

Single Shot Text Detector with Regional Attention

This work is based on using Attention mechanism for getting more accurate prediction for word level in natural images. Here author uses not only one attention, but 4 attentions from CNN features. Then concatenate them for output features using softmax outputs for text and non-text scores. So great idea is that we can use Attention to according to the features extracted from CNN to make the prediction more accurate. This is also based on SSD framework as feature extraction. Attention is great for some regions proposal adjust.

Pan He¹, Weilin Huang^{2, 3}, Tong He³, Qile Zhu¹, Yu Qiao³, and Xiaolin Li¹

¹National Science Foundation Center for Big Learning, University of Florida

²Department of Engineering Science, University of Oxford

³Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

Abstract

We present a novel *single-shot text detector* that directly outputs *word-level bounding boxes* in a natural image. We propose an *attention mechanism* which *roughly identifies text regions* via an *automatically learned attentional map*. This substantially suppresses background interference in the convolutional features, which is the *key* to producing *accurate* inference of words, particularly at extremely small sizes. This results in a single model that essentially works in a *coarse-to-fine* manner. It *departs* from recent *FCN-based text detectors* which cascade multiple FCN models to achieve an accurate prediction. Furthermore, we develop a *hierarchical inception module* which efficiently aggregates *multi-scale inception features*. This *enhances local details*, and also *encodes strong context information*, allowing the *detector* to *work reliably on multi-scale and multi-orientation text* with single-scale images. Our text detector achieves an *F-measure* of 77% on the ICDAR 2015 benchmark, advancing the state-of-the-art results in [18, 28]. Demo is available at: <http://sstd.whuang.org/>.



Figure 1: Illustrations of text attention mechanism in the proposed one-shot text detector. Our model automatically learns a rough text regional attention which is used to highlight text information in the convolutional features. This makes it possible to achieve *accurate* word-level predictions in one shot. Text can be precisely separated and accurately predicted at the *word level* in challenging cases.

1. Introduction

Reading text in the wild has attracted increasing attention in computer vision community, as shown in recent work [28, 4, 36, 38, 2, 6]. It has numerous potential applications in image retrieval, industrial automation, robot navigation and scene understanding. Recent work focuses on *text detection* in natural images, which remains a challenging problem [28, 4, 36, 38, 2]. The main difficulty lies in a vast diversity in text scale, orientation, illumination, and font, which often come with a highly complicated background.

Previous works in text detection have been dominated by *bottom-up approaches* [3, 9, 10, 33, 8], which often contain

multiple sequential steps, including *character or text component detection*, followed by *character classification or filtering*, *text line construction* and *word splitting*. *Character detection* and *filtering* steps play a *key role* in such bottom-up approaches. Previous methods typically identify character or text component candidates using connected component based approaches (e.g., stroke width [3, 9] or extremal region [22, 10, 33]), or sliding window methods [12, 30, 6]. However, both groups of methods commonly suffer from *two main limitations* which significantly reduce their efficiencies and performance. First, text detection is built on *identification of individual characters or components*, making it difficult to *explore regional context* information. This

often results in a low recall where ambiguous characters are easily discarded. It also leads to a reduction in precision, by generating a large number of false detections. Second, multiple sequential steps make the system highly complicated, and errors are easily accumulated in the later steps.

Deep learning technologies have advanced the performance of text detection considerably [28, 4, 36, 38, 7, 6]. A number of recent approaches [36, 4, 7, 32] were built on Fully Convolutional Networks (FCN) [19], by producing pixel-wise prediction of text or non-text. We refer this group of methods as pixel-based text detectors, which cast previous character-based detections into the problem of text semantic segmentation. This allows them to explore rich regional context information, resulting in a stronger capability for detecting ambiguous text, and also reducing the number of false detections substantially. *In spite of effectively identifying rough text regions, these FCN-based approaches fail to produce accurate word-level predictions with a single model. The main challenge is to precisely identify individual words from a detected rough region of text.* As indicated in [28], the task of text detection often requires a higher localization accuracy than general object detection. To improve the accuracy, a coast-to-fine detection pipeline was developed, by cascading two FCN models [7, 36]. The second FCN produces word- or character-level predictions on a cropped text region detected by the first one. This inevitably increases system complexity: (i) it is highly heuristic yet complicated to correctly crop out regions of texts, words or characters from a predicted heatmap [7, 36]; (ii) multiple bottom-up steps are still required for constructing text lines/words [36, 32].

Recently, another group of text detectors was developed for direct prediction of text bounding boxes, by extending from the state-of-the-art object detectors, such as Faster R-CNN [23] and SSD [17]. They all aim to predict text boxes directly by sliding a window through the convolutional features [18, 28, 16, 4, 37, 24], and we refer them as box-based text detectors. *The box-based text detectors are often trained by simply using bounding-box annotations, which may be too coarse (high-level) to provide a direct and detailed supervision, compared to the pixel-based approaches where a text mask is provided.* This makes the models difficult to learn sufficient word information in details, leading to accuracy loss in one-shot prediction of words, particularly for those small-scale ones. Therefore, they may again come up with multiple post-processing steps to improve the performance.

1.1. Contributions

These related approaches inspire current work that aims to directly estimate word-level bounding boxes in one shot. We cast the cascaded FCNs detectors into a single model by introducing a new attention module, which enables a di-

rect mask supervision that explicitly encodes detailed text information in training, and functions on an implicit text region detection in testing. This elegantly bridges the gap between the pixel-based approaches and the box-based text detectors, resulting in a single-shot model that essentially works in a coarse-to-fine manner. We develop a hierarchical inception module which further enhances the convolutional features. The main contributions of the paper are three-fold.

First, we propose a novel text attention module by introducing a new auxiliary loss, built upon the aggregated inception convolutional features. It explicitly encodes strong text-specific information using a pixel-wise text mask, allowing the model to learn rough top-down spatial attention on text regions. This text regional attention significantly suppresses background interference in the convolutional features, which turns out to reduce false detections and also highlight challenging text patterns.

Second, we develop a hierarchical inception module which efficiently aggregates multi-scale inception features. An inception architecture with dilated convolutions [34] is applied to each convolutional layer, enabling the model to capture multi-scale image content. The multi-layer aggregations further enhance local detailed information and encode rich context information, resulting in stronger deep features for word prediction.

Third, the proposed text-specific modules are seamlessly incorporated into the SSD framework, which elegantly customizes it towards fast, accurate and single-shot text detection. This results in a powerful text detector that works reliably on multi-scale and multi-orientation text with single-scale inputs. It obtains state-of-the-art performance on the standard ICDAR 2013 and ICDAR 2015 benchmarks, with about 0.13s running time on an image of 704×704 .

2. Related Work

Previous works on scene text detection mainly focus on bottom-up approaches, which detect characters or text components from images by using hand-crafted features [27, 33, 10, 31] or sliding window methods [12, 30]. They often involve pixel-level binary classification of text/non-text and generate a text salience map. Then multiple bottom-up steps are designed to group text-related pixels into characters, which are further formed character pairs, and then text lines. Each step may be followed by a text/non-text filter or classifier. Most of these steps are built on the heuristic or hand-crafted features, e.g., gradient, stroke width [3, 9], covariance descriptor [9], etc. These bottom-up approaches are complicated, and identification of individual characters using low-level features is neither robust nor reliable.

Deep learning technologies have significantly advanced the performance of text detection in the past years [28, 4, 36, 16, 7]. These approaches essentially work in a sliding window fashion, with two key developments: (i) they lever-

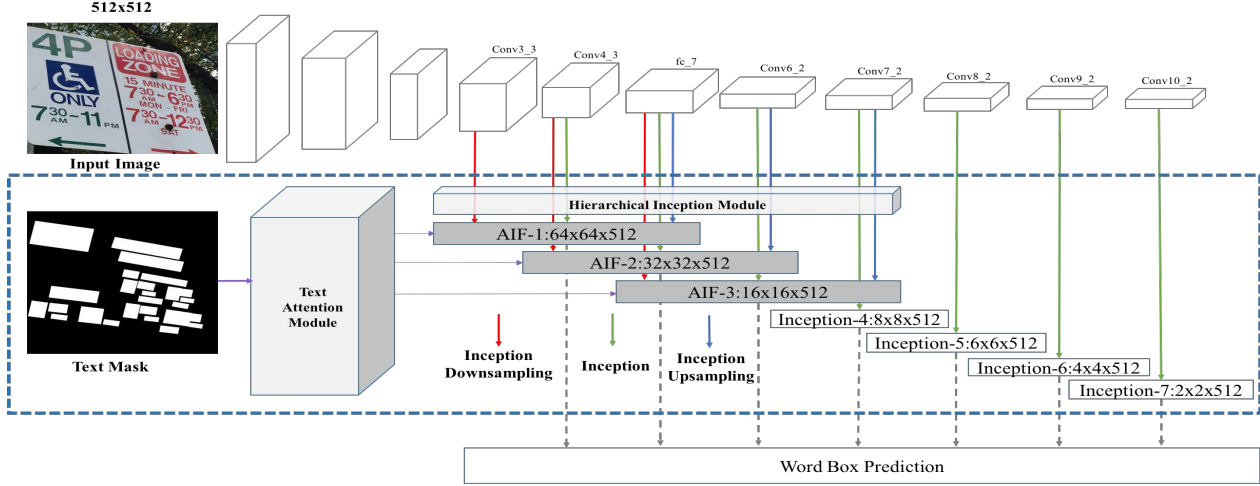


Figure 2: Our single-shot text detector contains three main parts: a convolutional part, a text-specific part, and a word box prediction part. We propose the text-specific part which comprises a Text Attention Module (TAM) and a Hierarchical Inception Module (HIM). The TAM introduces a new pixel-wise supervision of text, allowing the model to automatically learn text attention map which identifies rough text regions. The HIM aggregates multi-layer inception modules, and enhances the convolutional features towards text task.

age deep features, jointly learned with a classifier, to enable strong representation of text; (ii) sharing convolutional mechanism [19, 6] was applied for reducing the computational cost remarkably. With these two improvements, a number of Fully Convolutional Network (FCN) [19] based approaches have been proposed [36, 7, 32]. They compute pixel-wise semantic estimations of text or non-text, resulting in a fast text detector able to explore rich regional context information. However, these pixel-based text detectors are difficult to provide sufficient localization accuracy by using a single model. Furthermore, accurate segmentation of text from a predicted heatmap is complicated, and often requires a number of heuristic post-processing steps.

Our work also relates to most recent approaches [18, 28, 16, 4, 37, 24] which are extended from the state-of-the-art object detectors, such as SSD [17] and Fast R-CNN [23]. These approaches all aim to predict text bounding boxes from the convolutional features. Liao *et. al.* [16] proposed a TextBox by extending the SSD model for text detection, but their performance is limited by the SSD architecture which is designed for general object detection. Deep Matching Prior Network (DMPNet) was proposed in [18], by introducing quadrilateral sliding windows to handle multi-orientation text. Accurate text localization is achieved by using multi-step coarse-to-fine detection with post adjustments. Tian *et. al.* [28] proposed a Connectionist Text Proposal Network (CTPN) which detects a text line by predicting a sequence of fine-scale text components. The CTPN is difficult to work on multi-orientation text and requires bottom-up steps to group text components into text

lines. Gupta *et. al.* [4] developed a Fully Convolutional Regression Network (FCRN) which predicts word bounding boxes in an image. However, the FCRN requires three-stage post-processing steps which reduce the efficiency of the system considerably. For example, the post-processing steps take about 1.2s/image, comparing to 0.07s/image for bounding box estimations. Our work differs distinctly from these approaches by proposing two text-specific modules. It has a number of appealing properties that advance over these related methods: (i) it is a single-shot detector that directly outputs word bounding boxes, filling the gap between semantic text segmentation and direct regression of word boxes; (ii) it is highly efficient, and does not require any bottom-up or heuristic post-processing step; (iii) it works reliably on multi-orientation text; (iv) it is fast yet accurate, and significantly outperforms those related approaches on the standard ICDAR 2015 benchmark.

3. Methodology

We present details of the proposed single-shot text detector, which directly outputs word-level bounding boxes without post-processing, except for a simple NMS. Our detector is composed of three main parts: a convolutional component, a text-specific component, and a box prediction component. The convolutional and box prediction components mainly inherit from SSD detector [17]. We propose the text-specific component which contains two new modules: a text attention module and a hierarchical inception module. The convolutional architecture of the SSD is extended from the 16-layer VGGNet [25], by replacing the fully-connected

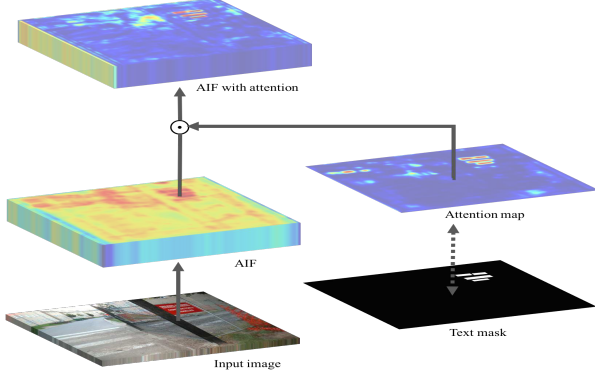


Figure 3: Text attention module. It computes a text attention map from Aggregated Inception Features (AIFs). The attention map indicates rough text regions and is further encoded into the AIFs. The attention module is trained by using a pixel-wise binary mask of text.

(FC) layers with several convolutional layers [17]. The proposed modules can be easily incorporated into the convolutional component and box prediction component of the SSD, resulting in an end-to-end trainable model, as shown in Fig. 2. The two text-specific modules elegantly customize the SSD framework towards accurate word detection. Compared to most recent methods [18, 28, 16], we show experimentally that our particular designs provide a more principled solution that generalizes better.

3.1. Text Attention Mechanism

Our attention module is able to automatically learn rough spatial regions of text from the convolutional features. This attention to text is then directly encoded back into the convolutional features, where text-related features are strongly enhanced by reducing background interference in the convolutional maps, as shown in Fig. 3.

The attention module is built on the Aggregated Inception Feature (AIF) (described in Sect. 3.2). It generates a pixel-wise probability heatmap which indicates the text probability at each pixel location. This probability heatmap is referred as the attention map which has an identical size of an input image and will be downsampled for each prediction layer. The attention module includes two 3×3 convolutional layers with pad 1, one deconvolution (upsampling with bilinear operation) layer which connects the AIF to the attention map. Then the attention map is generated by using a softmax activation function on the de-convoluted features. Specifically, given an input image of 512×512 , we get the first-layer AIF features, $F_{AIF_1} \in R^{64 \times 64 \times 512}$. The atten-

tion map, $\alpha^+ \in R^{512 \times 512}$, is computed as,

$$D_{AIF_1} = \text{deconv}_{3 \times 3}(F_{AIF_1}), \quad (1)$$

$$\bar{D}_{AIF_1} = \text{conv}_{1 \times 1}(D_{AIF_1}), \quad (2)$$

$$\alpha = \text{softmax}(\bar{D}_{AIF_1}). \quad (3)$$

where $D_{AIF_1} \in R^{512 \times 512 \times 512}$ is the de-convoluted feature maps, which are further projected to 2-channel maps, $\bar{D}_{AIF_1} \in R^{512 \times 512 \times 2}$ using 1×1 kernels, followed by a softmax function. Then the positive part of the softmax maps, α^+ , is the attention map, indicating pixel-wise possibility of text. The attention map is further encoded into the AIF by simply resizing it as with spatial size,

$$\hat{\alpha}^+ = \text{resize}(\alpha^+), \quad (4)$$

$$\hat{F}_{AIF_1} = \hat{\alpha}^+ \odot F_{AIF_1}. \quad (5)$$

where $\hat{\alpha}^+ \in R^{64 \times 64}$ is the resized attention map, and \odot indicates element-wise dot production across all channel of the AIF maps. \hat{F}_{AIF_1} is the resulted feature maps with encoded text regional attention. The AIFs with and without text attention are shown in Fig. 3, where text information is clearly presented when the attention is encoded.

The text attentional information is learned automatically in the training process. We introduce an auxiliary loss which provides a direct and detailed supervision of text via a binary mask that indicates text or non-text at each pixel location. A softmax function is used to optimize this attention map toward the provided text mask, explicitly encoding strong text information into the attention module. Notice that the proposed attention module is formulated in a unified framework which is trained end to end by allowing for computing back-propagations through all layers.

The proposed attention module sets this work apart from both previous pixel-wise and box-wise text detectors. It elegantly handles the main limitations of both groups of methods, resulting in an efficient single-shot text detector that produces accurate word-level text detection. Efficiency of the proposed attention module is demonstrated in Fig. 4, where detection results by a baseline model and a with-attention model are presented. Obviously, the proposed attention module improves the performance at three aspects: (i) it reduces the number of false detections; (ii) it allows the model to detect more ambiguous texts; (iii) it improves the word-level detection accuracy.

3.2. Hierarchical Inception Module

In a CNN model, convolutional features in a lower layer often focus on local image details, while the features in a deeper layer generally capture more high-level abstracted information. The SSD detector predicts object bounding boxes at multi-layer convolutional maps, allowing it to localize multi-scale objects from a single-scale input. Texts

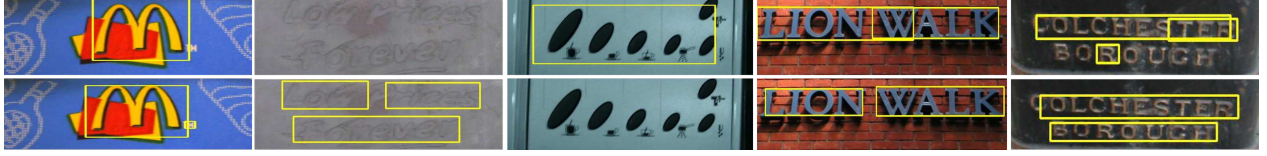


Figure 4: We compare detection results of the baseline model and the model with our text attention module (TAM), which enables the detector with stronger capability for identifying extremely challenging text with a **higher word-level accuracy**.

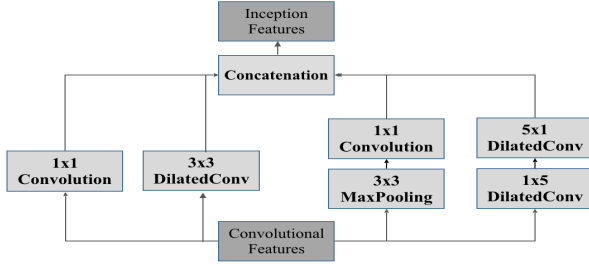


Figure 5: **Inception module**. The convolutional maps are processed through **four different convolutional operations**, with Dilated convolutions [34] applied.

often have large variations in scale with significantly different aspect ratios, making the single-scale text detection highly challenging. In recent work [28], a **layer-wise RNN** is incorporated into a convolutional layer, allowing the detector to explore rich context information throughout the whole line. This **RNN-based text detector** is **powerful** to detect **near-horizontal text lines** but is **difficult** to work reliably on **multi-orientation texts**.

Inspired by the inception architecture of GoogleNet [26], we propose a **hierarchical inception module** able to aggregate stronger convolutional features. First, we develop a similar inception module which is applied to each convolutional layer, as shown in Fig. 5. This allows it to capture **richer context information** by using **multi-scale receptive fields**. Second, we **aggregate the inception features** from **multiple layers** and **generate final Aggregated Inception Features (AIF)**.

Details of inception building block are described in Fig. 5. It is applied to those convolutional layers which are used to **predict word bounding boxes**. The convolutional maps in a layer are processed through four different convolutional operations: 1×1 -conv, 3×3 -conv, 3×3 -pool with 1×1 -conv, and 5×5 -conv. The 5×5 -conv is decomposed into 1×5 and 5×1 convolution layers. Dilated convolutions [34], which support an exponential expansion of receptive field without loss of **resolution or coverage**, are applied. Each convolutional operation reduces the number of feature channels to 128. The final inception features are generated by simply concatenating four 128-channel features, resulting in 512-channel inception features. By using mul-



Figure 6: Comparisons of baseline model and Hierarchical Inception Module (HIM) model. The HIM allows the detector to handle extremely challenging text, and also improves **word-level detection accuracy**.

iple convolutional operations with **channel concatenation**, the inception features have **multi-scale receptive fields** and thus can focus on image content in a wide range of scales.

Motivated from HyperNet [5], we further **enhance the convolutional features** by **aggregating multi-layer inception features**, which generates final AIFs at three key convolutional layers, as shown in Fig. 2. Each AIF is computed by fusing the inception features of current layer with **two directly adjacent layers**. Down-sampling and up-sampling are applied to the lower layer and higher layer, respectively. These sampling operations ensure same feature resolutions for **three inception features**, which are combined together using **channel concatenation**.

The proposed hierarchical inception module is related to the **skip architecture** developed in [19], which combines multi-layer convolutional features for handling multi-scale objects. We propose a **two-step aggregation approach** by leveraging efficient **layer-wise inception module**. This results in a **stronger AIF**, with richer local details encoded and a more powerful multi-scale capability. Both improvements are important for text task, allowing our model to identify very **small-scale text** and work reliably on the **multi-scale text** which often has a wider range of scales than the general objects. Compared to **layer-wise RNN method** developed in [28], the AIF encodes more local detailed information and generalizes better to multi-orientation text. The

Table 1: Scales and aspect ratios of the default box applied for each AIF or Inception layer.

Scales of Default Box						
AIF-1	AIF-2	AIF-3	Inc-4	Inc-5	Inc-6	Inc-7
7.7	38.4	69.1	102.4	194.6	286.7	378.9
17.9	48.6	79.4	133.1	225.3	317.4	409.6
28.2	58.9	89.6	163.8	256.0	348.2	440.3
Aspect Ratios: {0.5, 1, 2, 3, 5, 7, 9, 11}						

effectiveness of the proposed hierarchical inception module is demonstrated in Fig. 6.

3.3. Word Prediction Module

The proposed text-specific modules are directly incorporated into the SSD framework. It can be trained by simply following the SSD, with slight modifications on the box prediction part. As in [17], we use a softmax function for binary classification of text or non-text, and apply the smooth- l_1 loss for regressing 5 parameters for each word bounding box, including a parameter for box orientation. Our model predicts N word bounding boxes at each spatial location of the inception or AIF maps. The predictions are computed through all inception and AIF maps shown in Fig. 2. N is the number of the pre-defined default word boxes, which can be pre-computed. A default box is defined by its scale and aspect ratio. Since different inception maps have various sizes of receptive fields, the scale of a default box is varied over different inception or AIF maps. Its length is measured by the number of pixels in the input image. To better match the default boxes with ground truth ones, we use three different scales for each layer, and the scales for all inception and AIF layers are listed in Table 1.

Aspect ratios of a default box are set in a wide range (in Table 1), due to the significant variances of them for text. Same aspect ratios are used in all layers. In order to handle multi-orientation text, a scale can be the width or height of a default box. This results in 45 default boxes in total for each layer, which allow the detector to handle large shape variances of text. The center of a default box is identical to the current spatial location of the inception/convolution maps.

4. Experimental Results

Our methods are evaluated on three standard benchmarks, the ICDAR 2013 [14], ICDAR 2015 [15] and COCO-Text dataset [29]. The effectiveness of each proposed component is investigated by producing exploration studies. Full results are compared with the state-of-the-art performance on the three benchmarks.

4.1. Datasets and Implementation Details

Datasets. The ICDAR 2013 [14] consists of 229 training images and 233 testing images, with word-level annotations provided. It is the standard benchmark for evaluating near-horizontal text detection. We use two standard evaluation protocols: the new ICDAR13 standard [15] and the DetEval [14]. Our results were obtained by uploading the predicted bounding boxes to the official evaluation system. The ICDAR 2015 (Incidental Scene Text Challenge 4) [15] was collected by using Google Glass and it has 1,500 images in total: 1,000 images for training and the remained 500 images for testing. This new benchmark was designed for evaluating multi-orientation text detection. Word-level annotations are provided and each word is labeled by the coordinates of its four corners in a clockwise manner. This dataset is more challenging and has images with arbitrary orientations, motion blur, and low-resolution text. We evaluate our results based on the online evaluation system [15]. The COCO-Text [29] is the largest text detection dataset, which has 63,686 annotated images in total: 43,686 for the training and the rest 20,000 for testing.

Training datasets. Our training samples were collected from the training sets of the ICDAR 2013 and ICDAR 2015. We also added images harvested from Internet as the training data and manually labelled them with word level annotation. We have 13,090 training images in total. We did not use the training data from the COCO-Text.

Implementation details. Our detection network is trained end-to-end by using the mini-batch stochastic gradient descent algorithm, where the batch size is set to 32, with a momentum of 0.9. We initialize our model using the pre-trained model in [17]. The learning rate is set to 0.001 and is decayed to its $\frac{1}{10}$ after 15,000 iterations. The model is trained by fixing the first four convolutional layers, and the training is stopped when the loss no longer decreases.

Data augmentation is used by following the SSD [17]. We take a similar strategy by randomly sampling a patch from an image and set the minimum Jaccard overlap with ground truth word bounding boxes to {0.1, 0.3, 0.5, 0.7, 0.9}. The sampled patches are then re-sized to 704×704 and are randomly mirrored with color distortion. The NMS threshold and confidence threshold are set to 0.3 and 0.7, respectively. Our methods were implemented using Caffe [13], with TITAN X GPUs.

4.2. Exploration Study

We evaluate the effectiveness of the proposed text attentional module (TAM) and hierarchical inception module (HIM) individually. We compare them with our baseline model, which is extended from the SSD framework by simply modifying the word prediction module (as described in Sect. 3.3). Experimental results on the ICDAR 2013 are compared in Table 2, in the terms of precision, recall

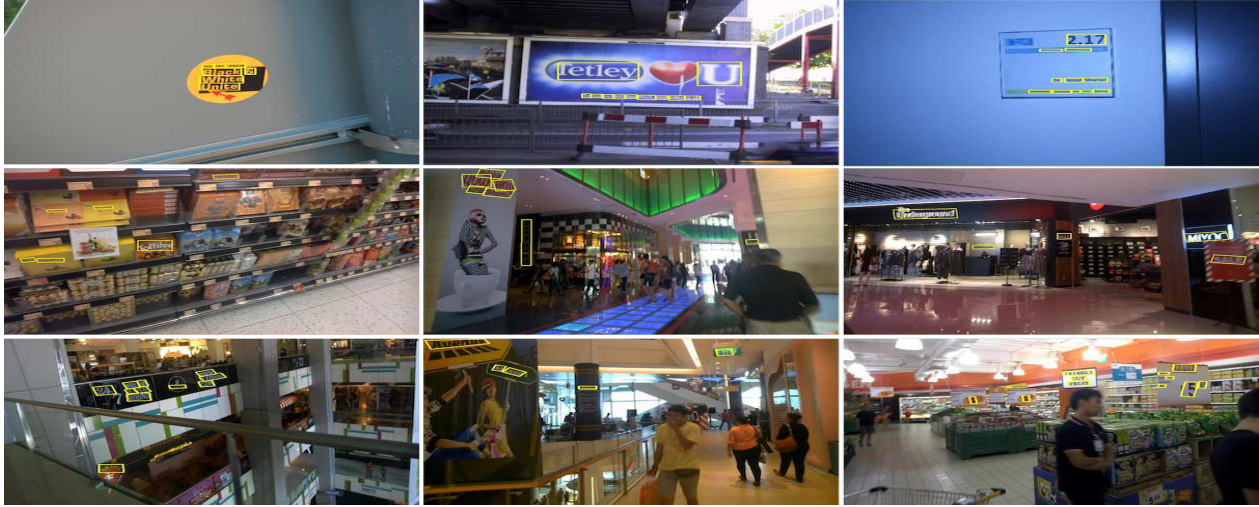


Figure 7: Detection results by the proposed single-shot text detector.

Table 2: Exploration study on the ICDAR 2013 dataset, based on the new ICDAR13 standard.

Method	Recall	Precision	F-measure
SSD [17]	0.60	0.80	0.68
Textboxes [16]	0.74	0.86	0.80
Baseline Model	0.78	0.87	0.82
TAM	0.83	0.87	0.85
HIM	0.82	0.87	0.84
TAM+HIM	0.86	0.88	0.87

and f-measure. First, we compare our baseline model with original SSD trained on text data, without any modification of the model. Our baseline model obtains significant performance improvements and outperforms recent TextBoxes [16], which is also extended from the SSD. Second, the proposed TAM and HIM both achieve large improvements in recall, indicating that the **proposed text-specific modules** can **increase model accuracy** and **improve** its **capability** for identifying **challenging words**, as shown in Fig. 4 and 6. Third, by incorporating both TAM and HIM into a single model, the **final single-shot detector** further **improves** the **performance in both recall and precision**, obtaining a final F-measure of 0.87. This suggests that both TAM and HIM can compensate for each other in our model. Notice that the performance on the ICDAR2013 is saturated, and the improvement obtained by each component is significant.

4.3. Comparisons with state-of-the-art methods

Qualitative results. Detection results on a number of very challenging images are demonstrated in Fig. 7, where our detector is able to correctly identify many extremely challenging words, some of which are even difficult for hu-

man. It is important to point out that the word-level detection by our detector is particularly accurate, even for those very small-scale words that are closed to each other.

On the ICDAR 2013. We compare our performance against recently published results in Table 3. All our results are reported on single-scale test images. On the ICDAR 2013 which is designed for near-horizontal text detection, our detector achieves the state-of-the-art performance on both evaluation standards. By using the ICDAR 2013 standard, which focuses on word-level evaluation, our method obtains an F-measure of 0.87, outperforming all other methods compared, including recent FCRN [4], CTPN [28], and TextBoxes [16]. By using the DetEval standard, our method is comparable with the CTPN, and again has substantial improvements over the others. It has to point out that the DetEval standard allows for text-line level evaluation as well, and is less strict than the ICDAR 2013 standard. Our method encourages a more accurate detection in word level, resulting in better performance in the ICDAR 2013 standard.

In addition, we further evaluate our method for end-to-end word spotting on the ICDAR 2013 by directly combining our detector with recent word recognition model presented in [6]. We obtain an accuracy of 0.83 on *generic* case, which is comparable to recent results: 0.79 in [11] and 0.85 in [4].

On the ICDAR 2015. The ability to work on multi-orientation text is verified on the ICDAR 2015. Our method obtains an F-measure of 0.77, which is a remarkable improvement over 0.61 achieved by CTPN [28]. This clearly indicates that our method generalizes much better to **multi-orientation** text than the CTPN. On the ICDAR 2015, our method got the state-of-the-art performance in terms of recall, precision, and F-measure, surpassing recently published results in [18] (0.71 F-measure) by a large margin.

Table 3: Comparisons of the state-of-the-art results on the ICDAR 2013 and ICDAR 2015. The results are reported in the terms of Recall (R), Precision (P) and F-measure (F)

ICDAR 2013 dataset								ICDAR 2015 dataset			
Method	ICDAR Standard			DetEval			Time(s)	Method	R	P	F
	R	P	F	R	P	F					
SSD [17]	0.60	0.80	0.68	0.60	0.80	0.69	0.10	Deep2Text-MO	0.32	0.50	0.39
Yin [33]	0.66	0.88	0.76	0.69	0.89	0.78	0.43	HUST_MCLAB	0.44	0.38	0.41
Neumann [20]	0.72	0.82	0.77	-	-	-	0.40	AJOU	0.47	0.47	0.47
Neumann [21]	0.71	0.82	0.76	-	-	-	0.40	NJU-Text	0.36	0.73	0.48
FASText [1]	0.69	0.84	0.77	-	-	-	0.15	CASIA USTB	0.40	0.62	0.48
Zhang [35]	0.74	0.88	0.80	0.76	0.88	0.82	60	StradVision1	0.46	0.53	0.50
TextFlow [27]	0.76	0.85	0.80	-	-	-	0.94	StradVision2	0.37	0.77	0.50
Text-CNN [8]	0.73	0.93	0.82	0.76	0.93	0.84	4.6	MCLAB.FCIN [36]	0.43	0.71	0.54
FCRN [4]	0.76	0.94	0.84	0.76	0.92	0.83	1.27	CTPN [28]	0.52	0.74	0.61
TextBoxes [16]	0.74	0.86	0.80	0.74	0.88	0.81	0.09	Yao <i>et. al.</i> [32]	0.57	0.72	0.64
CTPN [28]	0.73	0.93	0.82	0.83	0.93	0.88	0.14	DMPNet [18]	0.68	0.73	0.71
Proposed method	0.86	0.88	0.87	0.86	0.89	0.88	0.13	Proposed method	0.73	0.80	0.77

Table 4: Comparisons on the COCO-text dataset

COCO-text dataset			
Method	Recall	Precision	F-score
Baseline model C	0.05	0.19	0.07
Baseline model B	0.11	0.90	0.19
Baseline model A	0.23	0.84	0.36
Yao [32]	0.27	0.43	0.33
Proposed method	0.31	0.46	0.37

In all implementations, our method obtains a higher recall than the others. This suggests that the proposed text-specific modules enable our detector with stronger capability for detecting extremely challenging text in a high word-level accuracy, including the very small-scale, significantly-slant or highly-ambiguous words, as shown in Fig. 7.

On the COCO-Text. We further evaluate our detector on the COCO-Text [29], which is a large-scale text dataset. We achieve state-of-the-art performance with an F-score of 0.37, which improves recent state-of-the-art result [32] slightly. This demonstrates strong generalization ability of our method to work practically on **large-scale images** in the **unseen scenarios**. Again, our method achieves a significantly higher recall than all compared approaches.

Running time. We compare running time of various methods on the ICDAR 2013 (in Table 3). Our detector achieves running time of 0.13s/image using a single GPU, which is slightly faster than CTPN using 0.14s/image. It is slightly slower than TextBoxes [16], but has substantial performance improvements. In addition, the TextBoxes was not tested on the multi-orientation text. Besides, FCRN [4] predicts word boxes at 0.07s/image, but it takes 1.2s/image for post-processing the generated boxes. All recent CNN-

based approaches are compared on GPU time, which has become the main stream with success of deep learning technologies. The fastest CPU based text detector is FASText [1], using 0.15s/image. But its performance is significantly lower than recent CNN-based approaches.

5. Conclusion

We have presented a **fast yet accurate text detector** that predicts **word-level bounding boxes in one shot**. We proposed a novel text attention mechanism which encodes **strong supervised information of text in training**. This enables the model to automatically learn a text attentional map that implicitly identifies rough text regions in testing, allowing it to work essentially in a coarse-to-fine manner. We developed a hierarchical inception module that efficiently **aggregates multi-scale inception features**. This further enhances convolutional features by encoding more local details and stronger context information. Both text-specific developments result in a **powerful text detector** that works reliably on the **multi-scale and multi-orientation text**. Our method achieved new state-of-the-art results on the ICDAR 2013, ICDAR 2015 and COCO-Text benchmarks.

Acknowledgement

This work is supported in part by National Science Foundation (CNS-1624782, OAC-1229576, CCF-1128805), National Institutes of Health (R01-GM110240), Industrial Collaboration Project (Y5Z0371001), National Natural Science Foundation of China (U1613211, 61503367) and Guangdong Research Program (2015B010129013, 2015A030310289).

References

- [1] M. Busta, L. Neumann, and J. Matas. Fasttext: Efficient unconstrained scene text detector, 2015. In ICCV. 8
- [2] H. Cho, M. Sung, and B. Jun. Canny text detector: Fast and robust scene text localization algorithm, 2016. In CVPR. 1
- [3] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform, 2010. In CVPR. 1, 2
- [4] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images, 2016. In CVPR. 1, 2, 3, 7, 8
- [5] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization, 2015. In CVPR. 5
- [6] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang. Reading scene text in deep convolutional sequences, 2016. In AAAI. 1, 2, 3, 7
- [7] T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network, 2016. arXiv:1603.09423. 2, 3
- [8] T. He, W. Huang, Y. Qiao, and J. Yao. Text-attentional convolutional neural networks for scene text detection. *IEEE Trans. Image Processing (TIP)*, 25:2529–2541, 2016. 1, 8
- [9] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors, 2013. In ICCV. 1, 2
- [10] W. Huang, Y. Qiao, and X. Tang. Robust scene text detection with convolutional neural networks induced msr trees, 2014. In ECCV. 1, 2
- [11] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision (IJCV)*, 116:1–20, 2016. 7
- [12] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting, 2014. In ECCV. 1, 2
- [13] 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 ACM MM. 6
- [14] D. Karatzas et al. Icdar 2013 robust reading competition, 2013. In ICDAR. 6
- [15] D. Karatzas et al. Icdar 2015 competition on robust reading, 2015. In ICDAR. 6
- [16] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network, 2017. In AAAI. 2, 3, 4, 7, 8
- [17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector, 2016. In ECCV. 2, 3, 4, 6, 7, 8
- [18] Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection, 2017. In CVPR. 1, 2, 3, 4, 7, 8
- [19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation, 2015. In CVPR. 2, 3, 5
- [20] L. Neumann and J. Matas. Efficient scene text localization and recognition with local character refinement, 2015. In ICDAR. 8
- [21] L. Neumann and J. Matas. Real-time lexicon-free scene text localization and recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, 38:1872–1885, 2015. 8
- [22] L. Neumann and K. Matas. Real-time scene text localization and recognition, 2012. In CVPR. 1
- [23] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks, 2015. In NIPS. 2, 3
- [24] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments, 2017. In CVPR. 2, 3
- [25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. In ICLR. 3
- [26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions, 2015. In CVPR. 5
- [27] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, and C. Lim Tan. Text flow: A unified text detection system in natural scene images, 2015. In ICCV. 2, 8
- [28] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network, 2016. In ECCV. 1, 2, 3, 4, 5, 7, 8
- [29] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images, 2016. arXiv:1601.07140. 6, 8
- [30] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks, 2012. In ICPR. 1, 2
- [31] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images, 2012. In CVPR. 2
- [32] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction, 2016. arXiv:1606.09002. 2, 3, 8
- [33] 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, 8
- [34] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions, 2016. In ICLR. 2, 5
- [35] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes, 2015. In CVPR. 8
- [36] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks, 2016. In CVPR. 1, 2, 3, 8
- [37] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: An efficient and accurate scene text detector, 2017. In CVPR. 2, 3
- [38] S. Zhu and R. Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification, 2016. In CVPR. 1, 2