Multimedia Tools and Applications (2020) 79:29225–29242
https://doi.org/10.1007/s11042-020-09440-1

# Arbitrary-shaped text detection with adaptive convolution and path enhancement pyramid network

Qi Cheng[1] · Guodong Wang[1] · Qian Dong[2] · Bin Wei[3]

Received: 27 December 2019 / Revised: 22 July 2020 / Accepted: 28 July 2020 /
Published online: 10 August 2020

**Abstract**
Recently, scene text detection has become an active research field, which is an essential component of scene text reading. Especially, segmentation-based methods are commonly used, since the segmentation results can describe text of arbitrary shape. However, curve texts have a diversity of shapes, scales and orientations, which are difficult to locate, so the detector requires to adjust the local receptive fields size adaptively, which can aggregate multi-scale spatial information to accurately locate the curve text instance. Moreover, the low-level features are critical for localizing large text instances. When using Feature Pyramid Network (FPN) for multi-scale feature fusion, it will prevent the flow of accurate localization signals due to the long path from low-level to top-level. In order to solve these two problems, this paper proposes an Adaptive Convolution and Path Enhancement Pyramid Network (ACPEPNet), which can more accurately locate the text instances with arbitrary shapes. Firstly, an Adaptive Convolution Unit is introduced to improve the ability of backbone to aggregate multi-scale spatial information at the same stage. Specially, this unit is a lightweight component and without the cost of computations, based on this component we present a backbone network for text features extraction. Secondly, the original FPN structure is redesigned to build a short path from the low-level to top-level, in this way, we modify the path from one-way flow to two-way flow and add original features to the final stage of information fusion. Experiments on CTW1500, Total-Text, ICDAR 2015 and MSRA-TD500 validate the robustness of the proposed method. When there is no bells and whistles, this method achieves an F-measure of 80.8% without external training data on CTW1500.

**Keywords** Arbitrary shapes text detection · Adaptive convolution · Backbone network · Feature pyramid network · Multi-scale

✉ Guodong Wang
doctorwgd@gmail.com

Extended author information available on the last page of the article

## 1 Introduction

In recent years, scene text detection as a fundamental computer vision task has become an active research field, since it is an essential step in many applications such as automatic driving, scene understanding and text recognition. With the rapid development of Convolutional Neural Networks [7, 9, 13, 17, 46, 47], many progresses have been made [19, 20, 39, 43]. Scene text detection methods can be roughly formulated as two categories: regression-based methods and segmentation-based methods, especially segmentation-based methods have received much attention, since the segmentation results can describe text of arbitrary shape such as curve text. Some new approaches [19, 24, 27, 45] have been proposed to detect cure texts. On the one hand, many of these approaches usually employ classification networks as the backbone [7, 10, 42] network. However, due to the diversity of curve text in shape, scale and orientation, therefore, the detector requires to adjust the local receptive fields size adaptively to encode sufficient context information. Based on the fact, it is not optimal that simply transferring classification networks to text detection. On the other hand, Multi-scale detection is very important to text detection, as the high layers strongly respond to global features while the shallow layers are likely to retain local features. Most of previous methods use Feature Pyramid Network (FPN) [23] to extract multi-scale feature. Nevertheless, it will prevent the flow of accurate localization signals due to the long path from low-level to top-level.

To address above problems, an arbitrary shaped text detector is proposed in this paper, namely, Adaptive Convolution and Path Enhancement Pyramid Network (ACPEPNet). The proposed detector as a segmentation-based method which can makes arbitrary shape text detection. The pipeline of this method is as follows, which includes two steps: 1) Using segmentation network to obtain the segmentation maps. 2) Converting the segmentation maps to binarization maps and then reconstructing the text regions by post-processing algorithm. Firstly, in order to make detector adjust the local receptive fields size adaptively and improve non-linear aggregation capability, EfficientNet [42] is used as the backbone and Adaptive Convolution Unit is embedded into it. By redesigning the structure of EfficientNet, a set of backbone networks named ACNet-B0 and ACNet-B3 are proposed, which are designed for curve text detection. Acnet-b0 can achieve better efficiency / accuracy trade-off, while ACNet-B3 can achieve better accuracy. Compared with classification networks, ACNet can bring significant improvement to text detection tasks. Secondly, in order to make the low-level features flow into the top-level more smoothly, a Path Enhancement Feature Pyramid Network (PEFPN) is proposed, which constructs an extremely short path less than 10 layers and adds original feature maps to the final stage of information fusion. For high efficiency, we use depthwise separable convolution [4] instead of conventional convolution in PEFPN.

To show the effectiveness of the proposed method, experiments are carried out on four public benchmark datasets including CTW1500 [24], Total-Text [1], ICDAR2015 [16] and MSRA-TD500 [49]. Among these datasets, CTW1500 and Total-Text are specially designed for curve text detection, ICDAR2015 is multi-oriented text detection datasets and MSRA-TD500 is multi-lingual text datasets. On CTW1500, on ctw1500, when using ACNet-B3 as backbone network, the F-measure is 80.8%, which is 2.8% better than PSENet [45]. Meanwhile, this method also achieves promising performance on multi-oriented and multi-lingual text datasets.

The contributions can be summarized as:

- We introduced an adaptive convolution unit, which can adjust the local receptive fields size adaptively and nonlinear aggregate multi-scale spatial information.
- We proposed ACNet, a backbone network designed for text detection, which can improve the result of curve text detection significantly.
- We proposed PEFPN, a two-way feature pyramid network and it is benefit for cross-scale feature fusion.

## 2 Related work

Recent scene text detection task based on deep learning methods have achieved remarkable results. Scene text detection methods can be roughly formulated as two categories: regression-based methods and segmentation-based methods.

**Regression-based methods** usually based on general object detection benchmark [28, 29, 37, 53], which directly regress the bounding boxes of the text instances. TextBoxes [18] directly modified the anchor scales and the shape of convolutional kernels to deal with the text with different aspect ratios. TextBoxes++ [21] adopt quadrilaterals to regress the multi-oriented text. SSTD [36] introduced an attention mechanism to roughly recognizes the text regions. RRD [22] extracted rotation-invariant features for text classification and utilized rotation-sensitive features for text regression which are better for multi-oriented and long text detection. EAST [54] and DeepReg [11] are based on anchor-free, which utilize pixels to directly regress the multi-oriented text instances. SegLink [39] use the segments of bounding box for regression and studied their links, to handle long text detection.

However, most of the above-mentioned methods rely on complex anchor design, which makes these works heavy-footed and result in sub-optimal performance. In addition, these methods were specially proposed for multiple oriented text detection, which are limited to represent quadrilateral bounding boxes and may fall short when dealing with curve texts.

**Segmentation-based methods** are mainly joined pixel-level prediction benchmark [2, 3, 14, 25] and post-processing algorithms to get the bounding boxes. Zhang et al. [52] extracted text regions by semantic segmentation and adopted MSER to detect character candidate. Yao et al. [51] formulated one text block as three parts, then predicted the corresponding heat-maps by FCN [26]. Lyu et al. [32] adopted corner localization and represented the bounding box with irregular quadrangles. PixelLink [6] predicted pixel connections, to separate texts which are lying close to each other. TextSnake [27] represented curve text for arbitrary shapes text detection by model ordered disks. SPCNet [48] utilized instance segmentation benchmark and use context information to detect curve texts while suppressing false positives. PSENet [45] proposed progressive scale expansion algorithm to construct the text instances by setting multi-scale kernels.

The above methods have achieved remarkable performances over several horizontal and multi-oriented text benchmarks. Nonetheless, except for TextSnake [27], SPCNet [48] and PSENet [45], most of methods have not focused on curve text. However, these methods have not considered the significance of multi-scale receptive field and low-level feature for curve text detection.

# 3 Methods

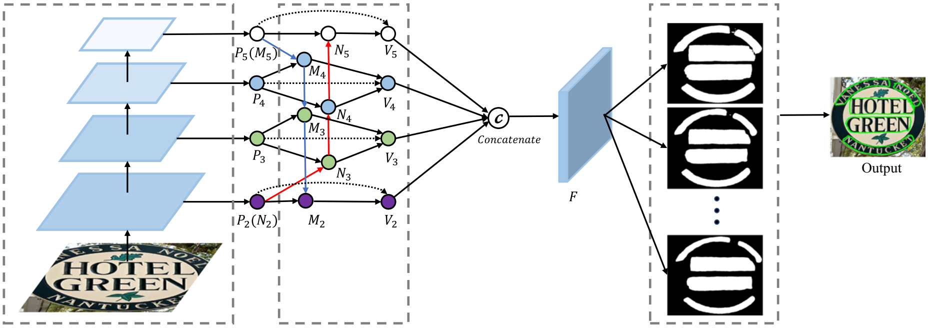
## 3.1 Overall architecture

The overall architecture of our method as shown in Fig. 1. Firstly, the input images are fed into ACNet, our proposed PEFPN serves as the feature extraction and cross-scale fusion network, which takes level 2–5 feature maps $\{P_2, P_3, P_4, P_5\}$ from the ACNet, and we aligned the feature maps to same dimension, which have stride of {4,8,16,32} pixels with respect to the input image. Secondly, feature maps are fused through top-down and bottom-up pathway, we denote the feature maps as $\{M_2, M_3, M_4, M_5\}$ and $\{N_2, N_3, N_4, N_5\}$ respectively, $M_i$ and $N_i$ are merged with the corresponding $P_i$ by element-wise addition. Thirdly, the outputs $V_i$ of PEFPN are upsampled to the same scale and concatenated to produce feature map $F$. Finally, we use progressive scale expansion algorithm [45] as post-processing to obtain the final results.

## 3.2 Adaptive convolution unit

To make the network adjust the local receptive fields size adaptively and improve non-linear aggregation capability, we introduced an operation which can select kernel size automatically. This operation is divided into three steps. We only use two branches in parallel as an example, however, it can be extended to multi-branch parallel easily. Figure 2 show the architecture of adaptive convolution unit. Next, we will discuss each step in detail.

Step 1: Given a feature map $X \in R^{H \times W \times C}$and projected $F$ into multiple branches in parallel with different kernel sizes for convolution operations, which can be formulated as two transformation $\mathcal{F}_1$ mapping $X \in R^{H \times W \times C}$ to $X_1 \in R^{H \times W \times C}$ and $\mathcal{F}_2$ mapping $X \in R^{H \times W \times C}$ to $X_2 \in R^{H \times W \times C}$. We take $\mathcal{F}_1$ and $\mathcal{F}_2$ as two convolution operators, to prevent channel dependencies, we use depthwise/dilated convolution [30, 31] for feature extraction and then followed by Batch Normalization [15] and ReLU [35] activation. Notably, the kernel sizes are $3 \times 3$ and $5 \times 5$, specially, we use $3 \times 3$ kernel and dilation size 2 to instead of $5 \times 5$ kernel. After this stage, the network is able to pay attention to the multi-scale feature in the same layer.



(a) ACNet (b) PEFPN (c) Post-processing

**Fig. 1** The overall architecture of our detector, **a** ACNet, the backbone network is designed for curve text detection. **b** PEFPN, the feature extraction and multi-scale fusion network. Different colored circles indicate different levels of feature maps. Top-down pathway is visualized in blue arrows, bottom-up pathway is displayed in red arrows, respectively. **c** Post-processing algorithm to obtain the result

Step 2: To enable the network to improve non-linear spatial aggregation capability, the kernels need to adaptively select their receptive fields size according to the different stimulate, we first fuse features from multi branches by element-wise addition:

$$X^{'} = X_1 + X_2, \tag{1}$$

Due to each of kernel only with a local receptive field, therefor, the output $X'$ is unable to exploit contextual information while it is essential for network sensitivity. Consequently, we obtained the global spatial information via global average pooling to generate channel-wise statistics, denote as $\widetilde{X} \in R^c$, shrinking $X'$ through its 2D spatial dimensions $H \times W$, specially, $\widetilde{X}_c$ is the *i-th* element of $\widetilde{X}$ can be calculated by:

$$\widetilde{X}_c = \frac{\sum_{i=1}^{H} \sum_{j=1}^{W} X_c^{'}(i, j)}{H \times W}, \tag{2}$$

To limit complexity, we introduce a dimensionality-reduction operation, which is composed of two fully connected layers. In particular, we use several 1 × 1 convolutions instead, they can be simply defined as $W_1 \in R^{d \times C}$, $W_{21} \in R^{C \times d}$ and $W_{22} \in R^{C \times d}$:

$$S_1 = W_{21}\alpha\left(\beta\left(W_1\widetilde{X}\right)\right), S_2 = W_{22}\alpha\left(\beta\left(W_1\widetilde{X}\right)\right), \tag{3}$$

where $\beta$ refers to the Batch Normalization [15] and $\alpha$ represents the ReLU [35] activation. we use $d$ to control the compactness of $S_i \in R^{C \times 1}$, $d$ takes the maximum of $C/r$ and $L$. $r$ denotes dimensionality-reduction ratio, $L$ is the minimum value of $d$ ($L = 8$):

$$d = max\left(\frac{C}{r}, L\right), \tag{4}$$

Step 3: In order to select the multi-scales spatial information adaptively, we adopted the SoftMax as self-attention function:

$$\mu_1 = \frac{e^{S_1}}{e^{S_1} + e^{S_2}}, \mu_2 = \frac{e^{S_2}}{e^{S_1} + e^{S_2}}, \tag{5}$$

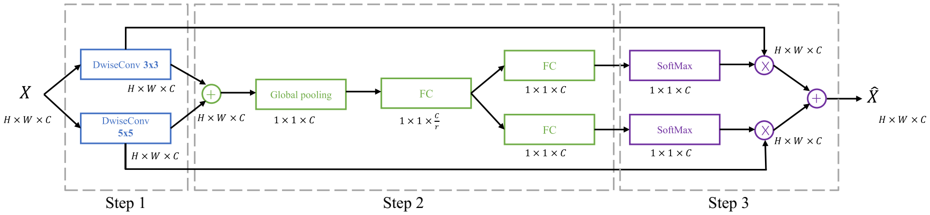


**Fig. 2** Adaptive Convolution Unit

Where $\mu_1 \in R^{C \times 1}$ and $\mu_2 \in R^{C \times 1}$, we conduct $\mu_1$ and $\mu_2$ as the attention scores, which represent the sensitivity of network to multi-scales spatial information. The final outputs $\widehat{X}$ are produced by channel-wise multiplication:

$$\widehat{X} = \mu_1 \cdot X_1 + \mu_2 \cdot X_2, \tag{6}$$

### 3.3 Backbone network design

Table 1 show the structure of our proposed backbone ACNet. We start from EfficientNet [42] for three reasons: 1) It is the state-of-the-art network with high efficiency on classification. 2) Compared with other networks, it has fewer parameters and FLOPs without losing accuracy. 3) The network architecture of EfficientNet [42] is obtained by reinforcement learning search algorithms, rather than artificial design, therefore, it has a better balance among depth, width and resolution. EfficientNet [42] is mainly composed of a stack of repeated mobile inverted bottleneck MBConv [38, 44] and we still follow this design. Moreover, in order for the model to adaptively select the receptive field size, we use Adaptive Convolution Unit instead of depthwise convolution in each MBConv [38, 44] block, and then each block consists of a sequence of 1 × 1 expand convolution, adaptive convolution unit and 1 × 1 project convolution. Adaptive convolution unit is imposed only a slight increase in parameter and computational cost.

As shown in Table 1, ACNet-B0 has seven stages with {1,2,2,3,3,4,1} MBconv [38, 44] blocks, respectively. In adaptive convolution unit, $K$ is the number of paths that controls the number of choices of different kernels to be aggregated, and the dimensionality-reduction ratio $r$ that determines the number of parameters in Step. 2(see Eq. (4)). AC [6, 50] is the typical setting in ACNet. In addition, we also design a deeper network based on EfficientNet-B3 [42] for better accuracy and named ACNet-B3, which has seven stages with {1,2,3,3,5,5,6,2} MBconv [38, 44] blocks.

### 3.4 Path enhancement feature pyramid

In the backbone network, low level has larger feature maps and richer spatial details, it is more likely to describe local texture and patterns. On the contrary, high level has smaller feature maps and strongly respond to entire text instances. Generally, localization is more sensitive to low level features, especially for arbitrary shapes text detection, due to the irregularity of text shapes, the network needs to capture more sensitive edge information. However, features are restricted by one-way flow in FPN [23], consequently, it is necessary to build a two-way path to propagate semantically strong features and enhance all features with reasonable classification capability. To address this problem, we further enhance the localization capability of the entire feature hierarchy by spreading the strong response of low-level information. we build an extremely short path, which includes of less than 10 layers to prevent the loss of local features after a lengthy backbone.

We takes level 2–5 feature maps $\{P_2, P_3, P_4, P_5\}$ from the backbone network, and we align the feature maps to same dimension, which have stride of {4,8,16,32} pixels with respect to the input image. Different from traditional FPN [23], we reduced the dimension to 64 for efficiency, and then fuse the multi-scale features through the following steps:

**Table 1** The architecture of ACNet-B0 and ACNet-B3 network with different channels and layers, each row refers to a stage $i$ with $\widehat{L}_i$ MBconv [38, 44] blocks, with input resolution $\widehat{H}_i \times \widehat{W}_i$ and the MBconv [38, 44] block of each stage $i$ includes the corresponding components listed in operator

| Stage<br>$i$ | Resolution<br>$\widehat{H}_i \times \widehat{W}_i$ | Operator<br>$\widehat{\mathcal{F}}_i$ | #Channels<br>$\widehat{C}_i$ | #Layers<br>$\widehat{L}_i$ |
|---|---|---|---|---|
| 1 | 640×640 | Conv3x3 | 32/40 | 1 |
| 2 | 320×320 | AC [$K=2$, $r=4$]<br>Conv1x1 | 16/24 | 1/2 |
| 3 | 320×320 | Conv1x1<br>AC [$K=2$, $r=4$]<br>Conv1x1 | 24/32 | 2/3 |
| 4 | 160×160 | Conv1x1<br>AC [$K=2$, $r=4$]<br>Conv1x1 | 40/48 | 2/3 |
| 5 | 80×80 | Conv1x1<br>AC [$K=2$, $r=4$]<br>Conv1x1 | 80/96 | 3/5 |
| 6 | 40×40 | Conv1x1<br>AC [$K=2$, $r=4$]<br>Conv1x1 | 112/136 | 3/5 |
| 7 | 40×40 | Conv1x1<br>AC [$K=2$, $r=4$]<br>Conv1x1 | 192/232 | 4/6 |
| 8 | 20×20 | Conv1x1<br>AC [$K=2$, $r=4$]<br>Conv1x1 | 320/384 | 1/2 |

**Firstly**, in the top-down path, we use the same approach as FPN [23]:

$$M_i = Conv(Up_{X2}(P_{i+1}) + M_{i+1}), \tag{7}$$

Where $M_i$ denotes the *i-th* level of fusion feature map in the top-down path, the value of $i$ is {2, 3, 4} and $M_5$ is simply $P_5$, $Up_{X2}$ refers to 2 times up-sampling. To further improve the efficiency, in *Conv*, we use depthwise separable convolution [4] instead of common 3 × 3 convolution. The structure as shown in Fig. 3a.

**Secondly**, we build a reverse path to return the low-level features to the high level:

$$N_i = Conv(Down_{X2}(P_{i-1}) + N_{i-1}), \tag{8}$$

Where $N_i$ denotes the *i-th* level of fusion feature map in the bottom-up path, the value of $i$ is {3, 4, 5} and $N_2$ is simply $P_2$, both $Down_{X2}$ and *Conv* are depthwise separable convolution [4] with factor 2 and 1 respectively. The architecture as shown in Fig. 3b.

**Thirdly**, after the above operations, we obtained two sets of feature maps, which are respectively called $\{M_2, M_3, M_4, M_5\}$ and $\{N_2, N_3, N_4, N_5\}$. We introduced the original feature maps $\{P_2, P_3, P_4, P_5\}$ at same level, then, $P_i$ is merged with the corresponding $M_i$ and $N_i$ by element-wise addition:

$$V_i = \begin{cases} M_i + P_i, & i = 2 \\ M_i + N_i + P_i, & 4 \geq i \geq 3 \\ N_i + P_i, & i = 5 \end{cases}, \tag{9}$$

Where $V_i$ represents the *i-th* level of final output. The operation as shown in Fig. 3c. With these optimizations, we name the efficient feature fusion network as path enhancement feature pyramid network (PEFPN).

### 3.5 Loss function

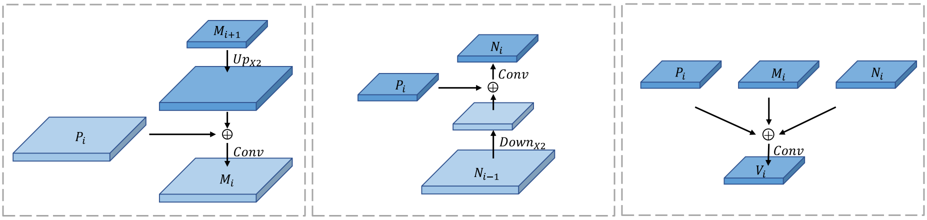
It is common that binary cross entropy [5] is used to optimize the network's weight. Nonetheless, the text instances usually occupy only an extremely small region in natural images, which leads to the prediction of detector bias to the regions which are non-text region. Thus, in order to obtain a better model during the learning phase, we utilize dice coefficient [34] in training stage, and it can be formulated as follow:

$$L(D_i, G_i) = \frac{2\sum_{x,y}\left(D_{i,x,y} \times G_{i,x,y}\right)}{\sum_{x,y} D_{i,x,y}^2 + \sum_{x,y} G_{i,x,y}^2}, \tag{10}$$

where $D_{i,\,x,\,y}$ denotes the value of pixel $(x, y)$ in detection result $D_i$, and $G_{i,\,x,\,y}$ indicates ground truth $G_i$.

In addition, to distinguish patterns, such as fences, lattices, which are similar to text strokes. We use Online Hard Example Mining (OHEM) [40] to improve the discernment of detector. Let us consider the training mask given by OHEM as $O$, and the final loss can be formulated as follow:

$$L^{'} = 1 - L(D_i \cdot O, G_i \cdot O), \tag{11}$$



(a) Top-down (b) Bottom-up (c) Final feature fusion

**Fig. 3** The illustration of Path Enhancement feature pyramid network. **a** Top-down path. **b** Bottom-up path. **c** Final feature fusion, where $4 \geq i \geq 3$

# 4 Experiment

## 4.1 Datasets

**CTW 1500** [24] is a popular challenging dataset for arbitrarily curve text detection. It includes 1000 training images and 500 testing images. Different from conventional text datasets (e.g. ICDAR 2017 MLT, ICDAR 2015), In order to describe the shape of an arbitrarily curve text that 14 points are used to label the text instances in CTW1500.

**Total-Text** [1] is also a newly released benchmark for long curve text detection. It consists of horizontal, multi-Oriented and curve text instances. The benchmark is divided into training set and testing set with 1255 and 300 images, respectively.

**ICDAR 2015** [16] is a multi-oriented benchmark for text detection. Scene text images in this dataset are taken by Google Glasses without taking care of positioning, image quality and viewpoint. It contains a total of 1500 images, 1000 images for training and another 500 images for testing. The text regions are labelled by 4 vertices of the quadrangle.

**MSRA-TD500** [49] is a commonly used dataset for text detection. It includes 300 training pictures and 200 test pictures with text line level annotations. It is a dataset with multi-oriented, multi-lingual and long text lines. We follow these works [27, 32] to tr model on HUST-TR400 [50], which include 400 images.

## 4.2 Evaluation metrices

In order to evaluate the performance of our detector, we use the Precision (P) and Recall (R) that have been utilized in information retrieval field. Meanwhile, we utilized the F-measure (F) which can be obtained as follows:

$$F = 2 \times \frac{P \times R}{P + R}, \tag{12}$$

where calculating the precision and recall are based on using the ICDAR 2015 intersection over union (IOU) metric [16], which is obtained for the *j-th* ground-truth and *i-th* detection bounding box as follow:

**Table 2** Results of backbone with different combinations of multiple kernels

| Kernel | | | $\widehat{X}$ | CTW 1500 | Total-Text |
|---|---|---|---|---|---|
| 3 × 3 | 5 × 5 | 7 × 7 | | F | F |
| √ | | | | 78.0 | 77.8 |
| | √ | | | 76.2 | 75.9 |
| | | √ | | 75.8 | 75.3 |
| √ | √ | | √ | 78.7 | 78.3 |
| √ | √ | | | 79.5 | 79.7 |
| √ | | √ | √ | 78.5 | 78.2 |
| √ | | √ | | 79.3 | 79.6 |
| | √ | √ | √ | 77.1 | 76.8 |
| | √ | √ | | 77.6 | 77.2 |
| √ | √ | √ | √ | 78.8 | 78.5 |
| √ | √ | √ | | 79.5 | 79.8 |

“F” means F-measure

**Table 3** Result of with or without PEFPN, “F” means F-measure

| Backbone | FPN | PEFPN | CTW 1500<br>F | Total-Text<br>F |
|---|---|---|---|---|
| ResNet50 | √ | | 76.5 | 75.7 |
| ResNet50 | | √ | 77.5 | 76.6 |
| ACNet-B0 | √ | | 78.7 | 78.3 |
| ACNet-B0 | | √ | 79.5 | 79.7 |

$$\text{IoU} = \frac{Area\left(G_j \cap D_i\right)}{Area\left(G_j \cup D_i\right)}, \tag{13}$$

Where $G_j$ and $D_i$ same as Eqs. (10) and (11). Meanwhile, a threshold of IoU > t is used for counting a correct detection.

### 4.3 Implementation details

We train our model from scratch with batch size 8 on 2 GPUs for 72 K iterations, and the initial learning rate is set to $1 \times 10^{-3}$ and is divided by 10 at 24 K and 48 K iterations. All the networks are optimized by using stochastic gradient descent (SGD). Note that no extra data set is used during training. We use a weight decay of $5 \times 10^{-4}$ and a Nesterov momentum [41] of 0.99 without dampening. We adopt the weight initialization introduced by [8].

During training, three data augmentation strategies are adopted in the all datasets: 1) rescaling images with the ratio {0.5, 1.0, 2.0, 3.0} randomly; 2) flipping the images in the range [$-10^\circ$, $10^\circ$] randomly; 3) All the images are re-sized to 640 × 640 for better efficiency. For quadrangular text, we calculate the minimal area rectangle to extract the bounding boxes. For curve text dataset, we use the result of progressive scale expansion algorithm [45] as the final output.

During the inference, for all test images, we set a suitable width and then re-scale the height through the aspect ratio. We use a batch size of 1 and a single 1080ti GPU to evaluate the inference speed (i.e. FPS) in a single thread. When calculating the inference speed, the model forward time cost and the post-processing time cost are included.

### 4.4 Ablation study

To investigate the effectiveness of our proposed module, we conduct an ablation study on the ICDAR 2015 and the CTW 1500, which is a quadrangle text dataset and a curve text dataset respectively. Note that, all experiments without any external dataset.

**Table 4** Result of whether the model uses the original feature map, $P_i$ denotes the original feature map at the $i$-th level, “F” means F-measure

| $P_i$ | CTW 1500<br>F | Total-Text<br>F |
|---|---|---|
| | 79.1 | 79.3 |
| √ | 79.5 | 79.7 |

**Table 5** Result of model with different backbone, "F" means F-measure

| Backbone | CTW 1500<br>F | Total-Text<br>F |
|---|---|---|
| ACNet-B0 | 79.5 | 79.7 |
| ACNet-B3 | 80.8 | 80.9 |

**Different kernels and different branches** In Section 3.1, we only take two size kernels as examples, therefor, in order to explore the effect of combination of different kernels and number of different branches, in Table 2, we use three different kernels, 3 × 3 denote the 3 × 3 depth-wise convolution, 5 × 5 denote the 3 × 3 depth-wise convolution with dilation 2, and 7 × 7 denote the 3 × 3 depth-wise convolution with dilation 3. We only use ACNet-B0 as backbone. If $\widehat{X}$ is ticked, it denotes that we use the attention mechanism which is the output of Adaptive Convolution unit, otherwise we only add up the feature maps without SoftMax attention. Considering the efficiency of the model, we did not use a convolution kernel with larger receptive field.

As shown in Table 2, we have the following conclusions: 1) When the number of branches N increases, in general the F-measure increases. 2) When using the attention mechanism, the performance is better than simple addition. 3) In the case of using the attention mechanism, the performance gain of the model from $N = 2$ to $N = 3$ is slight. For better efficiency, $N = 2$ is used.

**The effectiveness of PEFPN** We design a set of comparative experiments to verify the effectiveness of PEFPN. Considering the fairness of the comparisons, we employ our own designed network and ResNet-50 [9] as the backbone, then, we add the original FPN [23] and PEFPN after these two backbones, respectively. From Table 3, we can see that the F-measure can make improvement about 0.8%, 1.4% and 1.0%, 0.9% when using ACNet-B0 and ResNet-50 [9], respectively. It indicates that no matter which backbone is used, the performance of PEFPN is better than the original FPN [23].

**The influence of the original feature maps** In the PEFPN, we add the original feature map at the same level to the fused feature map as the final output (i.e.$P_i$ in Eq. 9). To verify the

**Table 6** The single-scale results on CTW 1500

| Method | Ext | Venue | CTW 1500 | | | |
|---|---|---|---|---|---|---|
| | | | P | R | F | FPS |
| CTPN [43] | – | ECCV'16 | 60.4* | 53.8* | 56.9* | 7.14 |
| SegLink [39] | – | CVPR'17 | 42.3* | 40.0* | 40.8* | 10.7 |
| EAST [54] | – | CVPR'17 | 78.7* | 49.1* | 60.4* | 21.2 |
| CTD + TLOC [24] | – | ICDAR'18 | 77.4 | 69.8 | 73.4 | 13.3 |
| TextSnake [27] | √ | ECCV'18 | 67.9 | 85.3 | 75.6 | – |
| PSENet [45] | – | CVPR'19 | 80.6 | 75.6 | 78.0 | 3.9 |
| Ours (ACNet-B0) | – | – | 82.3 | 76.8 | 79.5 | 5.2 |
| Ours (ACNet-B3) | – | – | 84.5 | 77.4 | 80.8 | 4.1 |

"P"," R" and" F" represent the precision, recall and F-measure respectively. "Ext" indicates external data. * indicates the results from [24]. ACNet-B0 and ACNet-B3 indicates the backbone network used

**Table 7** The single-scale results on Total-Text

| Method | Ext | Venue | Total-Text | | | |
|---|---|---|---|---|---|---|
| | | | P | R | F | FPS |
| SegLink [39] | – | CVPR'17 | 30.3* | 23.8* | 26.7* | – |
| EAST [54] | – | CVPR'17 | 50.0* | 36.2* | 42.0* | – |
| DeconvNet [1] | – | ICDAR'18 | 33.0 | 40.0 | 36.0 | – |
| TextSnake [27] | √ | ECCV'18 | 82.7 | 74.5 | 78.4 | – |
| PSENet [45] | – | CVPR'19 | 81.8 | 75.1 | 78.3 | 3.9 |
| Ours (ACNet-B0) | – | – | 83.2 | 76.4 | 79.7 | 5.2 |
| Ours (ACNet-B3) | – | – | 85.0 | 77.2 | 80.9 | 4.1 |

"P"," R" and" F" represent the precision, recall and F-measure respectively. "Ext" indicates external data. * indicates the results from [27]. ACNet-B0 and ACNet-B3 indicates the backbone network used

influence of the original feature map on the detection results, we remove the original feature map when fusing feature. We can find from Table 4 that without the original feature map, the F-measure is decrease. Thence, in the final fusion stage, it is necessary to add the original feature map, meanwhile, this operation will not bring too much extra computation cost.

**The influence of the backbone** To better analyze the capability of our model, we adopt ACNet-B0 and ACNet-B3 as backbone, respectively. As shown in Table 5, keep the same setting, it can obviously improve the performance by replace the deeper backbone. ACNet-B0 for faster inference and ACNet-B3 for better accuracy.

### 4.5 Comparisons with state-of-the-art methods

For efficiency, we just training on a single dataset and do not use the pre-training strategy which training on extra datasets adopted by PSENet [45]. It is worth mentioning that only comparing the detection results of training on a single dataset, our method has surpassed PSENet [45].

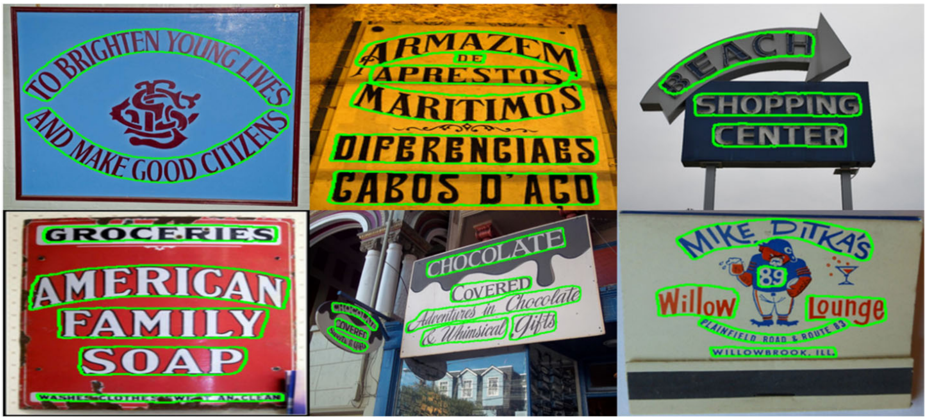


**Fig. 4** Some visualization results on CTW 1500

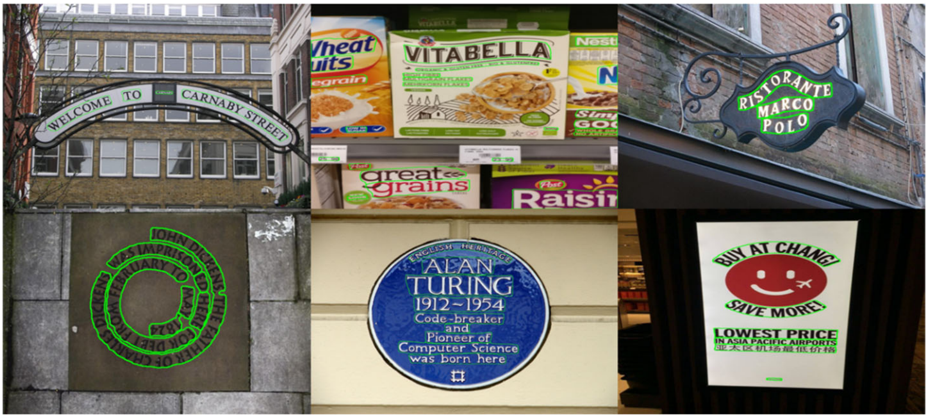


**Fig. 5** Some visualization results on Total-Text

**Curve text datasets detection** To evaluate the ability of curve text detection, we test our model on CTW 1500 and Total-Text, which mainly include curve texts, during inference stage, we re-scale the longer side of images to 1280 and evaluate the results using the same evaluation method with [45]. We report the single-scale performance of our model on CTW 1500 and Total-Text in Tables 6 and 7, respectively.

On CTW 1500, without external data pre-training, our model with ACNet-B0 achieves the F-measure of 79.5% and with ACNet-B3 achieves the F-measure of 80.8%. Especially, when using ACNet-B3, the performance surpasses most of the counterparts, Notably, we can find that the F-measure of our model with ACNet-B3 is 2.8% higher than PSENet [45], which was published on CVPR 2019.

On Total-Text, similar conclusions can be obtained. Without external data pre-training, our model with ACNet-B0 not only surpasses the PSENet [45] in F-measure (79.7%) but also leads in speed (5.2 FPS). Another model with ACNet-B3 outperforms the previous state-of-the-art method by 80.9%.

**Table 8** The single-scale results on ICDAR 2015

| Method | Ext | Venue | ICDAR 2015 | | | |
|---|---|---|---|---|---|---|
| | | | P | R | F | FPS |
| CTPN [43] | – | ECCV'16 | 74.2 | 51.6 | 60.9 | 7.1 |
| EAST [54] | – | CVPR'17 | 83.6 | 73.5 | 78.2 | 13.2 |
| SegLink [39] | √ | CVPR'17 | 73.1 | 76.8 | 75.0 | – |
| DeepReg [11] | – | ICCV'17 | 82.0 | 80.0 | 81.0 | – |
| SSTD [36] | √ | ICCV'17 | 80.2 | 73.9 | 76.9 | 7.7 |
| WordSup [12] | √ | CVPR'17 | 79.3 | 77.0 | 78.2 | – |
| RRPN [33] | – | TMM'18 | 82.0 | 73.0 | 77.0 | – |
| PixeLink [6] | – | AAAI'18 | 82.9 | 81.7 | 82.3 | 7.3 |
| TextSnake [27] | √ | ECCV'18 | 84.9 | 80.4 | 82.6 | 1.1 |
| PSENet [45] | – | CVPR'19 | 81.5 | 79.7 | 80.6 | 1.6 |
| Ours (ACNet-B0) | – | – | 82.4 | 80.6 | 81.5 | 2.8 |
| Ours (ACNet-B3) | – | – | 83.7 | 81.9 | 82.8 | 1.9 |

"P"," R" and" F" represent the precision, recall and F-measure respectively. "Ext" indicates external data. ACNet-B0 and ACNet-B3 indicates the backbone network used

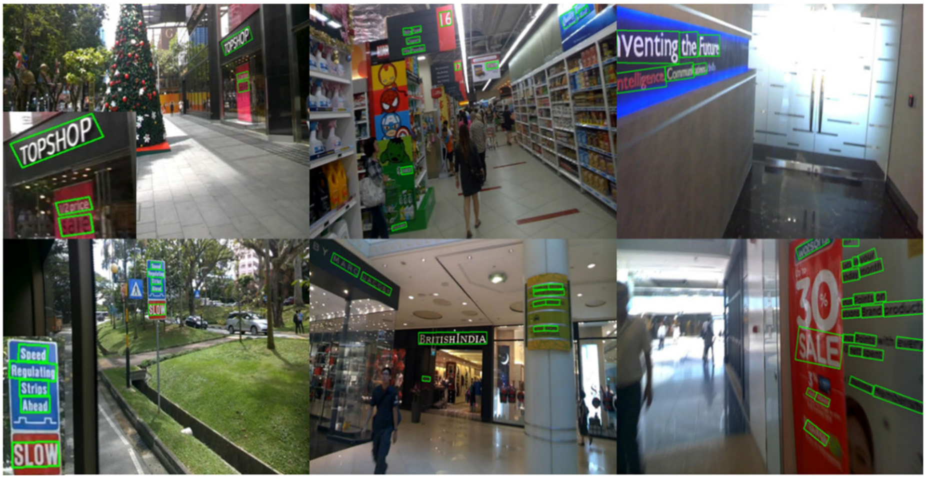


**Fig. 6** Some visualization results on ICDAR 2015

The performance on CTW 1500 and Total-Text demonstrates the solid superiority of our method to detect arbitrary-shaped text instances. We illustrate several detection results in Figs. 4 and 5. It is clearly demonstrated that our method can elegantly distinguish complex curve text instances.

**Oriented text datasets detection** We evaluate our method on the ICDAR 2015 to test its ability for oriented text detection. Same as previous experiments, we adopt the ACNet-B0 and ACNet-B3 as the backbone of our model. In the test stage, we scale the long side of images to 2240. Table 8 show the results that compares with other state-of-the-art methods. Our model with ACNet-B0 achieves the F-measure of 81.5% at 2.8 FPS, both F-measure and speed surpass the PSENet. When using ACNet-B3, although the speed is not as well as some methods, but our model has a significant improvement in F-measure over 2.2%. Moreover, we demonstrate some test illustrations in Fig. 6, our method can accurately detect the text instances with arbitrary orientations.

**Table 9** The single-scale results on MSRA-TD500

| Method | Ext | Venue | MSRA-TD500 | | | |
|---|---|---|---|---|---|---|
| | | | P | R | F | FPS |
| SegLink [39] | √ | CVPR’17 | 86.0 | 70.0 | 77.0 | 8.9 |
| EAST [54] | – | CVPR’17 | 87.3 | 67.4 | 76.1 | 13.2 |
| TextSnake [27] | √ | ECCV’18 | 82.7 | 74.5 | 78.4 | – |
| PixeLink [6] | √ | AAAI’18 | 83.0 | 73.2 | 77.8 | 3.0 |
| RRD [22] | √ | CVPR’18 | 87.0 | 73.0 | 79.0 | 10 |
| Ours (ACNet-B0) | √ | – | 86.6 | 77.1 | 81.6 | 2.8 |
| Ours (ACNet-B3) | √ | – | 87.5 | 78.3 | 82.6 | 1.9 |

“P”,” R” and” F” represent the precision, recall and F-measure respectively. “Ext” indicates external data. ACNet-B0 and ACNet-B3 indicates the backbone network used

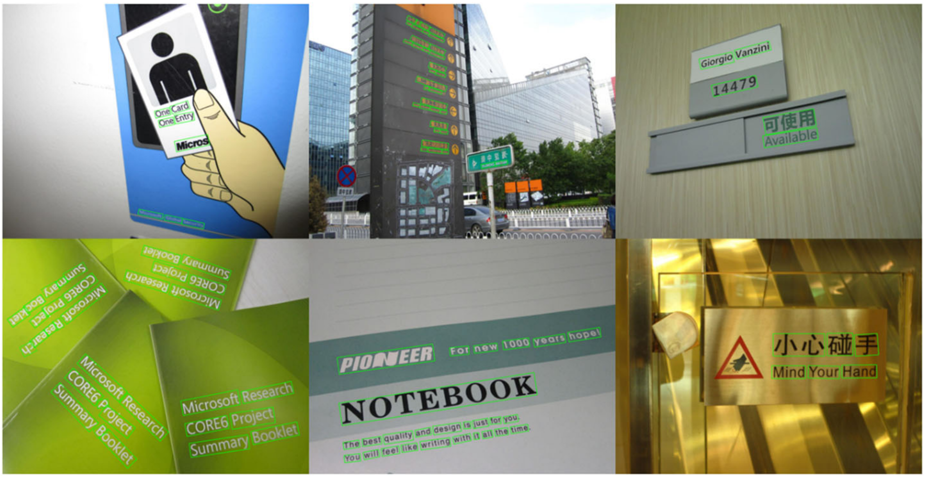
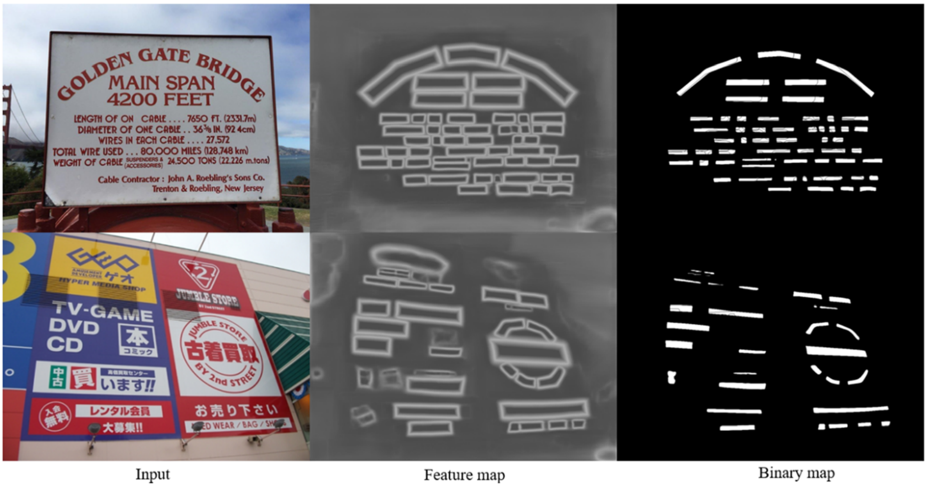


Fig. 7 Some visualization results on MSRA-TD500

**MultiLingual text datasets detection** To prove the robustness of our method to multiple languages and long straight, we test our model on MSRA-TD500 dataset. Considering the fairness of the comparisons, we also resize the longer edge of the test images to 2240 as ICDAR 2015. As shown in Table 9, our model achieves the F-measure of 81.6% and 82.6% with different backbone. Compare with other method, our model has a slight improvement indeed. Therefor, this proves that our method is robust for multiple languages and long straight text detection and can indeed be deployed in complex natural scenarios. We also shown some results in Fig. 7.



Input Feature map Binary map

Fig. 8 visualization of the feature maps and binary maps from the network

### 4.6 Effectiveness of ACPEPNet

To demonstrate the effectiveness of our proposed method, Fig. 8 provides a visualization of the feature maps from the network. In the encoding stage, due to the application of the Adaptive Convolution Unit, the ability of aggregate multi-scale spatial information is enhancement, and the response of the text areas are more sensitive. In the decoding stage, PEFPN allows texts with different scales to be effectively retained, which indicates that the text features have not lost in the bottom-up path.

## 5 Conclusion

In this paper, we focus on two design principles in text detection tasks: 1) larger receptive field 2) finer low-level features and proposed an efficiency detector for arbitrary shapes text detection. Firstly, we design a set of feature extraction networks with EfficientNet as the baseline, named ACNet-B0 and ACNet-B3. These backbones are embedded with the Adaptive Convolution Unit, which enables the network to adaptively adjust the receptive field, enhancing the ability of aggregate multi-scale spatial information while bringing minor extra computation. Then, In order for FPN [23] to obtain more finer low-level features during the feature fusion stage, we redesign the original FPN [23] information flow path, changing it from one-way flow to two-way flow, and adding original features to the final stage of information fusion, this design to shorten the distance of the low-level feature to the top-level while adding more original features. The experiment on scene text detection datasets demonstrate the superior performance of the previous methods. In the future, we will continue to explore the optimization problem of text detection from real-time and learnable post-processing.

**Acknowledgements** This work is supported by the Natural Science Foundation of Shandong Province (ZR2019MF050), the Shandong Province colleges and universities youth innovation technology plan innovation team project under Grant (No.2020KJN011).

## References

1. Ch’ng CK, Chan CS (2017) Total-text: a comprehensive dataset for scene text detection and recognition. In: Proc. ICDAR, pp 935–942
2. Chen X, Girshick R, He K, Dollár P (2019) TensorMask: a foundation for dense object segmentation. In: Proc. ICCV, pp 2061–2069
3. Chen K, Pang J, Wang J, Yu X, Li X, Sun S (2019) Hybrid task cascade for instance segmentation. In: Proc. CVPR, pp 4974–4983
4. Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proc. CVPR, pp 1251–1258
5. De Boer P-T, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on cross-entropy method. Ann Oper Res 134:19–67
6. Deng D, Liu H, Li X, Deng C (2018) Pixellink: detecting scene text via instance segmentation. In: Proc. AAAI, pp 6773–6780
7. Gao H, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proc. CVPR, pp 4700–4708
8. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proc. ICCV, pp 1026–1034
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proc. CVPR, pp 770–778

10. He K, Zhang X, Ren S, Sun J (2016) Identity mappings in deep residual networks. In: Proc. ECCV, pp 630–645
11. He W, Zhang X-Y, Yin F, Liu C-L (2017) Deep direct regression for multi-oriented scene text detection. In: Proc. ICCV, pp 745–753
12. Hu H, Zhang C, Luo Y, Wang Y, Han J, Ding E (2017) Wordsup: exploiting word annotations for character based text detection. In: Proc. ICCV, pp 4940–4949
13. Hu S, Wang G, Wang Y, Chen C, Pan Z (2020) Accurate image super-resolution using dense connections and dimension reduction network. Multimedia Tools and Application 79:1427–1443
14. Huang Z, Huang L, Gong Y, Huang C, Wang X (2019) Mask scoring r-cnn. In: Proc. CVPR, pp 6409–6418
15. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: PMLR, vol 37, pp 448–456
16. Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A, Iwamura M, Matas J, Neumann L, Chandrasekhar VR, Lu S, Shafait F, Uchida S, Valveny E (2015) ICDAR 2015 competition on robust reading. In: Proc. ICDAR, pp 1156–1160
17. Li X, Wang W, Hu X, Yang J (2019) Selective kernel networks. In: Proc. CVPR, pp 510–519
18. Liao M, Shi B, Bai X, Wang X, Liu W (2017) Textboxes: A fast text detector with a single deep neural network. In: Proc AAAI, pp 4161–4167
19. Liao X, Zheng Q, Ding L (2017) Data embedding in digital images using critical function. Signal Process Image Commun 58:146–156
20. Liao X, Li K, Yin J (2017) Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform. Multimedia Tools and Application 76:20739–20753
21. Liao M, Shi B, Bai X (2018) Textboxes++: A single-shot oriented scene text detector. IEEE Trans Image Process 27(8):3676–3690
22. Liao M, Zhu Z, Shi B, Xia G-s, Bai X (2018) Rotation-sensitive regression for oriented scene text detection. In: Proc. CVPR, pp 5909–5918
23. Lin T-Y, Doll'ar P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proc. CVPR, pp 2117–2125
24. Liu Y, Jin L, Zhang S, Zhang S (2017) Detecting curve text in the wild: New dataset and new solution. arXiv preprint arXiv:1712.02170
25. Liu S, Lu Q, Qin H, Shi J, Jia J (2018) Path aggregation network for instance segmentation. In: Proc. CVPR, pp 8759–8768
26. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proc. CVPR, pp 3431–3440
27. Long S, Ruan J, Zhang W, He X, Wu W, Yao C (2018) Textsnake: a flexible representation for detecting text of arbitrary shapes. In: Proc. ECCV, pp 20–36
28. Lu XK, Ma C, Ni B, Yang X, Reid I, Yang M-H (2018) Deep regression tracking with shrinkage loss. In: Proc. ECCV, pp 353–369
29. Lu X, Ma C, Ni B, Yang X (2019) Adaptive region proposal with channel regularization for robust object tracking. IEEE Transactions on Circuits and Systems for Video Technology
30. Lu X, Wang W, Ma C, Shen J, Shao L, Porikli F (2019) See more, know more: unsupervised video object segmentation with co-attention siamese networks. In: Proc. CVPR, pp 3623–3632
31. Lu XK, Wang W, Shen J, Tai Y-W, Crandall D, Hoi SCH (2020) Learning video object segmentation from unlabeled videos. In: Proc. CVPR, pp 8960–8970
32. Lyu P, Yao C, Wu W, Yan S, Bai X (2018) Multi-oriented scene text detection via corner localization and region segmentation. In: Proc. CVPR, pp 7553–7563
33. Ma J, Shao W, Ye H, Wang L, Wang H, Zheng Y, Xue X (2018) Arbitrary-oriented scene text detection via rotation proposals. IEEE Trans Multimedia 20(11):3111–3122
34. Milletari F, Navab N, Ahmadi S-A (2016) V-net: fully convolutional neural networks for volumetric medical image segmentation. In: 3D vision, pp 565–571
35. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proc. ICML, vol 807-814
36. Pan H, Huang W, He T, Zhu Q, Yu Q, Li X (2017) Single shot text detector with regional attention. In: Proc. ICCV, pp 3047–3055
37. Rezatofighi H, Tsoi M, Gwak JY, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: a metric and a loss for bounding box regression. In: Proc. CVPR, pp 658–666
38. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proc.CVPR, pp 4510–4520
39. Shi B, Bai X, Belongie S (2017) Detecting oriented text in natural images by linking segments. In: Proc. CVPR, pp 2550–2558
40. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: Proc. CVPR, pp 761–769

41. Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. In: Proc. ICML, vol 28, pp 1139–1147
42. Tan M, Le QV (2019) Efficientnet: rethinking model scaling for convolutional neural networks. In: PMLR, vol 97, pp 6105–6114
43. Tian Z, Huang W, He T, Pan H, Yu Q (2016) Detecting text in natural image with connectionist text proposal network. In: Proc. ECCV, pp 56–72
44. Tian M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, Quoc VL (2019) MnasNet: platform-aware neural architecture search for mobile. In: Proc. CVPR, pp 2820–2828
45. Wang W, Xie E, Li X, Hou W, Lu T, Yu G, Shao S (2019) Shape robust text detection with progressive scale expansion network. In: Proc. CVPR, pp 9336–9345
46. Wang Y, Wang G, Chen C, Pan Z (2019) Multi-scale dilated convolution of convolutional neural network for image denoising. Multimedia Tools and Application 78:19945–19960
47. Wang Y, Hu S, Wang G, Chen C, Pan Z (2020) Multi-scale dilated convolution of convolutional neural network for crowd counting. Multimedia Tools and Application 79:1057–1073
48. Xie E, Zang Y, Shao S, Yu G, Yao C, Li G (2019) Scene text detection with supervised pyramid context network. In: Proc. AAAI, pp 9038–9045
49. Yao C, Bai X, Liu W, Ma Y, Zhuowen T (2012) Detecting texts of arbitrary orientations in natural images. In: Proc. CVPR, pp 1083–1090
50. Yao C, Bai X, Liu W (2014) A unified framework for multioriented text detection and recognition. IEEE Trans Image Process 23(11):4737–4749
51. Yao C, Bai X, Sang N, Zhou X, Zhou S, Cao Z (2016) Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002
52. Zheng Z, Zhang C, Shen W, Yao C, Liu W, Bai X (2016) Multi-oriented text detection with fully convolutional networks. In: Proc. CVPR, pp 4159–4167
53. Zheng Q, Li Z, Zhang Z, Bao Y, Yu G, Peng Y, Sun J (2019) ThunderNet: towards real-time generic object detection. In: Proc. ICCV, pp 6718–6727
54. Zhou X, Yao C, He W, Wang Y, Zhou S, He W, Liang J (2017) East: an efficient and accurate scene text detector. In: Proc. CVPR, pp 5551–5560

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Qi Cheng[1] • Guodong Wang[1] • Qian Dong[2] • Bin Wei[3]**

[1] College of Computer Science and Technology, Qingdao University, Qingdao, China

[2] Department of Pediatric Surgery, The Affiliated Hospital of Qingdao University, Qingdao, China

[3] Shandong Key Laboratory of Digital Medicine and Computer Assisted Surgery, The Affiliated Hospital of Qingdao University, Qingdao, China