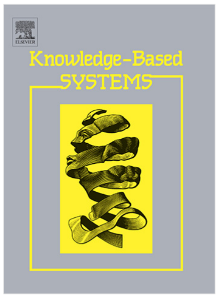
Journal Pre-proof



PRPN: Progressive region prediction network for natural scene text detection

Yuanhong Zhong, Xinyu Cheng, Tao Chen, Jing Zhang, Zhaokun Zhou, Guan Huang

PII: S0950-7051(21)00988-6
DOI: https://doi.org/10.1016/j.knosys.2021.107767
Reference: KNOSYS 107767

To appear in: *Knowledge-Based Systems*

Received date : 4 January 2021
Revised date : 12 November 2021
Accepted date : 14 November 2021

Please cite this article as: Y. Zhong, X. Cheng, T. Chen et al., PRPN: Progressive region prediction network for natural scene text detection, *Knowledge-Based Systems* (2021), doi: https://doi.org/10.1016/j.knosys.2021.107767.

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Published by Elsevier B.V.

Revised Manuscript (Clean Version) Journal Pre-proof Click here to view linked References

PRPN: Progressive Region Prediction Network for Natural Scene Text Detection

Yuanhong Zhong[a,*], Xinyu Cheng[a], Tao Chen[b], Jing Zhang[a], Zhaokun Zhou[a] and Guan Huang[a]

[a] *School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China*
[b] *State Grid Chongqing Electric Power Research Institute, Chongqing 401123, China*

ARTICLE INFO

Keywords:
natural scene text detection
progressive region
pixel prediction

ABSTRACT

With the development of deep learning, scene text detection methods have made great progress in recent years. Most text detection methods are based on bounding box prediction with a 0-1 discrete distribution; thus, separating adjacent text instances is difficult. Direct prediction of the bounding box also renders difficult the detection various shapes of text, such as quadrangular text and curved text. In this work, we design a 2D progressive kernel for describing the progressive variety of text regions. It transforms the original ground truth (GT) of bounding boxes into the GT of a 0-1 progressive probability distribution. We also propose a novel progressive region prediction network (PRPN) with directional pooling for predicting the probability distributions of text regions. Then, a postprocessing algorithm is used to transform the probability distributions of the text regions into bounding box output for text detection. Experiments on standard datasets, including ICDAR 2013, ICDAR 2015, MSRA-TD500, and SCUT-CTW1500, demonstrate that the proposed method outperforms state-of-the-art methods in terms of accuracy and robustness. The method obtains an F-measure of 86.0% on ICDAR 2015 and 81.4% on SCUT-CTW1500. The code is available at https://github.com/xinyu-ch/ProgressiveTextDetection.

1. Introduction

Text reading in natural scenes has attracted widespread attention due to its many applications, such as in image retrieval [1], autonomous driving [2], visual question answering [3], and blind navigation [4]. The pipeline of scene text reading is composed of text detection and text recognition. Text detection is generally the first step of scene text reading, which uses bounding boxes to locate words or text lines, and it is of great significance for text recognition. In this work, we focus on detecting natural scene text accurately and effectively.

In the early development stage of deep learning, many scene text detection methods [5, 6, 7, 8, 9, 10, 11, 12, 13, 14] represent each text instance using an axis-aligned bounding box that tightly surrounds the text instance. Numerous potential text bounding boxes are fed into an image classifier. The image classifier determines whether each bounding box contains a text instance. These methods can be categorized into two-stage methods and one-stage methods. Two-stage text detectors [5, 6, 11] set a series of anchors on the original image. In the first stage, a region proposal network (RPN) is used to classify and regress these anchors; then, many candidate regions are obtained. After extracting the characteristics of each candidate region, a classifier network is applied to classify and regress the candidate regions in the second stage. Finally, nonmaximal suppression (NMS) is used to obtain accurate detection results. Although these two-stage detectors have high accuracy, they have high computational complexity. One-stage text detectors [7, 8, 9, 10] set a series of anchors on the original image. They use a fully convolutional network (FCN) to classify and regress these anchors once and use NMS to obtain the text detection results. In general, the two-stage algorithms are more accurate, while the one-stage algorithms are faster. Although these early deep learning methods have made progress in terms of accuracy, they still directly predict the bounding boxes. This causes them to focus only on the information of boxes and to not combine the text features with network prediction, which indirectly reduces the robustness and generalization ability of the model.

Recently, many scene text detection methods that are based on image segmentation and divested from the setting of bounding boxes have been proposed [15, 16, 17, 18, 19, 20, 21]. Inspired by SegLink [13], Deng et al. [16] link all pixels within the same instance together to realize text segmentation. The segmentation result is applied to generate text bounding boxes without location regression. By text/nontext prediction and link prediction, this approach effectively converts the task of bounding box detection to the task of pixel prediction. It does not require extra location regression and can effectively segment adjacent text instances. Baek et al. [15] exploit a Gaussian kernel to obtain a mask of each character and a mask of affinity between characters; then, a network is trained for mask prediction. Since many available text datasets do not have character-level GT, they use weakly supervised learning to obtain character-level annotations for network training. This approach could overcome the problem of missing data, but it is extremely difficult for researchers to apply weakly supervised methods and improve their performance. These segmentation-based methods use the strategy of image segmentation to avoid the setting of anchors and promote the development of scene text detection. However, they do not take full advantage of the variability of text regions, which leads to their robustness not being signifi-

*Corresponding author
Email addresses: zhongyh@cqu.edu.cn (Y. Zhong)
ORCID(s): 0000-0001-5689-1146 (Y. Zhong)

Journal Pre-proof

(a)

(b)

(c)

Figure 1: Visualization of 2D progressive text masking. The image in the first column is from the ICDAR 2015 dataset, and the image in the second column is from the SCUT-CTW1500 dataset. (a) The original image with the GT bounding box. (b)The 2D progressive masks of text instances. (c) The weighted loss. In (b), from the edge of each text instance to its center, the value of the pixels progressively increases from 0 to 1, and the color gradually changes from blue to red. In (c), brighter pixels correspond to larger values.

cantly improved.

The purpose of this work is to find an effective strategy for simulating the variability of text regions for robust detection. In [15], the authors exploit a 2D Gaussian kernel to describe the variability of the character region. The 2D Gaussian kernel can effectively describe a circular region with a progressive probability of 0 to 1. Since the aspect ratio of characters is basically consistent and similar among datasets in this scenario, a 2D Gaussian kernel can describe the variability of character regions. However, for quadrangular or curved datasets, the aspect ratio of texts differs substantially from that of characters. It is difficult for a 2D Gaussian kernel to express the variability of text regions. Therefore, we design a novel 2D progressive kernel for expressing the variability. Figure 1 (b) is a visualization of 2D progressive text masking. If the pixel of the mask is closer to the center of the text, the value of the pixel will be close to 1, and the color will be brighter. First, we calculate the distance from the foreground to the background of text regions, which is used to describe the 0-1 progressive variety of the text regions. Then, we use a 1D Gaussian function to map this distance to a 2D mask, which has the Gaussian generalization characteristic. We name the process 2D progressive kernel. This shows that the 2D progressive mask can be quadrangular and curved to adapt to complex natural environments. Using a 2D progressive kernel, we transform the original bounding box detection task into a pixel value prediction task. The corresponding postprocessing algorithm will transform pixel predictions into bounding boxes without NMS.

In this paper, we also design a progressive region prediction network for predicting the 2D progressive mask of text. Based on the original U-Net [22], we add extra convolution layers for feature extraction and propose a novel directional pooling module for enlarging the receptive field. The directional pooling module is based on the study of Law et al. [23]. They proposed corner pooling for obtaining an extra feature from objects outside the bounding box. Similarly, the progressive mask in Figure 1 (b) shows that the masks of text regions exhibit obvious divergence. Hence, we improve the structure of corner pooling to gather features outside the boundaries of texts in both the horizontal and vertical directions to realize better regression prediction. The contributions of this paper are listed as follows:

- We propose a 2D progressive kernel for generating the progressive masks of text instances. 2D progressive kernel can effectively describe the progressive variety of text regions. In contrast to the 2D Gaussian kernel, the proposed 2D progressive kernel can satisfy the requirements of various quadrangular text and curved text detection tasks in natural scenes.
- A unique network with a directional pooling module is proposed for predicting the progressive masks of text instances. Since text instances have strong divergence and to better suit the task of prediction, pooling in different directions can be used to obtain more text information and can produce more accurate results.
- A new weighted loss is designed for network training. The text scale in natural scenes varies substantially, and there are many adjacent instances, which makes such text instances slightly difficult to handle. The weighted loss treats the losses of samples of different sizes equally, which enables the network to robustly detect those texts. Moreover, it treats hard negative samples and simple negative samples differently. This can effectively guide the network to focus more attention on hard samples.
- Experiments on ICDAR 2013, ICDAR 2015, MSRA-TD500 and SCUT-CTW1500 show the effectiveness of the method. In particular, PRPN shows a strong performance and competitive recall rate for quadrangular texts and curved texts.

2. Related Work

Text detection with traditional features. Text detection has been a hot research topic in recent years. Early methods rely mainly on sliding windows or connected components. Sliding window-based methods [24, 25, 26] slide windows of various sizes on the image and use a classification model to determine whether each window contains text. Wei et al. [26] adjusted the size of the original image to obtain the

Journal Pre-proof

Journal Pre-proof

gradient difference of a three-level pyramid image in various sliding windows. Then, K-means clustering was used to classify each gradient difference window into text and non-text. The main limitation of this approach is the high computational cost for processing numerous sliding windows. Based on low-level features (such as light intensity, color and gradient), connected component-based methods [27, 28, 29, 30] first aggregate the pixels of the image into connected components. Then, a classification model is used to filter the noise regions. Tian et al. [30] proposed a minimum cost flow network. It detects the connected components of candidate characters by the cascading boosting method, which effectively solves the problem of error accumulation in text detection. These traditional methods use handcrafted image features for text detection and need to design different features for different scenes. This causes them to be unsuitable for complex natural scenes.

Text detection with bounding boxes. In the early stage of deep learning, most text detection models focus on the prediction of bounding boxes, including two-stage and one-stage models. These methods [9, 12, 13, 14, 31, 32, 33, 34] rely on the precise setting of anchors to predict bounding boxes and determine text regions. Jaderberg et al. [32] trained a convolutional neural network (CNN) as a classifier to evaluate the probability that a sliding window contains text. Then, the sliding window with the highest probability was regarded as the bounding box for text detection. Their method contains multiple stages and components, such as candidate area aggregation and word segmentation. Hence, its performance is easily affected by complex backgrounds. Thus, some developed methods remove traditional features and use the design of anchors to exploit deep features for complex scenes. In [33], the authors classified the proposals that were generated by RPN and predicted the inclined minimum area boxes with features of various pooled sizes to detect arbitrarily oriented text. Liao et al. [12] adapted various properties of a single-shot multibox detector (SSD) network, such as the aspect ratio of the default box, the size of the convolution kernel and the cascading NMS, but these changes made the algorithm heavily dependent on the image scale. These methods with bounding boxes need to reset the anchor size in text datasets of different scales, which makes it difficult for them to adapt to text detection tasks in different environments and languages.

Text detection with image segmentation. Methods of this type [15, 16, 18, 19, 35, 36] are based on image segmentation and aim to identify text regions at the pixel level. Xu et al. [35] designed a direction field that is represented by an image of 2D vectors and is learned via FCN. It encodes both binary text mask and direction information to separate adjacent text instances. However, the generation process of the direction field mask and the morphology-based postprocessing of the predicted direction field are very complex. Liu et al. [36] imported a conditional spatial expansion (CSE) mechanism to improve the performance of curved text detection. CSE is a conditional prediction process that retrieves an instance-level text region by seeding and expanding. Based on the observation of the position of the image block and the context that is inferred from the merged region, it selectively expands its area starting from any internal point (seed) of the text region. Segmentation-based methods largely avoid the disadvantages of anchors under specific conditions to improve the performance of detectors. Segmentation-based methods have achieved remarkable performance in arbitrarily oriented text detection, but there are still various difficulties to be addressed, such as complex training steps [36] and unsatisfactory robustness [16]. Therefore, we propose a novel progressive region prediction algorithm that is based on image segmentation. This algorithm utilizes a 2D progressive kernel to transform bounding box detection into pixel-level probability prediction. The kernel is closely related to the variability of the text region, which can more easily and better represent arbitrarily oriented texts and curved texts. A simple postprocessing method is used to transform the pixel prediction of the network into a bounding box. On the ICDAR 2015 and SCUT-CTW1500 datasets, the experimental results demonstrate that the proposed algorithm outperforms established segmentation-based methods.

3. Methodology

The main purpose of this paper is to find a special way to describe the distance variation between the text center and the text side. This will help the network divide adjacent text instances and background areas. Inspired by [15], we propose a novel 2D progressive kernel for describing the variability of the text. As illustrated in Figure 2, we utilize a 2D progressive kernel to convert the GT bounding box into a GT 2D mask which has a progressive pixel probability. For predicting the pixel probability of text regions, we also propose a new progressive region prediction network (PRPN). The network output has only one channel for the pixel probability map, and the loss between the output and GT mask is calculated and used to train the network. In the testing stage, with the help of the Watershed algorithm and a simple filtering operation, the output that is predicted by the trained network can be easily transformed into bounding box results.

3.1. Two-Dimensional Progressive Kernel and Label Generation

Traditional detection methods that are based on bounding boxes and borrow directly from object detection methods can only detect scene text using binary discrete distributions. Since they do not combine the regional variability of texts, they have weak generalization ability and limited accuracy. Thus, we propose a 2D progressive kernel for combining text features effectively.

In contrast to binary image segmentation which tags each pixel discretely, we use the 2D kernel to continuously encode the probability of text pixels and generate a progressive mask, which is used to train the network. The mask representation method has been used in various fields, such as object detection [37] and pose estimation[38]. These applications mostly use the Gaussian kernel as a point coding method because of its high flexibility in handling circular GT regions.

Journal Pre-proof

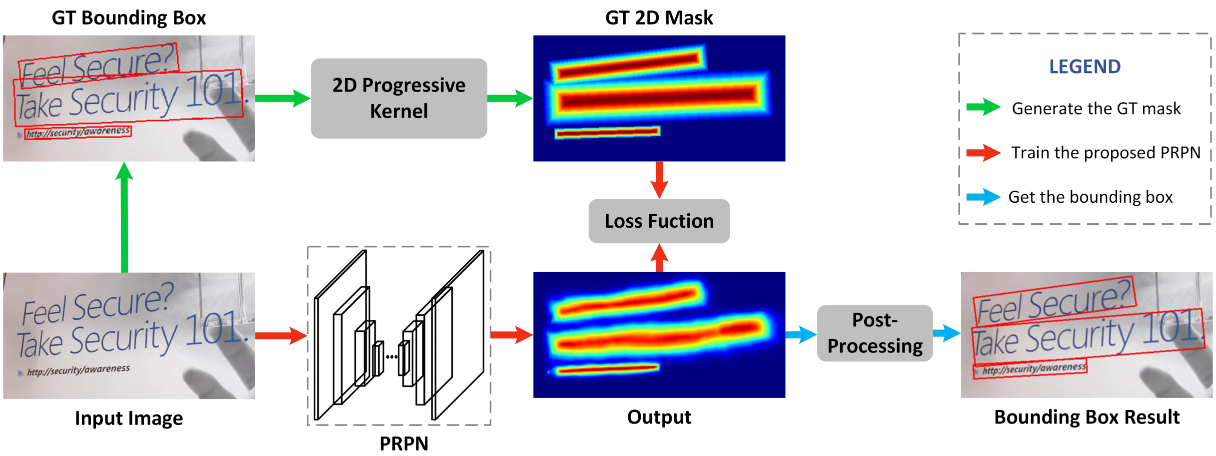


Figure 2: Overall workflow of PRPN. The complete workflow includes three stages: (1) A 2D progressive kernel is utilized to generate a GT 2D mask with pixel probability. (2) In the training stage, the loss between the GT mask and the output is calculated and used to train the PRPN. (3) In the testing stage, postprocessing is used to transform the output to bounding box results.

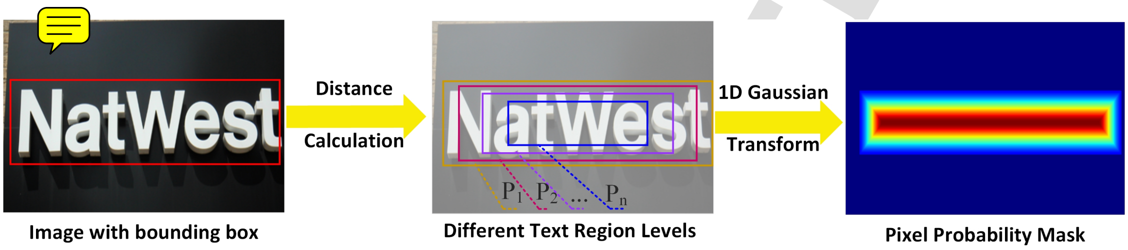


Figure 3: Illustration of the generation of a GT mask by the 2D progressive kernel. First, the distance transform algorithm is used to divide the text region into various distance levels. Then, a 1D Gaussian function is used to map the distance levels into a pixel-level mask, and the mask is normalized.

The mask that is generated by the 2D Gaussian kernel is generally effective for these applications. Although pixel coding with a 2D Gaussian kernel is suitable for character detection with a similar aspect ratio, describing the complex words or text lines in natural scenes is difficult.

To retain its flexibility, the design of the 2D progressive kernel is based on a 1D Gaussian function. Figure 3 summarizes the generation pipeline of the progressive mask. We divide the process of mask generation by a 2D progressive kernel into two steps: 1) the distance levels of pixels in various text regions are calculated; and 2) a 1D Gaussian function is used to map the distance values of the pixels to generate a probability mask with Gaussian properties.

Due to the obvious regional characteristics of text, a region segmentation strategy is very important for combining the variability. In a binary image, 1 represents foreground points, and 0 represents background points. Here, the points of the text region are regarded as the foreground points, and nontext points are regarded as the background points. We use the distance transform between a foreground point and the nearest background point as the measure of the distance level for region segmentation. We use the Euclidean distance to calculate the distance between two points. Compared with other calculation methods, the Euclidean distance has many advantages, such as high precision and consistency with the actual distance. It can be formulated as:

$$disf(c_1, c_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}, \tag{1}$$

where c_1 and c_2 are any two points and their coordinates are (x_1, y_1) and (x_2, y_2). The distance transform can be calculated as:

$$D(c) = min(disf(c, s)), \tag{2}$$

where c is in F and s is in B. F is the set of foreground points and B is the set of background points.

Through the distance transform, we calculate the transform distance of each pixel and divide the text region into various distance levels, such as P_1, P_2..., P_n in Figure 3. Here, n represents the number of distance levels, and text regions of different sizes have different values of n. Then, using n as a parameter, we can obtain the distribution parameters of the 1D Gaussian function for mapping the distance values. Finally, we normalize these values in the various text regions. With the 1D Gaussian function, the mapping value of each level is calculated as follows:

Journal Pre-proof

$$G_i = D(c_i) * e^{-\frac{(i-\frac{n}{2})^2}{(2*\sigma^2)}}, \tag{3}$$

where $i = 1, ..., n$ and $\sigma = 0.5 * Max(D(c_1), ..., D(c_n)) - 1$. The steps of label generation are described as Algorithm 1.

Algorithm 1 Generating the 2D progressive mask for text detection

Input: The bounding box of text regions for each image
Output: The 2D progressive mask with pixel value ranging between 0 and 1

1: Initializes a 2D mask D with zeros
2: **for** each label in all labels **do**
3: Create a temporary mask T with the same size as D
4: Draw the contour on T according to the label and fill with 1
5: Divide pixels in T into foreground points S1 and background points S2.
6: Calculate the distance between each foreground point (x, y) in S1 and the nearest background point in S2 with Eq. 2. The set is denoted as S3
7: Find the maximum Max and minimum Min in S3
8: For each foreground point, the transform distance G is calculated as follows:
$G(x, y) = 255 \times |S_3(x, y) - Min|/|Max - Min|$
9: According to different G, text regions are divided into n distance levels. Then by Eq. 3, mapping the distance values of pixel to mask with Gaussian properties and normalizing it
10: Adding T to D
11: **end for**
12: **return** 2D progressive mask D

The mask that is generated by the 2D progressive kernel is based on the bounding box labels. The mask does not represent the real text probability, and we just use it to separate the adjacent text instances. Moreover, the center point of text instance can hardly describe the text, but the center region can easily describe it. The proposed 2D progressive kernel can effectively encode the text region. With the predicted mask, we can easily separate adjacent texts and curved texts, as shown in Figure 1. Compared with previous segmentation-based methods, PRPN is simpler to build and train. These advantages are based on the unique and effective progressive mask.

3.2. Pipeline

PRPN is based on image segmentation, which requires the prediction of the pixel probability. Figure 4 illustrates the structure of PRPN. It outputs a feature heatmap, which determines the probability that pixels are away from the center of the text. To better predict the probability of a region, we need to associate more feature information in various directions beyond the text region.

Inspired by CornerNet [23], we design a directional pooling module to capture richer visual features in the horizontal and vertical directions. Figure 4 (a) illustrates the directional pooling module. Compared with corner pooling [23], we reconstruct the whole pooling architecture to expand the receptive field. First, we preserve CornerNet's residual structure, which enables the network to selectively use the features that are extracted by the directional pooling module. In the residual structure, a combination of directional pooling modules is used to replace the original convolution module. It uses two opposite 3×3 directional pooling modules (LeftPooling and RightPooling, TopPooling and BottomPooling) in two branches to process the output feature of the first Conv layer. Then, the outputs of the two branches are concatenated in channels, and finally, the 3×3 Conv-BN-ReLU module is used to extract the features. Via this method, the directional pooling module can help the network collect more feature information.

The framework of the proposed PRPN is illustrated in Figure 4 (b). The network follows the overall design of U-Net [22], and it can aggregate low-level features step by step. We replace the skip architecture with the directional pooling module in the decoding part. The network output has only one channel as a pixel probability map for text detection. Through the above design, PRPN is suitable for detecting arbitrarily oriented texts and curved texts.

3.3. Loss Function

For training, we need to calculate the loss between the GT pixel probability and the network output in Figure 2. The loss function is a weighted loss function, which is expressed as follows:

$$Loss = W * L_{pixel}, \tag{4}$$

where L_{pixel} is the loss of all pixels, which we use Smooth L1 [39] to calculate, and W is the weight of all pixels. It consists of two parts: the weight of positive pixels and the weight of negative pixels. Here, positive pixels and foreground points are positive samples, while negative pixels and background points are negative samples.

Positive Weight. The sizes of text instances are usually inconsistent in complex natural scenes. As shown in Figure 1 (a), the areas of other words are far smaller than the area of 'SALE'. If the same weight were used to calculate the loss of all positive pixels, it would impact the detection of text instances with small areas and decrease the performance of the model. We design a weighted loss to address this problem. It equally treats all instances in an image by giving the same weight to each instance. The weight of the i-th text instance is formulated as:

$$W_i = \frac{S}{N}, \forall i \in \{1, ..., N\}, \tag{5}$$

where N is the number of instances and $S = \sum_{n=1}^{N} S_i$, and S_i is the area of the i-th instance. However, we cannot directly use W_i to calculate the weights of pixels in an instance. This is because all pixels in an instance are different, and

Journal Pre-proof

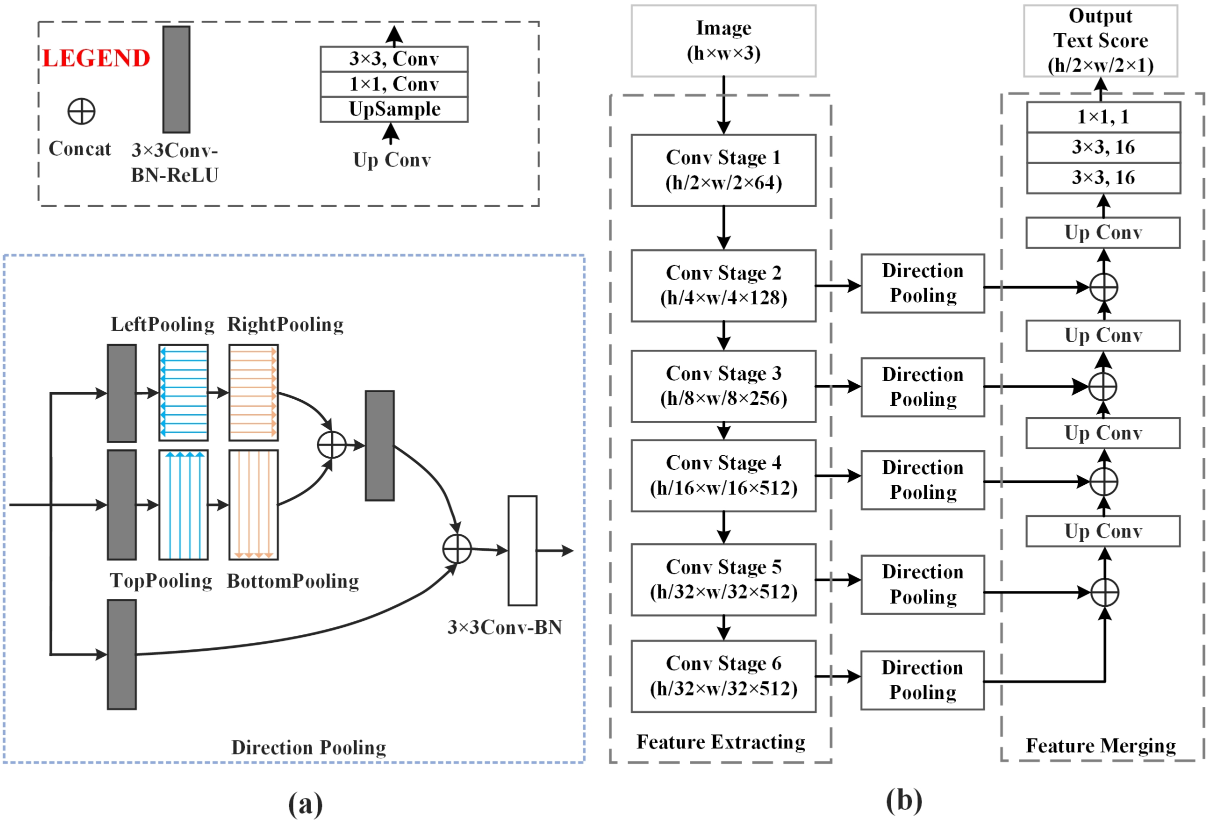


Figure 4: Structure of the proposed PRPN. The symbol ⊕ denotes the concatenation of channels by stacking, the pooling modules (left, right, top and bottom) refer to [23], and the upsampling operation is bilinear interpolation. (a) The directional pooling module. Pooling in various directions is used to expand the receptive field of the detector for better region segmentation. (b) The framework of PRPN. The backbone network is based on U-Net [22]. Before concatenating the original features, we use the directional pooling module to improve the feature extraction capabilities.

the scaling of loss computations is essential. The task of progressive prediction will make the pixel probability at the edges of text regions close to zero. Consequently, the network will be more inclined to learn from negative samples when calculating the edge loss. This will lead to unacceptable effects. If a pixel is closer to the text center, its impact is smaller. Therefore, we obtain the complement of the region mask and add 0.5 to the value of the complement probability. Via this approach, from the edge of the text to the center, the pixel values range from 1.5 to 0.5. Finally, multiplying by S_i/W_i, we obtain the loss weight of the positive sample. The weight of a positive pixel can be calculated by the following formula:

$$W_{pixel} = \frac{S_i}{W_i} * (1 - P_i + 0.5), \tag{6}$$

where P_i is the pixel probability value of the i-th progressive 2D mask, which depends on the pixel.

Negative Weight. The ratio of positive samples to negative samples in natural scenes is usually quite unbalanced. Consequently, we use the online hard example mining (OHEM) [40] method to select partial negative samples. In this work, the ratio of positive samples to negative samples is set to $1:3$ in training. Thus, some negative samples will not contribute to the network. In [40], the loss of all negative samples were calculated, and the top-k negative samples were selected for training. However, our method does not distinguish hard negative samples and simple negative samples. For text detection, scene images usually contain many simple samples, while the hard samples between adjacent text instances are difficult to separate. It is necessary to guide the network to select hard samples. With the label generation method that is described in Section 3.1, we regard text regions as background points and the nontext region as foreground points and execute Step 1-8 in Algorithm 1 to obtain the weight map.

Finally, we combine the weights of positive samples and negative samples to obtain the weighted loss, as illustrated in Figure 1 (c). A text instance with a smaller area has a brighter color, which corresponds to a larger value. Each region also darkens from the edge to the center, which indicates that the value gradually decrease. The weights of positive samples are much larger than the weights of negative samples. Since the differences among the negative sample values after normalization are small, the color of the nega-

Journal Pre-proof

tive samples shows little brightness variation.

3.4. Inference

As illustrated in Figure 2, the output of PRPN is a one-channel prediction heatmap, and the ultimate detection task is to obtain the coordinates of bounding boxes. Therefore, we design a postprocessing algorithm for generating bounding boxes. **Post-processing** is summarized as follows: First, we need to set two thresholds for filtering out unnecessary noise in the process of finding the maximum point (the high-prediction threshold and the low-prediction threshold). All pixels with values that exceed the high-prediction threshold are set to 1, and all pixels with values that are less than the low-prediction threshold are set to 0. In our experiments, we empirically set the high-prediction threshold to 0.5 and the low-prediction threshold to 0.15. Second, we binarize the feature map and find all maximum points in terms of the Euclidean distance. We utilize the characteristics of eight connected domains to analyze the connected components for implementing the watershed algorithm [41]. The watershed algorithm generates a series of region segmentation contours. Last, contours are transformed by traversing and filtering the segmentation results to obtain the bounding boxes for text detection.

4. Experiments

First, we evaluate the performance of the proposed method in quadrangular text detection on three public benchmarks: ICDAR 2013, ICDAR 2015, and MSRA-TD500. Second, we evaluate the performance of PRPN in detecting curved text on SCUT-CTW1500. Finally, we conduct ablation studies for the proposed PRPN.

4.1. Benchmark Datasets

ICDAR 2013 (IC13) [42] was released during the ICDAR 2013 Robust Reading Competition. It contains 229 training images and 233 test images that focus on horizontal text content of interest. They are all real-world images that show text on signs, books, posters, or other objects, and these texts are all in English.

ICDAR 2015 (IC15) [43] is the most commonly used benchmark for detecting natural scene text in arbitrary directions, which is Challenge 4 of the ICDAR 2015 Robust Reading Competition. Text regions in IC15 are annotated by four vertices of a quadrangle.

MSRA-TD500 (TD500) [44] contains 500 natural images which comprises 300 training images and 200 testing images. In contrast to the IC13 and IC15 datasets, the texts are written in Chinese and English. In TD500, the text regions are of arbitrary orientations and annotated at the sentence level.

SCUT-CTW1500 [45] was constructed by Yuliang et al., which includes 1000 training images and 500 testing images. It is a challenging dataset for curved text detection. In SCUT-CTW1500, the text regions are annotated by a polygon with 14 points that describes the shape of the arbitrarily curved text. The annotation of the dataset is extremely different from those of other datasets.

4.2. Implementation Details

In most datasets, the image sizes vary substantially. Therefore, we normalize all images to 720 pixels on the short side. We use data augmentation to enhance the robustness. The following data augmentation process is adopted for the training model: (1) the images are horizontally flipped, vertically flipped, and rotated in the range of $[-15°, 15°]$ randomly; (2) the color channel and brightness of the images are also changed randomly; (3) the images are scaled with a ratio in the range of $[0.8, 1.5]$ and cropped to 512×512 randomly; and (4) the channel means and standard deviations are used to normalize the images. On the image labels, we perform the same operation as on the image to maintain the consistency. Then, we use Algorithm 1 directly to generate the pixel probability mask.

The training process consists of two stages: First, we use the IC15 dataset to train PRPN for 300 epochs. Then, the model is fine-tuned to adapt to other benchmark datasets for 200 epochs. In the first stage, the CNN layers of the network are initialized by KaiMing Normal [46], and the original ReLU is replaced by LeakyReLU after batch normalization. The learning rate is set to 10^{-3} for the first 200 epochs and decayed to 10^{-4} for the remaining 100 epochs. The model is end-to-end trained by using the standard Adam algorithm [47]. Its momentum and weight decay are set to 0.9 and 5×10^{-4} respectively. For the fine-tuning stage, the learning rate is adjusted to 10^{-4} for the first 100 epochs and 10^{-5} for the last 100 epochs, and the other settings are consistent. For the IC15 and SCUT-CTW1500 datasets, "DO NOT CARE" texts are not included in the loss calculation of backpropagation. For details, please refer to https://github.com/xinyuch/Progressive-TextDetection. All experiments are conducted on the same workstation with 32G RAM, an NVIDIA GeForce RTX 2080Ti GPU and an Intel(R) Core(TM) CPU i9-9900KF (3.60 GHz).

4.3. Experimental Results

4.3.1. Detecting Quadrangular Text

First, we evaluate the proposed method on three quadrangle datasets, namely, IC13, IC15 and TD500, with the standard evaluation protocol [48]. Several detection results are shown in Figure 5. According to these results, PRPN could adapt to a variety of natural scenes and could also detect text instances that were close to each other.

Detecting Horizontal Scene Text on IC13. IC13 is one of the most popular horizontal text datasets. Because the resolution of images on the IC13 dataset varies from 422×102 to 3888×2592, we resize all the training images and testing images such that their short side is 720 pixels. Table 1 compares our PRPN and other recently published methods. Our method shows comparable performance with start-of-the-art method RRPN [11], which is the traditional two-stage CNN detector. Compared with two-stage detectors such as R2CNN [33] and Textbox++ [12], the proposed PRPN directly predicts the probabilities of pixels for the bounding box while maintaining high precision.

Detecting Incidental Scene Text on IC15. This dataset

Journal Pre-proof

Figure 5: Several representative results for PRPN on the IC13, IC15 and TD500 datasets of quadrangular text examples. The first row presents the prediction results of PRPN, and the second row presents the bounding boxes of the detection results after postprocessing.

Table 1: Results on IC13. 'P', 'R' and 'F', refer to precision, recall, and F-measure, respectively. All listed results for comparison are quoted from the corresponding original papers.

Methods	P(%)	R(%)	F(%)
R2CNN[33]	**93.6**	83.0	87.7
CTPN[31]	93.0	73.0	82.0
SegLink[13]	87.7	83.0	85.3
TextBoxes++[12]	91.0	84.0	87.0
RRPN[11]	90.0	72.0	80.0
Lyu et al.[7]	92.0	84.4	88.0
HAM[49]	89.6	81.3	85.3
The proposed PRPN	90.8	**86.1**	**88.4**

is more challenging than other datasets because it includes blurred images that were collected with poor focus. All the images on IC15 have the same resolution of 1280×720; thus, there is no need to adjust the sizes of the images. PRPN shows excellent performance with an F-measure of 86.0%. As presented in Table 2, our method outperforms many representative methods. In particular, Huang's method [21] with the pyramid attention network (PAN) performs similarly to our method in terms of F-measure. However, it has a more powerful backbone network, namely, ResneXt50 [51]. The result on IC15 demonstrates that the proposed PRPN is competent in adapting to various unstable scenes.

Detecting Long Scene Text On TD500. For the TD500 dataset, text annotations are in lines, including the spaces between the words in the bounding box. The resolutions of images on TD500 range from 1296×864 to 1920×1280; therefore, we keep the 720 pixels of the short side by the

Table 2: Results on IC15 Incidental Scene Text. All listed results for comparison are quoted from the corresponding original papers.

Methods	P(%)	R(%)	F(%)
R2CNN[33]	85.6	79.7	82.5
EAST[9]	83.3	78.3	80.7
CTPN[31]	74.2	51.7	60.9
SegLink[13]	74.7	76.5	75.6
RRPN[11]	84.0	77.0	80.0
PixelLink[16]	82.0	**85.5**	83.7
Deng et al.[14]	88.7	80.7	84.5
Huang et al.[21]	86.8	83.2	85.0
Lyu et al.[7]	**89.5**	79.7	84.3
TextField [35]	84.3	83.9	84.1
TextSnake[50]	84.9	80.4	82.6
The proposed PRPN	88.5	83.7	**86.0**

same scaling operation. We evaluate the results by the same method as in [44]. Moreover, we set the high-text threshold to 0.4 due to the line-level annotations, which differs from the threshold values for IC13 and IC15. The results in Table 3 show the performance differences between the state-of-the-art methods and the proposed PRPN. Our method achieves an outstanding F-measure value on TD500. RRPN [11] performs the worst among these methods due to its fixed anchors. The performance of EAST [9] is relatively reduced due to its high demand on the large receptive field.

4.3.2. Detecting Curved Text

Additionally, we evaluate our PRPN on a typical curved text dataset, namely, SCUT-CTW1500. Figure 6 shows the de-

Journal Pre-proof

Journal Pre-proof

Figure 6: Results of curved text detection on SCUT-CTW1500. The left dashed box represents a pair of results: the first row presents the network output without any processing, and the second row presents the bounding-boxes of the detection results after postprocessing. The images in second dashed box are additional detection results.

Table 3: Results on the TD500 Focused Scene Text task. 'P', 'R' and 'F', refer to precision, recall, and F-measure, respectively. All listed results for comparison are quoted from the corresponding original papers.

Methods	P(%)	R(%)	F(%)
RRPN[11]	82.0	69.0	75.0
EAST[9]	87.3	67.4	76.1
SegLink[13]	86.0	70.0	77.2
PixelLink[16]	83.0	73.2	77.8
Lyu et al.[7]	87.6	76.2	81.5
TextSnake[50]	83.2	73.9	78.3
CRAFT [15]	**88.2**	78.2	82.9
TextField [35]	88.0	79.0	83.0
HAM[49]	81.8	78.7	80.2
The proposed PRPN	85.5	**81.3**	**83.3**

tection results for several examples on SCUT-CTW1500.

SCUT-CTW1500 can assess the ability of the proposed model to detect arbitrarily shaped text. On this dataset, the resolutions of the images range from 320×144 to 5591×2614; thus, we perform the same scaling operation. We use the same evaluation method [45] to evaluate the results of the model. Table 4 presents the performance on SCUT-CTW1500. PRPN outperforms all the other detectors in terms of precision and F-measure. In particular, our model outperforms the CSE [36]. The CSE is specially designed to detect curved texts. Figure 6 shows the effectiveness of the model in detecting various curved or deformed text instances. The experiment clearly demonstrates that our PRPN can robustly detect curved text in complex scenes.

Table 4: Results on the SCUT-CTW1500 Focused Scene Text task. 'P', 'R', and 'F', refer to precision, recall, and F-measure, respectively. * indicates that the results are from [45].

Methods	P(%)	R(%)	F(%)
CTPN[31]	60.4*	53.8*	56.9*
SegLink[13]	42.3*	40.0*	40.8*
EAST[9]	78.7*	49.1*	60.4*
CTD+TLOC[45]	77.4	69.8	73.4
TextSnake[50]	67.9	**85.3**	75.6
PSENet[18]	82.0	79.3	80.6
CSE[36]	81.1	76.0	78.4
The proposed PRPN	**83.2**	79.7	**81.4**

Table 5: Results of various network structures and loss function networks. 'P', 'R', and 'F', refer to precision, recall, and F-measure, respectively. (PRPN_No_Direction_Pooling refers to the PRPN without directional pooling. PRPN_No_Weighted_Loss refers to the trained model without the weighted loss.)

Method	P(%)	R(%)	F(%)	FPS
PRPN_No_Direction_Pooling	80.4	77.8	79.1	7.2
PRPN_No_Weighted_Loss	81.5	75.6	78.4	5.1
PRPN	88.5	83.7	86.0	5.0

4.4. Ablation Study

Why is directional pooling necessary? To evaluate the effectiveness of the directional pooling module, we perform an

Journal Pre-proof

ablation study on the IC15 dataset. We adopt VGG-16 as the backbone network to extract the original feature maps. Then, we compare PRPN with directional pooling and PRPN without directional pooling in terms of prediction performance. The results are presented in Table 5. PRPN with directional pooling obtains a higher F-measure, and directional pooling can effectively combine the characteristics of various directions to improve the performance. Moreover, we evaluate the impact of our module on the computational complexity. As presented in Table 5, compared with PRPN_No_Direction_Pooling, the proposed PRPN realizes a speed of 5 fps, which is slightly lower for extra computation. However, PRPN with directional pooling has a large advantage in terms of F-measure. Hence, the slight speed loss could be acceptable.
Is the weighted loss truly effective? To resolve this doubt, we test the performance of PRPN trained with the proposed weighted loss and without the weighted loss. Table 5 presents the test results with the same number of training epochs. We determine that PRPN with the weighted loss leads in terms of precision, recall, and F-measure. A network also needs a suitable loss function to realize better performance. The proposed weighted loss assigns the same weights to positive samples of different sizes in an image; thus, all samples are treated equally in the model training. Meanwhile, it assigns different weights to negative samples, which caused the network to focus more attention on hard negative samples. The loss function is only used in the network training; thus, no additional computation is required in the network testing.

5. Conclusions and Future Work

Improving the performance of established text detection algorithms that are based on bounding box prediction is difficult. Additionally, robustly detecting the scene texts of different datasets without adjusting the parameters of the anchors if difficult. Inspired by segmentation-based detection methods, we propose a novel text detection method, namely, PRPN, that directly predicts a mask of pixels using a 2D progressive kernel instead of a bounding box. This two-dimensional kernel is developed from a 1D Gaussian function and overcomes the limitation of the 2D Gaussian kernel of being able to encode only targets with similar aspect ratios. The proposed kernel can freely encode long text lines or curved text. The distance level information that is generated can be used to better detect adjacent text instances. Moreover, we design a network with a directional pooling module to broaden the receptive field and design a weighted loss to improve the detection performance on hard samples. The experimental results on quadrangular text datasets (IC13, IC15, and TD500) and a curved text dataset (SCUT-CTW1500) demonstrate the superior performance of our PRPN. PRPN can be easily developed into an end-to-end scene text recognition algorithm. Moreover, it can be applied to similar visual tasks, such as image segmentation and object detection.

However, there are limitations that need to be overcome. Although the directional pooling module brings performance improvement, the computational complexity of the module leads to a decrease in the processing speed. In addition, PRPN has difficulty dealing with some cases of text embedding for rare training data. In the future, we hope to develop PRPN for end-to-end scene text recognition. A new network structure will be explored to improve the running speed while retaining the performance, such as [52, 53]. The design of the progressive kernel will also be optimized for hard text samples.

CRediT authorship contribution statement

Yuanhong Zhong: Conceptualization, Methodology, Writing review & editing, Supervision. **Xinyu Cheng**: Methodology, Software, Validation, Writing original draft, Visualization. **Tao Chen**: Validation, Investigation, Funding acquisition. **Jing Zhang**: Visualization, Project administration. **Zhaokun Zhou**: Conceptualization, Software. **Guan Huang**: Resources, Data Curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledge

The research was funded by the Natural Science Foundation of China [grant numbers 61501069] and the special project of technological innovation and application development of Chongqing [grant number No.cstc2019jscx-msxmX0167].

References

[1] N. Vo, L. Jiang, C. Sun, K. Murphy, L.-J. Li, L. Fei-Fei, J. Hays, Composing text and image for image retrieval-an empirical odyssey, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6439–6448.
[2] Y. Wu, S. Tang, S. Zhang, H. Ogai, An enhanced feature pyramid object detection network for autonomous driving, Applied Sciences 9 (2019) 4363.
[3] C. Alberti, J. Ling, M. Collins, D. Reitter, Fusion of detected objects in text for visual question answering, arXiv preprint arXiv:1908.05054 (2019).
[4] A. Meliones, D. Sampson, Blind museumtourer: A system for self-guided tours in museums and blind indoor navigation, Technologies 6 (2018) 4.
[5] Z. Zhong, L. Jin, S. Zhang, Z. Feng, Deeptext: A unified framework for text proposal generation and text detection in natural images, arXiv preprint arXiv:1605.07314 (2016).
[6] W. He, X.-Y. Zhang, F. Yin, C.-L. Liu, Deep direct regression for multi-oriented scene text detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 745–753.
[7] P. Lyu, C. Yao, W. Wu, S. Yan, X. Bai, Multi-oriented scene text detection via corner localization and region segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7553–7563.
[8] Y. Liu, L. Jin, Deep matching prior network: Toward tighter multi-oriented text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1962–1969.
[9] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the

Journal Pre-proof

IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 5551–5560.
[10] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, E. Ding, Wordsup: Exploiting word annotations for character based text detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4940–4949.
[11] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, X. Xue, Arbitrary-oriented scene text detection via rotation proposals, IEEE Transactions on Multimedia 20 (2018) 3111–3122.
[12] M. Liao, B. Shi, X. Bai, Textboxes++: A single-shot oriented scene text detector, IEEE transactions on image processing 27 (2018) 3676–3690.
[13] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2550–2558.
[14] L. Deng, Y. Gong, Y. Lin, J. Shuai, X. Tu, Y. Zhang, Z. Ma, M. Xie, Detecting multi-oriented text with corner-based region proposals, Neurocomputing 334 (2019) 134–142.
[15] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9365–9374.
[16] D. Deng, H. Liu, X. Li, D. Cai, Pixellink: Detecting scene text via instance segmentation, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[17] Y. Tang, X. Wu, Scene text detection using superpixel-based stroke feature transform and deep learning based region classification, IEEE Transactions on Multimedia 20 (2018) 2276–2288.
[18] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, J. Yang, Shape robust text detection with progressive scale expansion network, arXiv preprint arXiv:1806.02559 (2018).
[19] C. Xue, S. Lu, F. Zhan, Accurate scene text detection through border semantics awareness and bootstrapping, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 355–372.
[20] Z. Liu, G. Lin, S. Yang, J. Feng, W. Lin, W. L. Goh, Learning markov clustering networks for scene text detection, arXiv preprint arXiv:1805.08365 (2018).
[21] Z. Huang, Z. Zhong, L. Sun, Q. Huo, Mask r-cnn with pyramid attention network for scene text detection, in: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2019, pp. 764–772.
[22] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234–241.
[23] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 734–750.
[24] X. Chen, A. L. Yuille, Detecting and reading text in natural scenes, in: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004., volume 2, IEEE, 2004, pp. II–II.
[25] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 1457–1464.
[26] Y. C. Wei, C. H. Lin, A robust video text detection approach using svm, Expert Systems with Applications 39 (2012) 10832–10840.
[27] B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 2963–2970.
[28] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide-baseline stereo from maximally stable extremal regions, Image and vision computing 22 (2004) 761–767.
[29] L. Sun, Q. Huo, W. Jia, K. Chen, Robust text detection in natural scene images by generalized color-enhanced contrasting extremal region and neural networks, in: 2014 22nd International Conference on Pattern Recognition, IEEE, 2014, pp. 2715–2720.
[30] S. Tian, Y. Pan, C. Huang, S. Lu, K. Yu, C. L. Tan, Text flow: A unified text detection system in natural scene images, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 4651–4659.
[31] Z. Tian, W. Huang, T. He, P. He, Y. Qiao, Detecting text in natural image with connectionist text proposal network, in: European conference on computer vision, Springer, 2016, pp. 56–72.
[32] M. Jaderberg, A. Vedaldi, A. Zisserman, Deep features for text spotting, in: European conference on computer vision, Springer, 2014, pp. 512–528.
[33] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, Z. Luo, R2cnn: rotational region cnn for orientation robust scene text detection, arXiv preprint arXiv:1706.09579 (2017).
[34] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, J. Yan, Fots: Fast oriented text spotting with a unified network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5676–5685.
[35] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, X. Bai, Textfield: Learning a deep direction field for irregular scene text detection, IEEE Transactions on Image Processing 28 (2019) 5566–5579.
[36] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, W. L. Goh, Towards robust curve text detection with conditional spatial expansion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
[37] X. Zhou, J. Zhuo, P. Krahenbuhl, Bottom-up object detection by grouping extreme and center points, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 850–859.
[38] Z. Cao, T. Simon, S.-E. Wei, Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7291–7299.
[39] R. Girshick, Fast r-cnn, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448.
[40] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining (2016) 761–769.
[41] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (1991) 583–598.
[42] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, L. P. De Las Heras, Icdar 2013 robust reading competition, in: 2013 12th International Conference on Document Analysis and Recognition, IEEE, 2013, pp. 1484–1493.
[43] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, E. Valveny, Icdar 2015 competition on robust reading, in: International Conference on Document Analysis and Recognition, 2015.
[44] C. Yao, X. Bai, W. Liu, Y. Ma, Z. Tu, Detecting texts of arbitrary orientations in natural images, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 1083–1090.
[45] L. Yuliang, J. Lianwen, Z. Shuaitao, Z. Sheng, Detecting curve text in the wild: New dataset and new solution, arXiv preprint arXiv:1712.02170 (2017).
[46] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.
[47] D. Kingma, J. Ba, Adam: A method for stochastic optimization, Computer Science (2014).
[48] K. Wang, B. Babenko, S. Belongie, End-to-end scene text recognition, in: IEEE International Conference on Computer Vision, 2012.
[49] J.-B. Hou, X. Zhu, C. Liu, K. Sheng, L.-H. Wu, H. Wang, X.-C. Yin, Ham: Hidden anchor mechanism for scene text detection, IEEE Transactions on Image Processing 29 (2020) 7904–7916.
[50] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: A flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 20–36.
[51] S. Xie, R. Girshick, P. Dollr, Z. Tu, K. He, Aggregated residual

Journal Pre-proof

transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1492–1500.

[52] S. Heo, S. Cho, Y. Kim, H. Kim, Real-time object detection system with multi-path neural networks, in: 2020 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS), IEEE, 2020, pp. 174–187.

[53] Z. Liu, T. Zheng, G. Xu, Z. Yang, H. Liu, D. Cai, Training-time-friendly network for real-time object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 11685–11692.

Journal Pre-proof

Declarations of interest: none

Journal Pre-proof