MOST: A Multi-Oriented Scene Text Detector with Localization Refinement

Minghang He[1*]   Minghui Liao[1*]   Zhibo Yang[2]   Humen Zhong[3]
Jun Tang[2]   Wenqing Cheng[1]   Cong Yao[2]   Yongpan Wang[2]   Xiang Bai[1†]
Huazhong University of Science and Technology[1]
Alibaba Group[2]   Nanjing University[3]

{minghanghe,mhliao}@foxmail.com   {yaocong2010,yangzhibo450}@gmail.com   yongpan@taobao.com
xixing.tj@alibaba-inc.com   zhonghumen@smail.nju.edu.cn   {xbai,chenwq}@hust.edu.cn

Abstract

Over the past few years, the field of scene text detection has progressed rapidly that modern text detectors are able to hunt text in various challenging scenarios. However, they might still fall short when handling text instances of extreme aspect ratios and varying scales. To tackle such difficulties, we propose in this paper a new algorithm for scene text detection, which puts forward a set of strategies to significantly improve the quality of text localization. Specifically, a Text Feature Alignment Module (TFAM) is proposed to dynamically adjust the receptive fields of features based on initial raw detections; a Position-Aware Non-Maximum Suppression (PA-NMS) module is devised to selectively concentrate on reliable raw detections and exclude unreliable ones; besides, we propose an Instance-wise IoU loss for balanced training to deal with text instances of different scales. An extensive ablation study demonstrates the effectiveness and superiority of the proposed strategies. The resulting text detection system, which integrates the proposed strategies with a leading scene text detector EAST, achieves state-of-the-art or competitive performance on various standard benchmarks for text detection while keeping a fast running speed.

1. Introduction

Recently, scene text reading has become an active research topic in the computer vision community, due to the vital value in a wide range of applications, such as video indexing, signboard reading [25], and instant translation, which requires automatic textual information extraction from natural images.

Driven by deep neural networks and massive data, techniques and systems for scene text reading have evolved drastically over the past few years and numerous inspiring
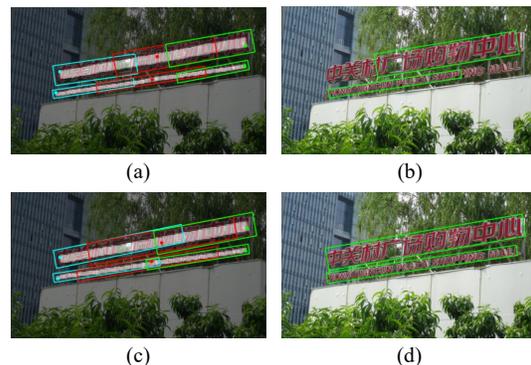
Figure 1. Illustration of the superiority of the proposed MOST in handling long text instances. (a) and (b) are from EAST [45]; (c) and (d) are from MOST. (a) and (c) show raw detection boxes predicted at 3 different locations inside the text region. The sampling point shares the same color with the corresponding detection box. (b) and (d) are the final detection results.

ideas have been proposed [20]. However, on account of real-world challenges such as diverse shapes, arbitrary orientations, various scales as well as complex illumination, severe blur, and perspective distortion, there is still large room for improvement of current text reading methods.

In particular, regarding scene text detection, the topic we are concerned with in this work, obvious weaknesses of existing algorithms can be observed. For instance, EAST [45], a representative one-stage scene text detector, has proven to be poor at detecting text instances with extreme aspect ratios (see Fig. 1 (a) and (b)). There are two main reasons: (1) The network has a limited receptive field and thus is incapable of making use of sufficient information needed to precisely predict the spatial extent of long text instances; (2) In the Non-Maximum Suppression (NMS) procedure of EAST, the detections are merged using their text/non-text classification scores as weight, which neglects their differences in quality caused by the network's limited receptive field and leads to biased geometric estimation.

To address these issues, we propose a Multi-Oriented Scene Text detector (MOST) with localization refinement.

The localization refinement part includes a Text Feature Alignment Module (TFAM) and a Position-Aware Non-Maximum Suppression (PA-NMS) module. The former aligns image features with the coarse detection results, which can dynamically adjust the receptive field for the localization prediction layer. The latter, on the other hand, adaptively merges the raw detections according to the positions at which they are predicted to focus on accurate predictions while abandoning inaccurate ones. Moreover, to improve the detection of small text instances, we design an Instance-wise IoU loss, which keeps the weight of each instance in the loss function the same.

The experiments demonstrate that the three proposed strategies can effectively enhance the detection performance. Specifically, they bring 4.0% and 9.5% performance gain on the MLT17 validation set for different IoU standards and 5.1% improvement on the MTWI test dataset. Moreover, our proposed text detector keeps a simple pipeline and runs quite fast.

The contributions of this paper are four-fold:

1. We propose TFAM, which dynamically adjusts the receptive field based on the coarse detections.

2. The proposed PA-NMS further refines the detections by merging reliable predictions based on the positions.

3. We introduce the Instance-wise IoU loss to strike a balanced training of text instances of different scales.

4. Our proposed MOST achieves state-of-the-art or competitive performance and a fast inference speed.

2. Related Work

According to the different pipelines, modern text detectors based on deep learning can be roughly divided into two categories: bottom-up methods and top-down methods.

Bottom-up methods consider scene text detection as a two-step process: (1) Detecting the fundamental elements; (2) Aggregating these elements to produce the detection results. SegLink [26] and its variant SegLink++ [29] take small segments of a text instance as the fundamental elements and link them together to form the bounding boxes. TextSnake [21] further improves the bottom-up methods by treating a set of circular plates along the text centerline as fundamental elements and solves the arbitrary-shape text detection problem. CRAFT [1] instead considers the character bounding boxes as basic elements and uses an affinity score map to aggregate detected characters. PSENet [31] and PAN [32] follow a segmentation pipeline, defining each pixel in the image as a basic element and aggregating them by Breadth-First Search. The aforementioned methods have achieved remarkable performance on several benchmarks. However, most of them suffer from a complicated post-processing algorithm for aggregating basic elements, which

could significantly degrade their efficiency and inference speed. Besides, a less powerful algorithm for aggregating basic elements may also affect the accuracy greatly since a text instance might be cut into several segments if the aggregating algorithm is not working as expected.

Top-down methods often follow a general object detection pipeline and directly output the word/line-level detection results. These methods can be further formulated as two sub-categories, i.e. one-stage text detectors and two-stage text detectors. One-stage text detectors like TextBoxes [15], EAST [45], TextBoxes++ [14], and RRD [17] directly regress the parameters of the text bounding boxes on the entire feature map and adopt NMS to produce the final results. Two-stage text detectors like Mask TextSpotter series [22, 12, 13], on the contrary, usually follow a MaskR-CNN [7]-style framework by using a region proposal network (RPN) to produce text proposals first and then regress the corresponding parameters based on those produced text proposals. These methods often have a relatively simple post-processing algorithm and can avoid the complex aggregating procedure.

Comparison with LOMO Aiming at improving the detection of long text instances, LOMO [42] proposes an Iterative Refinement Module (IRM) to perceive the entire long text by iterative refinement. It extracts the RoI feature multiple times based on preliminary proposals, forming a multi-stage detector. Different from the RoI transform [28] in LOMO, our proposed TFAM performs localization refinement by tailored deformable convolution operator [3], which achieves higher accuracy and is more efficient (see the comparisons in Sec. 4).

3. Methodology

The pipeline of our proposed MOST is shown in Fig. 2. It is composed of a ResNet-50 backbone with a feature pyramid structure [18], a text/non-text classification head, a position-sensitive map prediction head, a localization branch, and a Position-Aware Non-Maximum Suppression (PA-NMS) module. The localization branch contains a coarse localization head, a Text Feature Alignment Module (TFAM), and a refined localization head.

3.1. Network Design

The input of all the prediction heads (excluding the refined localization head) is a fused feature map from the feature-pyramid backbone, which is of shape $\frac{H}{4} \times \frac{W}{4} \times C$, where H and W are the height and width of the input image; C indicates the number of channels, set to 256. In the following description, Conv, BN, ReLU and Sigmoid indicate convolution, batch normalization [10] , rectified linear units [5] and the sigmoid function respectively.

Text/Non-Text Classification Head Firstly, the input feature map is fed into Conv (3×3)-BN-ReLU layers to reduce
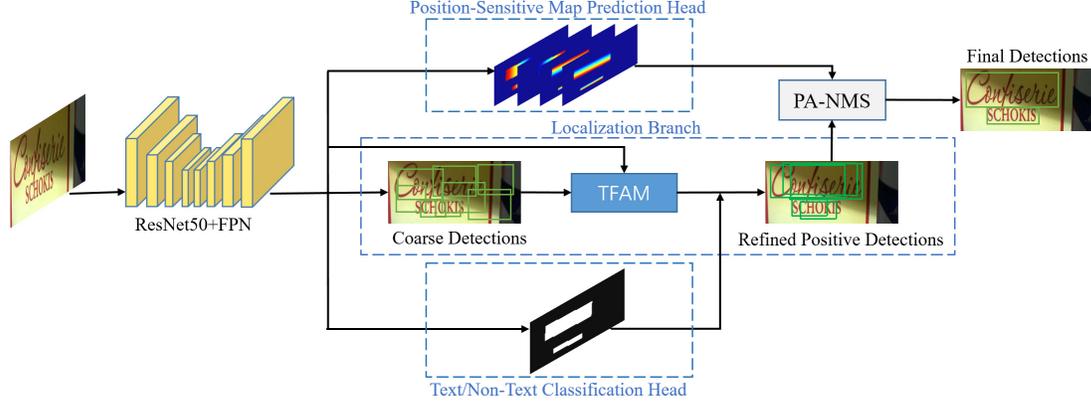
Figure 2. Overall architecture of our network, which is composed of a ResNet-50-FPN backbone, a text/non-text classification head, a localization branch and a position-sensitive map prediction head. The localization branch consists of a coarse localization head, a text feature alignment module (TFAM) and a refined localization head. PA-NMS indicates position-aware non-maximum suppression.

the number of channels to 64. Then, Conv (1×1)-Sigmoid layers are followed to generate the score map, which is of shape $\frac{H}{4} \times \frac{W}{4} \times 1$ and with values in the range of $(0, 1)$.

Position-Sensitive Map Prediction Head The position-sensitive map prediction head is similar to the text/non-text classification head in the network structure, with a different output shape. The position-sensitive map is of shape $\frac{H}{4} \times \frac{W}{4} \times 4$ and in the range of $(0, 1)$. The four channels represent the position-sensitive map in left, right, top, and bottom order, respectively, as shown in Fig. 5.

Localization Branch The localization branch consists of a coarse localization head, a Text Feature Alignment Module (TFAM), and a refined detection head. First, coarse detections are predicted by the coarse localization head. Then, the TFAM dynamically adjusts the receptive field of the text features based on the coarse detections to produce the aligned feature, which is fed into the refined localization head to predict the final detections.

The coarse localization head and the refined localization head share the same structure. They consist of Conv (3×3)-BN-Relu layers (reducing the number of channels to 64) and a Conv (1×1) layer. The output geometry map is of shape $\frac{H}{4} \times \frac{W}{4} \times 5$. The five channels represent the distances to the four sides of the text quadrangle and the rotation angle of the text quadrangle, respectively.

3.2. Text Feature Alignment Module

Due to the limited receptive field of CNN, it is hard for top-down, single-stage text detectors like EAST [45] to localize the text boundaries precisely, especially for text instances with large scales or extreme aspect ratios. LOMO [42] proposes to refine the localization stage by stage through aligning the image feature progressively with previous detections using ROI transform [28]. However, such a multi-stage network would bring heavy extra computation, especially when the number of the text instances is large.

To achieve better feature alignment while keeping a fast running speed, we propose the text feature alignment module (TFAM). An illustration of TFAM is shown in Fig. 3. First, the coarse detection results are used to generate sampling points. Then, the sampling points are applied to the deformable convolution operator [3] to obtain the aligned feature for the refined localization. The location p_0 of the aligned feature y can be calculated by:

$$y(p_0) = \sum_{p_n \in \hat{R}} w(p_n) * x(p_0 + p_n + \Delta p_n) \quad (1)$$

where x is the input feature map and w is the weight of the deformable convolution; $\hat{R}$ represents a regular sampling grid and p_n enumerates the locations in $\hat{R}$. An additional offset is added, termed as Δp_n, which is related to the sampling point selection strategies.

Feature-based Sampling Feature-based sampling is a sampling point selection strategy applied in the original deformable convolution layer, where Δp_n is predicted from the preceding feature maps via additional convolutional layers, given by:

$$\Delta p_n = Conv(x(p_0)) \quad (2)$$

An illustration of this sampling method is shown in Fig. 3 (a).

Localization-based Sampling Different from the feature-based sampling in the original deformable convolution layer, our localization-based sampling uses the coarse detections predicted by the coarse localization head to assign the sampling points. Δp_n is calculated by:

$$\Delta p_n = \Gamma(\hat{d_{c0}}, p_0) \quad (3)$$

where $\hat{d_{c0}}$ represents the coarse detection box at p_0 and the Γ function calculates the offsets required to make the sampling points distribute evenly in the coarse detection box, as shown in Fig. 3 (b).
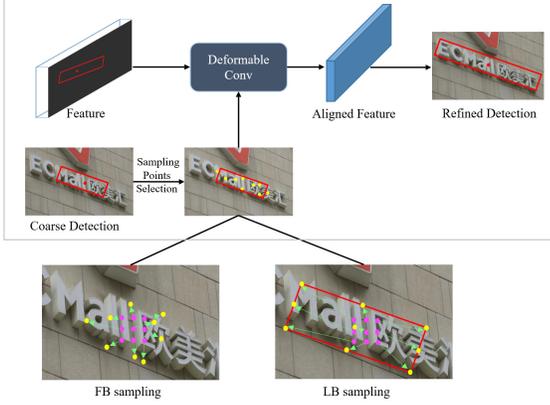
Figure 3. The description of TFAM. In (a) (b), the purple points represent regular sampling grid and the yellow points represent the deformed sampling locations. The additional offsets(Δp_n) are represented by light green arrows.
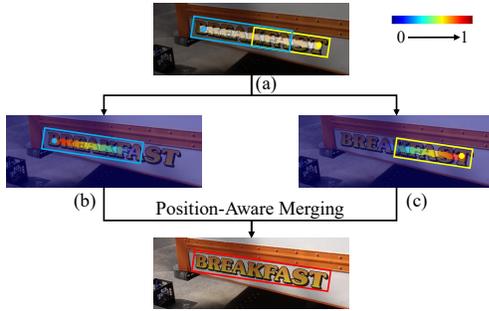

Figure 4. Visualization of position-aware merging. The score map in (a) is used to determine the positive boxes. (b) and (c) show the position-sensitive maps in left and right order, respectively.

By adopting the localization-based sampling method, TFAM can generate feature that aligns with the coarse detections, which can be used further by the refined localization head to generate refined detections that enclose the text region better than the coarse detections. The ablation study on the sampling method for TFAM is presented in Sec. 4.3.

TFAM can generate features that have an adaptive receptive field, the extent of which is determined by coarse detections' shape and scale. The whole process of feature alignment is completed by a tailored deformable convolution layer, which makes it fast and easy to implement.

3.3. Position-Aware Non-Maximum Suppression

NMS in EAST EAST [45] proposes locality-aware NMS to merge all the positive detection boxes predicted by the network to get final results. Compared with standard NMS, locality-aware NMS can generate more stable results while taking much less time, the working process of which can be divided into two steps: the weighted merging and the standard NMS. During the process of weighted merging, the detections are merged row by row and the detection box currently encountered is merged iteratively with the last merged ones. Given two detection boxes p and q and their

corresponding text/non-text classification scores $S(p)$ and $S(q)$, the weighted merging can be formulated as:

$$
\begin{aligned}
m_i &= (S(p)p_i + S(q)q_i)/S(m), \quad i = 1, \cdots, 4 \\
S(m) &= S(p) + S(q)
\end{aligned}
\tag{4}
$$

where m_i represents i-th coordinate of the merged box m and $S(m)$ represents the score of the merged box.

Proposed Position-Aware NMS For all the positive points inside a text region, the position of a point can affect its prediction of the detection box. As shown in Fig. 4 (a), the closer the point is to a text boundary, the more likely it can predict an accurate location of the text boundary. Thus, more convincing and more accurate values (distance to the boundaries) of the detected boxes can be merged if the positions of the boxes are taken into consideration.

We propose the position-aware NMS, where the merging process reserves accurate parts of the detected boxes while removing the inaccurate parts, according to the positions of the boxes, as shown in Fig. 4. The positions are given by the position-sensitive maps [2], which reflect the positions within text instances. Fig. 5 (c) - (f) show the position-sensitive maps in left, right, top, and bottom order respectively, which can be used as the weight for the prediction of left, right, top, and bottom boundary of the text instance in the box merging process. Given two boxes p and q (the indexes of 1, 2, 3, and 4 correspond to top-left, top-right, bottom-right, and bottom-left vertexes of the box respectively.) and position-sensitive maps in left, right, top, and bottom order, termed as L, R, T and B, the function of position-aware box merging can be formulated as:

$$
\begin{aligned}
m_i(x) &= (L(p)p_i(x) + L(q)q_i(x))/L(m), i = 1, 4 \\
m_j(x) &= (R(p)p_j(x) + R(q)q_j(x))/R(m), j = 2, 3 \\
m_k(y) &= (T(p)p_k(y) + T(q)q_k(y))/T(m), k = 1, 2 \\
m_l(y) &= (B(p)p_l(y) + B(q)q_l(y))/B(m), l = 3, 4 \\
\Psi(m) &= \Psi(p) + \Psi(q), \qquad \Psi \in \{L, R, T, B\}
\end{aligned}
\tag{5}
$$

where m is the merged box; $m_i(x)$ and $m_k(y)$ are the x and y coordinates of the i-th and the k-th vertex of the box m (likewise for p and q). $L(p)$ represents the value of the left-sensitive map at the corresponding location of the box p (likewise for R, T and B, likewise for q and m).

As suggested in Eq. (5), PA-NMS uses the value of the corresponding position-aware score, instead of the text/non-text classification score, as the weight for the box in the position-aware merging process, which can help the precise localization of text boundaries.

3.4. Label Generation

We follow the same process in EAST [45] to generate the score map and the geometry maps. The process of generating the position-sensitive maps is illustrated in this section.
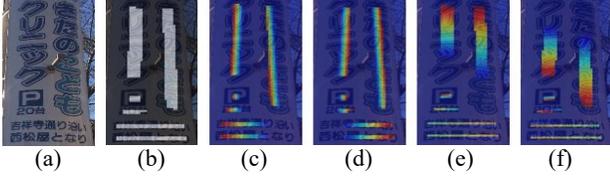
Figure 5. Visualization of the GT of position-sensitive maps. (a) input image; (b) score map; (c), (d), (e) and (f) are position-sensitive maps in left, right, top and bottom order respectively.

For each text instance, the value of the position-sensitive maps at a certain positive position i inside the text region can be formulated as:

$$F(i) = \begin{cases} 1 - \frac{Dist(i,f) - min(D_f)}{d_f - min(D_f)}, & \text{if } Dist(i,f) < d_f \\ 0, & \text{Otherwise.} \end{cases}$$
$$d_f = \alpha * (max(D_f) - min(D_f)) + min(D_f)$$
$$D_f = \{Dist(i,f) | i \in P\}$$
(6)

where f represents one of the four sides of a text region and F is its corresponding position-sensitive map, e.g. the right side corresponds to the right-sensitive map. P represents the set of all the positive samples in the text instance. $Dist(i,f)$ calculates the distance from the point i to the side f. d_f is the threshold of distances: if $Dist(i,f) > d_f$, the value of the point i in the corresponding position-sensitive map becomes 0. α is set to 0.75 in our experiment. The visualization of the position-sensitive maps is shown in Fig. 5.

3.5. Instance-wise IoU loss

The IoU loss [41] is adopted in EAST [45] to calculate the regression loss for the geometry prediction. The IoU loss is scale-invariant for each positive sample. However, as shown in Fig. 5 (b), the large text region contains far more positive samples than the small text region, which makes the regression loss bias towards large and long text instances. The IoU loss is formulated as:

$$L_{iou} = -\frac{1}{|\Omega|} \sum_{i \in \Omega} logIoU(\hat{d}_i, d_i^*)$$
(7)

where $\hat{d}_i$ and d_i^* represent the geometry prediction of the i-th sample and its corresponding ground truth, respectively. Ω represents the set of positive samples, and $|\Omega|$ is the number of samples in Ω.

For a more balanced training for text instances of different scales, we propose the Instance-wise IoU loss, as:

$$L_{ins-iou} = -\frac{1}{N_t} \sum_{j=1}^{N_t} \frac{1}{|S_j|} \sum_{k \in S_j} logIoU(\hat{d}_{jk}, d_{jk}^*)$$
(8)

where $\hat{d}_{jk}$ and d_{jk}^* represent the geometry prediction of the k-th sample in the j-th text instance and its corresponding

ground truth respectively. S_j represents the set of positive samples that belong to the j-th text instance and N_t is the total number of text instances.

As suggested in Eq. (8), the loss of each positive sample is normalized by the number of positive samples in the text instance that it belongs to. So each text instance, regardless of the number of positive samples it has, can contribute equally to the total instance-wise IoU loss.

3.6. Optimization

The loss for our network can be formulated as:

$$L = L_s + \lambda_{gc}L_{gc} + \lambda_{gr}L_{gr} + \lambda_p L_p$$
(9)

where L_s, L_{gc}, L_{gr} and L_p represent the losses for the score map, the geometry map predicted by the coarse localization head, the geometry map predicted by the refined localization head, and the position-sensitive maps respectively. λ_{gc}, λ_{gr} and λ_p balance the importance of the four losses, which are all set to 1 in our experiment.

Note that we only calculate L_{gc}, L_{gr} and L_p for the set of positive samples, termed as Ω.

Loss for Score Map We use the binary cross entropy loss as the object function for score map prediction, termed by L_s, OHEM [27] is adopted for L_s, in which the ratio between the negatives and positives is set to 3:1.

Loss for Geometry Maps The loss of the rotation angle is formulated as:

$$L_\theta = \frac{1}{|\Omega|} \sum_{i \in \Omega} 1 - \cos(\hat{\theta}_i - \theta_i^*)$$
(10)

where $\hat{\theta}_i$ and θ_i^* represent the prediction of the rotation angle and the corresponding ground truth for the i-th sample in Ω respectively.

L_{gc} and L_{gr} share the the same form of L_g, which is a combination of L_{iou}, $L_{ins-iou}$ and L_θ:

$$L_g = L_{iou} + \lambda_i L_{ins-iou} + \lambda_\theta L_\theta$$
(11)

The loss functions of L_{iou} and $L_{ins-iou}$ are given in Eq. (7) and Eq. (8) respectively. λ_i and λ_θ balance the three losses, which are set to 1 and 20 respectively in our experiment.

Loss for Position-sensitive Maps Smoothed-L1 loss [4] is adopted in the calculation of L_p:

$$L_p = \frac{1}{4|\Omega|} \sum_{i \in \Omega} \sum_{\Psi \in \{L,R,T,B\}} SmoothedL1(\hat{\Psi}_i - \Psi_i^*)$$
(12)

where $\hat{\Psi}_i$ and Ψ_i^* represent the prediction of the position-sensitive map Ψ for the i-th sample in Ω and the corresponding ground truth respectively.

4. Experiments

First, we briefly introduce all the datasets used in our experiments. Then, the implementation details of our method are given. Third, we show the ablation studies on the proposed contributions. Finally, we make comparisons with the state-of-the-art methods on four benchmark datasets.

Note that the "baseline" mentioned in our experiment refers to the model adapted from EAST [45]. Our network is simply established by adding the proposed TAFM, PA-NMS and instance-wise IoU loss to it.

4.1. Datasets

SynthText is a synthetic dataset containing 800k images, generated by a synthetic image generation engine [6]. This dataset is only used for pre-training.

ICDAR 2017 MLT (MLT17) is a dataset proposed in the competition of ICDAR 2017 on Multi-lingual scene text detection [24]. It contains 7200 images for training, 1800 images for validation and 9000 images for testing. Text instances of this dataset are from 9 different languages.

MTWI is a dataset proposed in the contest of ICPR 2018 on robust reading for multi-type web images [9], including 10000 training image and 10000 testing images. Text in this dataset are mainly in Chinese and English. All text instances are annotated at the line level.

ICDAR 2015 (IC15) is presented for the ICDAR 2015 Robust Reading Competition [11]. The dataset is annotated with word-level quadrangles, including 1000 training images and 500 test images.

MSRA-TD500 [40] is a multi-language dataset, containing both English and Chinese text. The dataset images are taken by pocket cameras from indoor and outdoor scenes. It is divided into 300 training images and 200 test images. Following the previous works [45][21][23], 400 training images from HUST-TR400 [39] are added for training.

4.2. Implementation Details

We adopt a ResNet-50 [8] backbone with a feature pyramid structure [18] in our model. During the pre-training stage, we train the model on SynthText[6] for 2 epochs using the Adam optimizer, whose learning rate is set to 10^{-4}. SGD optimizer is used to fine-tune the pre-trained model with the training set of each dataset. Following [44], we use the "poly" learning rate policy. The initial learning rate is set to 0.005 and the power is set to 0.9. The number of fine-tuning epochs for MLT17, MTWI, IC15, MSRA-TD500 is set as 300, 300, 1200 and 1200, respectively. For all datasets, the training batch size is 16. For data augmentation, text regions are randomly cropped and resized to 640×640. Besides, several common data augmentation techniques such as flipping, rotation, and color change are adopted for training. The model is implemented in PyTorch and trained with 2 Tesla V100 GPUs.

4.3. Ablation Study

The ablation study is conducted to demonstrate the effectiveness of each module. Since there are enough training data in MLT17 [24] and MTWI [9], we train the models directly on them without pre-training on SynthText [6].

Sampling Methods of TFAM As mentioned in section 3.2, there are two sampling methods proposed for TFAM: feature-based sampling (FB) and the localization-based sampling (LB). Besides, we also try to combine them together, named as "Combine"(CB). Half of CB's sampling points are assigned acoording to the rule of FB, while the other half the rule of LB. Tab. 1 clearly shows that both sampling methods bring performance improvements over the baseline in terms of f-measures and LB gets larger performance gain. The combination of the two sampling methods (CB) achieves the best, outperforming the baseline by 2.4% and 7.4% using IoU@0.5 and IoU@0.7 respectively. Therefore, we adopt the "Combined" sampling method in all the forthcoming experiments.

Table 1. Ablation study on sampling methods. The results are evaluated on the validation set of MLT17 [24]. "FB", "LB", and "CB" are short for feature-based, localization-based, and combined.

Sampling	IoU@0.5			IoU@0.7		
	P	R	F	P	R	F
FB	82.1	70.4	75.8	70.5	60.4	65.1
LB	84.5	**71.2**	77.3	74.3	**62.7**	68.0
CB	**85.7**	70.6	**77.4**	**76.1**	**62.7**	**68.8**
baseline	82.0	69.0	75.0	67.2	56.5	61.4

The Proposed Strategies The results of the ablation studies on the proposed strategies are given in Table 2.

(1) *TFAM* significantly improves the f-measure on MTWI by 4.2% over the baseline. In addition, the f-measure improvement on the validation set of MLT17 is 7.4% under the protocol of IoU@0.7. These performance enhancements demonstrate that TFAM can effectively handle text instances with large variation of aspect ratios, benefiting from the improved receptive fields.

(2) *Instance-wise IoU Loss* balances the training of text instances of different scales. On the validation set of MLT17, it leads to performance gains of 1.0% and 2.9% in f-measure under the protocol of IoU@0.5 and IoU@0.7 respectively over the baseline. When TFAM is also used, Instance-wise IoU loss can still boosts the performance by 1.1% and 1.0% respectively. Similar improvements can be observed on MTWI.

(3) *PA-NMS* incorporates position information into the merging of overlapping boxes, which improves the localization of text boundaries. It boosts the baseline by 2.7% in f-measure on MTWI. By adding PA-NMS to the model equipped with both TFAM and Instance-wise IoU loss, an additional gain of 0.4% can be achieved. Similar performance gains are achieved on the validation set of MLT17.
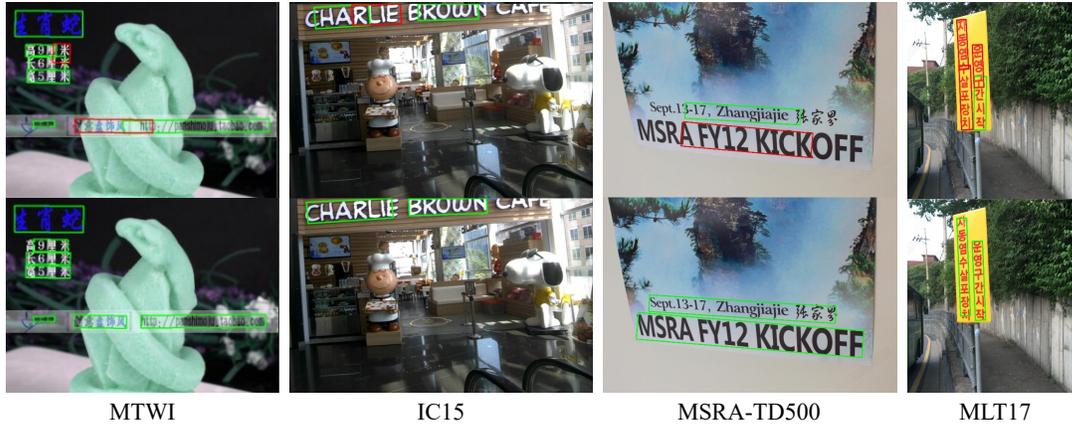
| MTWI | IC15 | MSRA-TD500 | MLT17 |

Figure 6. Visualization results from 4 benchmarks. The top row and the bottom row show results predicted by the baseline and MOST, repectively. The green boxes mean accurate detections while the red boxes indicate inaccurate detections.

Table 2. Ablation study on the three proposed strategies. TFAM, Ins-IoU, and PA-NMS indicate text feature alignment module, instance-wise IoU loss and position-aware NMS respectively.

| TFAM | Ins-IoU | PA-NMS | MLT17 val | | | | | | MTWI test | | |
| | | | IoU@0.5 | | | IoU@0.7 | | | | | |
			P	R	F	P	R	F	P	R	F
×	×	×	82.0	69.0	75.0	67.2	56.5	61.4	72.7	65.9	69.1
✓	×	×	**85.7**	70.6	77.4	76.1	62.7	68.8	76.6	70.3	73.3
×	✓	×	81.9	70.9	76.0	69.3	60.0	64.3	73.0	67.5	70.2
×	×	✓	83.5	70.0	76.1	72.5	60.7	66.1	76.1	68.0	71.8
✓	✓	×	85.0	72.9	78.5	75.6	64.8	69.8	76.8	71.1	73.8
✓	✓	✓	85.2	**73.7**	**79.0**	76.4	**66.1**	**70.9**	**77.3**	**71.3**	**74.2**

With all the proposed strategies, our model can improve significantly over the baseline by 4.0% and 9.5% in f-measure under the protocol of IoU@0.5 and IoU@0.7 respectively on the validation set of MLT17. In addition, a performance gain of 5.1% can be achieved by our model over the baseline on the test set of MTWI. Some visualization results are shown in Fig. 6.

4.4. Multi-Oriented Text Detection

We evaluate our method on IC15 [11] and compare it with other state-of-the-art methods. During inference, we set the short side of images to 1152 while keeping their aspect ratios. As shown in Tab. 3, our proposed method is on par with GNNets [36] in f-measure (88.2% vs 88.5%) at a much faster inference speed (10.0FPS vs 2.1FPS). As shown in Tab. 4, our proposed method can further boost the performance under a more strict threshold i.e. IOU@0.7, in which it outperforms previous methods by an even larger margin (75.9% vs 73.3%).

4.5. Multi-Lingual and Long Text Detection

To evaluate the performance of our method on detection of multi-lingual and long text, we compare our model with other state-of-the-art methods on MSRA-TD500 [40], MLT17 [24], and MTWI [9].

Table 3. Quantitative results on IC15 [11] under the protocol of IoU@0.5. † indicates using deformable-ResNet-50 as backbone and * indicates using external MLT17 data for pre-training.

Method	P	R	F	FPS
SegLink [26]	73.1	76.8	75.0	-
TextField [37]	84.3	80.1	82.4	6.0
TextSnake [21]	84.9	80.4	82.6	1.1
PSENet-1s* [31]	86.9	84.5	85.7	1.6
PAN [32]	84.0	81.9	82.9	**26.1**
ATRR [33]	90.4	83.3	86.8	10.0
CRAFT* [1]	89.8	84.3	86.9	8.6
DRRG [43]	88.5	84.7	86.6	3.5
ContourNet [34]	87.6	86.1	86.9	3.5
LOMO [42]	91.3	83.5	87.2	3.4
LSE [30]	88.3	85.0	86.6	3.0
DB† [16]	**91.8**	83.2	87.3	12.0
GNNets [36]	90.4	86.7	**88.5**	2.1
baseline	86.8	84.8	85.8	15.4
ours	89.1	**87.3**	88.2	10.0

MSRA-TD500 [40] We conduct experiments on MSRA-TD500 to demonstrate the robustness of our method for detecting long text instances. The long side of the image is set to 640 during inference. As shown in Tab. 5, our method achieves a f-measure of 86.4%, which outperforms previous state-of-the-art methods by 1.5% (86.4% vs 84.9%) in f-

Table 4. Quantitative results on IC15 [11] and MSRA-TD500 [40] under the protocol of IoU@0.7. PSE represents PSENet-1s. * indicates using released models from open-source repository.

Method	IC15			MSRA-TD500		
	P	R	F	P	R	F
PSE* [33]	74.5	72.2	73.3	-	-	-
PAN* [32]	62.3	57.3	59.7	75.4	73.2	74.3
DB* [16]	76.2	70.5	73.3	74.0	65.6	69.6
baseline	71.5	69.5	70.5	70.4	62.8	66.4
ours	**76.7**	**75.2**	**75.9**	**81.2**	**74.8**	**77.9**

Table 5. Quantitative results on MSRA-TD500 [40] under the protocol of IoU@0.5. * indicates using external MLT17 data for pre-training and † indicates using deformable-ResNet-50 as backbone.

Method	P	R	F	FPS
SegLink [26]	86	70	77	8.9
TextField [37]	87.4	75.9	81.3	-
TextSnake [21]	83.2	73.9	78.3	1.1
ATRR [33]	85.2	82.1	83.6	10.0
CRAFT* [1]	88.2	78.2	82.9	8.6
MSR [38]	87.4	76.7	81.7	-
LSE [30]	84.2	81.7	82.9	3.0
PAN [32]	85.7	**83.2**	84.5	30.2
DB† [16]	**91.5**	79.2	84.9	32.0
DRRG* [43]	88.1	82.3	85.1	-
baseline	87.0	77.6	82.0	**66.8**
ours	90.4	82.7	**86.4**	51.8

measure. Compared to methods like DRRG [43] which uses external MLT17 data for pre-training, our methods can still achieve a higher f-measure (86.4% vs 85.1%) while running at an impressive speed (51.8FPS). Besides, as shown in Tab. 4, it outperforms other methods by at least 3.6% (77.9% vs 74.3%) under a more strict IoU constraint.

Table 6. Quantitative results on the test set of MLT17 [24], * indicates the reported FPS is from CRAFT [1] and † indicates using deformable-ResNet-50 as backbone.

Method	P	R	F	FPS
Lyu et al. [23]	**83.8**	55.6	66.8	5.7*
DRRG [43]	75.0	61.0	67.3	-
LOMO [42]	78.8	60.6	68.5	-
SPCNet [35]	73.4	66.9	70.0	-
PSENet-1s [31]	73.8	68.2	70.9	-
CRAFT [1]	80.6	68.2	73.9	8.6*
GNNets [36]	79.6	70.1	74.5	-
DB† [16]	83.1	67.9	74.7	**19.0**
BDN [19]	83.6	70.1	76.3	2.3
baseline	75.9	67.0	71.1	15.2
ours	82.0	**72.0**	**76.7**	10.1

MLT17 [24] The results are listed in Tab. 6. During testing, we set both sides of the image in the range of (640, 1920) while keeping its aspect ratio. As shown in the table, MOST achieves a f-measure of 76.7%, which sur-

passes all other counterparts and has a large advantage especially in the terms of recall rate by at least 1.9% (72.0% vs 70.1%). Compared with one-stage methods, our method outperforms them with a large margin (76.7% vs 74.7%) while running at a competitive speed (10.1FPS). Besides, even compared with complicated two-stage methods like BDN [19], our method can still achieve a higher f-measure with a simpler pipeline.

MTWI [9] To demonstrate the generalization ability of the proposed method, we test our model on the MTWI dataset, containing multi-lingual text instances obtained from web images. The results are listed in Tab. 7. Note that we re-produce results on some methods [1] who did not provide official MTWI results in their papers, and these methods are marked with †. During testing, we set both sides of the image in the range of (640, 1280) while keeping its aspect ratio. Our method outperforms all of the counterparts by at least 1.2% (74.7% vs 73.5%), while achieving competitive inference speed (23.5FPS). The results on MTWI demonstrate the generalization ability of our method that the proposed modules can enhance the performance of detecting text in both natural scene images and web images.

Table 7. Quantitative results on MTWI [9]. Note that the original PAN uses a ResNet-18 backbone. We re-implement it with a ResNet-50 backbone for a fair comparison. * indicates the result obtained from SegLink++ [29] and † indicates reproducing from open-source repository.

Method	P	R	F	FPS
TextBoxes++* [14]	66.8	56.3	61.1	-
SegLink* [26]	70.0	65.4	67.6	-
SegLink++* [29]	74.7	69.7	72.1	-
PAN† [32]	**78.9**	68.9	73.5	16.9
BDN† [19]	77.3	70.0	73.4	2.7
baseline	74.0	66.0	69.8	**30.6**
ours	78.8	**71.1**	**74.7**	23.5

5. Conclusion

In this paper, we propose a set of strategies to address the major weaknesses of the existing algorithms for scene text detection: imprecise geometry prediction of extremely long text instances and defectiveness in handling significant scale variation. The comprehensive experiments have demonstrated that the proposed approach resolves these issues in a principled way and outperforms previous state-of-the-art methods on standard datasets in this field. Moreover, it is worth noting that the proposed strategies are actually quite general and thus could be readily extended to many other one-stage text detection methods. We would like to leave this for future research.

---

[1]The results are reproduced from the official open-source repository of PAN and BDN respectively.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019. 2, 7, 8

[2] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016. 4

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 2, 3

[4] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 5

[5] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011. 2

[6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016. 6

[7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[9] Mengchao He, Yuliang Liu, Zhibo Yang, Sheng Zhang, Canjie Luo, Feiyu Gao, Qi Zheng, Yongpan Wang, Xin Zhang, and Lianwen Jin. ICPR2018 contest on robust reading for multi-type web images. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 7–12. IEEE, 2018. 6, 7, 8

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 2

[11] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 6, 7, 8

[12] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(2):532–548, 2021. 2

[13] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Proc. ECCV*, volume 12356, pages 706–722, 2020. 2

[14] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018. 2, 8

[15] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Pro. AAAI*, pages 4161–4167, 2017. 2

[16] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020. 7, 8

[17] Minghui Liao, Zhen Zhu, Baoguang Shi, Guisong Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proc. CVPR*, pages 5909–5918, 2018. 2

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2, 6

[19] Yuliang Liu, Sheng Zhang, Lianwen Jin, Lele Xie, Yaqiang Wu, and Zhepeng Wang. Omnidirectional scene text detection with sequential-free box discretization. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3052–3058, 2019. 8

[20] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, pages 1–24, 2020. 1

[21] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 20–36, 2018. 2, 6, 7, 8

[22] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83, 2018. 2

[23] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7553–7563, 2018. 6, 8

[24] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. ICDAR2017 robust reading challenge on multi-lingual scene text detection and script identification-RRC-MLT. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017. 6, 7, 8

[25] Xuejian Rong, Chucai Yi, and Yingli Tian. Recognizing text-based traffic guide panels with cascaded localization network. In *European Conference on Computer Vision*, pages 109–121, 2016. 1

[26] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2550–2558, 2017. 2, 7, 8

[27] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 761–769, 2016. 5

[28] Yipeng Sun, Chengquan Zhang, Zuming Huang, Jiaming Liu, Junyu Han, and Errui Ding. Textnet: Irregular text reading from images with an end-to-end trainable network. In *Asian Conference on Computer Vision*, pages 83–99, 2018. 2, 3

[29] Jun Tang, Zhibo Yang, Yongpan Wang, Qi Zheng, Yongchao Xu, and Xiang Bai. Seglink++: Detecting dense and arbitrary-shaped scene text by instance-aware component grouping. *Pattern Recognition*, 96:106954, 2019. 2, 8

[30] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4234–4243, 2019. 7, 8

[31] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019. 2, 7, 8

[32] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8440–8449, 2019. 2, 7, 8

[33] Xiaobing Wang, Yingying Jiang, Zhenbo Luo, Cheng-Lin Liu, Hyunsoo Choi, and Sungjin Kim. Arbitrary shape scene text detection with adaptive text region representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6449–6458, 2019. 7, 8

[34] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11753–11762, 2020. 7

[35] Enze Xie, Yuhang Zang, Shuai Shao, Gang Yu, Cong Yao, and Guangyao Li. Scene text detection with supervised pyramid context network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9038–9045, Jul. 2019. 8

[36] Youjiang Xu, Jiaqi Duan, Zhanghui Kuang, Xiaoyu Yue, Hongbin Sun, Yue Guan, and Wayne Zhang. Geometry normalization networks for accurate scene text detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9137–9146, 2019. 7, 8

[37] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 28(11):5566–5579, 2019. 7, 8

[38] Chuhui Xue, Shijian Lu, and Wei Zhang. MSR: Multi-scale shape regression for scene text detection. *arXiv preprint arXiv:1901.02596*, 2019. 8

[39] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749, 2014. 6

[40] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1083–1090. IEEE, 2012. 6, 7, 8

[41] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016. 5

[42] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look more than once: An accurate detector for text of arbitrary shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10552–10561, 2019. 2, 3, 7, 8

[43] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9699–9708, 2020. 7, 8

[44] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 6

[45] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017. 1, 2, 3, 4, 5, 6