

Boosting up Scene Text Detectors with Guided CNN

Xiaoyu Yue¹

yuexiaoyu@sensetime.com

Zhanghui Kuang*¹

kuangzhanghui@sensetime.com

Zhaoyang Zhang¹

zhangzhaoyang@sensetime.com

Zhenfang Chen²

zfchen@cs.hku.hk

Pan He³

pan.he@ufl.edu

Yu Qiao⁴

yu.qiao@siat.ac.cn

Wei Zhang¹

wayne.zhang@sensetime.com

¹ SenseTime Research

² The University of Hong Kong

³ University of Florida

⁴ SIAT-Sensetime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

Abstract

Deep CNNs have achieved great success in text detection. Most of existing methods attempt to improve accuracy with sophisticated network design, while paying less attention on speed. In this paper, we propose a general framework for text detection called Guided CNN to achieve the two goals simultaneously. The proposed model consists of one guidance subnetwork, where a guidance mask is learned from the input image itself, and one primary text detector, where every convolution and non-linear operation are conducted only in the guidance mask. The guidance subnetwork filters out non-text regions coarsely, greatly reducing the computation complexity. At the same time, the primary text detector focuses on distinguishing between text and hard non-text regions and regressing text bounding boxes, achieving a better detection accuracy. A novel training strategy, called background-aware block-wise random synthesis, is proposed to further boost up the performance. We demonstrate that the proposed Guided CNN is not only effective but also efficient with two state-of-the-art methods, CTPN [1] and EAST [2], as backbones. On the challenging benchmark ICDAR 2013, it speeds up CTPN by 2.9 times on average, while improving the F-measure by 1.5%. On ICDAR 2015, it speeds up EAST by 2.0 times while improving the F-measure by 1.0%.

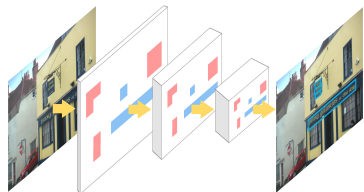
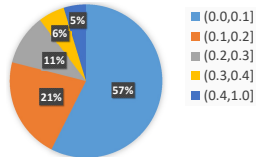
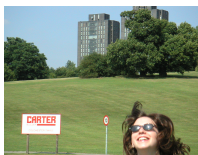


Figure 1: Illustration of guiding the primary text detector. Convolutions and non-linear operations are conducted only in the guidance mask indicated by the red and blue rectangles. The guidance mask (the blue) is expanded by background-aware block-wise random synthesis (the red) during training. When testing, the guidance mask is not expanded.

Figure 2: Text appears very sparsely in scene images. The left shows one example image. The right shows the text area ratio composition of ICDAR 2013 test set. Images with $(0\%,10\%]$, $(10\%,20\%]$, $(20\%,30\%]$, and $(30\%,40\%]$ text region account for 57%, 21%, 11%, and 6% respectively. Only 5% images have more than 40% text region.



1 Introduction

Reading text in natural images in the wild has attracted increasing attention recently, as shown in [3, 7, 10, 11, 13, 22, 27, 36, 37, 39, 40, 52, 55, 57, 62]. Large variance of text patterns and highly cluttered background pose the main challenge for text detection.

Inspired by recent advances in general object detection and semantic segmentation, such as Faster R-CNN [24], SSD [35] and FCN [15], recent text detection approaches [3, 4, 11, 12, 13, 14, 21, 22, 27, 36, 37, 52, 62] directly predict the bounding boxes of text, and improve text detection accuracy considerably. They focus on designing of better network architecture and objective function while paying less attention to the detection speed, which hinders deploying CNN-based text detection methods. Although there exist methods [4, 8, 29, 49, 62, 61] to accelerate general CNNs, they often achieve speedup at the cost of sacrificing accuracy, and are not tailored for text detection.

Text appears very sparsely in natural scene images, as shown in Figure 2. Our statistics on ICDAR 2013 test set show that only 9.7% of the whole region has text, and the rest is background. Except for those background regions nearby texts, background regions are useless for text detection and text recognition, as discussed in [11, 12], and thus are unnecessarily processed in text detectors. Filtering out them could potentially speed up text detectors. However, filtering out background without sacrificing accuracy itself is not trivial. First, this filtering operation should be fast with very little computation overhead. Second, we need to achieve high recall of text regions after filtering as the text filtered out cannot be detected.

To this end, in this paper, we make full use of the characteristic of sparseness of text in the wild, and propose a simple yet effective general framework for text detection, called Guided CNN, to speed up text detectors and improve their accuracy simultaneously. It mimics the procedure of the human visual system: first glancing at the full image and then focusing on salient regions. Specifically, it consists of two subnetworks: one guidance subnetwork, and one primary text detector. The guidance subnetwork is learned from the input images with text regions as supervision signals. It targets at finding text regions and filtering out most of non-text regions coarsely. The primary text detector uses any existing single forward CNN based text detection method as backbones, where every convolution and nonlinear operation are conducted only in the guidance mask predicted by the guidance subnetwork (See Figure 1). In this way, the primary text detector focuses on distinguishing between text and hard non-text, and predicting bounding boxes of text.

Improving speed and accuracy simultaneously is challenging. In order to achieve good

trade-off between guidance performance and computational load, we introduce a *context module* with pyramid pooling layers, to capture global context information in the guidance subnetwork. Training the primary text detector, only with the guidance mask derived from the ground truth, would lead to high false alarm rate, as it does not learn enough background patterns. To resolve this issue, we propose *background-aware block-wise random synthesis* to expand the guidance mask, by randomly synthesizing foreground blocks in background, with a fixed probability during training (See Figure 1). It can be interpreted as a special kind of *dropout*.

The proposed Guided CNN is a general framework, which can be easily plugged into existing single forward CNN based text detection methods. We conduct experiments on standard benchmarks, namely SWT [6], ICDAR 2011 [68], ICDAR 2013 [23] and ICDAR 2015 [24], with various backbones. The proposed method significantly speeds up the recent state-of-the-art methods [52, 64], while achieving a higher accuracy on average. Specifically, on ICDAR 2013, it speeds up CTPN [52] by 2.9 times, while improving accuracy by 1.5% on average. On ICDAR 2015, it speeds up EAST [64] by 2.0 times, while improving accuracy by 1.0%. It achieves an F-measure of 0.901 on ICDAR 2013, and 0.823 on ICDAR 2015 with a single scale test without bells and whistles.

2 Related Work

Existing methods in the literature for text detection can roughly be categorized into two approaches, namely traditional feature based approaches and deep learning based approaches.

Traditional feature based approaches. Hand-crafted features such as wavelet, gradient, and HOG, which are widely used in other computer vision applications, were introduced for text detection in [0, 28, 33, 47, 50]. Later on, Stroke Width Transform (SWT) was proposed to find the value of stroke width for each image pixel [8, 55]. It was extended by incorporating color cues of text pixels, leading to significantly enhanced performance on inter-component separation and intra-component connection [15]. Extremal Regions (ER) were employed to generate character candidates [39, 41], which was then improved by Maximally Stable Extremal Regions (MSERs) [16, 57, 58]. Different from all the above methods which use traditional hand-crafted features, the proposed general framework unifies feature extraction and text detection in a deep neural network, and both the feature extractor and detector are trained in an end-to-end fashion.

Deep learning based approaches. Earlier work [12, 17, 19, 53] in this stream detected text with CNNs on sliding windows or region proposals. They didn't share computation between sliding windows or region proposals, and thus led to slow training and testing. Inspired by the state-of-the-art object detection, and segmentation techniques, such as Faster R-CNN [24], SSD [35] and FCN [45], a number of text detection methods with a single forward pass were proposed. Yao *et al.* [56] and Zhang *et al.* [62] estimated text regions, individual characters and their relationship with FCN. Different from general objects, words tend to have large aspect ratios, and multiple orientations. To this end, TextBoxes [32] employed default boxes with big range aspect ratios and vertical offsets. CTPN [52] fixed the width of default boxes and only predicted height of each default box. Deep matching prior network [56] used quadrilateral sliding windows, and EAST [64] directly predicted quadrilateral shapes on convolutional feature maps. Ma *et al.* [57] proposed rotation proposals to detect arbitrary-oriented scene text while Jiang *et al.* [22] predicted axis-aligned bounding box and inclined bounding box as multi-task learning. *Different from all the above methods,*

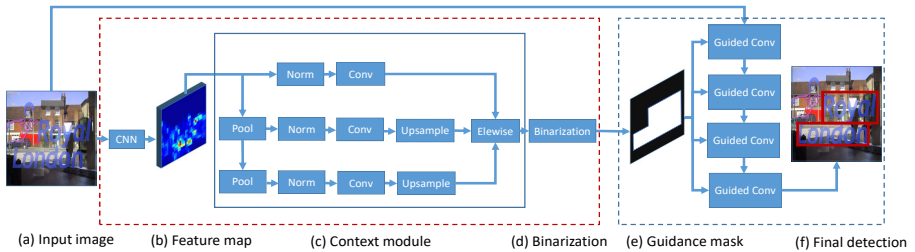


Figure 3: The Guided CNN framework for text detection. It consists of one guidance subnetwork indicated by the red dash rectangle, one primary text detector indicated by the blue dash rectangle. Given an input image (a), we first use CNN to get the convolutional feature map (b), then a context module (c) is used to capture multi-scale context and predict one guidance map, followed by binarizing (d) the guidance map to one guidance mask (e). Finally, the (expanded) guidance mask is used to guide the primary text detector to detect text (during training), resulting in the final detection (f).

the proposed *Guided CNN* is a general framework to improve both the accuracy and speed based on existing text detectors with a single forward pass.

Our work is related to the work of network acceleration. Linear decomposition was proposed to accelerate layers of CNNs [4, 18, 26], which was further extended by considering nonlinear approximation in [6]. Quantization, pruning, and Huffman coding were proposed to speed up CNNs and compress its weights in [8, 54]. Weights of CNNs were binarized while keeping the structure of network unchanged in [2, 6, 43, 49]. *Our proposed Guided CNN is orthogonal to those methods and can be combined with them for further acceleration.*

Our work is also related to cascaded CNNs. [61, 42, 59, 60] first used a sliding window method, a fully convolution network, or a region proposal network to generate face candidates, and then one by one fed each cropped face candidate into a series of cascaded CNNs for further filtering and bounding box refinement. [80] treated a single deep model as a cascade of several sub-models and processed harder regions progressively. *Different from the above methods, our Guided CNN has no object (i.e., text) candidates, but predicts a guidance mask which can be of any shape to tell the primary text detector where its convolution and nonlinear operations are conducted.*

3 Guided CNN

Our proposed Guided CNN is based on the observation that text appears very sparsely in natural scene images as shown in Figure 2. We target at improving the speed and accuracy of existing CNN base text detectors by filtering out most of the background. We achieve these two goals by proposing Guided CNN with a carefully-designed network architecture and its corresponding training procedure. An overview of our framework is illustrated in Figure 3. It consists of a guidance subnetwork, and a primary text detector. The guidance subnetwork is designed to coarsely predict the guidance mask, i.e., the text regions. The primary text detector conducts convolution and non-linear operations only in the guidance mask. Our Guided CNN is a general framework for text detection. Its primary text detector can use any existing single forward CNN based text detector as backbone. We denote Guided CNN with backbone using the nomenclature *Guided backbone*. Guided CNN with CTPN as backbone, for example, is called Guided CTPN.

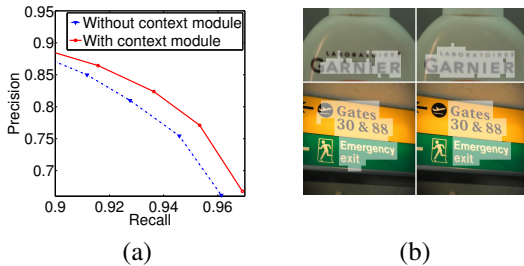


Figure 4: Quantitative and qualitative comparisons of the guidance subnetwork with and without the context module. (a) Recall precision curves of the guidance subnetwork with or without the context module. (b) The left and right columns show the input images superposed with the predicted guidance mask without and with the context module, respectively.

3.1 Guidance Subnetwork

The guidance subnetwork uses the pretrained PVANET [25] to extract the convolutional feature map as shown in Figure 3 (b). The final feature map, i.e., $conv5_4$, is of $\frac{1}{32}$ size of the input image. On the top of the map, we use a context module to extract multiple scale context and predict the guidance as shown in Figure 3 (c). Its output keeps $\frac{1}{32}$ of the input image, and is fed into a cross entropy loss layer during training. We generate the ground truth guidance mask using text bounding boxes of training data. We denote a rectangle by a four-tuple (r, c, h, w) that specifies its top-left corner (r, c) and its height and width (h, w) . For high recall, we set the ground truth label of the location (y, x) on the final prediction guidance map to be 1 as long as its corresponding rectangle $(32y - 16, 32x - 16, 32, 32)$ in the input image has intersection with any text bounding box. The predicted guidance map is binarized (Figure 3 (d)) with a threshold τ to a guidance mask as shown in Figure 3 (e).

Inspired by the recent advancement of image segmentation [24], we design the *context module* with pyramid pooling. As shown in Figure 3 (c), the context module consists of 3 levels. At the first level (the first row), feature vectors at each location are $l2$ normalized, followed by a convolutional layer to predict the guidance map. The latter two levels follow the same procedure, except that they use one average pooling layer with stride 2 to down-sample the feature map at the beginning and up-sample back after predicting the guidance map at the end. The normalization layer at each level is introduced to avoid the unbalance magnitude of the feature map. The predicted guidance maps of all levels are element-wise added together before feeding them to the loss layer.

The context module is essential to guarantee that the guidance subnetwork gets high recall and high precision. Figure 4 (a) compares the recall and precision of the guidance subnetwork with context module and those without it (i.e., the first level prediction) on ICDAR 2013. It has been shown that the guidance subnetwork with context module obtains much more accurate guidance prediction than that without it. Figure 4 (b) visualizes the guidance mask (binarized guidance map) predicted with context module and that without it. Obviously, context module can reduce false alarms greatly and increase recall. We emphasize that the cost of context module can be ignorable compared with that of the whole guidance subnetwork since the feature map size of context module is small enough (i.e., $\frac{1}{32}$ of the input image size).

3.2 Primary Text Detector

The primary text detector can use any existing single forward CNN based text detector as backbone. In existing text detectors, each convolution layer has only one input, convolution operation is done in the whole input feature map. Instead, in our primary text detector, each convolution layer has two inputs, namely, the input feature map and the guidance mask pre-

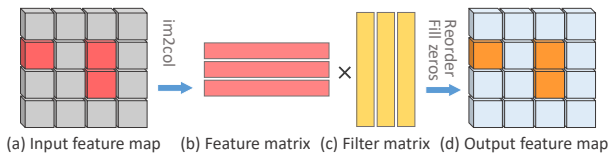


Figure 5: Illustration of efficient implementation of the guided convolution. The red indicates the feature vectors in the guidance mask.

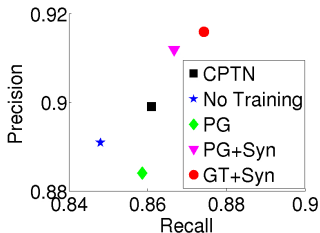


Figure 6: Performance improved by background-aware block-wise random synthesis. ■ indicates CTPN, ★ indicates applying the predicted guidance mask on CTPN during testing without retraining, ◆ indicates Guided CTPN trained with the predicted guidance mask, ▼ indicates Guided CTPN trained with the predicted guidance mask and random synthesis, and ● indicates Guided CTPN trained with the ground truth mask and random synthesis.

dicted by the guidance subnetwork, and convolution operation is done only in the guidance mask.

All the regions in the guidance mask are handled in the primary text detector with one pass but not one by one. To this end, we implement a new layer called *guided convolution* by modifying the “im2col” in caffe [24] as shown in Figure 5. Different from the original “im2col” which reorders the input feature map into a big matrix, our guided convolution layer reorders the input feature vectors only in the guidance mask (Figure 5 (a)) and generates a much smaller matrix (Figure 5 (b)). After matrix multiplication with the convolution filter matrix (Figure 5 (c)), its output is inserted back into a big output feature map with zero filling in background (Figure 5 (d)). From Figure 5, it can be concluded that Guided CNN can speed up its backbone by r times with ignoring the overhead of the guidance subnetwork if the guidance mask accounts for the ratio of $\frac{1}{r}$ over the whole feature map area.

3.3 Training with Background-aware Block-wise Random Synthesis

Training the primary text detector with guidance is not trivial. Directly training the primary text detector leads to a high false alarm rate. It is reasonable since the primary text detector fails to learn many background pattern variations during training.

To this end, we expand the predicted guidance mask by randomly synthesizing mask blocks with size 32×32 (since each location represents 32×32 rectangle in the input image) in the background with a probability p during training. We experimentally find that this simple approach is very effective, and can improve the final text detection performance greatly. When $p = 0$, the primary text detector is trained without randomly synthesis. When $p = 1$, it is degraded to its backbone without the guidance mask where convolution and other non-linear operations are conducted in the whole feature map.

Figure 6 compares the results on ICDAR 2013 with different training strategy. Firstly, we directly apply our predicted guidance mask on the original CTPN without retraining. It performs slightly worse than the original CTPN. It is straightforward since the contour patterns on the feature maps caused by predicted mask are never learned during training. Therefore, we then retrain Guided CTPN with the Predicted Guidance (PG). Its performance is improved but still worse than CTPN. Experimental results show that it has many false alarms. As we discussed before, totally dropping background patterns causes this issue. It is further validated in our third strategy, where the primary text detector is trained with both the predicted guidance mask and background-aware block-wise random Synthesis (Syn)

($p = 0.4$). Training with this strategy performs even better than the original CTPN, while speeding up considerably. Although the predicted guidance mask is with high recall, it might miss some small or big-scale texts. To resolve this issue, we instead choose the forth strategy, where the Ground Truth guidance mask (GT), together with random Synthesis (Syn), are used to train the primary text detector. We find that this training strategy further gives a better performance, in terms of recall and precision. The rest of experimental results is reported using this training strategy except as otherwise noted.

Advantages. Training the primary text detector with the ground truth guidance and random synthesis achieves two obvious advantages. First, the primary text detector can obtain better performance than its backbone while speeding up it greatly. Second, the training of guidance subnetwork, and that of the primary text detector are not correlated. Therefore, they can be trained individually and in parallel, leading to convenient deployment.

Connection with Dropout. One possible explanation for our background-aware block-wise random synthesis is to dropout [50] feature map with probability $1 - p$ in background. Following standard test procedure of dropout, we multiply the feature map in the primary text detector with a scale of p but not 0 in background regions during testing, the primary text detector can be further improved. For clarify, we called this testing method as *Guided backbone+*. For example, Guided CNN with CTPN as its backbone is called Guided CTPN+ if it follows the dropout test procedure when testing. We emphasize that Guided backbone+ would not accelerate its backbone. It can be used in the case when accuracy but not speed is most important.

The background-aware block-wise random synthesis is different from the standard dropout. First, the standard dropout drops feature activations independently while our method drop all activations of one block once. Second the standard dropout drops activation everywhere while our method only in the background (non masked regions). We did experiments by inserting one dropout layer before the detection layer, and didn't observe any improvement.

4 Experiments

We conduct comprehensive ablation experiments along with a thorough comparison of the proposed Guided CNN and the state-of-the-art text detection methods. We instantiate our Guided CNN with CTPN [52] and EAST [54] since they are the state-of-the-art text detection methods for near horizontal text and multi-orientation text respectively.

4.1 Datasets and Evaluation

The proposed method are evaluated on four text detection benchmarks, namely, SWT [5], ICDAR 2011 [58], ICDAR 2013 [43], and ICDAR 2015 [44]. ICDAR datasets consist of 229 images for training and 255 for testing, 229 for training and 233 for testing, and 1,000 for training and 500 for testing respectively. All the images are labeled in word level. ICDAR 2011, and ICDAR 2013 focuses on horizontal or near-horizontal text while ICDAR 2015 targets at incidental scene text with multiple orientations. The SWT datasets contains 307 color images with sizes ranging from 1024×368 to 1024×768 . The SWT dataset is more challenging than ICDAR 2011 and ICDAR 2013 because of extremely smaller texts, repeating patterns, various plants, etc. We follow previous work by using standard evaluation protocols which are provided by the dataset creators or competition organizers. Our training dataset contains the training sets of the ICDAR 2013 and ICDAR 2015. We also

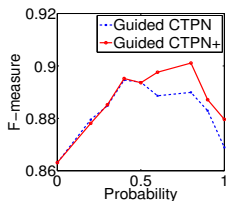


Figure 7: Impact of the probability p of background-aware block-wise random synthesis. As discussed in Section 3.3, if we interpret the proposed background-aware block-wise random synthesis as dropout, Guided CTPN is a kind of approximation of Guided CTPN+ with ignoring feature maps scaled by p in background. as p increases from 0 to 1, the approximation error becomes unignorable. Therefore, Guided CTPN achieves best trade-off at a smaller p (i.e., $p = 0.4$).

collect images from Internet as the training data as done in [11, 52]. We have 4000 training images in total.

4.2 Implementation Details

We use caffe [20] to implement all our experiments except the training related with EAST [54] which is done with tensorflow.

For the guidance subnetwork, its backbone is pretrained on ImageNet. We preprocess the input image as its corresponding primary text detector. We employ multi-step learning rate policy with the base learning rate being 0.01. The learning rate decays by 10 times every 6000 iterations, and the maximum iteration size is set to 30000. For stable training, we set *iter_size* to 10 to obtain stable gradients.

For the primary text detector, we finetune its corresponding backbone model with the ground truth guidance expanded by random synthesis. CTPN [53] is reimplemented by ourselves as its model and source code released by its original authors without side refinement¹. The source code and the model of EAST is reimplemented and released by the third party². The re-implemented EAST uses ResNet-50 [9] as its backbone model, while the original EAST VGG [48], PVANET [55] and PVANET with double channels in [53]. The input image preprocessing procedure keeps the same as that in [56, 52] during training and testing.

Runtime. As for testing the forward speed, we measure the runtime of all models on a PC with *i7* CPU with caffe [20] as text detection approaches are deployed on devices with CPU in most scenarios. The runtime of Guided CNN (or Guided CNN+) includes that of two subnetworks. For models related with EAST, we convert them to caffe models before testing. All results including accuracy and runtime are reported with a *single scale test*.

4.3 Ablation Experiments

We run a number of ablations to analyze the impact of hyper parameters and the effectiveness of each component of the proposed Guided CNN.

The impact of parameters τ and p . There are only two parameters (i.e., the threshold τ to binarize the predict guidance map as described in Section 3.1 and the probability p of background-aware block-wise random synthesis as described in Section 3.3) in Guided CNN. To demonstrate the impact of the parameters on the performance of the model, we conduct experiments on ICDAR 2013 based on CTPN under different parameter settings.

When τ is set to 0.1, 0.2, 0.3 and 0.4, the guidance subnetwork obtains the recall of 0.97, 0.95, 0.94, and 0.92 while Guided CTPN achieves speedup by 2.69, 2.90, 3.01, and 3.12 times respectively. We fix τ to 0.2 for all rest experiments since it obtains good trade-off between recall and speedup. Figure 7 shows the performance of Guided CTPN and Guided

¹<https://github.com/tianzhi0549/CTPN>

²<https://github.com/argman/EAST>

Method	SWT				ICDAR 2011				ICDAR 2013			
	Recall	Precision	F-measure	Speedup	Recall	Precision	F-measure	Speedup	Recall	Precision	F-measure	Speedup
CTPN	0.635	0.688	0.660	×1	0.858	0.867	0.862	×1	0.861	0.899	0.880	×1
Guided CTPN	0.669	0.664	0.667	×4.2	0.868	0.893	0.880	×2.9	0.874	0.916	0.895	×2.9
Guided CTPN+	0.660	0.691	0.675	×0.7	0.875	0.906	0.890	×0.7	0.880	0.924	0.901	×0.7

Table 1: Effectiveness of guidance on SWT, ICDAR 2011 and ICDAR 2013.

Method	Recall	Precision	F-measure	Speedup
EAST	0.773	0.847	0.808	×1
Guided EAST	0.789	0.848	0.818	×2.0
Guided EAST+	0.789	0.866	0.823	×0.8

Table 2: Effectiveness of guidance on ICDAR 2015. For fair comparison, we compare Guided EAST and Guided EAST+ with the EAST result re-implemented by the third party.

Method	ICDAR 2013			ICDAR 2015			Method	R	P	F
	IC13 Eval	DetEval		R	P	F				
	R	P	F	R	P	F				
Jiang [10]	-	-	-	0.915	0.922	0.919	SSTD [10]	0.739	0.802	0.769
WordSup [10]	-	-	-	0.875	0.933	0.903	SegLink [10]	0.768	0.731	0.750
SegLink [10]	-	-	-	0.830	0.877	0.853	WordSup [10]	0.770	0.793	0.782
Yao [10]	0.802	0.889	0.843	-	-	-	R2CNN* [10]	0.743	0.764	0.753
Zhang [10]	0.780	0.880	0.830	-	-	-	DMPNet [10]	0.682	0.732	0.706
He [10]	0.810	0.920	0.860	-	-	-	RRPN [10]	0.770	0.840	0.800
TextBoxes [10]	0.740	0.860	0.800	0.740	0.880	0.810	Original EAST* [10]	0.735	0.836	0.782
SSTD [10]	0.857	0.884	0.870	0.862	0.893	0.877				
Original CTPN [10]	0.737	0.928	0.822	0.830	0.930	0.877				
Guided CTPN	0.846	0.881	0.863	0.874	0.916	0.895	Guided EAST	0.789	0.848	0.818
Guided CTPN+	0.846	0.885	0.870	0.880	0.924	0.901	Guided EAST+	0.789	0.866	0.823

Table 3: Comparisons with state-of-the-art text detection methods on ICDAR 2013 and ICDAR 2015.

* indicates the best result with a single scale test.

CTPN+ with different p . It has been shown that Guided CTPN+ obtains its best result when $p = 0.8$ while Guided CTPN obtains its best result when $p = 0.4$. In the rest experiments, we report results of Guided CNN with $p = 0.4$ while those of Guided CNN+ with $p = 0.8$.

The effectiveness of guidance. We investigate the effectiveness of the guidance. We employ CTPN and EAST as backbone, and evaluate Guided CNNs and their corresponding backbones on benchmark datasets.

From Table 1, Guided CTPN consistently outperforms CTPN in terms of the F-measure while achieving considerable speedup. Guided CTPN+ performs best with sacrificing some speed. It can be used in the case when computation resource is adequate. Specifically, on ICDAR 2011 and ICDAR 2013, compared with CTPN, Guided CTPN improves the F-measure by 1.5 ~ 1.8% while speeding up 2.9 times. Guided CTPN+ improves the F-measure by 2.1 ~ 2.8% at the cost of 0.3 times slowing down. On SWT, Guided CTPN and Guided CTPN+ achieve 0.7% and 1.5% F-measure gain respectively, although its ground truth is annotated in text line level which is different from that of our training data (*e.g.*, training set of ICDAR 2013). Guided CTPN eventually speeds up 4.2 times. Surprisingly, Guided CTPN obtains a higher recall rate although it filters out most of regions of input image, we believe that it is because the guidance mechanism makes the text detector to focus on distinguishing between hard non-text and text regions, and bounding box regression, leading to better training and generalization. We also analyze the runtime for each component of Guided CTPN on ICDAR 2013. The guidance subnetwork accounts for 15% of the runtime while the primary text detector 85%.

From Table 2, it has been shown that Guided EAST improves the F-measure by 1.0% while speeding up 2.0 times compared with EAST. Guided EAST+ improves the F-measure by 1.5% while slowing down 20%.

Comparisons with state-of-the-art results. From Table 3, it has been shown that Guided CTPN and Guided CTPN+ are comparable with [10] and outperform others using

the IC13 evaluation protocol. They also outperform others except [24] using the DetEval evaluation protocol. Note that our Guided CTPN is based on a simple architecture of CTPN and accelerates CTPN significantly while [24] combined a segmentation and detection networks. Guided EAST and Guided EAST+ outperform others. Especially, Guided EAST+ outperforms the second best method RRPN by 2.3%.

5 Conclusion

We have proposed a general framework for text detection called Guided CNN to improve the accuracy and the speed of existing single forward CNN based text detectors simultaneously. We have designed a context module to capture multi-scale context in the guidance subnetwork, leading to effective guidance mask predictions. We have proposed a novel background-aware block-wise random synthesis training strategy, resulting in accuracy improvement and convenient deployment. It can be interpreted as a special kind of dropout. The proposed Guided CNN is a general framework which can be plugged into any existing single forward CNN based text detectors. We have demonstrated its general applicability by instantiating it with CTPN and EAST as backbone. Extensive experiments have evaluated its effectiveness and efficiency.

6 Acknowledgement

This work was supported in part by Shenzhen Basic Research Program (JCYJ2015092516300 5055) and International Partnership Program of Chinese Academy of Sciences (172644KYSB 20160033).

References

- [1] Xiangrong Chen and A. L. Yuille. Detecting and reading text in natural scenes. In *CVPR*, 2004.
- [2] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. BinaryConnect: training deep neural networks with binary weights during propagations. In *NIPS*, 2015.
- [3] Yuchen Dai, Zheng Huang, Yuting Gao, and Kai Chen. Fused text segmentation networks for multi-oriented scene text detection. In *arXiv preprint arXiv:1709.03272*, 2017.
- [4] Emily Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *NIPS*, 2014.
- [5] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *CVPR*, 2010.
- [6] Steve K Esser and John V Arthur. Backpropagation for energy-efficient neuromorphic computing. *NIPS*, 2015.

- [7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localization in natural images. In *CVPR*, 2016.
- [8] Song Han, Huizi Mao, and William J Dally. Deep compression: compressing deep neural networks with pruning, trained quantization and huffman coding. In *ICLR*, 2016.
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [10] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *AAAI*, 2016.
- [11] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *ICCV*, 2017.
- [12] Tong He, Weilin Huang, Yu Qiao, and Jian Yao. Text-attentional convolutional neural network for scene text detection. *TIP*, 2016.
- [13] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Deep direct regression for multi-oriented scene text detection. In *arXiv preprint arXiv:1703.08289*, 2017.
- [14] Han Hu, Chengquan Zhang, Yuxuan Luo, Yuzhuo Wang, Junyu Han, and Errui Ding. WordSup: exploiting word annotations for character based text detection. In *ICCV*, 2017.
- [15] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *ICCV*, 2013.
- [16] Weilin Huang, Yu Qiao, and Xiaoou Tang. Robust scene text detection with convolution neural network induced MSER trees. In *ECCV*, 2014.
- [17] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *ECCV*, 2014.
- [18] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. In *BMVC*, 2014.
- [19] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. In *IJCV*, 2016.
- [20] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: convolutional architecture for fast feature embedding. In *arXiv preprint arXiv:1408.5093*, 2014.
- [21] Fan Jiang, Zhihui Hao, and Xinran Liu. Deep scene text detection with connected component proposals. In *BMVC*, 2017.
- [22] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2CNN: rotational region CNN for orientation robust scene text detection. In *arXiv preprint arXiv:1706.09579*, 2017.

- [23] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluís Gomez I. Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluís Pere De Las Heras. ICDAR 2013 robust reading competition. *ICDAR*, 2013.
- [24] Dimosthenis Karatzas, Lluís Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. ICDAR 2015 competition on robust reading. *ICDAR*, 2015.
- [25] Kye-Hyeon Kim, Sanghoon Hong, Byungseok Roh, Yeongjae Cheon, and Minje Park. PVANET: deep but lightweight neural networks for real-time object detection. In *EMDNN*, 2016.
- [26] Vadim Lebedev, Yaroslav Ganin, Maksim Rakhuba, Ivan Oseledets, and Victor Lempitsky. Speeding-up convolutional neural networks using fine-tuned CP-Decomposition. *ICLR*, 2015.
- [27] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *arXiv preprint arXiv:1707.03985*, 2017.
- [28] Huiping Li, D Doermann, and O Kia. Automatic text detection and tracking in digital video. *TIP*, 2000.
- [29] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *CVPR*, 2017.
- [30] Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade. In *CVPR*, 2017.
- [31] Gang Li, Haoxiang and Lin, Zhe and Shen, Xiaohui and Brandt, Jonathan and Hua. A convolutional neural network approach for face detection. In *CVPR*, 2015.
- [32] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. TextBoxes: a fast text detector with a single deep neural network. In *AAAI*, 2017.
- [33] Rainer Lienhart and Axel Wernicke. Localizing and segmenting text in images and videos. *TCSVT*, 2002.
- [34] Wei Liu, Andrew Rabinovich, and Alexander C. Berg. ParseNet: looking wider to see better. In *arXiv preprint arXiv:1506.04579*, 2015.
- [35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.
- [36] Yuliang Liu and Lianwen Jin. Deep matching prior network: toward tighter multi-oriented text detection. In *CVPR*, 2017.
- [37] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary oriented scene text detection via rotation proposals. In *arXiv preprint arXiv:1703.01086*, 2017.

- [38] R Minetto, N Thome, M Cord, J Fabrizio, and B Marcotegui. Snooertext : a multiresolution system for text detection in complex visual scenes. In *ICIP*, 2010.
- [39] Lukas Neumann and Jiri Matas. Real-time scene text localization and recognition. In *CVPR*, 2012.
- [40] Lukas Neumann and Jiri Matas. Efficient scene text localization and recognition with local character refinement. *ICDAR*, 2015.
- [41] Lukáš Neumann and Jiří Matas. Real-time lexicon-free scene text localization and recognition. In *PAMI*, 2016.
- [42] Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu. Joint training of cascaded CNN for face detection. In *CVPR*, 2016.
- [43] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net : ImageNet classification using binary. In *ECCV*, 2016.
- [44] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *PAMI*, 2017.
- [45] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *PAMI*, 2016.
- [46] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, 2017.
- [47] Cunzhaoh Shi, Chunheng Wang, Baihua Xiao, Yang Zhang, Song Gao, and Zhong Zhang. Scene text recognition using part-based tree-structured character detection. In *CVPR*, 2013.
- [48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.
- [49] D Soudry, I Hubara, and R Meir. Expectation backpropagation: parameter-free training of multilayer neural networks with real and discrete weights. In *NIPS*, 2014.
- [50] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 2014.
- [51] S Tian, Y Pan, C Huang, S Lu, K Yu, and C L Tan. Textflow: a unified text detection system in natural scene images. In *ICCV*, 2015.
- [52] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 2016.
- [53] David J DJ Wu, Tao Wang, A Coates, and AY Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, 2012.
- [54] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *CVPR*, 2016.

- [55] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, 2012.
- [56] Cong Yao, Xiang Bai, Nong Sang, Xinyu Zhou, Shuchang Zhou, and Zhimin Cao. Scene text detection via holistic, multi-channel prediction. In *arXiv preprint arXiv:1606.09002*, 2016.
- [57] Xu-Cheng Yin, Xuwang Yin, Kaizhu Huang, and Hong-Wei Hao. Robust text detection in natural scene images. *PAMI*, 2014.
- [58] Alessandro Zamberletti, Lucia Noce, and Ignazio Gallo. Text localization based on fast feature pyramids and multi-resolution maximally stable extremal regions. In *ACCVW*, 2015.
- [59] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Senior Member, Yu Qiao, and Senior Member. Joint face detection and alignment using multi-task cascaded convolutional networks. *IEEE Signal Processing Letters*, 2016.
- [60] Kaipeng Zhang, Zhanpeng Zhang, Hao Wang, Zhifeng Li, Yu Qiao, and Wei Liu. Detecting faces using inside cascaded contextual CNN. In *CVPR*, 2017.
- [61] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun. Accelerating very deep convolutional networks for classification and detection. *PAMI*, 2016.
- [62] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenig Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *CVPR*, 2016.
- [63] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017.
- [64] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. EAST: an efficient and accurate scene text detector. In *CVPR*, 2017.