2963–2970. IEEE, 2010.

[8] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[9] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference o*, June 2016.

[10] D. He, X. Yang, Z. Zhou, D. Kifer, and L. Giles. Aggregating local context for accurate scene text detection. In *Asian Conference on Computer Vision*, pages 91–105, 2016.

[11] P. He, W. Huang, Y. Qiao, C. Loy, and X. Tang. Reading scene text in deep convolutional sequences. In *AAAI Conference on Artificial Intelligence*. 2016.

[12] T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. *arXiv preprint arXiv:1603.09423*, 2016.

[13] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1241–1248, 2013.

[14] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolution neural network induced mser trees. In *Proceedings of the 11th European Conference on Computer Vision*, pages 497–511. Springer, 2014.

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[16] M. jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

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

[18] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In *Proceedings of the 11th European Conference on Computer Vision*, pages 512–528. Springer, 2014.

[19] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *AISTATS*, volume 2, page 6, 2015.

[20] K. Li, B. Hariharan, and J. Malik. Iterative instance segmentation. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, June 2016.

[21] X. Liang, Y. Wei, X. Shen, J. Yang, L. Lin, and S. Yan. Proposal-free network for instance-level object segmentation. *arXiv preprint arXiv:1509.02636*, 2015.

[22] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015.

[23] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, pages 3431–3440, 2015.

[24] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *Document Analysis and Recognition (ICDAR), International Conference on*, page 682. IEEE, 2003.

[25] L. Neumann and J. Matas. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, pages 3538–3545. IEEE, 2012.

[26] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014.

[27] B. Romera-Paredes and P. H. Torr. Recurrent instance segmentation. *arXiv preprint arXiv:1511.08250*, 2015.

[28] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016.

[29] C. Shi, C. Wang, B. Xiao, Y. Zhang, and S. Gao. Scene text detection using graph model built upon maximally stable extremal regions. *Pattern recognition letters*, 34(2):107–116, 2013.

[30] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan. Text flow: A unified text detection system in natural scene images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4651–4659, 2015.

[31] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *Proceedings of the 11th European Conference on Computer Vision*, pages 56–72. Springer, 2016.

[32] K. Wang and S. Belongie. Word spotting in the wild. In *Proceedings of the 11th European Conference on Computer Vision*, pages 591–604. Springer-Verlag, 2010.

[33] 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*, pages 3304–3308. IEEE, 2012.

[34] C. Yao, X. Bai, and W. Liu. A unified framework for multi-oriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749, 2014.

[35] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, pages 1083–1090. IEEE, 2012.

[36] X.-C. Yin, W.-Y. Pei, J. Zhang, and H.-W. Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1930–1937, 2015.

[37] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR 2016*, 2016.

[38] A. Zamberletti, L. Noce, and I. Gallo. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In *Computer Vision-ACCV 2014 Workshops*, pages 91–105. Springer, 2014.

[39] Z. Zhang, S. Fidler, and R. Urtasun. Instance-level segmentation for autonomous driving with deep densely connected mrfs. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, June 2016.

[40] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, June 2015.

[41] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In *Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on*, June 2016.

[42] A. Zhu, R. Gao, and S. Uchida. Could scene context be beneficial for scene text detection? *Pattern Recognition*, 58:204–215, 2016.

[43] S. Zhu and R. Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–632, 2016.