TextBoxes: A Fast Text Detector with a Single Deep Neural Network

Minghui Liao*, Baoguang Shi*, Xiang Bai†, Xinggang Wang, Wenyu Liu
School of Electronic Information and Communications, Huazhong University of Science and Technology
{mhliao, xbai, xgwang, liuwy}@hust.edu.cn and shibaoguang@gmail.com

Abstract

This paper presents an end-to-end trainable fast scene text detector, named TextBoxes, which detects scene text with both high accuracy and efficiency in a single network forward pass, involving no post-process except for a standard non-maximum suppression. TextBoxes outperforms competing methods in terms of text localization accuracy and is much faster, taking only 0.09s per image in a fast implementation. Furthermore, combined with a text recognizer, TextBoxes significantly outperforms state-of-the-art approaches on word spotting and end-to-end text recognition tasks.

Introduction

Scene text is one of the most general visual objects in natural scenes. It frequently appears on road signs, license plates, product packages, *etc*. Reading scene text facilitates a lot of useful applications, such as image-based geolocation. Despite the similarity to traditional OCR, scene text reading is much more challenging, due to the large variations in both foreground text and background objects, as well as uncontrollable lighting conditions, *etc*.

Owing to the inevitable challenges and complexities, traditional text detection methods tend to involve multiple processing steps, *e.g.* character/word candidate generation (Neumann and Matas 2012; Jaderberg et al. 2016), candidate filtering, and grouping. They often end up struggling to get each module working properly, requiring much effort in tuning parameters and designing heuristic rules, also slowing down detection speed. Inspired by the recent developments in object detection (Liu et al. 2016; Ren et al. 2015), we propose to detect texts by *directly* predicting word bounding boxes via a single neural network that is end-to-end trainable.

Our key contribution in this paper is a fast and accurate text detector called *TextBoxes*, which is based on fully-convolutional network (LeCun et al. 1998). TextBoxes directly outputs the coordinates of word bounding boxes at multiple network layers by jointly predicting text presence and coordinate offsets to *default boxes* (Liu et al. 2016). The final outputs are the aggregation of all boxes, followed by a standard non-maximum suppression process. To handle the large variation in aspect ratios of words, we design several novel, inception-style (Szegedy et al. 2015) output layers that utilize both irregular convolutional kernels and default boxes. Our detector delivers both high accuracy and high efficiency with only a single forward pass on single-scale inputs, and even higher accuracy with multiple passes on multi-scale inputs.

Furthermore, we argue that word recognition is helpful to distinguish texts from backgrounds, especially when words are confined to a given set, *i.e.* a lexicon. We adopt a successful text recognition algorithm, CRNN (Shi, Bai, and Yao 2015), in conjunction with TextBoxes. The recognizer not only provides extra recognition outputs, but also regularizes text detection with its semantic-level awareness, thus further boosting the accuracy of word spotting considerably. The combination of TextBoxes and CRNN yields the state-of-the-art performance on word spotting and end-to-end text recognition tasks, which appears to be a simple yet effective solution to robust text reading in the wild.

To summarize, the contributions of this paper are threefold: First, we design an end-to-end trainable neural network model for scene text detection. Second, we propose a word spotting/end-to-end recognition framework that effectively combines detection and recognition. Third, our model achieves highly competitive results while keeping its computational efficiency.

Related Works

Intuitively, scene text reading can be further divided into two sub-tasks: text detection and text recognition. The former aims to localize text in images, mostly in the form of word bounding boxes; The latter transcripts cropped word images into machine-interpretable character sequences. We cover both tasks in this paper but pay more attention to detection.

Based on a basic detection target, previous methods for text detection can be roughly categorized into three categories:

1) Character-based: Individual characters are first detected and then grouped into words (Neumann and Matas 2012; Pan, Hou, and Liu 2011; Yao et al. 2012; Huang, Qiao, and Tang 2014). For example, (Neumann and Matas 2012) lo-

*Authors contribute equally.
†Corresponding author.

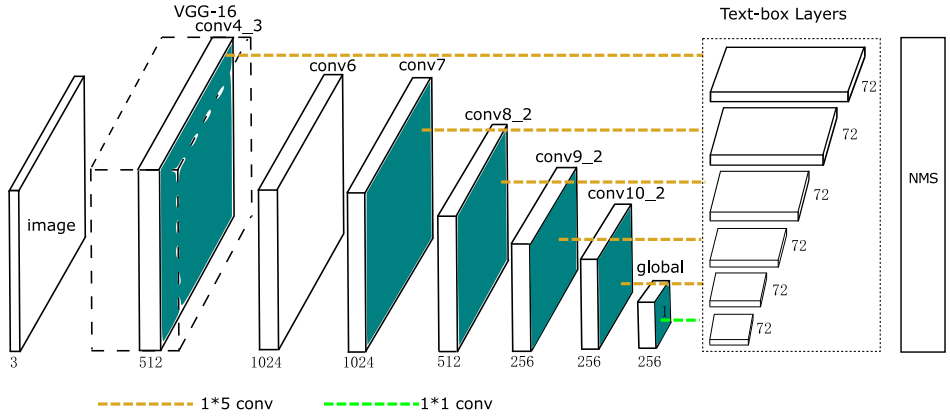

Figure 1: TextBoxes Architecture. TextBoxes is a 28-layer fully convolutional network. Among them, 13 are inherited from VGG-16. 9 extra convolutional layers are appended after the VGG-16 layers. Text-box layers are connected to 6 of the convolutional layers. On every map location, a text-box layer predicts a 72-d vector, which are the text presence scores (2-d) and offsets (4-d) for 12 default boxes. A non-maximum suppression is applied to the aggregated outputs of all text-box layers.

cates characters by classifying Extremal Regions. After that, the detected characters are grouped by an exhaustive search method;
2) Word-based: Words are directly hit with the similar manner of general object detection (Jaderberg et al. 2016; Zhong et al. 2016; Gomez-Bigorda and Karatzas 2016). (Jaderberg et al. 2016) proposes an R-CNN-based (Girshick et al. 2014) framework.First, word candidates are generated with class-agnostic proposal generators. Then the proposals are classified by a random forest classifier. Finally, a convolutional neural network for bounding box regression was adopted to refine the bounding boxes. (Gupta, Vedaldi, and Zisserman 2016) improves over the YOLO network (Redmon et al. 2016) while it still adopts the filter and regression steps for further removing the false positives;
3) Text-line-based: Text lines are detected and then broken into words. For example, (Zhang et al. 2015) proposes to detect text lines utilizing their symmetric characteristics. Furthermore, (Zhang et al. 2016) localizes text lines with fully convolutional networks (Long, Shelhamer, and Darrell 2015).

TextBoxes is *word-based*. In contrast to (Jaderberg et al. 2016), which comprises three detection steps and each further includes more than one algorithm, TextBoxes enjoys a much simpler pipeline. We only need to train one network end-to-end.

TextBoxes is inspired by SSD (Liu et al. 2016), a recent development in object detection. SSD aims to detect general objects in images but fails on words that have extreme aspect ratios. We propose text-box layers in TextBoxes to solve this problem, which significantly improve the performance.

We adopt a text recognizer called CRNN (Shi, Bai, and Yao 2015) in conjunction with TextBoxes for word spotting and end-to-end recognition. CRNN directly outputs character sequences given input images and is also end-to-end trainable. Besides, we use the confidence scores of CRNN to regularize the detection outputs of TextBoxes. Note that it is also possible to adopt other recognizers, such as (Jaderberg et al. 2016).

Detecting text with TextBoxes

Architecture

The architecture of TextBoxes is depicted in Fig. 1. It inherits the popular VGG-16 architecture (Simonyan and Zisserman 2014), keeping the layers from `conv1_1` through `conv4_3`. The last two fully-connected layers of VGG-16 are converted into convolutional layers by parameters downsampling (Liu et al. 2016). They are followed by a few extra convolutional and pooling layers, namely `conv6` to `pool11`.

Multiple output layers, which we call *text-box layers*, are inserted after the last and some intermediate convolutional layers. Their outputs are aggregated and undergo a non-maximum suppression (NMS) process. Output layers are also convolutional. All together, TextBoxes consists of only convolutional and pooling layers, thus *fully-convolutional*. It adapts to arbitrary-size images in both training and testing.

Text-box layers

Text-box layers are the key component of TextBoxes. A text-box layer simultaneously predicts text presence and bounding boxes, conditioned on its input feature map. At every map location, it outputs the classification scores and offsets to its associated default boxes in a convolutional manner. Suppose that image and feature map sizes are respectively $(w_{\text{im}}, h_{\text{im}})$ and $(w_{\text{map}}, h_{\text{map}})$. On a map location (i, j) which associates a default box $\mathbf{b}_0 = (x_0, y_0, w_0, h_0)$, the text-box layer predicts the values of $(\Delta x, \Delta y, \Delta w, \Delta h, c)$, indicating that a box $\mathbf{b} = (x, y, w, h)$ is detected with confidence c, where

$$\begin{aligned} x &= x_0 + w_0 \Delta x, \\ y &= y_0 + h_0 \Delta y, \\ w &= w_0 \exp(\Delta w), \\ h &= h_0 \exp(\Delta h). \end{aligned} \tag{1}$$

In the training phase, ground-truth word boxes are matched to default boxes according to box overlap, follow-

ing the matching scheme in (Liu et al. 2016). Each map location is associated with multiple default boxes of different sizes. They effectively divide words by their scales and aspect ratios, allowing TextBoxes to learn specific regression and classification weights that handle words of similar size. Therefore, the design of default boxes is highly task-specific.

Different from general objects, words tend to have large aspect ratios. Therefore, we include "long" default boxes that have large aspect ratios. Specifically, we define 6 aspect ratios for default boxes, including 1,2,3,5,7, and 10. However, this makes the default boxes dense on the horizontal direction while sparse vertically, which causes poor matching boxes. To solve this issue, each default box is set with vertical offsets. The design of the default boxes is illustrated in Fig. 2.

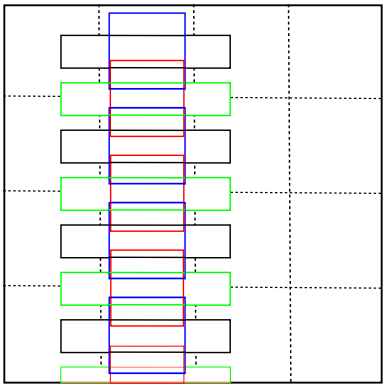

Figure 2: Illustration of default boxes for a 4*4 grid. For better visualization, only a column of default boxes whose aspect ratios 1 and 5 are plotted. The rest of the aspect ratios are 2,3,7 and 10, which are placed similarly. The black (aspect ratio: 5) and blue (ar: 1) default boxes are centered in their cells. The green (ar: 5) and red (ar: 1) boxes have the same aspect ratios and a vertical offset(half of the height of the cell) to the grid center respectively.

Moreover, in text-box layers we adopt irregular 1*5 convolutional filters instead of the standard 3*3 ones. This inception-style (Szegedy et al. 2015) filters yield rectangular receptive fields, which better fit words with larger aspect ratios, also avoiding noisy signals that a square-shaped receptive field would bring in.

Learning

We adopt the same loss function as (Liu et al. 2016). Let x be the match indication matrix, c be the confidence, l be the predicted location, and g be the ground-truth location. Specifically, for the i-th default box and the j-th ground truth, $x_{ij} = 1$ means matching while $x_{ij} = 0$ otherwise. The loss function is defined as:

$$L(x, c, l, g) = \frac{1}{N}(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g)), \quad (2)$$

where N is the number of default boxes that match ground-truth boxes, and α is set to 1. We adopt the smooth L1 loss (Girshick 2015) for L_{loc} and a 2-class softmax loss for L_{conf}.

Multi-scale inputs

Even with the optimizations on default boxes and convolutional filters, it may be still difficult to robustly localize the words of extreme aspect ratios and sizes. To further boost detection accuracy, we use multiple rescaled versions of the input image for TextBoxes. An input image is rescaled into five scales, including (width*height) 300*300, 700*700, 300*700, 500*700, and 1600*1600. Note that some scales squeeze image horizontally, so that some "long" words are shortened. Multi-scale inputs boost detection accuracy while slightly increasing the computational cost. On ICDAR 2013 , they further improve f-measure of detection by 5 percents. Detecting all five scales takes 0.73s per image, and 0.24s if we remove the last 1600*1600 scale. The running time is measured on a single Titan X GPU. Note that, different from testing, we only use single-scale input (300*300) for training.

Non-maximum suppression

Non-maximum suppression is applied to the aggregated outputs of all text-box layers. We adopt an extra non-maximum suppression for multi-scale inputs on the task of text localization.

Word spotting and end-to-end recognition

Word spotting is to localize specific words that are given in a lexicon. End-to-end recognition concerns both detection and recognition. Although both tasks can be achieved by simply connecting TextBoxes with a text recognizer, we propose to improve detection with recognition. We argue that a recognizer can help eliminating false-positive detection results that are unlikely to be meaningful words, *e.g.* repetitive patterns. Particularly, when a lexicon is present, a recognizer could effectively removes the detected bounding boxes that do not match any of the given words.

We adopt the CRNN model (Shi, Bai, and Yao 2015) as our text recognizer. CRNN uses CTC (Graves et al. 2006) as its output layer, which estimates sequence probability conditioned on input image, *i.e.* $p(\mathbf{w}|I)$, where I is an input image and $\mathbf{w}$ represents a character sequence. We treat the probability as a matching score, which measures the compatibility of an image to a particular word. The detection score is then the maximum score among all words in a given lexicon:

$$s = \max_{\mathbf{w} \in \mathcal{W}} p(\mathbf{w}|I) \quad (3)$$

where $\mathcal{W}$ is a given lexicon. If the task specifies no lexicon, we use a generic lexicon that consists of 90k English words.

We replace the original TextBoxes detection score with the one in Eq. 3. However, evaluating Eq. 3 on all boxes would be time-consuming. In practice, we first use TextBoxes to produce a redundant set of word candidates by detecting with a lower score threshold and a high NMS overlap threshold, preserving about 35 bounding boxes per image with a high recall of 0.93 with multi-scale inputs for ICDAR 2013 . Then we apply Eq. 3 to all candidates to re-evaluate their scores, followed by a second score thresholding and a NMS. When dealing with multi-scale inputs,

Table 1: Text localization on ICDAR 2011 and ICDAR 2013. P, R and F refer to precision, recall and F-measure respectively. FCRNall+filts reported a time consumption of 1.27 seconds excluding its regression step so we assume it takes more than 1.27 seconds.

Datasets	ICDAR 2011						ICDAR 2013						Time/s
Evaluation protocol	IC13 Eval			DetEval			IC13 Eval			DetEval			
Methods	P	R	F	P	R	F	P	R	F	P	R	F	
Jaderberg (Jaderberg et al. 2016)	–	–	–	–	–	–	–	–	–	–	–	–	7.3
MSERs-CNN (Huang, Qiao, and Tang 2014)	0.88	0.71	0.78	–	–	–	–	–	–	–	–	–	–
MMser (Zamberletti, Noce, and Gallo 2014)	–	–	–	–	–	–	0.86	0.70	0.77	–	–	–	0.75
TextFlow (Tian et al. 2015)	0.86	0.76	0.81	–	–	–	0.85	0.76	0.80	–	–	–	1.4
FCRNall+filts (Gupta, Vedaldi, and Zisserman 2016)	–	–	–	**0.92**	0.75	0.82	–	–	–	**0.92**	0.76	0.83	>1.27
FCN (Zhang et al. 2016)	–	–	–	–	–	–	0.88	0.78	0.83	–	–	–	2.1
SSD (Liu et al. 2016)	–	–	–	–	–	–	0.80	0.60	0.68	0.80	0.60	0.69	0.1
Fast TextBoxes	0.86	0.74	0.80	0.88	0.74	0.80	0.86	0.74	0.80	0.88	0.74	0.81	**0.09**
TextBoxes	**0.88**	**0.82**	**0.85**	0.89	**0.82**	**0.86**	**0.88**	**0.83**	**0.85**	0.89	**0.83**	**0.86**	0.73

we generate candidates separately on each scale and perform the above steps on candidates of all the scales. Here we also adopt a slightly different NMS scheme. A lower overlap threshold is employed for boxes that are recognized as the same word, so that stronger suppression is imposed on boxes of the same word.

Experiments

We verify the effectiveness of TextBoxes on three different tasks, including text detection, word-spotting, and end-to-end recognition.

Datasets

SynthText (Gupta, Vedaldi, and Zisserman 2016) contains 800k synthesized text images, created via blending rendered words with natural images. The synthesized images look realistic, as the location and transform of text are carefully chosen with a learning algorithm. This dataset is used for pre-training our model.

ICDAR 2011 (IC11) (Shahab, Shafait, and Dengel 2011) There are real-world images with high resolution in the ICDAR 2011 dataset. The test set of the ICDAR 2011 dataset is used to evaluate our model.

ICDAR 2013 (IC13) (Karatzas et al. 2013) The ICDAR 2013 dataset is similar to the ICDAR 2011 dataset. We use the training set of the ICDAR 2013 for training when we do experiments on the ICDAR 2011 dataset and the ICDAR 2013 dataset. The ICDAR 2013 dataset gives 3 lexicons of different sizes for the task of word spotting and end-to-end recognition. For each test image, it gives 100 words as a lexicon, which is called a strong lexicon. For the whole test set, it gives a lexicon containing hundreds of words, which is called a weakly lexicon. It also gives a generic lexicon which contains 90k words.

Street View Text (SVT) (Wang and Belongie 2010) The SVT dataset is more challenging than the ICDAR datasets due to the lower resolution of the images. There exist some unlabeled texts in the images. Thus, we only use this dataset for word spotting, in which a lexicon containing 50 words is provided for each image.

Implementation details

TextBoxes is trained with 300*300 images using stochastic gradient descent (SGD). Momentum and weight decay are set to 0.9 and 5×10^{-4} respectively. Learning rate is initially set to 10^{-3}, and decayed to 10^{-4} after 40k training iterations. On all the datasets except SVT, we first train TextBoxes on SynthText for 50k iterations, then finetune it on ICDAR 2013 training dataset for 2k iterations. On SVT, the finetuning is performed on the SVT training dataset. All training images are augmented online with random crop and flip, following the scheme in (Liu et al. 2016). All the experiments are carried out on a PC with one Titan X GPU. The whole training time is about 25 hours. Text recognition is performed with a pre-trained CRNN (Shi, Bai, and Yao 2015) model[1], which is implemented and released by the authors.

Text localization

TextBoxes is tested on ICDAR 2011 and ICDAR 2013 for evaluating its text localization performance. The results are summarized and compared with other methods in Table. 1. Results are evaluated under two different evaluation protocols, the DetEval (Wolf and Jolion 2006) and the ICDAR 2013 evaluation (Karatzas et al. 2013).

Since there is a trade-off between precision and recall rate, f-measure is the most accurate measurement of detection performance. TextBoxes consistently outperforms competing methods in terms of f-measure. On ICDAR 2011, TextBoxes outperforms the second best methods (Gupta, Vedaldi, and Zisserman 2016), by 4 percents. On ICDAR 2013, TextBoxes also outperforms competing methods by at

[1] https://github.com/bgshih/crnn

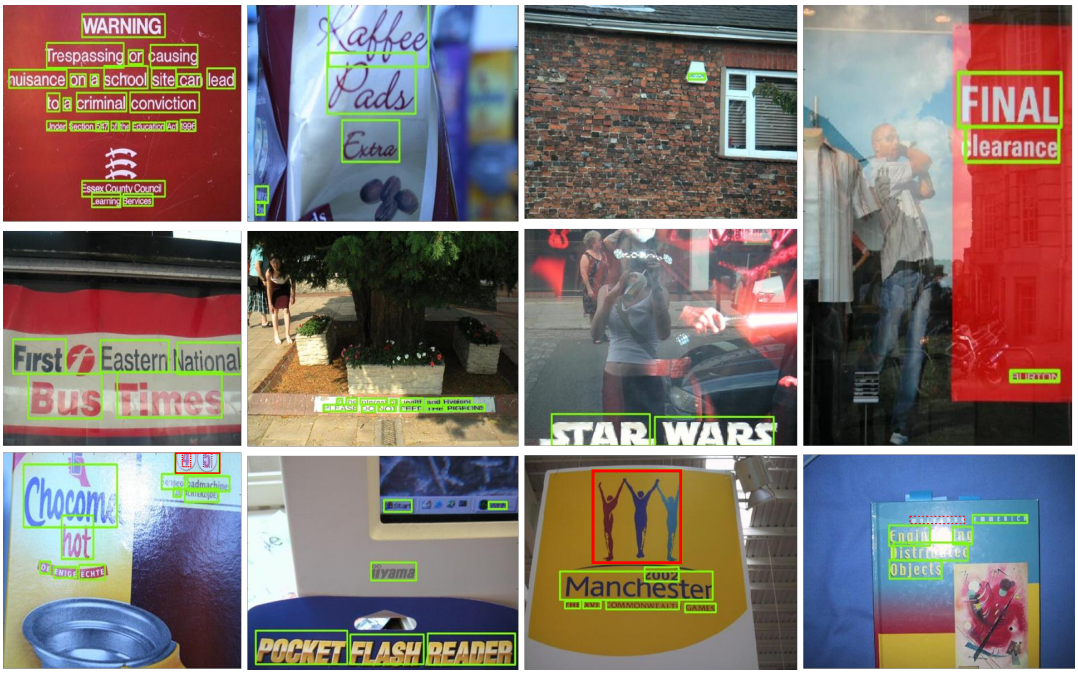

Figure 3: Examples of text localization results. The green bounding boxes are correct detections; Red boxes are false positives; Red dashed boxes are false negatives.

Table 2: Word spotting and end-to-end results. The values in the table are F-measure. For ICDAR 2013, strong, weak and generic mean a small lexicon containing 100 words for each image, a lexicon containing all words in the whole test set and a large lexicon respectively. We use a lexicon containing 90k words as our generic lexicon. The methods marked by "*" are published on the ICDAR 2015 Robust Reading Competition website `http://rrc.cvc.uab.es`.

Methods	IC11 spotting	SVT spotting	SVT-50 spotting	IC13 spotting			IC13 end-to-end		
				strong	weak	generic	strong	weak	generic
Alsharif (Alsharif and Pineau 2013)	–	–	0.48	–	–	–	–	–	–
Jaderberg (Jaderberg et al. 2016)	0.76	0.56	0.68	–	–	0.76	–	–	–
FCRNall+filts (Gupta, Vedaldi, and Zisserman 2016)	0.84	0.53	0.76	–	–	0.85	–	–	–
Deep2Text II+*	–	–	–	0.85	0.83	0.80	0.82	0.79	0.77
SRC-B-TextProcessingLab*	–	–	–	0.90	0.88	0.81	0.87	0.85	0.80
Adelaide_ConvLSTMs*	–	–	–	0.91	0.90	0.83	0.87	0.86	0.80
TextBoxes	**0.87**	**0.64**	**0.84**	**0.94**	**0.92**	**0.87**	**0.91**	**0.89**	**0.84**

least 2 percents. TextBoxes ranks the first in term of testing speed, even with the multi-scale version, which takes only 0.73s per image. Meanwhile, a fast implementation of TextBoxes takes merely 0.09s per image, without much loss in accuracy.

In order to further verify the effectiveness of TextBoxes, we also report the results of SSD (Liu et al. 2016) for the comparison in Table. 1, which is the most relevant and the state-of-the-art detector for general objects. Here, SSD is trained using the same procedures as TextBoxes. SSD achieves competitive performance, but still falls short of other state-of-the-art methods. In particular, we observe that SSD cannot achieve good results when detecting words with large aspect ratios while TextBoxes performs much better, benefiting from the proposed text-box layers which are designed in order to overcome the length variation of words.

Word spotting and end-to-end recognition

The performance of word spotting is evaluated by detection results that are refined by recognition, while the evaluation of end-to-end performance concerns both detection and recognition results. We test TextBoxes on ICDAR 2011, SVT, and ICDAR 2013.

As shown in Table. 2, our method outperforms all the existing methods, including the most recent competition results published on the website. On ICDAR 2011 and ICDAR 2013, our method outperforms the second best method at least 2 percents with all the evaluation protocol listed

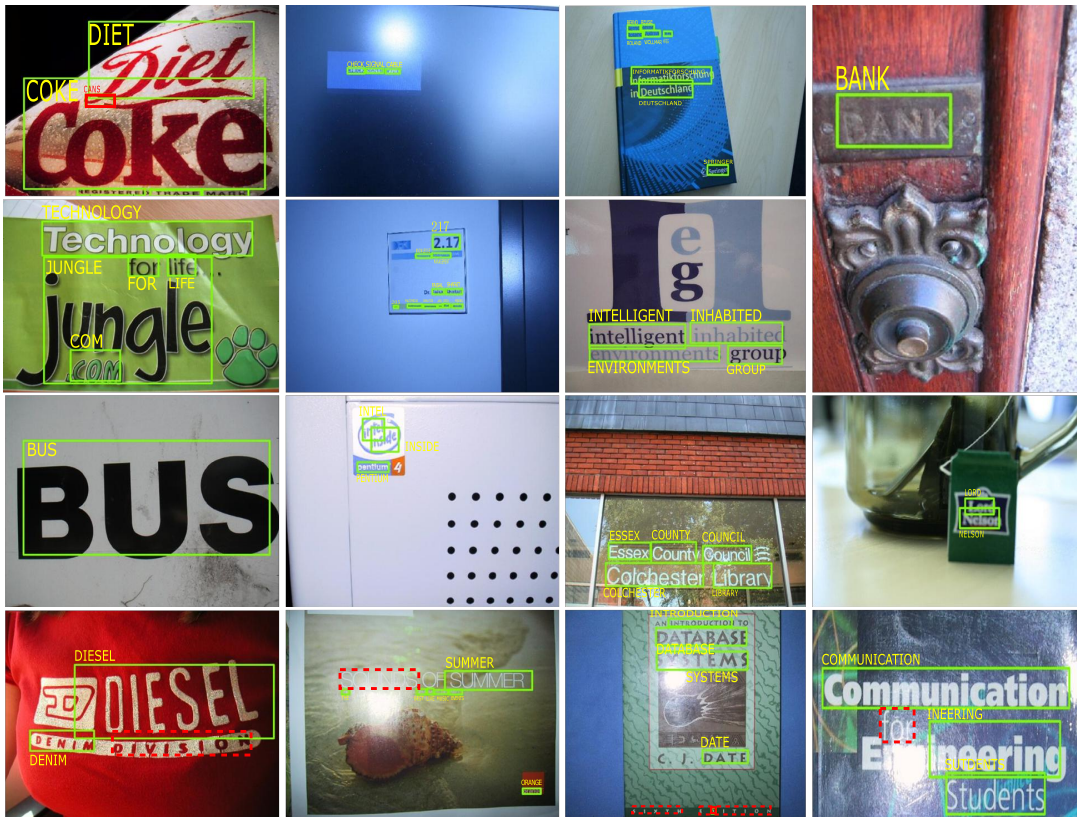

Figure 4: Examples of word spotting results. Yellow words are recognition results. Words less than 3 letters are ignored, following the evaluation protocol. The box colors have the same meaning as Fig. 3.

in Table. 2. The performance gap on SVT is even larger. TextBoxes outperforms the leading method (Gupta, Vedaldi, and Zisserman 2016), by over 8 percents on both SVT and SVT-50. The reason is likely to be that TextBoxes is more robust to the low resolution images in SVT, since TextBoxes is trained on relatively low resolution images.

Coupled with a recognition model, TextBoxes achieves the state-of-the-art performance on end-to-end recognition benchmarks. On ICDAR 2013, TextBoxes breaks the records recently made by Adelaide_ConvLSTMs* on all the lexicon settings. More specifically, TextBoxes generates about 35 proposals per image when using multi-scale inputs on ICDAR 2013, with a recall of 0.93. With a strong lexicon for the recognition model, 3.8 bounding boxes per image are reserved, achieving a recall of 0.91 and a precision of 0.97. We employ a 90k-lexicon for SVT and ICDAR 2011, and a 50-word lexicon per image on SVT-50. Note that even though Jaderberg (Jaderberg et al. 2016) and FCRNall+filts (Gupta, Vedaldi, and Zisserman 2016) adopt a much smaller lexicon(50k words), their results are still inferior to our method.

Running speed

Most existing methods detect texts in a multi-step manner, making them hard to run efficiently. Most of the computation of TextBoxes is spent on the convolutional forward passes, which are very fast when running on GPU devices. TextBoxes takes only 0.09s per image with $700 * 700$ single-scale images, resulting in an f-measure of 0.80 on ICDAR 2013, which is still very competitive. When running on 5 input scales, TextBoxes achieves 0.85 f-measure on ICDAR 2013, taking 0.73 second per image with the batch size setting to 1. We remove the $1600 * 1600$ scale when testing on SVT, since the SVT image resolutions are relatively low. Testing on the the remaining scales takes merely 0.24 second per image.

The speed comparisons are listed in Table. 1. (Jaderberg et al. 2016) adopts two proposal generation methods, a random forest classifier, and a CNN regression model. They each takes 1-3 seconds, about 7s in total. (Gupta, Vedaldi, and Zisserman 2016) proposes a YOLO-like model called FCRN, followed by the same random forest classifiers and a CNN regression model. It takes 1.27s excluding the regression step, whose running time is not reported. TextBoxes achieves the highest detection accuracy while being the fastest among them.

Weaknesses

TextBoxes performs well in most situations. However, it still fails to handle some difficult cases, such as overexposure and large character spacing. Some failure cases are shown in Fig. 3 and Fig. 4.

Conclusion

We have presented TextBoxes, an end-to-end fully convolutional network for text detection, which is highly stable and efficient to generate word proposals against cluttered backgrounds. Comprehensive evaluations and comparisons on benchmark datasets clearly validate the advantages of Textboxes in three related tasks including text detection, word spoting and end-to-end recognition. In the future, we are interested to extend TextBoxes for multi-oriented texts, and combine the networks of detection and recognition into one unified framework.

Acknowledgements

This work was partly supported by National Natural Science Foundation of China (61222308, 61573160, 61572207 and 61503145), and Open Project Program of the State Key Laboratory of Digital Publishing Technology (F2016001).

References

[Alsharif and Pineau 2013] Alsharif, O., and Pineau, J. 2013. End-to-end text recognition with hybrid HMM maxout models. *CoRR* abs/1310.1811.

[Girshick et al. 2014] Girshick, R.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*.

[Girshick 2015] Girshick, R. B. 2015. Fast R-CNN. In *Proc. ICCV*.

[Gomez-Bigorda and Karatzas 2016] Gomez-Bigorda, L., and Karatzas, D. 2016. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *CoRR* abs/1604.02619.

[Graves et al. 2006] Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*, 369–376.

[Gupta, Vedaldi, and Zisserman 2016] Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proc. CVPR*.

[Huang, Qiao, and Tang 2014] Huang, W.; Qiao, Y.; and Tang, X. 2014. Robust scene text detection with convolution neural network induced mser trees. In *Proc. ECCV*.

[Jaderberg et al. 2016] Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *IJCV* 116(1):1–20.

[Karatzas et al. 2013] Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and de las Heras, L. P. 2013. Icdar 2013 robust reading competition. In *ICDAR*, 1484–1493.

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

[Liu et al. 2016] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; and Reed, S. E. 2016. SSD: single shot multibox detector. In *Proc. ECCV*.

[Long, Shelhamer, and Darrell 2015] Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*.

[Neumann and Matas 2012] Neumann, L., and Matas, J. 2012. Real-time scene text localization and recognition. In *Proc. CVPR*, 3538–3545.

[Pan, Hou, and Liu 2011] Pan, Y.-F.; Hou, X.; and Liu, C.-L. 2011. A hybrid approach to detect and localize texts in natural scene images. *IEEE T. Image Proc.* 20(3):800–813.

[Redmon et al. 2016] Redmon, J.; Divvala, S. K.; Girshick, R. B.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proc. CVPR*.

[Ren et al. 2015] Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. NIPS*.

[Shahab, Shafait, and Dengel 2011] Shahab, A.; Shafait, F.; and Dengel, A. 2011. Icdar 2011 robust reading competition challenge 2: Reading text in scene images. In *Proc. ICDAR*, 1491–1496.

[Shi, Bai, and Yao 2015] Shi, B.; Bai, X.; and Yao, C. 2015. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *CoRR* abs/1507.05717.

[Simonyan and Zisserman 2014] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

[Szegedy et al. 2015] Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proc. CVPR*.

[Tian et al. 2015] Tian, S.; Pan, Y.; Huang, C.; Lu, S.; Yu, K.; and Lim Tan, C. 2015. Text flow: A unified text detection system in natural scene images. In *Proc. ICCV*.

[Wang and Belongie 2010] Wang, K., and Belongie, S. 2010. Word spotting in the wild. In *Proc. ECCV*, 591–604.

[Wolf and Jolion 2006] Wolf, C., and Jolion, J. 2006. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *IJDAR* 8(4):280–296.

[Yao et al. 2012] Yao, C.; Bai, X.; Liu, W.; Ma, Y.; and Tu, Z. 2012. Detecting texts of arbitrary orientations in natural images. In *Proc. CVPR*, 1083–1090.

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

[Zhang et al. 2015] Zhang, Z.; Shen, W.; Yao, C.; and Bai, X. 2015. Symmetry-based text line detection in natural scenes. In *Proc. CVPR*, 2558–2567.

[Zhang et al. 2016] Zhang, Z.; Zhang, C.; Shen, W.; Yao, C.; Liu, W.; and Bai, X. 2016. Multi-oriented text detection with fully convolutional networks. In *Proc. CVPR*.

[Zhong et al. 2016] Zhong, Z.; Jin, L.; Zhang, S.; and Feng, Z. 2016. Deeptext: A unified framework for text proposal generation and text detection in natural images. *CoRR* abs/1605.07314.