Shape Robust Text Detection with Progressive Scale Expansion Network

Xiang Li[1]*, **Wenhai Wang**[2 1]*, **Wenbo Hou**[2], **Ruo-Ze Liu**[2], **Tong Lu**[2], **Jian Yang**[1]
[1]DeepInsight@PCALab, Nanjing University of Science and Technology
[2]National Key Lab for Novel Software Technology, Nanjing University

Abstract

The challenges of shape robust text detection lie in two aspects: 1) most existing quadrangular bounding box based detectors are difficult to locate texts with arbitrary shapes, which are hard to be enclosed perfectly in a rectangle; 2) most pixel-wise segmentation-based detectors may not separate the text instances that are very close to each other. To address these problems, we propose a novel Progressive Scale Expansion Network (PSENet), designed as a segmentation-based detector with multiple predictions for each text instance. These predictions correspond to different "kernels" produced by shrinking the original text instance into various scales. Consequently, the final detection can be conducted through our progressive scale expansion algorithm which gradually expands the kernels with minimal scales to the text instances with maximal and complete shapes. Due to the fact that there are large geometrical margins among these minimal kernels, our method is effective to distinguish the adjacent text instances and is robust to arbitrary shapes. The state-of-the-art results on ICDAR 2015 and ICDAR 2017 MLT benchmarks further confirm the great effectiveness of PSENet. Notably, PSENet outperforms the previous best record by absolute 6.37% on the curve text dataset SCUT-CTW1500. Code will be available in https://github.com/whai362/PSENet.

1 Introduction

Recently, natural scene text detection has attracted extensive attention for its numerous applications, such as scene understanding, product identification, automatic driving and target geolocation. However, due to the large variations in foreground texts and background objects, and the diverse text variabilities in shapes, colors, fonts, orientations and scales, along with the extreme illumination and occlusion, text detection in natural scene is still faced with considerable challenges.

Nevertheless, great progress has been made in recent years with the amazing development of Convolutional Neural Networks (CNNs) [6, 10, 22]. Based on bounding box regression, a list of methodologies [8, 9, 12, 17, 19, 23, 26, 29, 30] has been proposed to successfully locate the text targets in forms of rectangles or quadrangles with certain orientations. Unfortunately, these frameworks cannot detect the text instances with arbitrary shapes (e.g., the curve texts), which also often appear in natural scenes (see Fig. 1 (b)). Naturally, semantic segmentation-based methods can be taken into consideration to explicitly handle the curve text detection problems. Although pixel-wise segmentation can extract the regions of arbitrary-shaped text instances, it may still fail to separate two text instances when they are relatively close, because their shared adjacent boundaries will probably merge them together as one single text instance (see Fig. 1 (c)).

To address these problems, in this paper, we propose a novel instance segmentation network, namely, Progressive Scale Expansion Network (PSENet). There are two advantages of the proposed PSENet.

*Equal contribution. Please contact xiang.li.implus@njust.edu.cn and wangwenhai362@163.com.

Preprint. Work in progress.

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

Figure 1: The results of different methods, best viewed in color. (a) is the original image. (b) refers to the result of bounding box regression-based method, which displays disappointing detections as the red box covers nearly more than half of the context in the green box. (c) is the result of semantic segmentation, which mistakes the 3 text instances for 1 instance since their boundary pixels are partially connected. (d) is the result of our proposed PSENet, which successfully distinguishs and detects the 4 unique text instances.

Firstly, as a segmentation-based method, PSENet is able to locate texts with arbitrary shapes. Secondly, we put forward a progressive scale expansion algorithm, with which the closely adjacent text instances can be identified successfully (see Fig. 1 (d)). Specifically, we assign each text instance with multiple predicted segmentation areas. For convenience, we denote these segmentation areas as "kernels" in this paper and for one text instance, there are several corresponding kernels. Each of the kernels shares the similar shape with the original entire text instance, and they all locate at the same central point but differ in scales. To obtain the final detections, we adopt the progressive scale expansion algorithm. It is based on Breadth-First-Search (BFS) and is composed of 3 steps: 1) starting from the kernels with minimal scales (instances can be distinguished in this step); 2) expanding their areas by involving more pixels in larger kernels gradually; 3) finishing until the largest kernels are explored.

The motivations of the progressive scale expansion are mainly of four folds. Firstly, the kernels with minimal scales are quite easy to be separated as their boundaries are far away from each other. Therefore, it overcomes the major drawbacks of the previous segmentation-based methods; Secondly, the largest kernels or the complete areas of text instances are indispensable for achieving the final precise detections; Thirdly, the kernels are gradually growing from small to large scales, and thus the smoonth surpervisions would make the networks much easier to learn; Finally, the progressive scale expansion algorithm ensures the accurate locations of text instances as their boundaries are expanded in a careful and gradual manner.

To show the effectiveness of our proposed PSENet, we conduct extensive experiments on three competitive benchmark datasets including ICDAR 2015 [13], ICDAR 2017 MLT [27] and SCUT-CTW1500 [18]. Among these datasets, SCUT-CTW1500 is explicitly designed for curve text detection, and on this dataset we surpass the previous state-of-the-art result by absolute 6.37%. Furthermore, the proposed PSENet achieves better or at least comparable performance on the ordinary quadrangular text datasets: ICDAR 2015 and ICDAR 2017 MLT, when compared with the existing state-of-the-art methods.

The main contributions of this paper are as follows:

- We propose a novel Progressive Scale Expansion Network (PSENet) which can precisely detect text instances with arbitrary shapes.
- We propose a progressive scale expansion algorithm which is able to accurately separate the text instances standing closely to each other.
- Our proposed PSENet significantly surpasses the state-of-the-art methods on the curve text detection dataset SCUT-CTW1500. Furthermore, it also achieves competitive results on the regular quadrangular text benchmarks: ICDAR 2015 and ICDAR 2017 MLT.

2 Related Work

Text detection has been an active research topics in computer vision for a long period of time. [15, 29] successfully adopted the pipelines of object detection into text detection and obtained good performance on horizontal text detection. After that, [8, 9, 12, 17, 23, 30] took the orientation of text line into consideration and made it possible to detect arbitrary-oriented text instances. Recently,

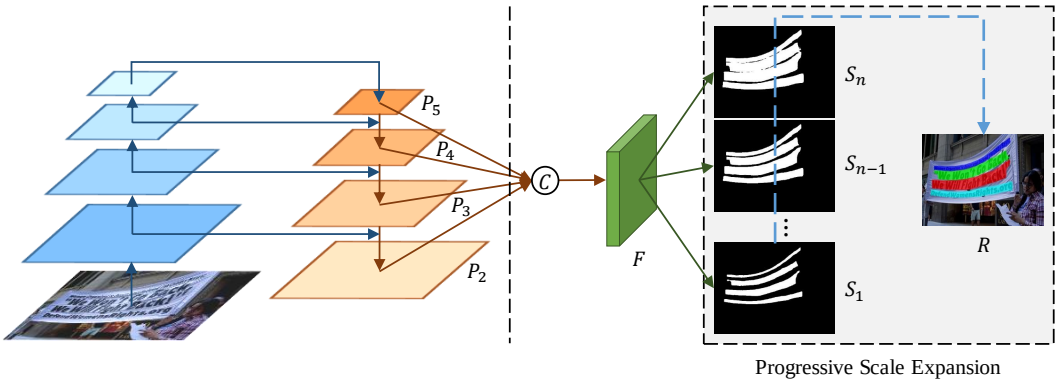


Figure 2: Illustration of our overall pipeline. The left part is implemented from FPN [16]. The right part denotes the feature fusion and the progressive scale expansion algorithm.

[19] utilized corner localization to find suitable irregular quadrangles for text instances. The detection manners are evolving from horizontal rectangle to rotated rectangle and further to irregular quadrangle. However, besides the quadrangular shape, there are many other shapes of text instances in natural scene. Therefore, some researches began to explore curve text detection and obtained certain results. [18] tried to regress the relative positions for the points of a 14-sided polygon. [31] detected curve text by locating two end points in the sliding line which slides both horizontally and vertically. A fused detector was proposed in [1] based on bounding box regression and semantic segmentation. However, since their current performances are not very satisfied, there is still a large space for promotion in curve text detection, and the detectors for arbitrary-shaped texts still need more explorations.

3 Proposed Method

In this section, we first introduce the overall pipeline of the proposed Progressive Scale Expansion Network (PSENet). Next, we present the details of progressive scale expansion algorithm, and show how it can effectively distinguish the adjacent text instances. Further, the way of generating label and the design of loss function are introduced. At last, we describe the implementation details of PSENet.

3.1 Overall Pipeline

The overall pipeline of the proposed PSENet is illustrated in Fig. 2. Inspired by FPN [16], we concatenate low-level feature maps with high-level feature maps and thus have four concatenated feature maps. These maps are further fused in F to encode informations with various receptive views. Intuitively, such fusion is very likely to facilitate the generations of the kernels with various scales. Then the feature map F is projected into n branches to produce multiple segmentation results $S_1, S_2, ..., S_n$. Each S_i would be one segmentation mask for all the text instances at a certain scale. The scales of different segmentation mask are decided by the hyper-parameters which will be discussed in Sec. 3.3. Among these masks, S_1 gives the segmentation result for the text instances with smallest scales (i.e., the minimal kernels) and S_n denotes for the original segmentation mask (i.e., the maximal kernels). After obtaining these segmentation masks, we use progressive scale expansion algorithm to gradually expand all the instances' kernels in S_1, to their complete shapes in S_n, and obtain the final detection results as R.

3.2 Progressive Scale Expansion Algorithm

As shown in Fig. 1 (c), it is hard for segmentation-based method to separate the text instances that are close to each other. To solve this problem, we propose the progressive scale expansion algorithm.

Here is a vivid example (see Fig. 3) to explain the procedure of progressive scale expansion algorithm, whose central idea is brought from the Breadth-First-Search (BFS) algorithm. In the example, we have 3 segmentation results $S = \{S_1, S_2, S_3\}$ (see Fig. 3 (a), (e), (f)). At first, based on the

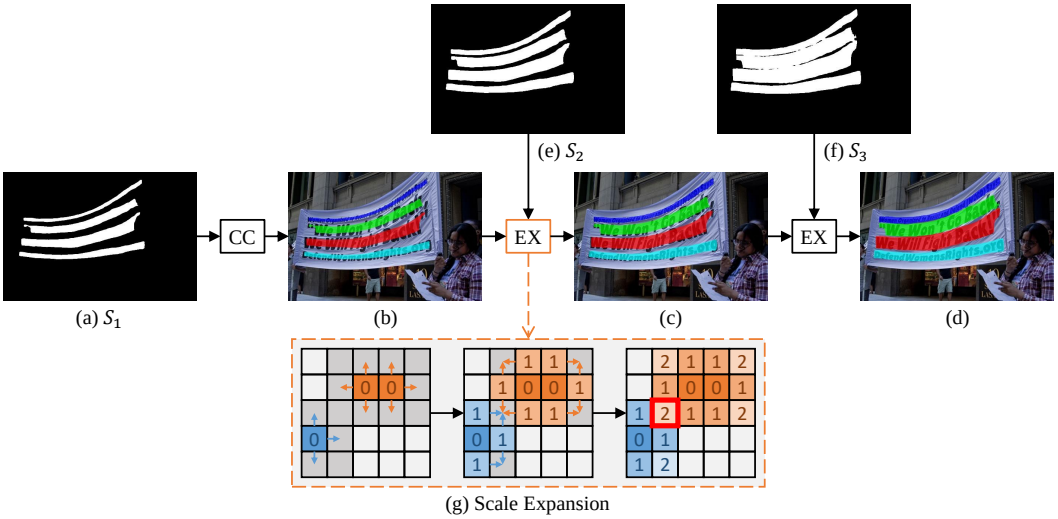


Figure 3: The procedure of progressive scale expansion algorithm. CC refers to the function of finding connected components. EX represents the scale expansion algorithm. (a), (e) and (f) refer to S_1, S_2 and S_3, respectively. (b) is the initial connected components. (c) and (d) is the results of expansion. (g) shows the illustration of expansion. The red box in (g) refers to the conflicted pixel.

Algorithm 1 Scale Expansion Algorithm

Require: Kernels: C, Segmentation Result: S_i
Ensure: Scale Expanded Kernels: E

function EXPANSION(C, S_i)
  $T \leftarrow \emptyset$; $P \leftarrow \emptyset$; $Q \leftarrow \emptyset$
  for each $c_i \in C$ **do**
    $T \leftarrow T \cup \{(p, label) \mid (p, label) \in c_i\}$; $P \leftarrow P \cup \{p \mid (p, label) \in c_i\}$
    Enqueue(Q, c_i) // push all the elements in c_i into Q
  end for
  while $Q \neq \emptyset$ **do**
    $(p, label) \leftarrow$ **Dequeue**(Q) // pop the first element of Q
    if $\exists q \in$ **Neighbor**(p) and $q \notin P$ and $S_i[q] =$ True **then**
      $T \leftarrow T \cup \{(q, label)\}$; $P \leftarrow P \cup \{q\}$
      Enqueue$(Q, (q, label))$ // push the element $(q, label)$ into Q
    end if
  end while
  $E =$ **GroupByLabel**(T)
  return E
end function

minimal kernels' map S_1 (see Fig. 3 (a)), 4 distinct connected components $C = \{c_1, c_2, c_3, c_4\}$ can be found as initializations. The regions with different colors in Fig. 3 (b) represent these different connected components, respectively. By now we have all the text instances' central parts (i.e., the minimal kernels) detected. Then, we progressively expand the detected kernels by merging the pixels in S_2, and then in S_3. The results of the two scale expansions are shown in Fig. 3 (c) and Fig. 3 (d), respectively. Finally, we extract the connected components which are marked with different colors in Fig. 3 (d) as the final predictions for text instances.

The procedure of scale expansion is illustrated in Fig. 3 (g). The expansion is based on Breadth-First-Search algorithm which starts from the pixels of multiple kernels and iteratively merges the adjacent text pixels. Note that there may be conflicted pixels during expansion, as shown in the red box in Fig. 3 (g). The principle to deal with the conflict in our practice is that the confusing pixel can only be merged by one single kernel on a first-come-first-served basis. Thanks to the "progressive" expansion procedure, these boundary conflicts will not affect the final detections and the performances. The detail of scale expansion algorithm is summarized in Algorithm 1. In the pseudocode, T, P are the intermediate results. Q is a queue. Neighbor$(\cdot)$ represents the neighbor pixels of p. GroupByLabel$(\cdot)$ is the function of grouping the intermediate result by label. "$S_i[q] =$ True" means that the predicted value of pixel q in S_i belongs to the text part.

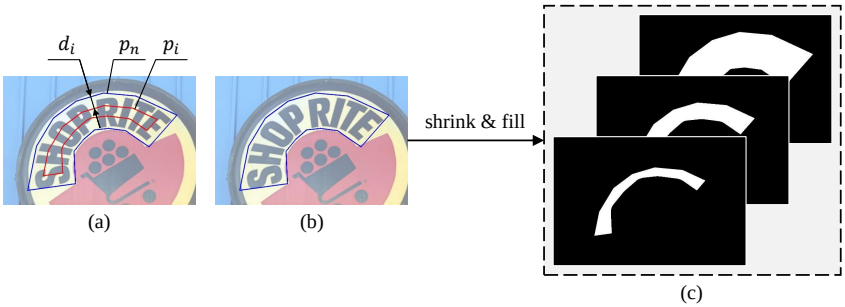


Figure 4: The illustration of label generation. (a) contains the annotations for d, p_i and p_n. (b) shows the original text instances. (c) shows the segmentation masks with different kernel scales.

3.3 Label Generation

As illustrated in Fig. 2, PSENet produces segmentation results (e.g. $S_1, S_2, ..., S_n$) with different kernel scales. Therefore, it requires the corresponding ground truths with different kernel scales as well during training. In our practice, these ground truth labels can be conducted simply and effectively by shrinking the original text instance. The polygon with blue border in Fig. 4 (b) denotes the original text instance and it corresponds to the largest segmentation label mask (see the rightmost map in Fig. 4 (c)). To obtain the shrunk masks sequentially in Fig. 4 (c), we utilize the Vatti clipping algorithm [28] to shrink the original polygon p_n by d_i pixels and get shrunk polygon p_i (see Fig. 4 (a)). Subsequently, each shrunk polygon p_i is transferred into a 0/1 binary mask for segmentation label ground truth. We denote these ground truth maps as $G_1, G_2, ..., G_n$ respectively. Mathematically, if we consider the scale ratio as r_i, the margin d_i between p_n and p_i can be calculated as:

$$d_i = \frac{\text{Area}(p_n) \times (1 - r_i^2)}{\text{Perimeter}(p_n)}, \tag{1}$$

where Area$(\cdot)$ is the function of computing the polygon area, Perimeter$(\cdot)$ is the function of computing the polygon perimeter. Further, we define the scale ratio r_i for ground truth map G_i as:

$$r_i = 1 - \frac{(1 - m) \times (n - i)}{n - 1}, \tag{2}$$

where m is the minimal scale ratio, which is a value in $(0, 1]$. Based on the definition in Eqn. (2), the values of scale ratios (i.e., $r_1, r_2, ..., r_n$) are decided by two hyper-parameters n and m, and they increase linearly from m to 1.

3.4 Loss Function

For learning PSENet, the loss function can be formulated as:

$$L = \lambda L_c + (1 - \lambda) L_s, \tag{3}$$

where L_c and L_s represent the losses for the complete text instances and the shrunk ones respectively, and λ balances the importance between L_c and L_s.

It is common that the text instances usually occupy only an extremely small region in natural images, which makes the predictions of network bias to the non-text region, when binary cross entropy [2] is used. Inspired by [20], we adopt dice coefficient in our experiment. The dice coefficient $D(S_i, G_i)$ is formulated as in Eqn. (4):

$$D(S_i, G_i) = \frac{2 \sum_{x,y} (S_{i,x,y} * G_{i,x,y})}{\sum_{x,y} S_{i,x,y}^2 + \sum_{x,y} G_{i,x,y}^2}, \tag{4}$$

where $S_{i,x,y}$ and $G_{i,x,y}$ refer to the value of pixel (x, y) in segmentation result S_i and ground truth G_i, respectively.

Furthermore, there are many patterns similar to text strokes, such as fences, lattices, etc. Therefore, we adopt Online Hard Example Mining (OHEM) [24] to L_c during training to better distinguish these patterns.

L_c focuses on segmenting the text and non-text region. Let us consider the training mask given by OHEM as M, and thus L_c can be written as:

$$L_c = 1 - D(S_n \cdot M, G_n \cdot M), \tag{5}$$

L_s is the loss for shrunk text instances. Since they are encircled by the original areas of the complete text instances, we ignore the pixels of non-text region in the segmentation result S_n to avoid a certain redundancy. Therefore, L_s can be formulated as follows:

$$L_s = 1 - \frac{\sum_{i=1}^{n-1} D(S_i \cdot W, G_i \cdot W)}{n-1}, \quad W_{x,y} = \begin{cases} 1, & if\ S_{n,x,y} \geq 0.5; \\ 0, & otherwise. \end{cases} \tag{6}$$

Here, W is a mask which ignores the pixels of non-text region in S_n, and $S_{n,x,y}$ refers to the value of pixel (x, y) in S_n.

3.5 Implementation Details

The backbone of PSENet is implemented from FPN [16]. We firstly get four 256 channels feature maps (i.e. P_2, P_3, P_4, P_5) from the backbone. To further combine the semantic features from low to high levels, we fuse the four feature maps to get feature map F with 1024 channels via the function $C(\cdot)$ as: $F = C(P_2, P_3, P_4, P_5) = P_2 \parallel \text{Up}_{\times 2}(P_3) \parallel \text{Up}_{\times 4}(P_4) \parallel \text{Up}_{\times 8}(P_5)$, where "$\parallel$" refers to the concatenation and $\text{Up}_{\times 2}(\cdot)$, $\text{Up}_{\times 4}(\cdot)$, $\text{Up}_{\times 8}(\cdot)$ refer to 2, 4, 8 times upsampling, respectively. Subsequently, F is fed into Conv$(3, 3)$-BN-ReLU layers and is reduced to 256 channels. Next, it passes through multiple Conv$(1, 1)$-Up-Sigmoid layers and produces n segmentation results $S_1, S_2, ..., S_n$. Here, Conv, BN, ReLU and Up refer to convolution [14], batch normalization [11], rectified linear units [4] and upsampling.

We set n to 6 and m to 0.5 for label generation and get the scales $\{0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. During training, we ignore the blurred text regions labeled as DO NOT CARE in all datasets. The λ of loss balance is set to 0.7. The negative-positive ratio of OHEM is set to 3. The data augmentation for training data is listed as follows: 1) the images are rescaled with ratio $\{0.5, 1.0, 2.0, 3.0\}$ randomly; 2) the images are horizontally fliped and rotated in range $[-10°, 10°]$ randomly; 3) 640×640 random samples are cropped from the transformed images; 4) the images are normalized using the channel means and standard deviations. For quadrangular text dataset, we calculate the minimal area rectangle to extract the bounding boxes as final predictions. For curve text dataset, the Ramer-Douglas-Peucker algorithm [21] is applied to generate the bounding boxes with arbitrary shapes.

4 Experiment

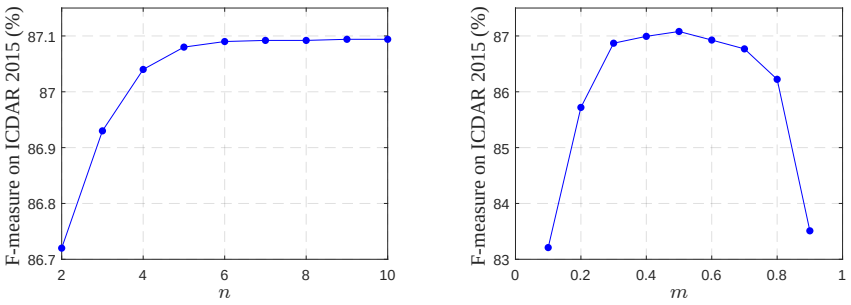
In this section, we first conduct ablation studies for PSENet. Then, we evaluate the proposed PSENet on three recent challenging public benchmarks: ICDAR 2015, ICDAR 2017 MLT and SCUT-CTW1500 and compare PSENet with many state-of-the-art methods.

4.1 Benchmark Datasets

ICDAR 2015 (IC15) [13] is a commonly used dataset for text detection. It contains a total of 1500 pictures, 1000 of which are used for training and the remaining are for testing. The text regions are annotated by 4 vertices of the quadrangle.

ICDAR 2017 MLT (IC17-MLT) [27] is a large scale multi-lingual text dataset, which includes 7200 training images, 1800 validation images and 9000 testing images. The dataset is composed of complete scene images which come from 9 languages. Similarly with ICDAR 2015, the text regions in ICDAR 2017 MLT are also annotated by 4 vertices of the quadrangle.

SCUT-CTW1500 is a challenging dataset for curve text detection, which is constructed by Yuliang et al. [18]. It consists of 1000 training images and 500 testing images. Different from traditional text datasets (e.g., ICDAR 2015, ICDAR 2017 MLT), the text instances in SCUT-CTW1500 are labelled by a polygon with 14 points which can discribe the shape of an arbitrarily curve text.



(a) (b)

Figure 5: Ablation study on m and n (Eqn. (2)). These results are based on PSENet-1s (Table 2).

Table 1: Comparison to the traditional semantic segmentation baseline with the same backbone on ICDAR 2015. "P", "R", "F" refer to Precision, Recall, F-measure respectively.

Method	P (%)	R (%)	F (%)
PSENet-1s ($n = 1, m = 1.0$, semantic segmentation baseline)	77.41	61.53	68.56
PSENet-1s ($n = 6, m = 0.5$)	88.71	85.51	**87.08**

4.2 Training

We use the FPN with ResNet [7] pre-trained on ImageNet dataset [3] as our backbone. All the networks are trained by using stochastic gradient descent (SGD). In the experiments on ICDAR datasets, we use 1000 IC15 training images, 7200 IC17-MLT training images and 1800 IC17-MLT validation images to train the model and report the precision, recall and F-measure on the test set of both datasets at the end of training. We use batch size 16 and train the models for 300 epochs. The initial learning rate is set to 1×10^{-3}, and is divided by 10 at 100 and 200 epoch. On SCUT-CTW1500, we use the 1000 training images to fine-tune the model from the trained model for ICDAR datasets for 400 epochs. The batch size is set to 16 for fine-tuning. The initial learning rate is set to 10^{-4}, and is divided by 10 at 200 epoch. At the end of fine-tuning, we report the precision, recall and F-measure on the test set. We use a weight decay of 5×10^{-4} and a Nesterov momentum [25] of 0.99 without dampening. We adopt the weight initialization introduced by [5].

4.3 Ablation Study

Why are the multiple kernel scales necessary? To answer this question, we investigate the effect of the number of scales n on the performance of PSENet. Specifically, we hold the minimal scale m constant and train PSENet with different n. In details, we set m to 0.5 and let n increase from 2 to 10. The models are evaluated on ICDAR 2015 dataset. Fig. 5 (a) shows the experimental results, from which we can find that with the growing of n, the F-measure on the test set keeps rising and begins to level off when $n \geq 6$. The informative result suggests that the design of multiple kernel scales is essential and effective, and we also donot need too many scales for the purpose of the efficiency. The original ablation study for n starts from $n = 2$ and fixes $m = 0.5$. Additionally, there is an extreme case when $n = 1$ and $m = 1$, which means we only use the traditional semantic segmentation method to deal with this task. Here we conduct the experiment by setting $n = 1$ and $m = 1$ to serve as a baseline with only one segmentation mask result for predictions. Table 1 shows the huge performance gap between these two settings, and it further validate the effectiveness of the design of multiple kernel scales.

How minimal can these kernels be? We then study the effect of the minimal scale m by setting the number of scales n to 6 and let the minimal scale m vary from 0.1 to 0.9. The models are also evaluated on ICDAR 2015 dataset. We can find from Fig. 5 (b) that the F-measure on the test set drops when m is too large or too small. When m is too large, it is hard for PSENet to separate the text instances standing closely to each other. When m is too small, PSENet often splits a whole text line into different parts incorrectly and the training can not converge very well.

4.4 Comparisons with State-of-the-Art Methods

Detecting Quadrangular Text. We evaluate the proposed PSENet on the ICDAR 2015 and ICDAR 2017 MLT datasets to test its ability of detecting the oriented quadrangular text. During testing, we resize the longer side of input images to 2240 and 3200 for ICDAR 2015 and ICDAR 2017 MLT, respectively. For fair comparisons, we report all the single-scale results on these two datasets.

We compare our method with other state-of-the-art methods in Table 2. Our method outperforms almost all the state-of-the-art methods in the aspect of F-measure. On ICDAR 2015, PSENet achieves the best recall (85.51%) and obtains the comparable F-measure with the FOTS [17]. Furthermore, our results on ICDAR 2017 MLT are even more encouraging with the best F-measure (72.45%), which surpass the second best method FOTS [17] by absolute 5.2%. The competitive results on the both ICDAR datasets validate the effectiveness of PSENet in the mainstream quadrangular text detection tasks. In addition, we demonstrate some test examples in Fig. 6 (a) (b), and PSENet can accurately locate the text instances with various orientations. Furthermore, we also test the speed (FPS) on NVIDIA GTX 1080 Ti (Table 2) and confirm the satisfactory efficiency of PSENet.

Table 2: The single-scale results on ICDAR 2015, ICDAR 2017 MLT and SCUT-CTW1500. "P", "R" and "F" refer to precision, recall and F-measure respectively. * indicates the results from [18]. "1s", "2s" and "4s" means the width and height of the output map are $1/1$, $1/2$ and $1/4$ of the input test image. The best, second-best F-measure are highlighted in red and blue, respectively.

Method	IC15				IC17-MLT			SCUT-CTW1500		
	P (%)	R (%)	F (%)	FPS	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
CTPN [26]	74.22	51.56	60.85	7.1	-	-	-	60.4*	53.8*	56.9*
SegLink [23]	74.74	76.50	75.61	-	-	-	-	42.3*	40.0*	40.8*
SSTD [8]	80.23	73.86	76.91	7.7	-	-	-	-	-	-
WordSup [9]	79.33	77.03	78.16	-	-	-	-	-	-	-
EAST [30]	83.27	78.33	80.72	6.52	-	-	-	78.7*	49.1*	60.4*
R^2CNN [12]	85.62	79.68	82.54	-	-	-	-	-	-	-
FTSN [1]	88.65	80.07	84.14	-	-	-	-	-	-	-
SLPR [31]	85.5	83.6	84.5	-	-	-	-	80.1	70.1	74.8
linkage-ER-Flow [27]	-	-	-	-	44.48	25.59	32.49	-	-	-
TH-DL [27]	-	-	-	-	67.75	34.78	45.97	-	-	-
TDN SJTU2017 [27]	-	-	-	-	64.27	47.13	54.38	-	-	-
SARI FDU RRPN v1 [27]	-	-	-	-	71.17	55.50	62.37	-	-	-
SCUT DLVClab1 [27]	-	-	-	-	80.28	54.54	64.96	-	-	-
Lyu et al. [19]	89.5	79.7	84.3	3.6	83.8	55.6	66.8	-	-	-
FOTS [17]	91.00	85.17	87.99	7.5	80.95	57.51	67.25	-	-	-
CTD+TLOC [18]	-	-	-	-	-	-	-	77.4	69.8	73.4
PSENet-4s (ours)	87.98	83.87	85.88	12.38	75.98	67.56	71.52	80.49	78.13	79.29
PSENet-2s (ours)	89.30	85.22	87.21	7.88	76.97	68.35	72.40	81.95	79.30	80.60
PSENet-1s (ours)	88.71	85.51	87.08	2.33	77.01	68.40	72.45	82.50	79.89	81.17

Detecting Curve Text. To test the ability of detecting arbitrarily shaped text, we evaluate our method on SCUT-CTW1500, which mainly contains the curve texts. In test stage, we resize the longer side of images to 1280 and evaluate the results using the same evaluation method with [18]. We report the single-scale performance on SCUT-CTW1500 in Table 2, in which we can find that the precision (82.50%), recall (79.89%) and F-measure (81.17%) achieved by PSENet significantly outperform the ones of other competitors. Remarkably, it surpasses the second best record by 6.37% in F-measure. The result on SCUT-CTW1500 demonstrates the solid superiority of PSENet when detecting curve or arbitrarily shaped texts. We also illustrate several challenging results in Fig. 6 (c) and make some visual comparisons to the state-of-the-art CTD+TLOC [18] in Fig. 6 (1) (2). The comparisons clearly demonstrate that PSENet can successfully distinguish very complex curve text instances.

4.5 More Comparisons on SCUT-CTW1500

In this section, we show more comparisons on SCUT-CTW1500 in Fig. 7, 8, 9. It is interesting and amazing to find that in Fig. 7, our proposed PSENet is able to locate several text instances where the groundtruth labels are even unmarked. This highly proves that our method is quite robust due to its strong learning representation and distinguishing ability. Fig. 8 and 9 demonstrate more examples

(a) ICDAR 2015 (b) ICDAR 2017 MLT (c) SCUT-CTW1500 (1) CTD+TLOC (2) PSENet

Figure 6: PSENets' results on three benchmarks and several representative comparisons of curve texts on SCUT-CTW1500. More examples are provided in the **supplementary materials**.

where PSENet can not only detect the curve text instances even with extreme curvature, but also separate those close text instances in a good manner.

Original Image Ground Truth CTD+TLOC PSENet (ours)

Figure 7: Comparisons on SCUT-CTW1500. The proposed PSENet produces several detections that are even missed by the groundtruth labels.

4.6 More Detected Examples on ICDAR 2015 and ICDAR 2017 MLT

In this section, we demonstrate more test examples produced by the proposed PSENet in Fig. 10 (ICDAR 2015) and Fig. 11 (ICDAR 2017 MLT). From these results, it can be easily observed that with the progressive scale expansion mechanism, our method is able to separate those text instances that are very close to each other, and it is also very robust to various orientations. Meanwhile, thanks to the strong feature representation, PSENet can as well locate the text instances with complex and unstable illumination, different colors and variable scales.

5 Conclusion and Future Work

We propose a novel Progressive Scale Expansion Network (PSENet) to successfully detect the text instances with arbitrary shapes in natural scene images. By gradually expanding the detected areas from small kernels to large and complete instances via multiple semantic segmentation maps, our method is robust to shapes and can easily distinguish those text instances which are very close or

Figure 8: Comparisons on SCUT-CTW1500.

Figure 9: Comparisons on SCUT-CTW1500.

even partially intersected. The experiments on scene text detection benchmarks demonstrate the superior performance of the proposed method.

There are multiple directions to explore in the future. Firstly, we will investigate whether the scale expansion algorithm can be trained along with the network in an end-to-end manner. Secondly, the progressive scale expansion algorithm can be introduced to the general instance-level segmentation tasks, especially in those bencnmarks with many crowded object instances. We are cleaning our codes and will release them soon.

References

[1] Yuchen Dai, Zheng Huang, Yuting Gao, and Kai Chen. Fused text segmentation networks for multi-oriented scene text detection. *arXiv preprint arXiv:1709.03272*, 2017.

[2] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 2005.

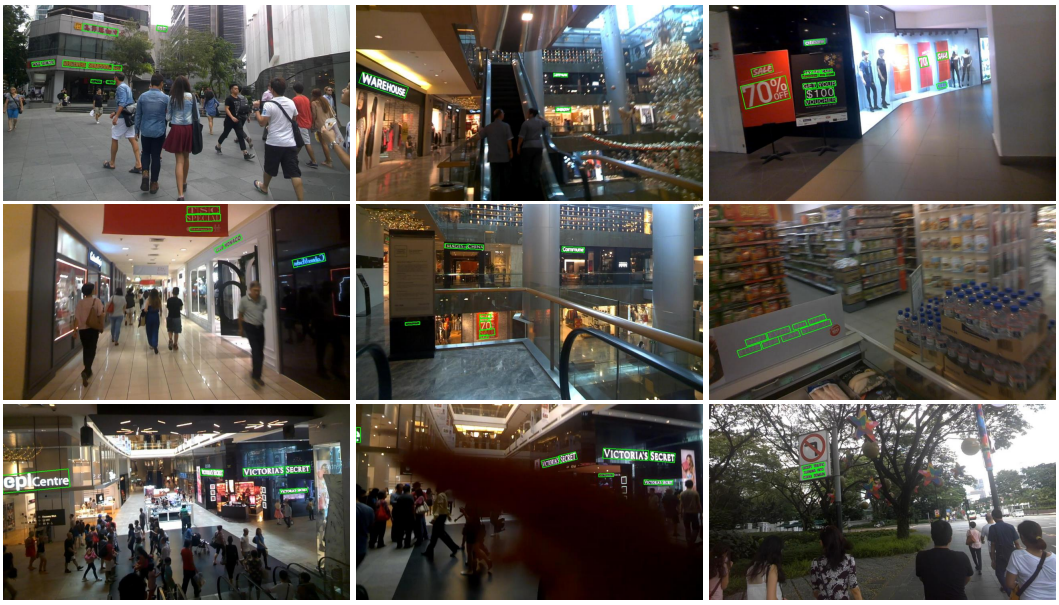


Figure 10: Test examples on ICDAR 2015 produced by PSENet.

Figure 11: Test examples on ICDAR 2017 MLT produced by PSENet.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *ICAIS*, 2011.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[8] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *ICCV*, 2017.

[9] Han Hu, Chengquan Zhang, Yuxuan Luo, Yuzhuo Wang, Junyu Han, and Errui Ding. Wordsup: Exploiting word annotations for character based text detection. In *ICCV*, 2017.

[10] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[11] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[12] 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.

[13] 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 *ICDAR*, 2015.

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

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

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[17] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. *arXiv preprint arXiv:1801.01671*, 2018.

[18] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, and Sheng Zhang. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017.

[19] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. *arXiv preprint arXiv:1802.08948*, 2018.

[20] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *IC3DV*, 2016.

[21] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *CGIP*, 1972.

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[23] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, 2017.

[24] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, 2016.

[25] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[26] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, 2016.

[27] "`http://rrc.cvc.uab.es/?ch=8&com=introduction`". Icdar2017 competition on multi-lingual scene text detection and script identification. *None*, 2017.

[28] Bala R Vatti. A generic solution to polygon clipping. *Communications of the ACM*, 1992.

[29] Zhuoyao Zhong, Lianwen Jin, Shuye Zhang, and Ziyong Feng. Deeptext: A unified framework for text proposal generation and text detection in natural images. *arXiv preprint arXiv:1605.07314*, 2016.

[30] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. *arXiv preprint arXiv:1704.03155*, 2017.

[31] Yixing Zhu and Jun Du. Sliding line point regression for shape robust scene text detection. *arXiv preprint arXiv:1801.09969*, 2018.