ABSTRACT
Scene text detection is a critical prerequisite for many fascinating applications for vision-based intelligent robots. Existing methods detect texts either using the local information only or casting it as a semantic segmentation problem. They tend to produce a large number of false alarms or cannot separate individual words accurately. In this work, we present an elegant segmentation-aided text detection solution that predicts the word-level bounding boxes using an end-to-end trainable deep convolutional neural network. It exploits the holistic view of a segmentation network in generating the text attention map (TAM) and uses the TAM to refine the convolutional features for the MultiBox detector through a multiplicative gating process. We conduct experiments on the large-scale and challenging COCO-Text dataset and demonstrate that the proposed method outperforms state-of-the-art methods significantly.

Index Terms— Text Detection, Natural Scenes, Deep Learning

1. INTRODUCTION
The ability to read texts in a natural scene is a highly desirable capability in many interesting and practical applications, such as assistance to visually impaired people, environment understanding and automatic navigation for smart robots, and visual translation, etc. Thus, research on scene text detection and recognition has drawn increasing attention in the computer vision community. Scene text detection is a crucial prerequisite for numerous subsequent recognition tasks. Generally speaking, text detection is a challenging task, which should provide word-level bounding boxes as the desired output. Its challenges originate from variabilities in fonts, scales and layout, as well as complex backgrounds and perspective distortion.

Conventionally, hand-crafted features were adopted in a text detection system [1, 2]. Recently, deep-learning-based methods [3, 4, 5] are dominating in this field. They have achieved remarkable performance on several well-known benchmark datasets [6, 7]. They can be divided into two categories. The first category follows the path of generic object detection and predicts word-level bounding boxes directly. Examples include [8, 9]. However, they cannot produce satisfactory results without the use of heavy post-processing techniques for the removal of false alarms. This is because predictions are made based on local regions with limited access to the contextual cues. The second category [10, 11, 12] formulates the text detection problem as a semantic segmentation problem. These methods predict the probability of each pixel belonging to a bounding box of a text block using the fully convolutional networks (FCNs). They are robust with respect to complex backgrounds because of the adoption of a more global view. Nevertheless, they often perform poorly in separating individual words.

In this work, we propose a novel segmentation-aided text detection solution, where the segmentation and detection networks are integrated into one single network. As a result, it combines a segmenter’s robustness and a detector’s deftness. The segmentation module aims to produce a “Text Attention Map” (TAM), which is a dense heat map indicating the probability of text’s presence at each pixel. Upon obtaining the TAM, we refine the feature maps for the detection module through a multiplicative gating process. The training process is to learn the TAM generation and MultiBox text detection jointly. By injecting the semantic localization information into the input convolutional features of detectors, the false positives can be effectively suppressed. Thus, the accuracy of the finally detected word-level bounding boxes is significantly improved. The proposed method achieves the state-of-the-art performance on the COCO-Text dataset [13], which is today’s largest and most challenging benchmark dataset. The overall system is end-to-end trainable and highly efficient.

2. RELATED WORK
Traditionally, scene text detection heavily relied on hand-crafted features. For example, the Maximally Stable Extremal Regions (MSER) [1] and the Stroke Width Transform (SWT) [2] were used to extract text-specific low-level features.

With the great success of convolutional neural networks (CNNs) in many computer vision tasks, CNNs have become popular in text detection. Recent text detection methods [14, 8, 9] have been inspired by the generic object detection frameworks, e.g. [15, 16, 17, 18]. However, due to a large variation in text fonts, scales, and aspect ratios, etc, a direct application of general object detection algorithms without domain-specific modification fails to produce satisfactory results. Jaderberg et al. [14] followed the R-CNN [15] framework yet employed the combination of two means in word proposal generation. Based on the Faster R-CNN [16] framework, CPTN [8] adopted a vertical anchor mechanism to generate text-component (rather than word-level) proposals and constructed a joint CNN-RNN model to predict the horizontal span of text lines. TextBoxes [9] finetuned the SSD framework [18] and re-designed the aspect ratios and the tiling scheme of pre-defined default boxes. On one hand, as compared with traditional methods, these methods achieved better performance on relatively simple benchmark datasets [6, 7]. On the other hand, detection results of these methods are made based on local regions in the image without enough contextual cues. When the environment becomes more challenging, they are not robust against text-like patterns such as fences, windows, leaves, etc. Post-processing is usually needed to remove false positives.

Text detection can be cast as a semantic segmentation problem. Recent advances in semantic segmentation offer new tools for text detection. Fully convolutional networks (FCNs) [19, 20] preserve the spatial information in the feature map and enable robust pixel-wise prediction for input images of an arbitrary size. Both Zhang et al. [10] and CCTN [11] adopted an FCN to find coarse text block regions and cascaded another FCN to refine segmentation results for each text block. Yao et al. [12] trained an FCN to predict the binary
mask for the text line block and individual characters simultaneously. Since FCNs have a large receptive field and incorporate more context information, they are robust against complex backgrounds and produce very few false positives. At the same time, it is difficult to detect fine-scale individual text lines or words due to the use of a large receptive field. Extra character-level bounding box annotation can be used to mitigate this shortcoming [10, 12]. However, tremendous post-processing efforts are still required to split text lines, group characters, and partition words for satisfactory performance.

It is apparent that the detection network and the semantic segmentation network are complimentary to each other in terms of recall and precision scores. It is natural to combine them into one single system. Qin et al. [21] adopted a straightforward cascade approach. That is, text block regions produced by an FCN are cropped out of the image and resized to a square shape of a fixed size. Then, a YOLO-like [17] detection network is applied to each cropped patch to detect word-level bounding boxes.

In this work, we take an unprecedented perspective by integrating the segmenter and the detector in a holistic way. Our method does not process each cropped text block image patch individually. It is more efficient in both computation and memory. Thus, it offers a more favorable solution in the resource-limited applications such as intelligent robots.

3. METHODOLOGY

In the proposed segmentation-aided text detection system, DeepLab V2 [20] and SSD [18] are integrated seamlessly into one CNN in the form of two modules: text attention map generator and text-attentional MultiBox detector. The details of the two modules and the overall architecture are explained in this section.

3.1. Text Attention Map (TAM) Generation

A text attention map (TAM) is a two-dimensional heat map with values ranging in [0, 1], which represent the probability of the existence of text. We adopt the recently developed segmentation network DeepLab V2 [20] to generate the TAM. On the top of several fully convolutional layers, we introduce the Atrous Spatial Pyramid Pooling (ASPP) layer as in DeepLab V2 [20]. The ASPP layer produces multi-scale representations by combining feature responses from parallel atrous convolution layers with different sampling rates. The softmax function is used to generate the TAM. This process is illustrated in the upper part of Fig. 1.

The ground-truth attention mask is converted from the bounding boxes annotations. We labeled class “1” for the pixels inside the text bounding boxes and class “0” elsewhere. The loss function \( L_{\text{seg}} \) is the mean of the cross-entropy terms for each spatial position in the predicted mask.

3.2. Text-attentional MultiBox Text Detection

This module aims to predict the class label and location of word-level bounding boxes. The architecture of this module is illustrated in the lower part of Fig. 1.

Text Attention Gating. Inspired by the gating mechanism of long short-term memory (LSTM), we design the text attention gating layer. The generated TAM in Sec. 3.1 now serves as a soft attention gating signal map, where the value at \((i, j)\) describes how much attention the detector should pay to the location \((i, j)\) in the input feature map. Zero value indicates no attention necessary while one means full attention need to be drawn. Given the input convolutional feature blob of dimension \(W_f \times H_f \times C_f\), the attention gating layer first resizes the TAM to \(W_f \times H_f \times 1\) using bilinear interpolation. Then for each channel in the input convolutional feature blob, we perform an element-wise multiplication operation between the feature map and the resized TAM. The output is a refined attention-aware feature map, which is the key component for the false positive suppression in the subsequent text detection.

MultiBox Text Detector. Given the refined attention-aware feature maps, we follow SSD [18] to attach several convolutional layers to predict class confidence scores and spatial offsets for some predefined default bounding boxes. At each feature map cell location \((i, j)\), which associates with a default box \(b_0 = (x_0, y_0, w_0, h_0)\), we apply two \(3 \times 3\) convolutional filters to either predict a confidence score \(c\) for categories (text or non-text) of the associated box, or regress the spatial offsets \(t = (\Delta x, \Delta y, \Delta w, \Delta h)\) relative to the associated box. The predicted box is \(b = (x, y, w, h)\), where

\[
\begin{align*}
x &= x_0 + \Delta x, \quad y = y_0 + \Delta y, \quad w = w_0 \exp(w), \quad h = h_0 \exp(h).
\end{align*}
\]

Training Objective. We define and associate a set of default boxes with various scales and aspect ratios to each feature map cell, for several feature maps extracted from base CNN at different depths. In the training stage, the ground truth boxes are matched to the default boxes according to the Jaccard overlap, following the matching strategy described in [18]. The training loss function \(L_{\text{det}}\) of the detector is a weighted sum of the average classification loss \((L_{\text{cls}})\) and localization loss \((L_{\text{loc}})\) for each matched default box:

\[
L_{\text{det}} = \frac{1}{N} (L_{\text{cls}} + \alpha L_{\text{loc}}),
\]

where \(N\) is the number of matched default boxes, and \(\alpha\) is empirically set to 1. Specifically, \(L_{\text{cls}}\) is the softmax loss over 2-class confidence scores \(c\), and \(L_{\text{loc}}\) is the smooth L1 loss [22] between the predicted box and the corresponding ground truth box location parameters. Note that we have to insert a Stop Gradient layer prior to the attention gating operation to isolate the two modules in the backward pass, preventing gradients back-propagation from the detector to the TAM generator.

3.3. Overall Network Architecture.

The overall end-to-end trainable architecture of the segmentation-aided text detection network is illustrated in Fig. 2.
Inspired by DSSD [23], we use the fully convolutional part of ResNet-101 [24] as the base network, removing the last global average pooling layer and the 1000-way fully connected layer. In order to increase the resolution of the feature map, we change the effective output stride of the conv5 block to 16 pixels instead of 32 pixels, which is beneficial for the dense prediction tasks like segmentation and detection [20]. We append four more residual blocks (namely, conv6 to conv9) after the conv5 block so that we can detect text at various scales. The input feature map of the TAM generation module is pulled from the conv5 block. We employ the attention gating for the feature maps from conv3, and conv5 to conv9, respectively, and perform MultiBox text detection based on the refined feature maps.

The overall loss function is a weighted sum of the segmentation loss ($L_{\text{seg}}$) and the detection loss ($L_{\text{det}}$):

$$L = \beta L_{\text{seg}} + L_{\text{det}},$$

where $\beta$ is set to 10 by cross-validation.

At the very end of the network, the non-maximum suppression (NMS) technique is adopted to aggregate and post-process the predictions from different layers to get final detection results.

4. EXPERIMENTS

To verify the effectiveness of the proposed segmentation-aided text detector and compare with existing methods, we conducted experiments on the most challenging benchmark to date, i.e. COCO-Text [13].

4.1. Datasets and Implementation Details

The COCO-Text is the largest publicly available dataset for text detection and recognition in natural images. The images are harvested from the Microsoft COCO dataset [25]. The full set of COCO-Text v1.4 annotated a total of 63,686 images with 145,859 text instances (training: 43,686/118,309, validation: 10,000/27,550, test: 10,000/no public annotations).

The annotated text regions are classified into two categories according to the legibility, which indicates whether a text can be read. Illegible texts are usually too blurry or far away from the viewpoint. In the smart robot applications, detecting legible texts are more of interest. Therefore, we selected a subset of the images that have at least one legible text instances and filtered the ground truth annotations to keep legible text instances only. The resulting training set has 14,324 images, and the validation set has 3,346 images. We will refer this subset as “COCO-Text-Legible” while the original dataset as “COCO-Text-Full” in the rest of the paper.

We implemented the proposed segmentation-aided text detector in Tensorflow [26] version 1.0. We reused some code from a re-implementation of the original VGG-16-based SSD in Tensorflow\footnote{https://github.com/balancap/SSD-Tensorflow}. We substituted the VGG base network with the ResNet-101 model\footnote{https://github.com/tensorflow/models/tree/master/slim}. Both the VGG-16 and the ResNet-101 model were pretrained on the ImageNet dataset [27]. We also conducted experiments with the default boxes that have text-specific aspect ratios and vertically denser tiling strategy proposed in TextBoxes [9].

The training process has two stages. In the first stage, we trained the text attention map generation module for 50k iterations, using Adam [28] optimizer with a fixed learning rate $10^{-5}$. The second stage initializes the network from the trained network in the first stage. We added the text-attentional MultiBox detectors and randomly initialized their associated filter weights. We applied the polynomial decay with power 0.5 to the learning rate, and it decays from $10^{-5}$ to $10^{-7}$ in 100 epochs. We trained the overall network end-to-end for 270k iterations.

In the training stage, data augmentation and hard negative example mining techniques as in [18] were used. At inference time, we resized the input images to $513 \times 513$. The NMS threshold was 0.45, and the top 30 detections were kept. The total inference time is 0.3s per image on average.

The experiments were conducted using a workstation with the following configurations: a single NVIDIA TitanX GPU, 3.5 GHz 6-Core CPU, 32GB RAM, and Linux 64-bit OS.
4.2. Experimental Results and Discussion

Quantitative Results on COCO-Text-Legible Dataset. The performance of the proposed method on the COCO-Text-Legible dataset is shown in Table 1. By just using ResNet-101 in place of VGG-16 as the base network, there is already 4.17% improvement in the F-Score. We denote this network as ResNet-SSD as our baseline. By adding the text attention gating module, the recall of the proposed approach is significantly improved (13.57% gain) over the baseline. Overall, the proposed text detector achieved 4.16% F-score gain over the baseline.

In ResNet-TextBoxes, default boxes with more text-specific aspect ratios and vertically denser tiling scheme as in TextBoxes [9] are used. With more default boxes per cell in the feature maps, both the recall and false detection rate increase, so the overall F-score is slightly higher than ResNet-SSD. Adding the proposed text attention gating mechanism boosts the performance by a remarkable margin, 3.67% in the recall, 8.09% in the precision and 5.98% in the F-score.

Quantitative Results on COCO-Text-Full Dataset. The performance of the proposed method on the COCO-Text-Full dataset is depicted in Table 2. Comparing with our baseline ResNet-SSD, which only gets 27.17% in the F-Score, adding the text attention gating mechanism makes it slightly better than the state-of-the-art method [12]. The F-score of ResNet-TextBoxes is just comparable with the state-of-the-art method [12], while with the help of the proposed text attention gating mechanism, our model substantially outperforms it. Specifically, the recall is improved by 14.7%, and the overall F-score gain is 4.22%. This confirms the effectiveness of the proposed method. We also list the baseline methods from [13], but they are not directly comparable since they were used in the data annotation pipeline when the dataset was built.

Table 1: Evaluations on COCO-Text-Legible validation set (in %)

<table>
<thead>
<tr>
<th>Models</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-SSD</td>
<td>30.38</td>
<td>42.01</td>
<td>35.26</td>
</tr>
<tr>
<td>ResNet-SSD</td>
<td>34.42</td>
<td>46.14</td>
<td>39.43</td>
</tr>
<tr>
<td>ResNet-SSD + Proposed</td>
<td>47.99</td>
<td>39.93</td>
<td>43.59</td>
</tr>
</tbody>
</table>

Table 2: Evaluations on COCO-Text-Full validation set (in %)

<table>
<thead>
<tr>
<th>Models</th>
<th>Recall</th>
<th>Precision</th>
<th>F-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yao et al. [12]</td>
<td>23.1</td>
<td>43.23</td>
<td>33.31</td>
</tr>
<tr>
<td>ResNet-SSD</td>
<td>35.4</td>
<td>31.03</td>
<td>27.17</td>
</tr>
<tr>
<td>ResNet-SSD + Proposed</td>
<td>40.7</td>
<td>28.59</td>
<td>33.57</td>
</tr>
<tr>
<td>ResNet-TextBoxes [9]</td>
<td>35.9</td>
<td>30.89</td>
<td>33.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Baselines from [13]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>23.3</td>
<td>83.78</td>
<td>36.48</td>
</tr>
<tr>
<td>B</td>
<td>10.7</td>
<td>89.73</td>
<td>19.14</td>
</tr>
<tr>
<td>C</td>
<td>4.7</td>
<td>18.56</td>
<td>7.47</td>
</tr>
</tbody>
</table>

Qualitative Results. Some sample detection results of the proposed method are shown in Fig. 3. TAMs clearly highlight the text regions, and thus yield higher detection accuracy with reduced false positives.

5. CONCLUSION

We have presented a robust and efficient scene text detection system. Our proposed system combines the strengths of the segmentation network and the detection network in a unified end-to-end trainable network. A text attention map (TAM) is produced to robustly determine the probability of the presence of text in each spatial locations in the images. The input convolutional features of the detection network are re-weighted adaptively according to the response values in the TAM. The MultiBox detectors are applied onto the semantically refined features to predict and localize the bounding boxes for each individual word. The experiments on the most challenging benchmark COCO-Text confirm that the proposed segmentation-aided text detector substantially outperforms previous methods. Without any post-processing and repeated computations, our system is more efficient and practical for intelligent robotics than previous methods.

Possible future directions worthy of exploring include: 1) combining the current work with text recognition module to build an end-to-end scene text reading system; 2) further optimizing the proposed method for the embedded systems; 3) extending the proposed idea to other detection problems, such as pedestrian detection.
6. REFERENCES


