Location-Aware Feature Selection for Scene Text Detection

Zengyuan Guo[1] Zilin Wang[1*] Zihui Wang[1†] Wanli Ouyang[2] Haojie Li[1] Wen Gao[3]
[1]Dalian University of Technology [2]The University of Sydney [3]Peking University

Abstract

Direct regression-based natural scene text detection methods have already achieved promising performances. However, for the bounding box prediction, they usually utilize a fixed feature selection way to select features used to predict different components, such as the distance to the boundary or the rotation angle, which may limit selection flexibility of each component prediction and thus degrades the performance of the algorithm. To address this issue, we propose a novel method called Location-Aware Feature Selection (LAFS). It separately learns the confidence of different locations features for each component and then selects the features with the highest confidence to form a combination of the most suitable features. In other words, LAFS uses a learnable feature selection way to flexibly pinpoint feature combinations used to predict more accurate bounding boxes. After adding LAFS, our network has a large performance improvement without efficiency loss. It achieved state-of-the-art performance with single-model and single-scale testing, outperforming all existing regression-based detectors.

(a) Without LAFS (b) With LAFS

Figure 1. (a) The bounding boxes predicted by a single feature is inaccurate. (b) Unbinding the features used to predict different components via LAFS significantly improves the accuracy of the bounding box. The color line refers to the distance or angle predicted by the location of the corresponding color point.

*Equal contribution
†Corresponding author: zhwang@dlut.edu.cn

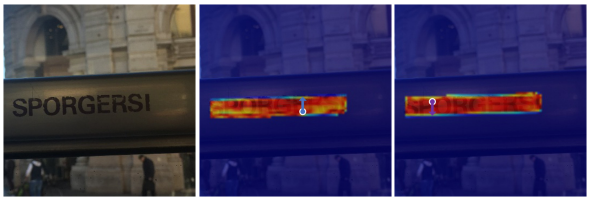
1. Introduction

Scene text detection based on deep learning has drawn much attention because it serves as a prerequisite for various downstream applications like self-driving car, instance translation, scene understanding and product search. Nowadays, segmentation-based text detection methods [3, 18, 22, 10, 1] have achieved outstanding performance. However, they require a complex post-process to determine the scope of each text instance precisely. In contrast, direct regression-based approaches [32, 31, 16, 13, 27] perform boundary regression by predicting the offsets from a given point, with the advantages of simple network structure, high efficiency and no need of complex post-process. Unfortunately, its hard for them to accurately locate the bounding boxes, especially for large and long text.

In this work, we argue that one of reason for the inaccurate bounding boxes regression of the direct regression-based approaches is the neglect of the Feature Adaptability. Here Feature Adaptability means the features from different locations are suitable to predict different components of bounding boxes, such as distance to the boundary or rotation angle. In the existing text detection networks, the features used to predict different components always are fixed to a single location or the locations close to predicted targets. For example, [32, 11, 31] use a single feature to predict all components. The prediction of different components of a bounding box was bounded to the same location. They neglected the fact that the features required by different components may be different and using single feature to predict different components will inherently prone to reduce the accuracy of the bounding boxes for text instances. As shown in Figure 1, the distribution of the most suitable features for each component is inconsistent and it is not advisable to predict all components with one feature obviously.

Different from them, Lyu et al [19] uses the feature close to targets. It predicts the four corners information of bounding box separately and then combines them to form the final bounding box. The position of each corner is predicted by the feature close to itself. Nevertheless, we find that as shown in (a) in Figure 2, the most suitable features predicted to the top boundary appear in the lower half of the text area,

which shows that the best features are not necessarily distributed near the predicted target. In addition, in Figure 2, we can also observe that the best features of (d) (predict the left boundary distance) and (e) (predict the right boundary distance) are obviously in different positions, which also shows that the positions of the best features of different components are inconsistent, and it is likely to be unreasonable to use a single feature for box prediction. In order to further explore the relationship between the distribution of the best features and the predicted target location, we made statistic on the distribution of the best features, and the results are shown in Figure 3. From Figure 3, we can see that the optimal features of different components are not always distributed in a solid single or multiple area. Especially for the best features of the top and bottom boundaries and angle, the distribution of their best features does not have an inevitable causal relationship with the location of the predicted target, and it is difficult to summarize a stable distribution law. Therefore, it is also not robust and reasonable to use only the features close to the target for prediction.

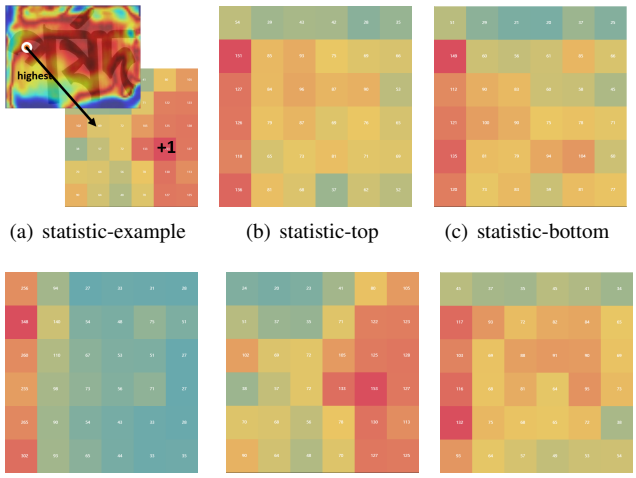


(a) text instance (b) top (c) bottom

(d) left (e) right (f) angle

Figure 2. (a) is a text instance and (b)-(f) are the confidence (the accuracy of prediction) visualization maps about the distance to the four boundaries and rotation angle respectively in different locations. Red pixels mean higher confidence and blue is lower. We can observe that: (1) The best features of different components are distributed in different locations; (2) The most appropriate feature is not always located at the location close to the predicted target (especially for top and bottom).

In order to better grasp the feature adaptability and determine the position of the most appropriate feature on the feature map for predicting each component, we propose an effective text detection pipeline with learnable feature selection. Our framework, named Location-Aware feature Selection text detection Network (LASNet), is designed with a fully convolution network, which can learn the location distribution of the most suitable feature. The architecture is shown as in Figure 4. We first use the network to separately predict the confidence of each component of the bounding box. Then, we combine the components with the highest confidence from the same text instance into a more accurate bounding box via the proposed Location-Aware Feature Selection (LAFS) mechanism. Specifically, with the predicted confidence, LAFS finds the top-K features of each component and fuse them to the best bounding box by weighting their confidence scores. Confidences indicate whether corresponding features are suitable to predict the components, which is exactly the embodiment of feature adaptability.



(a) statistic-example (b) statistic-top (c) statistic-bottom

(d) statistic-left (e) statistic-right (f) statistic-angle

Figure 3. Statistics results. After resizing all text instances to 32×32, we counted the distributions of the features with the highest confidence and visualized them.

After adding LAFS, LASNet has a significant performance improvement without efficiency loss. Our approach has reached the state-of-the-art performance, i.e. Hmean of 87.4%, 69.6% and 83.7% on ICDAR2015, ICDAR2017-MLT and MSRA-TD500 respectively, outperforming all other regression based detectors. Whats more, under more strict evaluation criteria, our method has more enormous accuracy improvement as shown in Table 1.

The contributions of this paper are two-fold: (1) We propose an LAFS mechanism, which can address accuracy insufficient problem of the direct regression-based methods by taking advantage of feature adaptability. (2) The proposed LASNet achieves the performance of state-of-the-art on several benchmarks and has more significant accuracy improvement for more rigorous criteria.

2. Related Work

Driven by the deep neural network and large-scale datasets, scene text detection method has made great progress in the past few years. The mainstream text detection methods based on deep learning can be roughly divided

into segmentation-based text detection and regression-based detection. Regression-based methods can be divided into indirect regression-based methods and direct Regression-based methods according to whether the anchor is used or not.

2.1. Segmentation-Based Text Detection

Most of the segmentation-based text detection methods improve FCN [17] and Mask R-CNN [5] on the task of text detection. PixelLink [3] uses instance segmentation to determine whether the pixels in different directions are in a text region. Then, Textsnake [18] uses a number of disks with different sizes and directions to cover the annotation text, which can be used to detect arbitrary shape text. To distinguish adjacent text instances, PSENet [10] uses the method of progressive scale expansion, and LSAE [22] regards text detection as instance segmentation and map image pixels into embedded feature space. Different from these methods, CRAFT [1] first detects a single character (character region score) and a connection relationship (affinity score) between characters, followed by determining the final text line according to the connectivity between characters. Segmentation-based methods are faced with an essential problem, which is how to determine the scope of text instances. As a result, they often require a complex post-processing process while having degraded efficiency.

2.2. Indirect Regression-Based Text Detection

Indirect Regression-Based methods mostly draw lessons from general object detection networks, such as Faster R-CNN [20] and SSD [15]. They generate text boxes via predicting the bounding box offsets from anchors. For example, SLPR [33] adds coordinate regression of intersection point of horizontal and vertical uniform sliding line and text polygon on the traditional object detection framework Faster R-CNN. It realized text detection of arbitrary shape. As two improvement methods of SSD in text detection, TextBoxes [12] modified the shape of anchor and convolution kernel of SSD to make it more suitable for text line detection. Seglink [21] no longer detects the entire text line at one time. It first detected text segments, and then connected segments together to get the final bounding box. Besides, RRD [13] abandoned the way of sharing feature map of classification and regression task which were used in previous detection framework, but adopted the way of independent feature extraction. Despite their great success, indirect regression-based methods have to design anchor, which is not robust to changes in datasets and text scales.

2.3. Direct Regression-Based Text Detection

Direct regression-based approaches perform boundary regression by predicting the offsets from a given point. Among them, EAST [32] uses FCN [17] to predict text score map and combines bounding box predictions in text area to determine bounding boxes. TextMountain [34] predicts text center-border probability (TCBP) and text center-direction (TCD) to easily separate text instances. In TCBP, mountaintop means the center of text instance and mountain foot means the border. An anchor-free RPN was proposed in AF-RPN [31], which aims to replace the original anchor-based RPN in the Faster R-CNN framework to address the problems caused by the anchor. Direct regression-based methods are characterized by simple network structure and high detection efficiency. Nevertheless, they also have some disadvantages. Its hard for direct regression-based methods to get the exact boundaries of bounding boxes, especially for large and long text.

Because the direct regression-based methods have the advantages of simple architecture and high detection efficiency, we think they have greater potential. To address their performance defect, we focus on how to effectively find the most suitable features for different components separately. Via considering the feature adaptability, we propose an effective approach, Location-Aware Feature Selection, to improve the detection accuracy. Our network learns the confidence of different locations' features for components prediction, selecting the most suitable features to form the superb feature combinations rather than bind them into the same location compulsively or just use the features closed to the predicted targets like most of other methods. The LASNet not only keep the efficiency advantage of direct regression-based methods, but also achieved excellent accuracy, which is competitive with segmentation-based methods.

3. Our Method

3.1. Network Architecture

3.1.1 Network Architecture

Figure 4 illustrates the architecture of the LASNet. First, we extract feature from intermediate layers of ResNet-50 [6]. Next comes an FPN [14], by which we can get feature maps of different scales. Then it is a Feature Fusion Module (FFM). It can selectively fuse different stages' features. For the output branches, apart from 1-channel classification score map and 5-channel geometry map, we especially predict a 5-channel confidence map used in LAFS at inference stage. Classification score map is a shrunk text foreground mask for the region of text instance, confirming that which features predictions are relatively accurate. Geometry map, containing 4 distances d_t, d_l, d_b, d_r from the point to the box boundaries and a rotation angle θ, is used for enclosing the word from the view of each pixel. In this work, the confidence map plays the key role in the proposed LAFS. Its channel number is the same as geometry map and it indicates the accuracy of the corresponding regression values

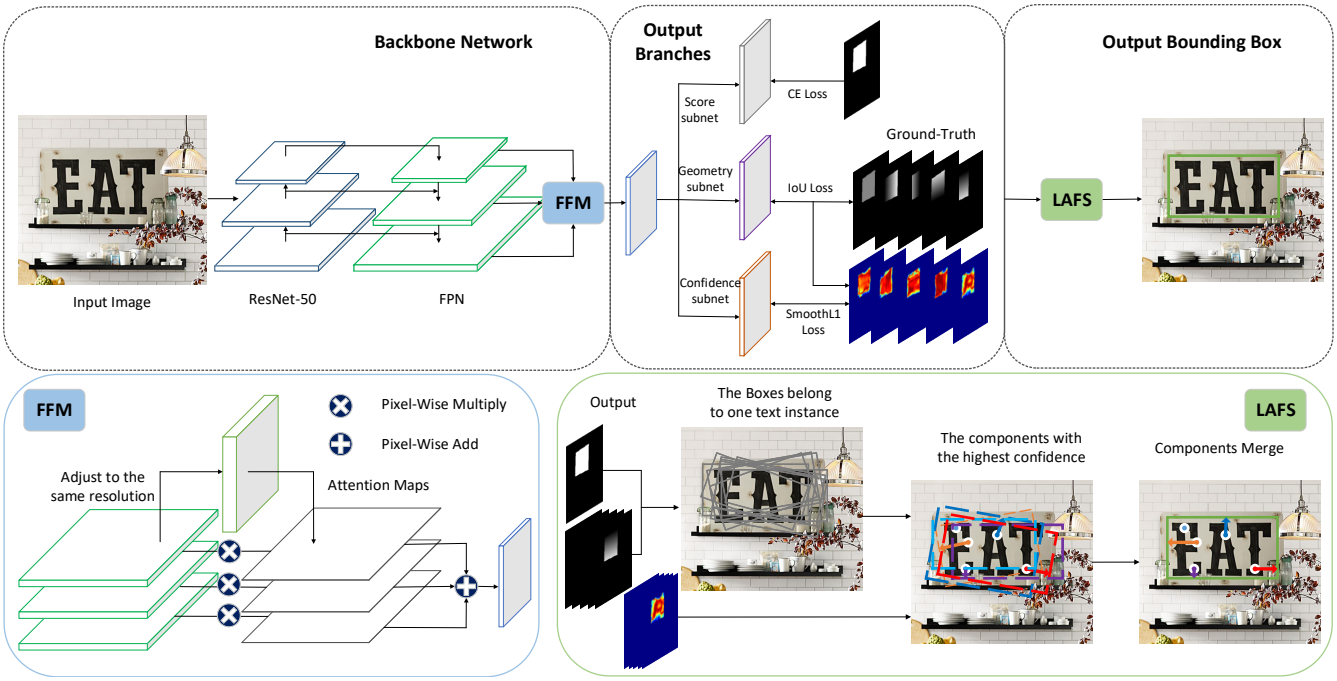


Figure 4. The architecture of LASNet. LASNet uses a Feature Pyramid Network [17] backbone on top of a feedforward ResNet architecture. The FPN outputs from different stages will be interpolated to the same resolution. In Feature Fusion Module (FFM), they are fused to one feature map with more suitable feature map by attention mechanism. Then there are three output branches, score, geometry and confidence. At the time of inference, proposed Location-Aware Feature Selection (LAFS) mechanism selects the feature with high confidence for each component and uses the most suitable feature combination to predict the final bounding box.

in geometry map.

3.1.2 Feature Fusion Module

As we all know, the abundance of feature map information is very helpful to improve the prediction accuracy of the network. We design a module called Feature Fusing Module (FFM). FFM uses the attention mechanism. Specifically, it predicts an attention map for each stage (three stages in this work) and selectively weight features from different stages to control their impact on the final prediction.

3.2. Location-Aware Feature Selection

To select the most suitable feature, we predict an extra confidence branch, whose five channels respectively correspond to the five channels of geometry branch. The meaning of confidence is the accuracy of the corresponding geometry branch's output. In the inference stage, we will disassemble five kinds of components of the bounding boxes which predict the same text instance and then reassemble the best 5 components into the best bounding box. The components selection is based on predicted confidence and the higher one will be reserved. The specific algorithm flow is showed in Algorithm 1. First of all, we determine which locations in the feature map predict the same text instance. Then, we rank all the components of the bounding boxes according to their confidence. Each kind of component selects the top-K confidence prediction results, $5 \times K$ prediction results in all. Then the confidence is used as weight to fuse the K prediction results of each component category. Finally, we combine the five best results to form the final bounding box.

The specific details of function $merge()$ is in Equation 1:

$$merge() = \frac{\sum conf_k \times pred_k}{\sum conf_k}, \tag{1}$$

where k is in range $[1, K]$.

Confidence map is defined as the confidence coefficient of each prediction of a text instance. In Equation 2. T means the text region. For a feature point t_i, its confidence $conf_i$ is defined as follows:

$$conf_i = \begin{cases} 1 - \dfrac{gap_i - min(Gap)}{\max(Gap) - min(Gap)} & if \quad t_i \in T \\ 0 & otherwise, \end{cases}$$
$$gap_i = abs(pred_i - label_i), \tag{2}$$

where gap_i is the gap between prediction and ground truth. As in the formula, we normalize the $conf_i$, so its range is

Algorithm 1 The specific algorithm flow of Location-Aware Feature Selection

Input: $B = [b_1, ..., b_N]$, is $N \times 5$ matrix of initial detection boxes. $b_n = [d_t, d_b, d_l, d_r, \theta] \in B$. $S = [b_s]$, which is the collection of bounding boxes predicting the same text instance with b_s (a random selected bounding box). K means that we choose top-K components. $U = []$ is the union of each kind of component with top-K confidence. $U.clear()$ means empty U to load next kind of component.

Output: Box is the final output box of the text instance which S stands for.

for each $n \in [1, N]$ **do**
  if $IoU(b_s, b_n) > threshold$ **then**
    $S \leftarrow S \cup b_n$
  end if
end for
for each $comp \in [d_t, d_b, d_l, d_r, \theta]$ **do**
  for $k = 1; k <= K; k++$ **do**
    $idx \leftarrow max(S.comp.conf)$
    $U \leftarrow U \cup S[idx].comp$
    $S[idx].comp.conf \leftarrow 0$
  end for
  $Box.comp \leftarrow merge(U)$
  $U.clear()$
end for
return Box

[0, 1].

In the process of inference, the key is how to determine which locations predict the same text instance. Common method is to instance segment the feature map [3, 22]. However, the instance segmentation often requires additional annotation information and network branches, which greatly increases the amount of computation. In this paper, we directly use the Intersection of Union (IoU) to deal with it. If the IoU of two locations predictions exceeds the threshold, we'll assume that they predict the same text instance. During the inference, the score map of the predicted text area is only used to determine whether the text instance exists, and does not participate in the LAFS process. This is because the distributions of classification score and confidence are not consistent.

3.3. Loss Functions

The loss function can be expressed as:

$$L_{sum} = L_{cls} + \gamma L_{geo} + \lambda L_{conf}, \tag{3}$$

where L_{cls}, L_{geo} and L_{conf} represent classification, regression and confidence loss respectively. γ and λ represent the hyper-parameter between different losses. According to the results of multiple experiments, the loss balance parameter can be set into γ = 1 and λ = 10.

3.3.1 Loss for Score Map

In the classification branch, we use Dice Loss, which is usually used in segmentation tasks. Dice Loss can reduce the bad influence brought by the number unbalance between the negative and positive samples:

$$L_{cls} = 1 - (2 \times \frac{\sum \hat{S} \times S}{\sum \hat{S} + \sum S + \epsilon}), \tag{4}$$

where $\hat{S}$ is the predicted value of the network, and S represents the ground truth.

3.3.2 Loss for Geometries

Geometries Loss is bipartition. μ is a hyper-parameter. In our experiments μ = 10:

$$L_{geo} = L_{IoU} + \mu L_{\theta}, \tag{5}$$

where L_{IoU} represents the loss of the distance to the four boundaries predicted by the network. We adopt IoU Loss:

$$L_{IoU} = -logIoU(\hat{R} \times R) = -log\frac{\hat{R} \cap R}{\hat{R} \cup R}, \tag{6}$$

where $\hat{R}$ and R represent the area of predicted bounding box and ground truth respectively, $\hat{R} \cap R$ means the intersection of areas.

$$\hat{R} \cup R = \hat{R} + R - \hat{R} \cap R. \tag{7}$$

The loss of rotation angle is:

$$L_{\theta} = 1 - cos(\hat{\theta} - \theta). \tag{8}$$

We use cosine function to measure the angle gap between a prediction result and a ground truth.

3.3.3 Loss for Confidence

We select Smooth L1 loss as the confidence loss:

$$L_{conf} = SmoothL1(\hat{C} - C), \tag{9}$$

where $\hat{C}$ is prediction and C is the ground tru

4. Experiment

4.1. Datasets

The datasets we used for the experiments are briefly introduced below:

ICDAR2015(IC15) comprises 1,000 training images and 500 testing images which is of different orientations. It is collected for the ICDAR 2015 Robust Reading Competition. The images are taken by Google Glasses without taking care of positioning, image quality, and viewpoint. IC15 only includes English text instances. The annotations are at the word level using quadrilateral boxes.

ICDAR2017-MLT(IC17) is a large-scale multi-lingual text dataset, which includes 7200 training images, 1800 validation images and 9000 test images with texts in 9 languages. The text regions in IC17 are also annotated by 4 vertices of the quadrangle.

MSRA-TD500 includes 500 images in total, 300 for training and 200 for testing, with text instances of different orientations. It is consisting of both English and Chinese text and annotated on text line level.

4.2. Training strategy

4.2.1 Experiment Settings

Our model is trained on double TITAN RTX GPUs by Adam optimizer with the batch size which is 32. Similar to EAST [32], we use the exponential decay learning rate strategy that the learning rate is multiplied by 0.94 for every 10,000 iterations, and the initial learning rate is set to 1.0×10^{-3} for all experiments. As the training sets of ICDAR2015 and MSRA-TD500 are too small, we directly pre-train our model on the union of MSRA-TD500, ICDAR2015, and ICDAR2017-MLT training datasets for 200,000 iterations, and then fine-tune the model with the training set of ICDAR2015, MSRA-TD500 and ICDAR2017 MLT respectively. ResNet-50 is taken as the backbone network, and the scale of training images is resized to 640.

4.2.2 Data Augmentation

In our experiments, we follow the data augmentation of EAST [32]. First, the sizes of images are randomly rescaled with ratio 0.5, 1.0, 2.0, 3.0. Second, we crop images from the transformed images and the ratio of foreground to background is 3:1. Finally, the cropped images are resized to 640×640 for training.

4.3. Ablation Study

We conduct ablation experiments to verify the performance of LAFS module on performance. In this part, we use the validation set of ICDAR2017. During the experiments, the longer side of the input picture is set to 1024.

4.3.1 Assessment criteria of text detection

In natural scene text detection, Recall, Precision and Hmean are usually used to measure the performance of the detectors. Hmean is calculated by the Recall and Precision, and the detailed process can be referred to the Wang et al [24]. In the calculation of recall rate and accuracy, an important concept "match", namely the matched degree between ground truth and prediction, is involved. In the evaluation method of ICDAR dataset, if the Intersection of Union (IoU) between ground truth bounding box and prediction exceeds 0.5, they are supposed to be matched. Obviously, the higher the matching threshold, the more favorable for downstream work. As shown in the Figure 6 (a) and (c), when IoU is set to 0.5, part of the text area will be missed during recognition. Therefore, we also evaluated the high threshold in the following experiments.

4.3.2 Verification of LAFS

With LAFS, bounding box becomes more accurate by purposeful selection of features. In order to show the effectiveness of the proposed LAFS, we conducted confirmatory experiments on the thresholds of 0.5, 0.6, 0.7 and 0.8 in the ablation experiments. It can be concluded from Table 1 that when the criteria for prediction box become further strict, our method show greater performance advantages. When we set the IoU threshold as 0.8, with LAFS, we can get 3.1% improvement on Recall, 5.5% improvement on Precision and 4.0% improvement on Hmean. This result is a further evidence of the remarkable improvement in performance for bounding box prediction based on our method. Figure 6 provides the visualization examples of the improvement.

4.3.3 Ceiling analysis

To measure the ceiling performance of the LAFS, we replace the predicted confidence with the ground truth values and evaluate performance on the ICDAR 2017-MLT validation dataset. As shown in Table 1, we can find that using ground truth confidence improves the Hmean from 71.3% to 77.5% for IoU threshold = 0.5 and from 44.6% to 56.3% for IoU threshold = 0.8. This demonstrates that there is still improvement room for the performance of LAFS.

4.3.4 Feature Selection Area

In the INTRODUCTION part, we discussed the distribution of confidence, as shown in Figure 3. Our conclusion is that for different components of text bounding box, the most suitable features are not only distributed in the close area, but also in the area far away from the target. In order to further confirm our observation results, we have carried out

Figure 5. Qualitative results on ICDAR2015, ICDAR2017-MLT and MSRA-TD500 by the proposed LASNet.

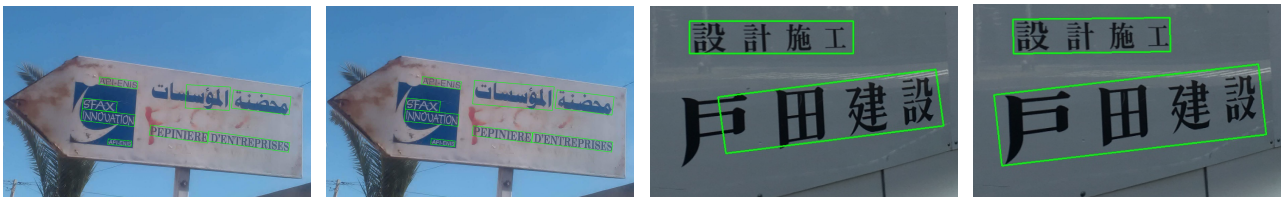
Table 1. We conducted the experiments on the validation set of IC17. Firstly, we add LAFS to the network. We can find that under more strict evaluation criteria, our method has more significant accuracy improvement. Secondly, we replace the predicted confidence with the ground-truth values, the results suggest there is still room for improvement in regressing confidence. The two values in "Gain" represent the improvement of "With LAFS" and "Confidence Ground-Truth" on "Without LAFS" respectively

IC17 MLT	Without LAFS			With LAFS			Confidence Ground-Truth			Gain
IoU Threshold	Recall	Precision	Hmean	Recall	Precisioin	Hmean	Recall	Precisioin	Hmean	Hmean
0.5	61.5%	84.8%	71.3%	63.1%	84.0%	**72.1%**	69.7%	87.3%	77.5%	0.9%/6.2%
0.6	59.3%	76.9%	66.9%	59.8%	79.6%	**68.3%**	66.8%	83.6%	74.3%	1.4%/7.4%
0.7	52.7%	68.4%	59.5%	54.5%	72.5%	**62.2%**	62.2%	77.8%	69.1%	2.7%/9.6%
0.8	39.4%	51.1%	44.6%	42.5%	56.6%	**48.6%**	50.7%	63.4%	56.3%	4.0%/11.7%

experiments. In the process of LAFS, we added constraint to the prediction of the top and bottom boundaries: only the features of the close areas can be selected. The experimental results are shown in Table 2, and the accuracy of prediction is descended. However, when the constraint was added to the left and right boundaries prediction, the accuracy of prediction increased slightly. We believe the reason for this result is that text examples are often rectangles with large aspect ratio, and their left and right boundaries are far away, so it is not appropriate to use the feature far away from the boundary to predict. We can also see this distribution law in (d) and (e) of Figure 3. To sum up, for different instances or even different components of the same instance, the distribution of confidence is not constant. On the other hand, it also shows that it is a possible research direction for the follow-up work to continue to optimize the feature selection of LAFS according to the characteristics of prediction objects.

4.3.5 Multi-group of Components

In the previous part of the ablation experiment, we only choose the best set of components within the same instance to generate our prediction box. However, only choosing the best set of components can lead to poor robustness. The way to deal with it is that we can consider the top-K group of components, and then through the confidence of the components, weighted and integrated them, getting our final prediction box. We set the group number K as a hyper-parameter, and Table 3 shows the ablation experiment we conducted for the hyper-parameter K. If more groups are considered, the performance may improve. But if we choose too many groups, the performance will decrease instead. We think this is because of the introduction of inaccurate information. It also proves that LAFS is highly accurate in choosing the most suitable location. In the inference stage, in order to balance the speed and accuracy, K is set to 1.



(a) (b) (c) (d)

Figure 6. (a) and (c) are the predicted bounding boxes if we only choose a single locations' prediction. (b) and (d) are the predicted bounding boxes after considering the feature adaptability, choosing the best location's feature in the feature map for each component. (b) and (d) are more accurate than (a) and (c).

Table 2. The feature selection area constraint experiments' results on ICDAR2017-MLT validation set.

constraint	up / down boundaries			Left / right boundaries			Compared with LASF
IoU Thresh	Recall	Precision	Hmean	Recall	Precision	Hmean	Hmean
0.5	63.1%	83.9%	72.0%	62.9%	84.1%	72.0%	-0.1%/ -0.1%
0.6	59.7%	79.4%	68.1%	59.7%	79.8%	68.3%	-0.2%/0%
0.7	54.3%	72.1%	61.9%	54.8%	73.2%	62.7%	-0.3%/0.4%
0.8	42.4%	56.4%	48.4%	57.7%	43.2%	49.4%	-0.2%/0.8%

Table 3. The selection of hyper-parameter K. We conducted the experiments on the validation set of ICDAR2017-MLT. The IoU threshold is 0.8.

K	**Recall**	**Precision**	**Hmean**
1	42.5%	56.6%	48.6%
2	42.8%	56.8%	48.8%
4	43.0%	56.9%	49.0%
6	43.0%	56.6%	48.9%

Table 4. Quantitative results of different methods evaluated on ICDAR2015. † means segmentation-based method.

Method	**Recall**	**Precision**	**Hmean**
TextSnake† [18]	80.4	84.9	86.8
PixelLink† [3]	82.0	85.5	83.7
FTSN† [2]	80.0	88.6	84.1
TextMountain† [34]	84.2	88.5	86.3
LSAE† [22]	85.0	88.3	86.6
IncepText† [27]	80.6	90.5	85.3
CRAFT† [1]	84.3	89.8	86.9
SPCNet† [25]	85.8	88.7	87.2
PSENet† [10]	85.2	89.3	87.2
LOMO† [29]	83.5	91.3	87.2
EAST [32]	73.5	83.6	78.2
R2CNN[9]	79.7	85.6	82.5
CRPN [4]	80.7	88.8	84.5
SLPR [33]	83.6	85.5	84.5
LASNet(ours)	84.0	91.2	87.4

Table 5. Quantitative results of different methods evaluated on ICDAR2017-MLT. † means segmentation-based method.

Method	**Recall**	**Precision**	**Hmean**
Lyu et al.† [19]	56.6	83.8	66.8
LOMO† [29]	60.6	78.8	68.5
SPCNet† [25]	66.9	73.4	70.0
PSENet† [10]	68.4	77.0	72.5
CRAFT† [1]	68.2	80.6	73.9
Huang et al.† [8]	69.8	80.0	74.3
He et al. [7]	57.9	76.7	66.0
Border [26]	62.1	77.7	69.0
LASNet(ours)	61.0	81.1	69.6

4.4. Experiments on Scene Text Benchmarks

4.4.1 ICDAR2015

We evaluated the performance of our method on predicting multi-orientation natural scene text. Because ICDAR2015 has too few training images, we use both ICDAR2015 and ICDAR2017-MLT training sets to fine tune 50k steps. In the test process, we use single-scale test (length of text long side is 2048), which can achieve state-of-the-art performance (R: 84.0%, P: 91.2%, H: 87.4%) as shown in Table 4. In ICDAR2015, text annotation is words level. The results show that our method can well solve the detection task of multi-orientation word level annotation text in natural scenes.

4.4.2 The comparison between segmentation-based methods and regression-based methods

Segmentation-based methods sometimes have better performance than regression-based methods. However, in order to separate text instances with small distances, they always have complex post process. As a result, the efficiency of them is lower than regression-based methods. We compare the efficiency between segmentation-based methods and regression-based methods. As is shown in Figure 7, regression based method has incomparable advantages on efficiency.

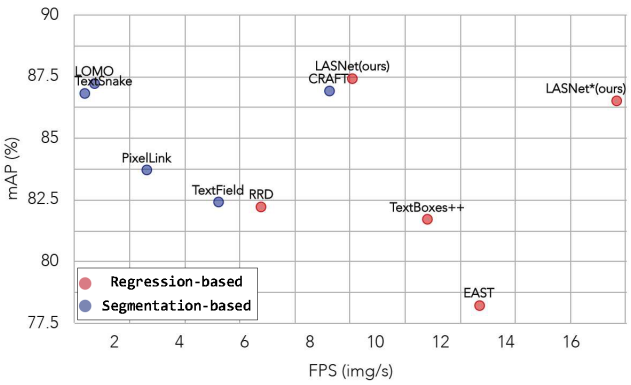


Figure 7. The efficiency comparison between segmentation-based methods (blue) and regression-based methods (red) on ICDAR2015. LASNet* means the length of text long side is 1280. Due to influences of factors such as GPU and CPU, the FPS value is only for reference. But it can be seen that regression-based methods have a great advantage over segmentation-based methods in efficiency.

4.4.3 ICDAR2017-MLT

We evaluated the performance of our method to more complex scenarios on ICDAR2017-MLT. We use the training set to fine tune 50k steps, and use a single-scale test (length of long side is 2048). Compared with ICDAR2015, the text instances of ICDAR2017-MLT have more diversity, larger scale changes and multiple languages. Our method can greatly improve the performance of regression-based detectors on this dataset as shown in Table 5. We achieve the performance of 69.6% of Hmean, which is state-of-the-art in the regression-based methods. But compared with the current segmentation-based text detection method, there is still a gap. The regression-based methods predict the text bounding boxes by regressing the absolute distances, while the segmentation-based methods predict the bounding boxes by classifying the pixels. Therefore, the regression-based methods have some disadvantages for the dramatic scale changes of the text instances. However, what it is worth to note that, as shown in Figure 7, the regression-based methods has remarkable advantage on efficiency in comparison with segmentation-based methods. This indicates the promising potential of our proposed regression-based method for taking both accuracy and efficiency into account.

4.4.4 MSRA-TD500

We evaluated the adaptability of our method to long text on MSRA-TD500. Unlike ICDAR2015 and ICDAR2017-MLT, MSRA-TD500 annotations are text lines rather than words. We use MSRA-TD500 training set to fine tune 50K steps and single-scale test (length of long side is 1024) is used in the test process. As shown in Table 6, we achieve state-of-the-art performance (R: 80.7%, P: 87.0%, H: 83.7%). The result also indicates that our method has a good performance in dealing with long text for MSRA-TD500 has a large quantity of long text instances.

Table 6. Quantitative results of different methods evaluated on MSRA-TD500. † means segmentation-based method.

Method	Recall	Precision	Hmean
Lyu et al.† [19]	76.2	87.6	81.5
FTSN† [2]	77.1	87.6	82.0
LSAE† [22]	81.7	84.2	82.9
Zhang et al. [30]	67.0	83.0	74.0
Yao et al. [28]	75.3	76.5	75.9
EAST [32]	67.4	87.3	76.1
SegLink [21]	70.0	86.0	77.0
RRD [13]	73.0	87.0	79.0
ITN [23]	72.3	90.3	80.3
LASNet(ours)	80.7	87.0	83.7

4.5. Conclusion and Future Work

We propose a regression-based scene text detection method called location-Aware Feature Selective (LAFS). Our network separately learns the confidence of different locations features and selects the suitable features for the components of the bounding box. We evaluate our method on ICDAR2015, ICDAR2017-MLT and MSRA-TD500, proving the huge improvement brought by our method. As for future work, in our opinion, LAFS is a beginning to consider feature adaptability in detection tasks. There is still more works can be done in this regard. From the ceiling analysis experiment, we can see that its potential is huge. We can continue to study in the design and the prediction of feature confidence. At the same time, we also think that the idea of feature adaptability is not only effective in detection tasks, but also can be explored in other directions of computer vision, such as classification and segmentation tasks.

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. 1, 3, 8

[2] Yuchen Dai, Zheng Huang, Yuting Gao, Youxuan Xu, Kai Chen, Jie Guo, and Weidong Qiu. Fused text segmentation networks for multi-oriented scene text detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3604–3609. IEEE, 2018. 8, 9

[3] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *Thirty-second AAAI conference on artificial intelligence*, 2018. 1, 3, 5, 8

[4] Linjie Deng, Yanxiang Gong, Yi Lin, Jingwen Shuai, Xiaoguang Tu, Yuefei Zhang, Zheng Ma, and Mei Xie. Detecting multi-oriented text with corner-based region proposals. *Neurocomputing*, 334:134–142, 2019. 8

[5] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 3

[6] 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. 3

[7] Wenhao He, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Multi-oriented and multi-lingual scene text detection with direct regression. *IEEE Transactions on Image Processing*, 27(11):5406–5419, 2018. 8

[8] Zhida Huang, Zhuoyao Zhong, Lei Sun, and Qiang Huo. Mask r-cnn with pyramid attention network for scene text detection. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 764–772. IEEE, 2019. 8

[9] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017. 8

[10] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, and Jian Yang. Shape robust text detection with progressive scale expansion network. *arXiv preprint arXiv:1806.02559*, 2018. 1, 3, 8

[11] Yuan Li, Yuanjie Yu, Zefeng Li, Yangkun Lin, Meifang Xu, Jiwei Li, and Xi Zhou. Pixel-anchor: A fast oriented scene text detector with combined networks. *arXiv preprint arXiv:1811.07432*, 2018. 1

[12] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 3

[13] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5909–5918, 2018. 1, 3, 9

[14] 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. 3

[15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[16] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018. 1

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 3, 4

[18] 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. 1, 3, 8

[19] 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. 1, 8, 9

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[21] 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. 3, 9

[22] 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. 1, 3, 5, 8, 9

[23] Fangfang Wang, Liming Zhao, Xi Li, Xinchao Wang, and Dacheng Tao. Geometry-aware scene text detection with instance transformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1381–1389, 2018. 9

[24] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011. 6

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

[26] Chuhui Xue, Shijian Lu, and Fangneng Zhan. Accurate scene text detection through border semantics awareness and bootstrapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 355–372, 2018. 8

[27] Qiangpeng Yang, Mengli Cheng, Wenmeng Zhou, Yan Chen, Minghui Qiu, Wei Lin, and Wei Chu. Inceptext:

A new inception-text module with deformable psroi pooling for multi-oriented scene text detection. *arXiv preprint arXiv:1805.01167*, 2018. 1, 8

[28] Cong Yao, Xiang Bai, Nong Sang, Xinyu Zhou, Shuchang Zhou, and Zhimin Cao. Scene text detection via holistic, multi-channel prediction. *arXiv preprint arXiv:1606.09002*, 2016. 9

[29] 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. 8

[30] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4159–4167, 2016. 9

[31] Zhuoyao Zhong, Lei Sun, and Qiang Huo. An anchor-free region proposal network for faster r-cnn-based text detection approaches. *International Journal on Document Analysis and Recognition (IJDAR)*, 22(3):315–327, 2019. 1, 3

[32] 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, 3, 6, 8, 9

[33] Yixing Zhu and Jun Du. Sliding line point regression for shape robust scene text detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3735–3740. IEEE, 2018. 3, 8

[34] Yixing Zhu and Jun Du. Textmountain: Accurate scene text detection via instance segmentation. *arXiv preprint arXiv:1811.12786*, 2018. 3, 8