

Accurate Text Localization in Natural Image with Cascaded Convolutional Text Network

Tong He^{1, 2}, Weilin Huang^{1, 3}, Yu Qiao^{1, 3}, and Jian Yao²

¹ Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

²School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China

³Department of Information Engineering, The Chinese University of Hong Kong, Shatin, Hongkong

{tong.he,wl.huang,yu.qiao}@siat.ac.cn; jian.yao@whu.edu.cn

Abstract

We introduce a new top-down pipeline for scene text detection. We propose a novel Cascaded Convolutional Text Network (CCTN) that joints two customized convolutional networks for coarse-to-fine text localization. The CCTN fast detects text regions roughly from a low-resolution image, and then accurately localizes text lines from each enlarged region. We cast previous character based detection into direct text region estimation, avoiding multiple bottom-up post-processing steps. It exhibits surprising robustness and discriminative power by considering whole text region as detection object which provides strong semantic information. We customize convolutional network by developing rectangle convolutions and multiple in-network fusions. This enables it to handle multi-shape and multi-scale text efficiently. Furthermore, the CCTN is computationally efficient by sharing convolutional computations, and high-level property allows it to be invariant to various languages and multiple orientations. It achieves 0.84 and 0.86 F-measures on the ICDAR 2011 and ICDAR 2013, delivering substantial improvements over state-of-the-art results [23, 1].

1. Introduction

Text detection and recognition in natural images have recently gained increasing attention from computer vision community, as substantiated by recent work [8, 7, 30, 9, 4, 1, 31, 23, 34, 32]. This paper focuses on text detection sub task. Though tremendous efforts have been devoted to improving its performance, accurately locating text in unconstrained environments is still extremely challenging, due to significant diversity of text patterns and highly complicated background. For example, text can be in very small size, low quality, or low contrast, and even regular ones can be distorted considerably by numerous real-world affects, such as perspective transform, strong lighting, large-scale occlu-



Figure 1. Two-step coarse-to-fine text localization results by the proposed Cascaded Convolutional Text Network (CCTN). A coarse text network detects text regions (which may include multiple or single text lines) from an image, while a fine text network further refines the detected regions by accurately localizing each individual text line. The ORANGE bounding box indicates a detected region by the coarse text network. We have two options for each text region: (i) directly output the bounding box as a final detection (solid ORANGE); (ii) refine the detected region by the fine text network (dashed ORANGE), and generate an accurate location for each text line (RED solid central line). The refined regions may include multiple text lines or an ambiguous text line (e.g., very small-scale text).

sion, or blurring. These pose fundamental challenges of this task where correct character detection is difficult, and multiple post-processing steps that group character candidates into text lines are highly complicated and unreliable.

Most existing scene text detection methods are built on bottom-up strategy that sequentially processes: *stroke or character candidate detection, filtering, text line construction* and *classification* [9, 31, 7, 32, 6, 17]. These approaches commonly suffer from a number of limitations. First, detecting strokes or characters by exploring low-level image cues is not robust, e.g., by using the widely-used SWT [3] or MSERs [16] detector. Second, it is easy to generate a large amount of non-text candidates, which can be many orders of magnitude more than the true text candidates. This makes it extremely challenging to filter out these non-text false detections robustly by using a character level classifier. Third, grouping the retained character candidates into text lines is complicated. It often explores a number of low-level heuristic properties and geometric information, and also requires manually setting a number of low-level grouping rules. Fourth, as indicated in [23], the bottom-up strategy is not reliable, where error in each step can be accumulated sequentially.

These limitations severely harm the performance of current systems, and recent efforts have been given to tackle one or some of them. Building on recent advances of deep learning models for image representation, Jaderberg *et al.* [9] and Wang *et al.* [26] designed their Convolutional Neural Networks (CNN) to detect character information in a sliding window fashion. The CNN models were also applied for filtering out non-character candidates by Huang *et al.* [7], or for assigning character confident scores in [23]. These approaches achieved the state-of-the-art performance by leveraging strong representation-capability of the CNN. However, these approaches commonly employ the CNN models for character-level classification, which is neither robust nor discriminative. It is more principled to jointly identify a group of text strings (e.g. a text region) where local neighbouring text is greatly helpful to make a reliable decision. Therefore, current text models did not fully explore excellent potential of the CNN as exhibited in generic object detection or segmentation.

In this work, we fill this gap by introducing CNN to direct text region estimation. Conventional CNN architecture includes a number of stacked convolutional layers followed by several fully-connected (FC) layers. The FC representation, which discards spatial information, is particularly efficient for classification task, but is not effective for localization task. Long *et al.* replaced the FC connections with 1×1 convolutions, and achieved fully convolutional networks for semantic segmentation [14]. This inspires us to utilize the fully convolutional properties which preserve coarse spatial information of the image. In the convolutional architecture, pooling operation can reduce computational complexity, and also introduce invariance to local transformations. However, these advantages come at the price of reduced spatial accuracy, which is of particular importance for ac-

curate text localization. To overcome these limitations, we propose a two-stage coarse-to-fine pipeline that tailors general convolutional networks towards our problem.

In this work, we tackle problem of text localization from an alternative perspective. We propose a Cascaded Convolutional Text Network (CCTN) for direct text region estimation. We develop an efficient top-to-down pipeline that localize text in a coarse-to-fine manner (see Fig. 1), departing from previous approaches building on character based detection. It makes the following major contributions.

First, this is the first attempt to cast previous character based detection approaches into direct text region estimation. It provides surprising robustness and discriminative power by considering whole text region as detection object. Our approach dose not include any post-processing, providing a significant simplification of existing methods.

Second, we customize general convolutional networks towards our text task. We design rectangle convolutions and multiple in-network fusions to handle multi-shape and multi-scale text lines. This allows our network to work reliably on a single-scale input image, compared to the 24 scales used in [34]. Besides, our fully convolutional design further reduces computational complexity.

Third, we develop a coarse-to-fine strategy which improves localization accuracy lost in pooling and up-sampling operations. We design a fine text network that estimates locations of central line and text line area. It separates text regions into text lines accurately, and also reduces false region detections, as shown in Fig. 2.

Fourth, our CCTN computes high-level deep features, giving excellent generalization ability. It achieves promising results on the standard ICDAR 2011 and 2013 datasets, with F-measure of 0.84 and 0.86, outperforming previous methods substantially. It also obtains excellent performance on the multilingual and multi-orientation MSRA-TD500 [29], by just training on English text.

2. Related Work

There are two groups of methods for text detection and localization in natural images: connected-component and sliding-window based approaches. The connected-component approaches detect text information at pixel-level by using a fast low-level detector, and then group the detected pixels into text component candidates. While the sliding-window methods scan the image densely by using a multi-scale sub-window with a pre-designed classifier.

The connected-component approaches have recently achieved promising performance [32, 7, 31, 6, 11, 13, 29, 3]. Among them, the Stroke Width Transform (SWT) [3] and Maximally Stable Extremal Regions (MSERs) [16] are two representative low-level methods for detecting text component candidates. The MSERs algorithm is powerful to detect challenging text patterns, resulting in a good re-

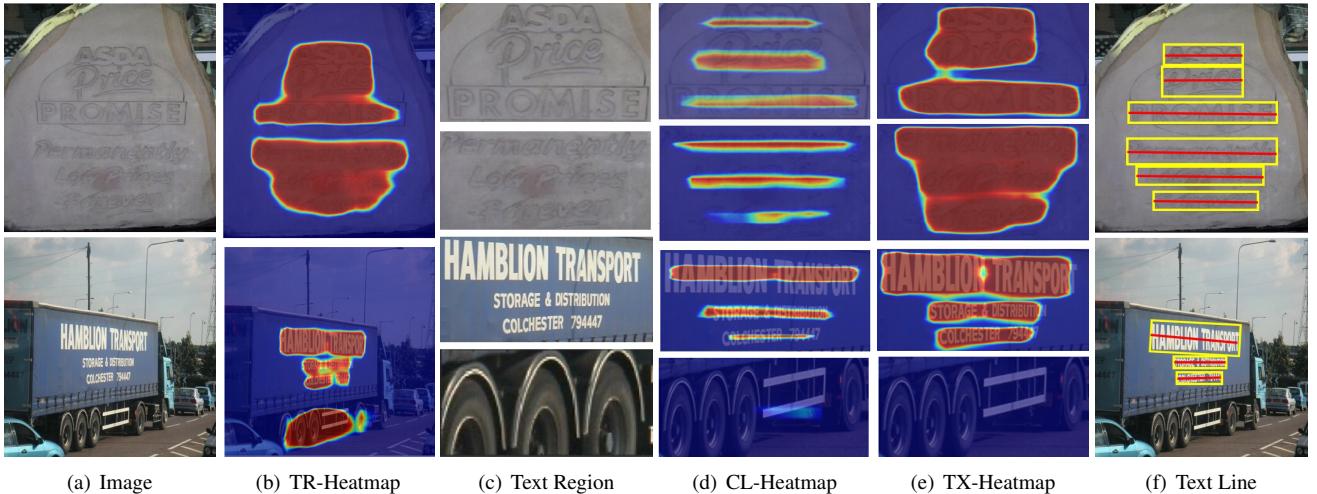


Figure 2. Pipeline of the Cascaded Convolutional Text Network (CCTN), which includes a coarse text network and a fine text network. (a) The input image (resized to 500×500); (b) The coarse text network detects text regions roughly by generating a text region heat-map (TR-Heatmap); (c) The detected regions are cropped out and enlarged to 500×500 ; (d-e) The fine text network refines them, and generate a central line area heat-map (CT-Heatmap) and a text line area heat-map (TL-Heatmap) for each region; (f) Final detection results.

call in character detection. This leads to a number of high-performance systems [32, 7, 11]. Recent efforts on developing the low-level text detector include Stroke Feature Transform (SFT) [6], Charcterness [13], and EdgeBox [35, 8]. Generally, the connected-component approaches based on these low-level detectors have a great advantage in speed by tracking image pixels in one pass computation. However, these detectors often generate a large amount of non-text components due to their low-level nature, raising main difficulties on filtering these non-text components, and grouping components into text lines. Consequently, they often require a number of bottom-up post-processing steps to reach a good performance. In fact, developing these post-processing methods is difficult itself. Previous methods generally develop a number of heuristic rules by exploring various low-level image properties, making the whole systems highly complicated and unreliable.

The sliding-window based methods have been developed rapidly [9, 23, 34, 24, 25, 26, 2]. One obvious advantage is that it computes global feature from a scanned window, so that the feature is invariant to various low-level distortions or transformations. However, they have a number of drawbacks. (i) A major challenge is the high computational cost. It requires multi-scale windows to handle various font sizes of text, resulting in a large number of the scanning windows. (ii) Designing a discriminative feature and training a powerful text/non-text classifier are difficult. (iii) Existing approaches are mostly built on character level detection, which is not robust and unreliable. They also require a number of bottom-up steps to refine detection results, such as identifying and grouping character candidates into text lines, increasing complexity of the systems considerably

[9, 23, 26, 2].

Current leading performance are achieved by incorporating CNN models to improve one or multiple components of the systems [8, 7, 5, 34, 23, 9, 26]. CNN is powerful to compute a high-level image feature, which is particularly efficient for classification task. Therefore, most of current approaches apply a CNN model for character/non-character classification. For example, Huang *et al.* [7] and He *et al.* [5] both exploited a CNN classifier to filter out non-character MSERs components, while sliding-window approaches built on a CNN character classifier were developed in [9, 26]. Although these CNN models have largely improved the performance of previous manually-designed features, the potential of CNN is not fully explored by these methods. Applying a CNN model for character classification is unreliable and inefficient. We customize general CNN towards our text localization task, resulting in a much higher accuracy.

Recently, Tian *et al.* proposed TextFlow, which simplifies multiple post-processing steps by utilizing a minimum cost flow network [23]. They applied cascade boosting algorithm with a number of manually-designed features for detecting character candidates. Then a trained CNN was used to assign a score to each character candidate. This approach is still built on character-level detection and classification. Alternatively, our method discriminates text and background information using a whole text region, which is more robust and discriminative. Our work is also related to Zhang *et al.*'s work where a text line is discriminated with a group of neighboring characters [34]. They designed a number of low-level symmetry features for detecting text line components, and also proposed multiple

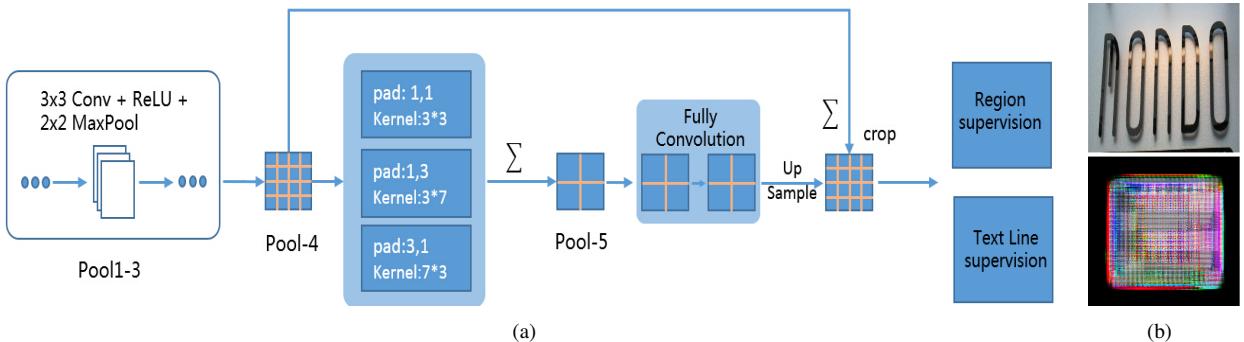


Figure 3. (a) Structure of the Convolutional Text Network, which is built on 16-layer VggNet [22]. (b) An resized 500×500 input image, and the actual receptive field of new *Pool-5*, which is computed as the response area in the input image by propagating the error of a single neuron in the new *Pool-5*.

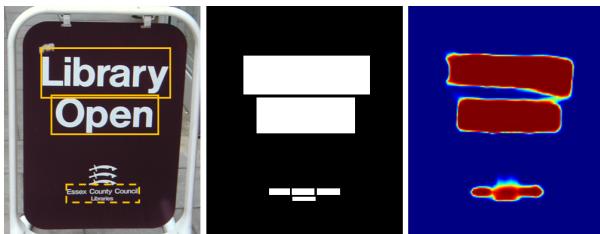


Figure 4. Text region mask (middle) and heat-map (right).

post-processing approaches with a number of heuristic rules and low-level properties to group the detected components into text lines, and post refine final results. Besides, the symmetry features are difficult to be generalized to oriented and multi-scale text lines. Here, we provide a more efficient approach that generalizes better.

3. Cascaded Convolutional Text Network

In this section, we present details of the proposed Cascaded Convolutional Text Networks (CCTN). It includes a coarse text network for rough text region detection from the whole image, and a fine text network used to accurately estimate a central line and a fine-grained region for each text line within an enlarged region. Our pipeline is presented in Fig. 2. The coarse network directly outputs a per-pixel heat-map, densely indicating the location and probability of text information. The fine network generates two such heat-maps for each cropped region. One heat-map presents the location of central line for each text line, and the other one gives separated areas of the text lines. In this work we show effectiveness of exploring convolutional networks to directly output accurate text regions, departing from previous approaches that apply a CNN classifier in a sliding window fashion for text detection, which require a number of complicated post-processing steps.

3.1. Coarse Text Network

The goal of our coarse network is to fast locate coarse text regions through a whole low-resolution image, with powerful discriminative ability and strong robustness against complicated background information. Most previous CNN based approaches for text localization train a character level CNN classifier and apply it for scanning an image densely with multi-scale sliding windows. They generate a corresponding heat-map that indicates the probabilities and locations of the text [9, 26]. These approaches are built on character level detection, which suffers from two limitations. First, it is extremely difficult to robustly distinguish isolated characters from complicated background outliers, imposing a large number of false detections, as shown in Fig. 5. Second, it requires a number of complicated post-processing steps to remove the false detections, and to bottom up the detected characters into text lines, making their systems highly complicated. Third, it is difficult to handle large diversity of the font sizes efficiently, and exploring multi-scale windows largely increases computational demands. To circumvent these problems, we present a new convolutional architecture that directly outputs a reliable text region heat-map. We cast conventional character classifier CNN into our text region estimation network. The output text heat-maps are shown in Fig. 2.

Our convolutional text network is built on the widely-used 16-layer VggNet architecture which has 16 convolutional layers separated by 5 pooling layers [22]. Inspired by design of fully convolutional networks (FCN) for semantic segmentation [14], we apply 1×1 convolutional layers for replacing the FC layers of original VggNet. This makes our network fully convolutional and capable of preserving rough spatial information. The structure of our convolutional text network is presented in Fig. 3, where we highlight our customized designs that make it adaptive to our text detection task.

Text Region Supervision. Our network is encouraged

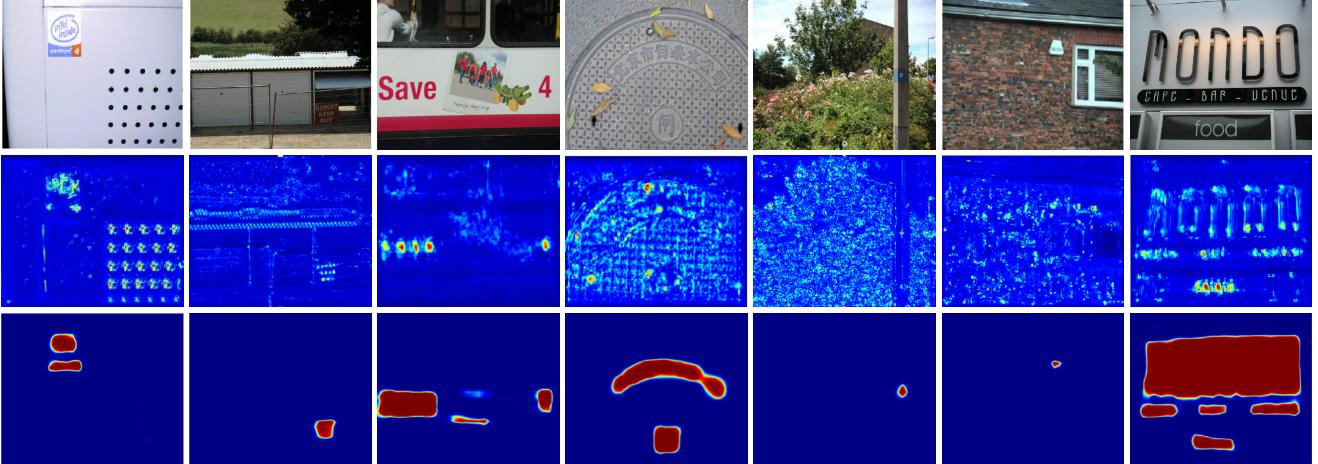


Figure 5. Comparisons of our heat-maps (Bottom) with those generated by CNN based character detector with a sliding-window (Middle). The CNN character/non-character classifier is provided by the authors of [9]. We select the best map of six scales for comparisons, while our coarse text network is just run on the single-scale image. Obviously, the results suggest that our method is strongly robust against cluttered background, and is also powerful to identify ambiguous text. It works reliably on both small-scale and large-scale text.

to directly output a text region estimation, e.g., a heat-map indicating the probabilities of text in all pixel locations. We consider all pixels within a text line area as text pixels. They include those pixels which are not exactly located on text strokes, but are inside the bounding box area of a text line. Because respective fields (RFs) of these pixels certainly include meaningful local text information around them. Thus we use a pixel-wise text region mask as our supervision information for region estimation. This information is obtained cheaply by just labeling all pixels within a text line bounding box. A exemplar region mask is presented in Fig. 4. The network is trained with pixel-wise softmax loss. This simple labeling scheme allows our detector to jointly consider local neighboring information, making it surprisingly advanced over low-level pixel based detectors, such as SWT and MSERs. It also significantly superior to those character based CNN detectors used in [9, 26], as compared in Fig. 5.

Text Rectangle Kernels. We consider a text line as our detection object. We found that appearance of the text lines has obvious characteristics. Their shapes are commonly in square (e.g., a word just including one or two characters), horizontal rectangle or vertical rectangle (e.g., oriented text). To handle such text-specific properties, we design three parallel convolutional layers with various kernel shapes to replace original three convolutional layers between the *Pool-4* and *Pool-5* layers. The three layers in original VggNet are sequentially connected by using a same 3×3 convolutional kernel. As shown in Fig. 3, sizes of the three kernels are set to 3×3 , 3×7 and 7×3 respectively in our text network. Differing from the original sequential architecture, we parallelize the three layers, and design different padding sizes for them, which allow them to output three equally-sized maps. This design allows the activations

in three layers to have their RFs in a wide range of shapes and aspect ratios, making them capable of detecting text in various shapes efficiently.

Multi-Layer Fusion. We design a two-step fusion strategy. In the first step, we fuse output maps of the designed three parallel layers into a single layer, by using element-wise summarization. Then the fused maps are further max-pooled with a 2×2 kernel to procure new *Pool-5* maps. This operation enlarges the RFs by 2×2 in this layer. To achieve multi-scale capability, a straightforward approach is to combine current feature maps with the output maps of previous layers, which capture more local fine-scale features by using smaller RFs. In the second step, to have a equal map size as the previous layer (e.g., the *Pool-4* maps), the *Pool-5* maps are first $2 \times$ -upsampled (after passing two 1×1 fully convolutional layers), and then are element-wisely combined with the *Pool-4* maps. Notice that this up-sampling operation does not change the sizes of RFs in *Pool-5*. In practice, in all our implementations we resize each input image to 500×500 . In our architecture, the actual RFs in the new *Pool-5* is 403×403 , which is able to cover most area of input image, as shown in Fig. 3 (b). The actual RFs is computed as the response area in the input image by propagating the error of a single neuron in the new *Pool-5*. After the two-step combination, the feature maps in the new *Pool-5* are able to capture both multi-shape (square, horizontal or vertical rectangle) and multi-scale text lines. *Therefore, our customized text network is strongly robust against cluttered background, and is powerful to identify ambiguous text. It works reliably on both small-scale and large-scale text with a single-scale input, successfully avoiding expensive computational cost raised by exploring multi-scale sliding-windows.* Examples and

comparisons are presented in Fig. 5.

Text Region/ Text Line Extraction. We observed that some large-scale text lines which do not have closed neighbouring lines can be localized accurately in the coarse heat-map. However, the estimated areas of multiple closing text lines or small-scale ones are easily merged and generally inaccurate. This may due to the max-pooling operation and low-resolution image used, as shown in Fig. 4 (c). For those accurately detected text lines, we directly extract their bounding boxes from the coarse heat-map. The ambiguous regions are cropped out and further refined by the proposed fine network, as described in Fig. 1. We design a simple rule that extracts the text lines or text regions as follow: (1) We binary the heat-map with a threshold of 0.3. (2) We compute area ratio and borderline ratio. (3) We directly output a text line bounding box if the area ratio > 0.7 and borderline ratio > 5 , while leaving the others (text regions) for a further refinement. (4) We crop the remained text regions. A text region is cropped by enlarging it in square, with a side length of $1.2 \times$ longer side.

Our text network directly outputs a pixel-wise heat-map which labels the input image densely. It essentially works in a sliding window fashion. So we compare our results to those of Jaderberg *et al.*'s method [9] in Fig. 5. They also used the sliding-window approach with a character/non-character CNN classifier for generating the text heat-maps. As can been seen, our heat-maps are surprisingly better than those generated by character based CNN detector. Our convolutional text network is very robust to complicated background outliers, and strongly discriminate ambiguous text regions. It is able to handle multi-scale and multi-shape text reliably in a single scale. The excellent performance are mainly ascribed to our region based detection strategy and multiple in-network fusions. They allow our detector to evaluate each single pixel reliably by inherently incorporating multi-scale and multi-shape image content. Furthermore, our method does not require any additional post-processing steps to bottom up character candidates into text lines. This grouping step is difficult when it is implemented on the character heat-maps, where many false detections exist, as shown in Fig. 5.

Despite with these appealing properties, the coarse network can not provide accurate bounding boxes for all text lines. The localization accuracy is significantly reduced by multiple pooling and sub-sampling operations, so that a further refinement is necessary.

3.2. Fine Text Network

Our coarse network is able to detect text information in rough regions reliably with few false detection. Our ultimate goal is to find accurate text area in text line or word level. Although the detected coarse regions can be used to find final detections of some isolate large-scale text lines,



Figure 6. Central line area (middle) and text line area (right).

localization accuracies on small-scale text fonts or multiple closed text lines are significantly reduced by using the low-resolution image (e.g. 500×500 in our experiments) and multiple pooling operations. While these strategies are indeed beneficial for speed and robustness by allowing larger RFs to scan image content, it is principled to develop a fine detection network to refine the coarse detection results (e.g., the cropped text regions). The goal of our fine network is to correctly separate all isolated text lines within a text region, and also remove false detections in the coarse detections (despite the number is small).

We look for a discriminative property that can separate multiple closed text lines correctly. A straightforward approach is to use text area within each text line bounding box, which is ideally separable. In practice, it may not work well in two situations. First, the text line areas are easily overlapped between two closed yet oriented text lines. Second, even the areas of two text lines are clearly separable in the original image, they may be confused in the estimated heat-map by multiple pooling and up-sampling operations, as shown in Fig. 4. Thus purely estimating the area of a text line is not reliable, and a more fine-grained text line localization is required. Therefore, we enlarge each detected text region to refine text line locations by using a fine text network. The fine text network has a same architecture as the coarse text network (in Fig. 3 (a)), but with a different output layer which we will describe bellow. Since text lines in an enlarged region are generally in large scale, e.g., covering most spatial area of a text region, the RFs of the network should be able to cover full area of an input region. Thus we resize each text region into a fix size of 500×500 with 50-pixel (zero values) padding on each side, so that full image content is constrained into the central area of 400×400 , which is fully covered by a RFs of 403×403 .

Text Line Supervision. Motivated from the symmetry-based detector in [34], we found that a central line is more reliable to define an unique text line, providing a better choice for our design. We aim to locate each text line separately by using a bounding box, so that we also need to measure the height of a text line if we know the location of its central line. With these considerations, we design a fine text network that is able to jointly estimate both *central line area* and *text line area*, as shown in Fig. 6. The *central line area* is defined by using a Gaussian distribution with its maximum value (e.g., 1) in the middle of a bounding box, which is decreased to 0 in a radius of $0.25 \times H$, where H is the height of bounding box. We use half height of the bounding box as the height of central line area. This

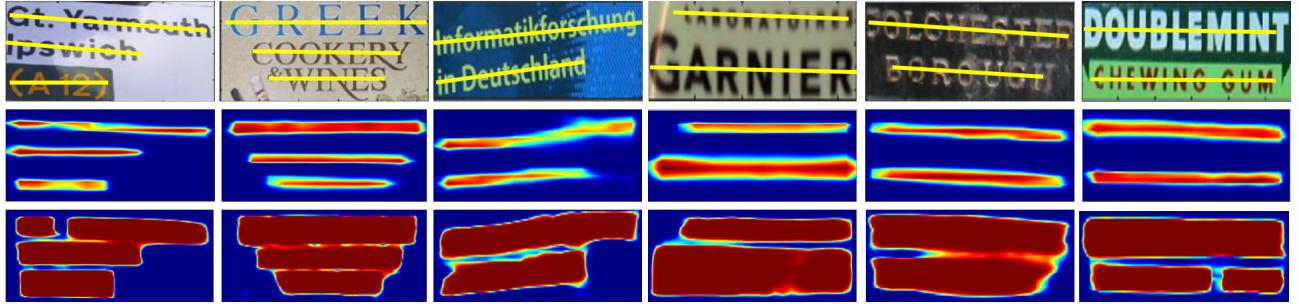


Figure 7. Results of the fine text network. **Top Row:** image and central line detection; **Middle Row:** central line area heat-map (CT-Heatmap); **Bottom Row:** text line area heat-map (TL-Heatmap).

design allows it to include both central line location and height information of a text line, leading to a better separation between closed text lines than using a full height. The *text line area* is all pixels within its bounding box. It is used to measure the height of a text line. Therefore, the fine text network is trained with both supervision masks by jointly computing per-pixel cross-entropy and softmax loss; and it should be able to estimate both the central line area and text line area heat-maps given an cropped region. The output heat-maps are presented in Fig. 2 (d) and (e).

Text Line Extraction For accurately locating a detected text line, we compute the central line and height of bounding box by using both central line area heat-map (CL-Heatmap) and text line area heat-map (TL-Heatmap). (1) We binary both heat-maps by using a threshold of 0.5. (2) Then we compute a minimum area rectangle (MAR) from the binarized CL-Heatmap, and compute its central line location C_{CL} , and height, H_{CL} . (3) We define a preliminary detected bounding box as $(C_{CL}, H_{CL} \times 2)$. (4) Similarly, we compute the MAR from the binarized TL-Heatmap, and compute its top and bottom sides, T_{top} and T_{bottom} . (5) We refine the preliminary detected bounding box with T_{top} and T_{bottom} , and generate final bounding box, as shown in Fig. 2 (f). More results of the fine text network are presented in Fig. 7.

4. Experiment and Results

The proposed CCTN is evaluated on three benchmarks for text localization in natural image: ICDAR 2011 [20], ICDAR 2013 [12] and MSRA-TD500 [29].

4.1. Datasets and Experimental Setting

Benchmarks. The ICDAR 2011 [20] includes 229 and 255 images for training and testing, respectively. There are 299 training images and 233 test images in the ICDAR 2013 [12]. The images in both datasets are varied from 307×93 to 1280×960 . The MSRA-TD500 database [29] has 500 images containing multi-orientation text lines in different languages (e.g. Chinese, English or mixture of both). The

training set contains 300 images, and the rest is used for testing. We follow standard evaluation protocol of the ICDAR 2011 [20, 27] and ICDAR 2013 [12]. Evaluation on the MSRA-TD500 was originally proposed by [29], which are measured by minimum area bounding boxes.

Training Data. Our training samples were generated from the training sets of the ICDAR 2011 and USTB [31] databases. We randomly cropped image patches with a fixed size of 500×500 from the training images. Then to increase the number of the training samples and their diversities, the cropped patches were further implemented with multiple data augmentations by rotating, flipping, and adding random noise. We finally generate 6611 training samples in total. Notice that our training samples only include English text, some of which are oriented (from the USTB dataset). We used these samples to train coarse text network. For fine text networks, we further cropped square text regions from the 6611 training samples, and enlarged them to 500×500 .

We used the pre-trained 16-layer VggNet [22] as initialization of our networks. The newly-developed rectangle convolutional layers are initialized with a Gaussian distribution of mean 0 and standard deviation 0.01 in the training process. The learning rate and momentum are set to 10^{-10} and 0.99 respectively. The coarse text network was trained with 200K iterations, while the fine text network was fine-tuned on the coarse text networks with a further 100K iterations. In the test processing, each input image is resized into two scales. One scale is 500×500 . The other is to fix the long side to 500, while keeping shape ratio unchanged. Therefore, in our cascaded model, we use both scales for the coarse detection, and use a single-scale 500×500 for the fine network, due to our square region cropping scheme.

4.2. Evaluation on component text networks

We investigate the performance of individual convolutional text networks without the cascaded architecture. The experiments were conducted on the ICDAR 2011 dataset by using each individual model for text detection. The results, including running time of each component, are reported in



Figure 8. Full detection results on changeling examples.

Table 1. Evaluation on individual Coarse Text Network or Fine Text Network on the ICDAR 2011.

	<i>P</i>	<i>R</i>	<i>F</i>	<i>Time</i>
Coarse Text Network	0.62	0.43	0.50	0.5s
Fine Text Network	0.70	0.51	0.60	0.3s
CCTN	0.88	0.79	0.84	1.3s

Table 1, where *R*, *P* and *F* indicate Recall, Precision and F-measure, respectively. All models were implemented in Caffe framework[10] with Matlab by using a single GPU.

As can be found, each individual model can not obtain reasonable performance. Obviously, our coarse-to-fine strategy significantly improves the performance, indicating that our cascaded design is indeed beneficial. It is difficult for each individual model to find accurate locations of the text lines, especially for those small-scale or multiple closing text lines. The average running time of the individual model on a single-scale image (500×500) is about 0.3s. The total running time for the CCTN is about 1.3s, including 0.5s for the two-scale coarse network, and 0.8s for the single-scale fine network. The time by the fine network is increased due to multiple cropped regions generated from an image. The coarse network detect totally 537 text regions from all 255 text images, with *only 35 false detections in all images* (We consider an false detection as a region that dose not include any text information), compared to *1000~2000 per image* error detections reported in [32, 34]. This suggests that our detector is extremely ro-

bust against background noise and outliers by considering a whole text region as a detection object, as shown in the heat-maps in Fig. 5. These false detections can be further removed by the fine text networks, as shown in Fig. 2.

4.3. Evaluation on full text detection

The full text detection results on a number of challenging images are presented in Fig. 8. The results demonstrate high performance of our CCTN, which has powerful discriminative ability to detect extremely ambiguous text lines, with strong robustness against multiple text variations and significantly cluttered background.

We evaluate the CCTN on two benchmarks: ICDAR 2011 and ICDAR 2013. The performance is compared extensively with most recent results in Table 2 and 3. Our CCTN obtains the best results, with 0.84 and 0.86 F-measure on the ICDAR 2011 and 2013 datasets, surpassing all previous results compared by a large margin. In the ICDAR 2013, it outperforms the cloest TextFlow [23] substantially with a 6% improvement on F-measure. The large improvement mainly comes from significantly higher recall by our detector. This can be ascribed to our text region detection scheme which has strong capability for detecting highly ambiguous and low-quality text lines by jointly considering neighboring text strings, resulting in a more reliable detection, as shown in Fig. 1 and 8.

Although Zhang *et al.*'s approach also detect a group of characters, they may lost recall in the bottom-up steps which connect the detected discrete short lines into text-

Table 2. Experimental results on the ICDAR 2011 dataset.

Method	Year	<i>P</i>	<i>R</i>	<i>F</i>
CCTN	–	0.88	0.79	0.84
TextFlow <i>et al.</i> [23]	2015	0.86	0.76	0.81
Jaderberg <i>et al.</i> [8]	2015	-	-	0.81
Zhang <i>et al.</i> [34]	2015	0.84	0.76	0.80
MSERs-CNN [7]	2014	0.88	0.71	0.78
Yin <i>et al.</i> [32]	2014	0.86	0.68	0.76
Neumann & Matas [17]	2013	0.85	0.68	0.75
SFT-TCD [6]	2013	0.82	0.75	0.73
Shi <i>et al.</i> [21]	2013	0.83	0.63	0.72

Table 3. Experimental results on the ICDAR 2013 dataset. Our performance is compared to the last published results in [23].

Method	Year	<i>P</i>	<i>R</i>	<i>F</i>
CCTN	–	0.90	0.83	0.86
TextFlow <i>et al.</i> [23]	2015	0.85	0.76	0.80
Zhang <i>et al.</i> [34]	2015	0.88	0.74	0.80
Lu <i>et al.</i> [15]	2015	0.89	0.70	0.78
Neumann and Matas [18]	2015	0.82	0.72	0.77
FASTText [1]	2015	0.84	0.69	0.77
iwrr2014 [33]	2014	0.86	0.70	0.77
USTB TexStar [32]	2014	0.88	0.66	0.76
Text Spotter [19]	2012	0.88	0.65	0.75

line candidates. Furthermore, Zhang *et al.*’s detector builds on symmetry property of the text line, which is sensitive to text orientations. It require a large number of detector scales (e.g., 14 scales) to reach a good recall, which raises its computational cost substantially, e.g., resulting in about 60s per image. Alternatively, our top-to-down coarse-to-fine strategy provides an more efficient and accurate approach for this task.

Although we got the highest Precision (0.90), as discussed, our coarse detector generates a very small number of false positives, so that it should be expected a higher value. However, we observed that additional false detections may happen in the refinement stage when we implement the fine text network on the enlarged text regions. Our detector does not have any post-processing step for removing these false detections, which reduces its precision in certain degree. In fact, most current approaches include such a post-processing step to improve their performance, such as [34, 8, 31, 32, 6].

We further evaluate generalization ability of our CCTN to multi-language and multi-orientation text lines on the MSRA-TD500. We train the CCTN with all English text lines without using the training set of MSRA-TD500. Surprisingly, our method generalizes well to other languages, such as Chinese, and is invariant to oriented text lines, as shown in Fig. 8. As can be found in Table 4, our results are compared favorably against the best performance achieved

Table 4. Comparisons of CCTN with recent methods specifically designed for multi-language and multi-orientation text lines.

	Year	<i>R</i>	<i>P</i>	<i>F</i>
MSRA-TD500				
CCTN	–	0.65	0.79	0.71
Yin <i>et al.</i> [31]	2015	0.63	0.81	0.71
Yin <i>et al.</i> [32]	2014	0.61	0.71	0.66
Yao <i>et al.</i> [28]	2014	0.62	0.64	0.61
Kang <i>et al.</i> [11]	2014	0.62	0.71	0.66
Yao <i>et al.</i> [29]	2012	0.63	0.63	0.60
ICDAR2011				
CCTN	–	0.79	0.88	0.84
Yin <i>et al.</i> [31]	2015	0.66	0.84	0.74
ICDAR2013				
CCTN	–	0.83	0.90	0.86
Yin <i>et al.</i> [31]	2015	0.65	0.84	0.73

by Yin *et al.* [31], which is specifically designed for multi-language/orientation text lines detection. On the other hand, as compared to their results on the standard ICDAR 2011 and 2013 (reported in [31]), we got more than 10% improvements on both datasets. This indicates that our approach is more principled to solve fundamental problem of scene text detection.

5. Conclusion

We have presented a Cascaded Convolutional Text Network (CCTN) for text localization in natural image. We introduce a new top-to-down coarse-to-fine pipeline that casts previous character-based detection into direct text region estimation. It overcomes main limitations of previous bottom-up approaches, and achieves surprising robustness and discriminative power. We develop convolutional text network by designing multiple rectangle convolutions and multiple in-network fusions, which customizes general convolutional networks towards our task. The proposed CCTN is able to handle multi-shape and multi-scale text robustly, and is invariant to multi-language and multi-orientation text. It works reliably on both small-scale and large-scale text in single-scale images, making it computationally attractive with sharing convolutional computation. Extensive experimental results show that our method has achieved the state-of-the-art performance on three standard benchmarks.

References

- [1] M. Busta, L. Neumann, and J. Matas. Fastext: Efficient unconstrained scene text detector, 2015. In IEEE International Conference on Computer Vision (ICCV). 1, 9
- [2] X. Chen and A. Yuille. Detecting and reading text in natural scenes, 2004. In IEEE Computer Vision and Pattern Recognition (CVPR). 3

- [3] B. Epshtain, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform, 2010. In IEEE Computer Vision and Pattern Recognition (CVPR). 2
- [4] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang. Reading scene text in deep convolutional sequences, 2016. The 30th AAAI Conference on Artificial Intelligence (AAAI-16). 1
- [5] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural networks for scene text detection, 2015. arXiv:1510.03283. 3
- [6] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors, 2013. In IEEE International Conference on Computer Vision (ICCV). 2, 3, 9
- [7] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolutional neural networks induced mser trees, 2014. In European Conference on Computer Vision (ECCV). 1, 2, 3, 9
- [8] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision (IJCV)*, 2015. 1, 3, 9
- [9] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting, 2014. In European Conference on Computer Vision (ECCV). 1, 2, 3, 4, 5, 6
- [10] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding, 2014. In Proceedings of the ACM International Conference on Multimedia. 8
- [11] L. Kang, Y. Li, and D. Doermann. Orientation robust text line detection in natural images, 2014. In IEEE Computer Vision and Pattern Recognition (CVPR). 2, 3, 9
- [12] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras. Icdar 2013 robust reading competition, 2013. In International Conference on Document Analysis and Recognition (ICDAR). 7
- [13] Y. Li, W. Jia, C. Shen, and A. van den Hengel. Characterness: An indicator of text in the wild. *IEEE Trans. Image Processing (TIP)*, 23:1666–1677, 2014. 2, 3
- [14] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation, 2015. In IEEE Computer Vision and Pattern Recognition (CVPR). 2, 4
- [15] S. Lu, T. Chen, S. Tian, J.-H. Lim, and C.-L. Tan. Scene text extraction based on edges and support vector regression. *International Journal on Document Analysis and Recognition (IJDAR)*, 18(2):125–135, 2015. 9
- [16] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing (IVC)*, 22:761–767, 2004. 2
- [17] L. Neumann and J. Matas. On combining multiple segmentations in scene text recognition, 2013. In International Conference on Document Analysis and Recognition (ICDAR). 2, 9
- [18] L. Neumann and J. Matas. Efficient scene text localization and recognition with local character refinement., 2015. In International Conference on Document Analysis and Recognition (ICDAR). 9
- [19] L. Neumann and K. Matas. Real-time scene text localization and recognition, 2012. In IEEE Computer Vision and Pattern Recognition (CVPR). 9
- [20] A. Shahab, F. Shafait, and A. Dengel. Icdar 2011 robust reading competition challenge 2: Reading text in scene images, 2011. In International Conference on Document Analysis and Recognition (ICDAR). 7
- [21] 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*, 34:107–116, 2013. 9
- [22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. In International Conference on Learning Representation (ICLR). 4, 7
- [23] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. L. Tan. Text flow: A unified text detection system in natural scene images, 2015. In IEEE International Conference on Computer Vision (ICCV). 1, 2, 3, 8, 9
- [24] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition, 2011. In IEEE International Conference on Computer Vision (ICCV). 3
- [25] K. Wang and S. Belongie. Word spotting in the wild, 2010. In European Conference on Computer Vision (ECCV). 3
- [26] T. Wang, D. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks, 2012. In International Conference on Pattern Recognition (ICPR). 2, 3, 4, 5
- [27] C. Wolf and J. Jolion. Object count / area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis*, 8:280–296, 2006. 7
- [28] C. Yao, X. Bai, and W. Liu. A unified framework for multioriented text detection and recognition. *Image Processing, IEEE Transactions on*, 23(11):4737–4749, 2014. 9
- [29] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images, 2012. In IEEE Computer Vision and Pattern Recognition (CVPR). 2, 7, 9
- [30] Q. Ye and D. Doermann. Text detection and recognition in imagery: A survey. In *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 37:1480–1500, 2015. 1
- [31] X. C. Yin, W. Y. Pei, J. Zhang, and H. W. Hao. Multi-orientation scene text detection with adaptive clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 37:1930–1937, 2015. 1, 2, 7, 9
- [32] X. C. Yin, X. Yin, K. Huang, and H. W. Hao. Robust text detection in natural scene images. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 36:970–983, 2014. 1, 2, 3, 8, 9
- [33] A. Zamberletti, L. Noce, and I. Gallo. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions, 2014. In Workshop of Asian Conference on Computer Vision (ACCV). 9
- [34] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes, 2015. In IEEE Computer Vision and Pattern Recognition (CVPR). 1, 2, 3, 6, 8, 9
- [35] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges, 2014. In European Conference on Computer Vision (ECCV). 3