A Character Flow Framework for Multi-oriented Scene Text Detection. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY

A Character Flow Framework for Multi-oriented Scene Text Detection

Abstract Scene text detection plays a significant role in various applications, such as object recognition, document management, and visual navigation. The instance segmentation-based method has been mostly used in existing research due to its advantages in dealing with multi-oriented texts. However, a large number of non-text pixels exist in the labels during the model training, leading to text mis-segmentation. In this paper, we propose a novel multi-oriented scene text detection framework, which includes two main modules: character instance segmentation (one instance corresponds to one character), and character flow construction (one character flow corresponds to one word). We use feature pyramid network (FPN) to predict character and non-character instances with arbitrary directions. A joint network of FPN and bidirectional long short-term memory (BLSTM) is developed to explore the context information among isolated characters, which are finally grouped into character flows. Extensive experiments are conducted on ICDAR2013, ICDAR2015, MSRA-TD500 and MLT datasets to demonstrate the effectiveness of our approach. The F-measures are 92.62%, 88.02%, 83.69% and 77.81%, respectively.

Keywords multi-oriented scene text detection, character instance segmentation, character flow, FPN, BLSTM

1 Introduction

Multi-oriented scene text detection in the wild has gained increasing attention with the popularization of mobile devices. It is challenging to detect texts from natural scene images, since the texts are usually in arbitrary orientations and scales, and various completeness and tightness.

Recently, up-to-date deep learning methods have been reported to achieve promising performance for multi-oriented scene text detection, which can be divided into two categories: bounding box regression-based and instance segmentation-based. In the first category, anchors need to be appropriately designed, considering their significant impact on the performance of text detection. The rectangular box is not always suitable for matching scene text [1], so that researchers handcraft multi-scale anchor boxes to regress multi-oriented text proposals. Liao et al. [2] presented a text box descriptor based on single shot multi-box detector (SSD) [3] to output diverse text boxes. The performance improvement lies in quadrilateral or oriented-rectangular anchors. Liu et al. [4] developed a deep matching prior network, and then applied quadrilateral boxes during the proposal generation to adapt to multi-oriented texts. Ma et al. [5] proposed rotation region proposal networks (RRPN) with a set of rotated anchor boxes to localize text regions. These methods can effectively address texts with long space between words or low contrast to the background; the downside is that the intensive manual work is inevitable.

The instance segmentation-based approach does not require handcrafted anchors during multi-oriented text proposal generation. Instead, it extracts text line instances directly without considering their directions. For example, EAST [6] uses fully convolutional networks (FCN) [7] and multi-channel feature map fusion to train a model, which can directly predict words or lines in any direction and quadrilateral shape from natural scene images. Mask TextSpotter [8] designs a mask text spotter based on MaskRCNN [9] to predict a character-level probability map for text spot recognition. The utilization of popular object segmentation methods may fail to distinguish different instances, therein inspiring efforts to solve the problem. In PSENet [10], after initial segmentation of text in-

Regular Paper

stances, the progressive scaling algorithm is used to regenerate different instances, in order to handle text lines with short distance. The inaccurate labels may also affect the segmentation results, since the rectangular calibration box contains too many background pixels for multi-oriented text. Therefore, SPCNet [11] uses outsourcing polygons to reduce the background pixels in ground truth, and advance the segmentation accuracy in the training stage. Another solution is to explore the relationship between characters separately; in other words, the segmentation of character instances rather than text lines. SegLink [12] uses SSD to detect text segments and text links simultaneously. Those segments are then combined into text lines by text links. PixelLink [13] applies feature pyramid network (FPN) [14] to predict text/non-text pixels and links, where text instances are grouped to words by links. CRAFT [15] also uses FPN to predict character-center and non-character-center pixels and affinity among characters. It introduces an effective weakly supervised learning method to enlarge the training dataset, and achieves outstanding performance.

Several related works based on other methods have recently been reported. Tian et al. [16] presented a connectionist text proposal network (CTPN) to detect the fixed-width text fragments. A bi-directional long short-term memory (BLSTM) network [17] was utilized to extract context information and combine the fragments into text lines. Lyu et al. [18] extracted candidate text proposals by corner point detection, and segmented position-sensitive text instances by FCN.

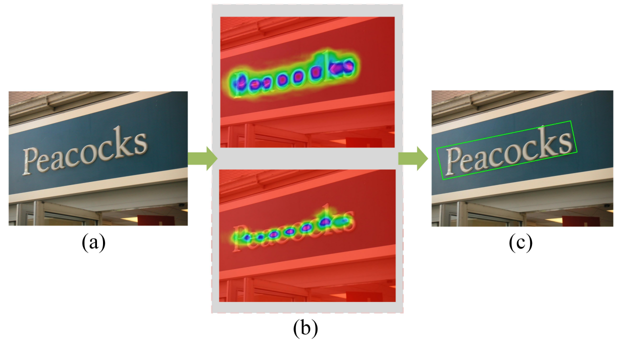


Fig.1. Construction of a character flow. (a) input image, (b) the segmented character instances (top) and the affinity of adjacent characters (bottom), (c) a character flow

In this paper, we apply FPN to segment the multi-oriented and multi-scale character instances, rather than text line instances. The advantage is that the combination of character instances into text line instances could facilitate the detection of different text lines from a real image. On the other hand, we integrate FPN and BLSTM to explore the sequential context between characters. It effectively groups single characters into character flows, i.e., words, which include all the individual characters in a word. Fig. 1 gives an example of character flow construction, where the affinity between adjacent characters is evaluated by their connection and context, so as to group the sequence of isolated characters into a character flow. Experimental results on four benchmark datasets demonstrate the superiority of our approach. The main contributions of this work are as follows:

- In order to detect multi-oriented texts from wild images, we propose a new text detection framework to localize text regions by character instance segmentation, and derive text lines by character flow construction.
- In order to reduce the background pixel interfer-ence during training, we focus on the segmenta-tion of character instances instead of text line in-stances.

- In order to construct character flows from character instances, we present a unified FPN-BLSTM network to measure the affinity of characters, without manual grouping rules.

2 Related Work

2.1 Connected components-based text detection

Detecting texts by extracting connected components from natural scene images has been developed over the years. Epshtein et al. [19] proposed a stroke width transform algorithm to calculate minimum path distance of two border pixels, and then group pixels with similar values into character candidates. Wu et al. [20] proposed a multi-scale adaptive color clustering scheme for text extraction. They assumed similar colors between characters in the same text line, and used a K-means color clustering algorithm to extract character candidates. In the work [21], maximally stable extremal regions (MSER) [22] were employed to detect text from natural images with low contrast and complex background. However, many overlapped text components could be generated, resulting in low detection accuracy and high computational cost. In order to eliminate redundant components, Yin et al. [23] proposed a MSER pruning algorithm by replacing the minimal variation with regularized variation, and improved character extraction accuracy and efficiency.

2.2 Regression-based text detection

Most methods in this category are inspired by the popular object detectors. Unlike objects in general, texts usually exhibit in arbitrary scales, orientations, and irregular shapes. To address these problems, Ma et al. [5] designed a set of rotated anchors in three sizes, three ratios and six directions, to extract text candidates with different directions and scales. Liao et al. [1] used large aspect ratio anchor boxes and irregular convolutional kernels to fit for scene text with different aspect ratios. They also combined a text recognition network named convolutional recurrent neural network (CRNN) [24] to improve text detection accuracy. Liu et al. [4] used quadrilateral sliding windows to locate text regions, and designed a sequential protocol to regress four vertices of polygon text box, so as to detect texts with perspective distortion. Liao et al. [25] used rotating filters to active convolution features with rotation-sensitive, which can detect multi-oriented texts.

2.3 Segmentation-based text detection

Many segmentation-based text detection frameworks are derived from instance segmentation, so as to deal with multi-oriented, multi-scale and arbitrary-shaped texts. Zhang et al. [26] generated saliency maps by FCN to predict text blocks, from which character candidates were extracted by MSER. The FCN was also applied to detect various angled discs, which were further concatenated into text lines [27]. Lyu et al. [8] applied MaskRCNN to detect text, consisting of four main networks: feature extraction network, text candidate region generation network, text bounding box regression network, and text instance and character segmentation network, which improved accuracy of text detection, especially for curved texts. Since inaccurate labels could lead to the generation of wrong samples, Xie et al. [11] proposed a text context module and rescore module to suppress false sample detection. Deng et al. [13] predicted text instances and links by FPN, however, the instance segmentation may result in incorrect classification. Therefore, Li et al. [10] proposed a progressive scale expansion algorithm to distinguish different text instances. A full word is constructed by a series of ordered characters; although characters have less receptive field, they not only maintain the advan-

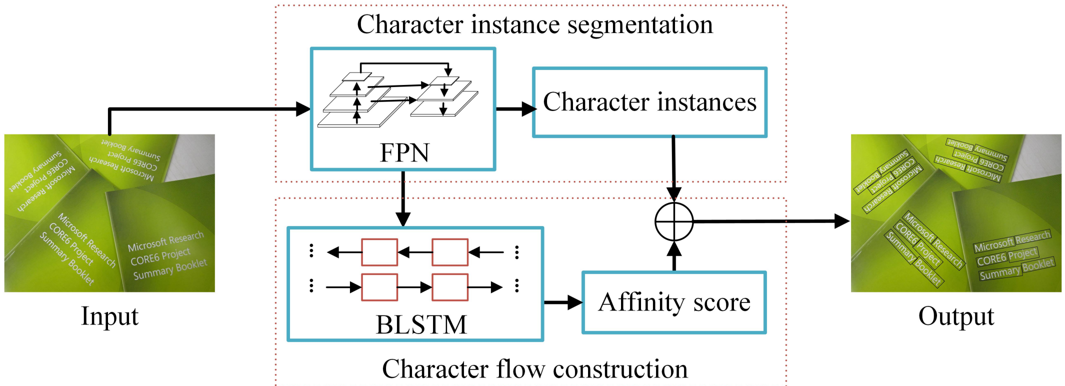


Fig.2. The proposed framework. Given an input image, a FPN model is trained to predict character/non-character instances, then a joint network is trained to combine the high affinity characters into character flows.

tages of text instance segmentation, but also overcome the difficulty to distinguish text instances. Motivated by this, Baek et al. [15] segmented every single character using FPN, then predicted the affinity among them.

3 Proposed Method

The overview of our framework for multi-oriented scene text detection is illustrated in Fig. 2. It consists of two major parts: character instance segmentation and character flow construction.

3.1 Character instance segmentation

Character instance segmentation is to separate text pixels from image background. Unlike other objects in natural scene images, texts often appear in arbitrary scales and orientations, as well as various colors and languages. Some traditional methods use the watershed algorithm [28] to extract text proposals. However, a large number of non-text proposals could also be generated, leading to difficulties for subsequent text classification. Although there are some excellent pruning algorithms like [23], it is still challenging to obtain a high-performance text classifier. With the development of object detection and instance segmentation algorithms that are based on deep learning, many practical frameworks have been proposed to effectively improve the text detection accuracy. The direct utilization of instance segmentation on the inclined texts may fail to classify multiple text instances [10], therein inspiring research on character instance segmentation.

In order to distinguish the text and non-text pixels, we utilize a FPN with ResNet-50 [29] to extract character instances. It is a top-down architecture that unifies both high-level and low-level semantic feature maps at all scales. As shown in Fig. 3, the network takes $h \times w \times 3$ sized inputs, and the convolutional stage1–stage5 is the infrastructure of residual network. Each stage has an up-sampling operation via bilinear interpolation and a skip connection. After adding feature maps bit by bit, each fused map is fed into Kernel(3×3)-BN-ReLU layers and reduced to 256 channels. Then it passes through n Kernel(1×1)-Up-Sigmoid layers, where the text region proposals are extracted. A BLSTM network is applied subsequently, resulting in an end-to-end trainable model. It is pre-trained on SynthText dataset [30], and automatically learns character features like fonts, color, size, stroke, etc. When the distributions of different types of character data have been learned, the character pixels can be gradually separated from the background. Fig. 4 shows the results of character instance segmentation.

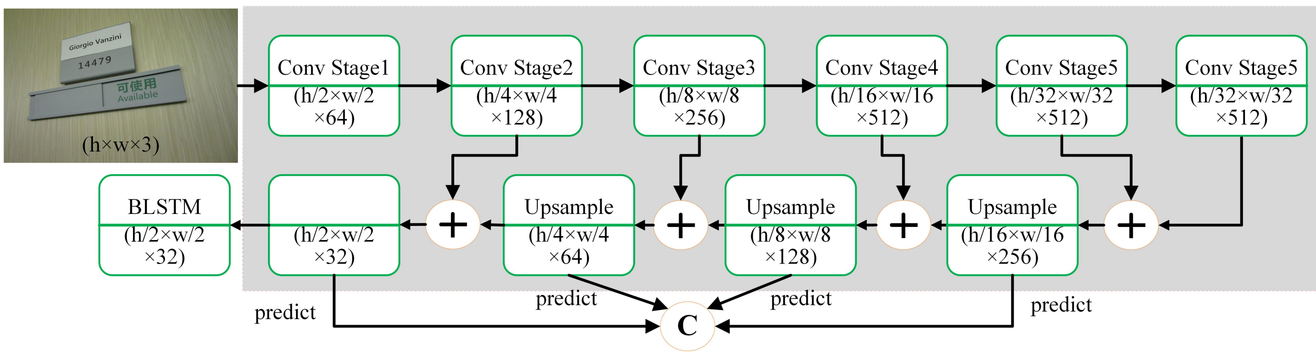


Fig.3. The detailed illustration of our network architecture, where →, ⊕ and Ⓒ represent the convolution operation, bitwise addition and fusion map of prediction results, respectively.

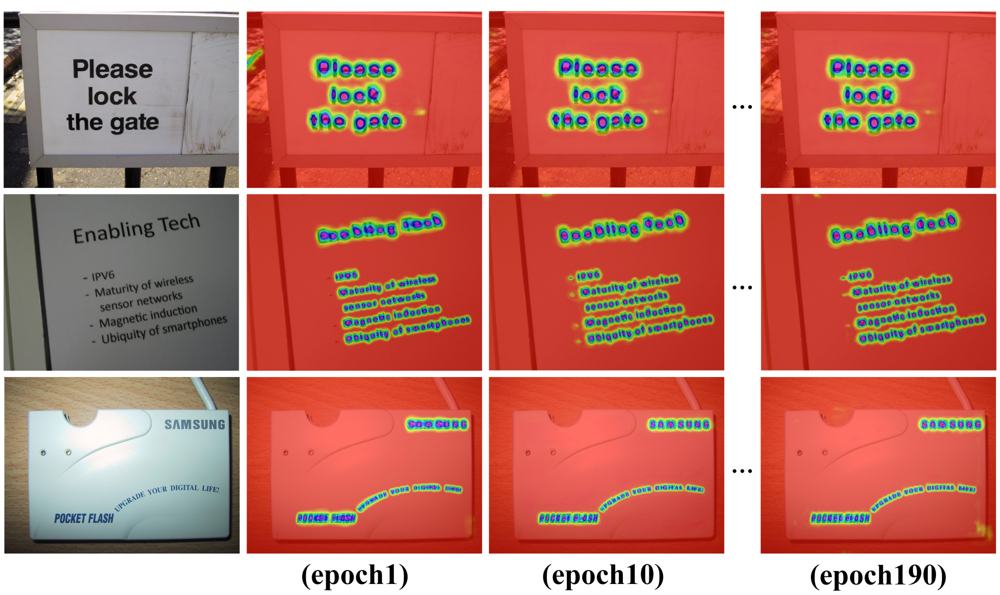


Fig.4. Character instances extracted at different epochs.

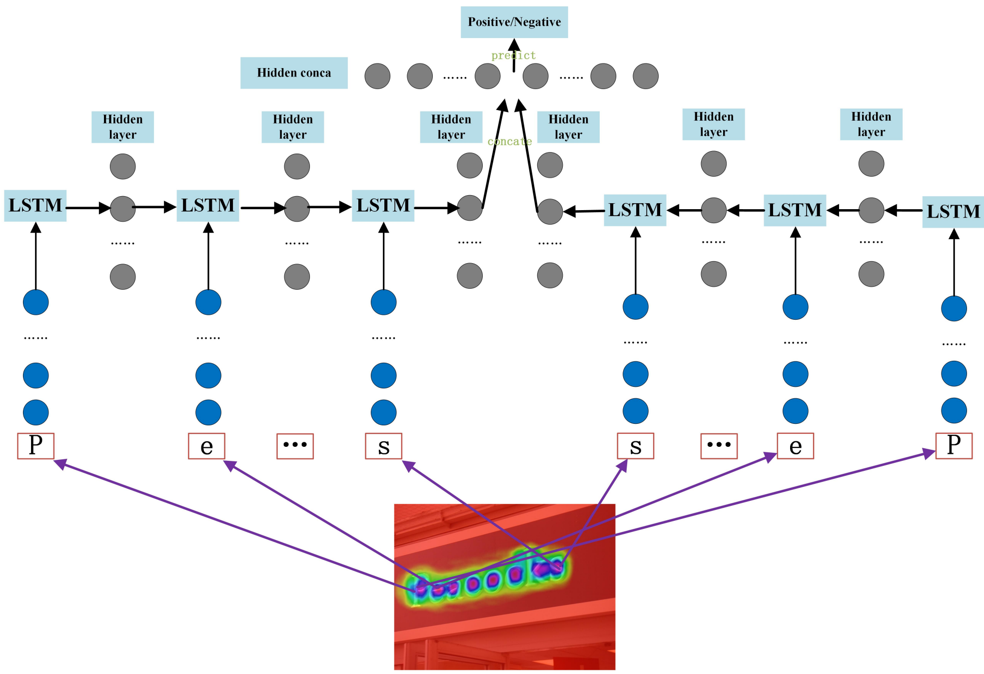


Fig.5. The context learning by BLSTM structure. A character sequence is input into bi-directional LSTM networks separately, where the sequential context information is learned to facilitate the text line construction.

3.2 Character flow construction

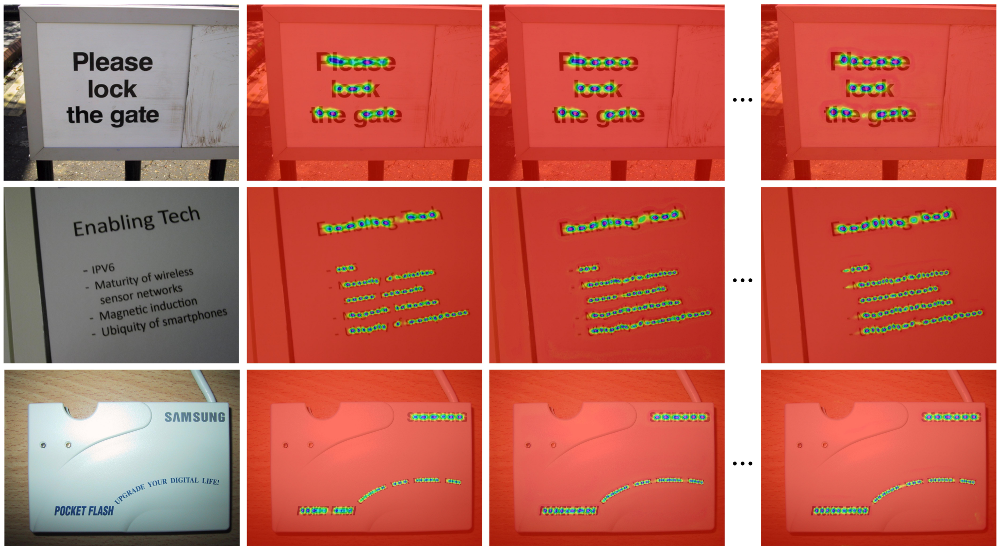
In order to combine the isolated characters, the measurement of their intrinsic connection is essential. Conventional grouping rules of text lines include spacing, aspect ratio, color clustering, etc. For example, Text Flow [31] uses a cost function to unify the geometrical features among characters. The minimal cost flow corresponds to a text line. CTPN [16] designs three rules (horizontal distance, vertical overlap, and proposal pairing) to join text segments into a text line. Those handcrafted rules could reduce the flexibility of deriving text lines, therefore, SegLink [12] traines a link model to connect adjacent characters.

Sequential context information exists in the character instances, which contributes to the construction of text lines. The BLSTM [17] is bi-directional LSTMs [32], a recurrent architecture to encode the context in opposite directions along the input sequence. It can effectively distinguish texts of arbitrary length from the background. As shown in Fig. 5, we apply a BLSTM, followed by a fully connected layer and a softmax classifier, to evaluate the character affinity and then construct character flows. A character flow corresponds to a complete word, i.e., a sequence of characters. The sequential features between adjacent characters, regarding to the color, space and orientation, are exploited to predict the connection relationship. Fig. 6 shows the affinity of adjacent characters. As can be seen, their connection becomes more and more obvious during the training procedure.

3.3 Label generation

Given an input image, its corresponding ground truth contains $C = \{c_1 = (cc_1, cl_1), c_2 = (cc_2, cl_2), \ldots, c_m = (cc_m, cl_m)\}$, where c_i is a horizontal rectangular box that represents the localization of a character region, cc_i and cl_i are the category and location of a character, respectively.

Here two types of mask maps are generated for segmentation network. One is the character instance



(epoch1) **(epoch10)** **(epoch190)**

Fig.6. The affinity between characters at different epochs.

mask, where the coordinates of character boxes are defined by the ground truth C. We generate this mask by drawing the normalized horizontal rectangles on a zero-initialized mask and filling the rectangles with the value of 1. In other words, pixels inside the bounding boxes are labeled positive; if overlap exists, only non-overlapping pixels are positive.

The other is the character affinity mask, where the coordinates of character affinity boxes can be computed by Eq. 1 and Eq. 2. To be more specific, for an affinity box, its center, width and height are the midpoint of adjacent character centers, the distance between adjacent character centers, and half of the larger height of adjacent character boxes, respectively. Similar pixel annotation is performed in order to construct character flows. The label generation for segmentation of character instances and character affinity is illustrated in Fig. 7.

$$A_x = Center(cc_i, cc_j)_x \tag{1}$$

$$A_y = Center(cc_i, cc_j)_y \pm Width_{sum}(cc_i, cc_j)/2 \tag{2}$$

where A_x and A_y are the coordinates of character affinity box, $Center(cc_i, cc_j)$ represents the center point of two adjacent characters in ground truth C, and $Width_{sum}(cc_i, cc_j)$ represents the sum of two adjacent character widths.

3.4 Optimization

Our loss function defined on each proposal is the sum of two quantities: the character/non-character pixel classification loss L_{class}, and the adjacent/non-adjacent character pair connection loss $L_{connect}$. It is designed to jointly optimize the model weights in multi-task learning. The overall loss is given as:

$$L_{total} = \sum_{p}(\lambda L_{class} + L_{connect}) \tag{3}$$

where λ is the weight to balance the two losses, and is set to 2.

According to PixelLink [13], L_{class} is the matrix of instance-balanced cross-entropy loss on character and non-character prediction, computed as:

(a) Input image (b)

Fig.7. Annotation of (a) character instances and (b) character affinity. For this input image, 4 character boxes and 3 character affinity boxes are generated for "BEER", and 6 character boxes and 5 character affinity boxes for "GARDEN".

$$L_{class} = \frac{W}{4NS} \sum_{p} -(y_p \cdot log(P_p) \quad + \\ (1 - y_p) \cdot log(1 - P_p)) \tag{4}$$

where S, W and N are the area of character instances, weight matrix [13] and the sum of pixels for a image, respectively, p is the pixel, y_p is the label of pixel defined as Eq. 5, and P_p is the probability of predicting a positive pixel.

$$y_p = \begin{cases} 1 & if \quad p \in positive \\ 0 & otherwise \end{cases} \tag{5}$$

$L_{connect}$ is computed as Eq. 6, where S_p^* is the confidence map of the ground truth defined as Eq. 7, and S_p is the predicted region score. The parameters c, $R(c)$, and p denote the ground truth of character, the annotated region of character affinity box, and the pixel in $R(c)$, respectively. The generation of ground-truth label for S_p^* is presented in Sec. 3.3.

$$L_{connect} = -||S_p^* - S_p||_2^2 \tag{6}$$

$$S_p^* = \begin{cases} 1 & if \quad p \in R(c) \\ 0 & otherwise \end{cases} \tag{7}$$

4 Experiments and Discussions

4.1 Datasets

ICDAR2013 dataset [33] contains 462 real scene images. Among them, 229 images are selected for training and the remaining 233 images for testing. It is the benchmark in the 2013 Robust Reading Competition, which focuses on the horizontal text detection in the wild. The ground truth is annotated at both word level and character level. All text regions are annotated by 4 vertices of a quadrangle.

ICDAR2015 dataset [34] contains 1500 real scene images. Among them, 1000 images are selected for training and the remaining 500 images for testing. It is the benchmark in the 2015 Robust Reading Competition, which focuses on the arbitrary-oriented text detection in the wild. The ground truth is annotated at the word level, and the text regions are annotated by 4 vertices of a quadrangle.

MSRA-TD500 dataset [35] contains 500 real scene images. Among them, 300 images are selected for training and the remaining 200 images for testing. It is a multi-language dataset that focuses on English and Chinese. The ground truth is annotated at the word level, and the text regions are annotated by 4 vertices of a quadrangle.

MLT dataset [36] contains 18000 real scene images. Among them, 9000 images are selected for training and the remaining 9000 images for testing. It is the benchmark of the 2017 Robust Reading Competition, which focuses on the multi-oriented, multi-scripting, and multi-lingual text detection in the wild. The ground truth is annotated at the word level, and the text regions are annotated by 4 vertices of a quad-

Table 1. Performance of different network settings on four benchmarks.

Networks	ICDAR2013			ICDAR2015			MSRA-TD500			MLT		
	R	P	F	R	P	F	R	P	F	R	P	F
FPN (with ResNet-50)	85.61%	92.44%	88.89%	74.15%	85.04%	82.66%	74.92%	77.48%	76.18%	66.24%	77.41%	71.39%
FPN-BLSTM (with VGGNet-16)	87.95%	92.64%	90.23%	83.73%	86.60%	85.14%	79.01%	81.33%	80.16%	69.11%	79.24%	73.83%
FPN-BLSTM (with ResNet-50)	**91.01%**	**94.29%**	**92.62%**	**86.86%**	**89.22%**	**88.02%**	**82.08%**	**85.37%**	**83.69%**	**73.93%**	**82.13%**	**77.81%**

rangle.

4.2 Implementation details

Our model is pre-trained on SynthText dataset [30], which contains around 857,500 synthetic images, with both word-level and character-level annotations. Text regions are annotated by 4 vertices of the quadrangle. The batch size is set to 128 on 4 GPUs for 50K iterations. The initial learning rate is 0.00003 and decreased to 0.000024 at 20K iterations. All the images are resized to 768×768. We use a weight decay of $5 \times 10^{-4}$and the training processes are optimized using the ADAM [37] optimizer. In order to quantitatively evaluate the performance, three metrics, namely, Recall (R), precision (P) and F-measure (F), are used.

4.3 Ablation study

4.3.1 Different backbone networks

To investigate the influence of backbone networks in our framework, both ResNet-50 and VGGNet-16 [38] are applied as the backbone of FPN. It is not surprising that with the deeper architecture and the residual structure, ResNet is expected to be effectively trained and derive discriminative text features, thus performing better than VGGNet (shown in Table 1). This experiment demonstrates that other different networks can also be embedded into our framework.

4.3.2 Influence of the BLSTM network

We verify the effectiveness of context learning in scene text detection by removing the BLSTM from our framework. Table 1 tabulates its behaviors on four datasets, where all three metrics decrease substantially without BLSTM. The possible reason lies in its special structure, which enables to discover more connection information between characters, so as to evaluate their affinity more accurately. Thus, we set BLSTM as an in-network architecture in the following experiments.

4.4 Comparison with the state of the art

We evaluate the proposed approach on ICDAR2013, ICDAR2015, MSRA-TD500 and MLT benchmarks. Some randomly chosen results are shown in Fig. 8– 11, respectively. It can be observed that our method is effective in detecting multi-oriented and multi-lingual texts. The comparison against several recent works is given in Table 2–4.

Table 2. Comparison results on ICDAR2013.

Methods	R	P	F
RRPN [5]	71.89%	90.22%	80.02%
SegLink [12]	83.00%	87.70%	85.30%
R2NN [39]	82.60%	93.60%	87.70%
CTPN [16]	83.00%	93.00%	88.00%
SSTD [40]	86.00%	89.00%	88.00%
PixelLink [13]	87.50%	88.60%	88.10%
TextBoxes++ [2]	86.00%	92.00%	89.00%
SPCNet [11]	90.50%	93.80%	92.10%
CRAFT [15]	**93.10%**	**97.40%**	**95.20%**
Ours	91.01%	94.29%	92.62%

4.4.1 Horizontal text detection

To evaluate the performance of horizontal text detection, we initialize the network with a pre-trained model and then fine-tune it on ICDAR2013 dataset. The model requires 12K iterations of training. As can be seen Table 2, our approach achieves competitive results: R, P and F are 91.01%, 94.29% and 92.62%,

Fig.8. Examples of text detection results on ICDAR2013.

Fig.9. Examples of text detection results on ICDAR2015.

Fig.10. Examples of text detection results on MSRA-TD500.

Fig.11. Examples of text detection results on MLT.

Table 3. Comparison results on ICDAR2015 and MSRA-TD500.

Methods	ICDAR2015			MSRA-TD500		
	R	*P*	*F*	*R*	*P*	*F*
RRPN [5]	73.23%	82.17%	77.44%	68.00%	82.00%	74.00%
SegLink [12]	76.80%	73.10%	75.00%	70.00%	86.00%	77.00%
R2NN [39]	85.62%	79.68%	82.54%	–	–	–
CTPN [16]	51.56%	74.22%	60.85%	–	–	–
SSTD [40]	73.00%	80.00%	77.00%	–	–	–
EAST [6]	78.30%	83.30%	80.70%	67.40%	87.30%	76.10%
MaskTextSpotter [8]	81.20%	85.80%	83.40%	–	–	–
PixelLink [13]	82.00%	85.50%	83.70%	73.20%	83.0%	77.8%
TextSnake [27]	80.40%	84.90%	82.60%	73.90%	83.20%	78.30%
TextBoxes++ [2]	78.50%	87.80%	82.90%	–	–	–
SPCNet [11]	85.80%	88.70%	87.20%	–	–	–
CRAFT [15]	84.30%	**89.80%**	86.90%	78.20%	88.20%	82.90%
SAE [41]	84.50%	85.10%	84.80%	81.70%	84.20%	82.90%
DB [42]	82.70%	88.20%	85.40%	73.20%	85.70%	79.00%
PSENet [10]	84.50%	86.92%	85.69%	–	–	–
Ours	**86.86%**	89.22%	**88.02%**	**82.08%**	**85.37%**	**83.69%**

respectively. Compared with the bounding box regression methods [5, 39, 2], our work shows a significant improvement on all metrics, indicating that most of character instances have been detected. This may be because we do not need to deal with large amount of candidate proposals generated by the anchor boxes, which may have a lot of overlap, leading to a decrease in detection rate. Compared with the instance segmentation-based methods [12, 13, 11], we also achieve an improvement on all the metrics. Similar observations can be found from the comparison with the text segment connection method [16]. This may be because the instance-based segmentation method produces misclassification, for example, multiple text instances are mistakenly classified as one text instance. We can effectively avoid such situation by dividing a single character instance and then combining it into a line of text. By taking advantages of weakly supervised learning and pixel-level character connection, CRAFT [15] achieves superior performance in horizontal text detection.

4.4.2 Oriented text detection

To evaluate the performance of multi-oriented text detection, we initialize the network with a pre-trained model and then fine-tune it on ICDAR2015 and MSRA-TD500 datasets. The model requires 19K iterations of training. From Table 3, we can see that our approach achieves almost the best results. Compared with the bounding box regression methods [5, 39, 2], our work detects the inclined character instances automatically without handcrafted anchors, and receives a significant increase. Compared with the instance segmentation-based methods [12, 13, 11, 27, 41, 10], ours shows an improvement on all metrics, resulting from extracting text features by applying the pyramid network structure, which enables to take advantage of the global feature information to locate character regions. Besides, high resolution feature map can focus more on the information of small character areas. Both our method and CRAFT [15] segment a single character, but our overall performance is better on both datasets. It is because the character context information facilitates the combination of isolated texts, which improves the detection effect of characters in multiple directions.

4.4.3 Multi-language text detection

To evaluate the performance of multi-language text detection, we initialize the network with a pre-trained model and then fine-tune it on MLT dataset. As shown in Table 4, our method surpasses the previous state-of-the-art method, where R, P and F are 73.93%, 82.13% and 77.81%, respectively. Compared with the instance segmentation based methods [43, 44, 45, 10, 15, 46, 11, 47], our work shows an improvement on all the metrics. Most of text instance segmentation methods split complete words or text lines, but there are gaps between words and characters, which will accumulate errors during the training process continuously. Our method can directly segment individual characters, so that naturally avoid those problems. Therefore, we obtain the highest text segmentation results.

Table 4. Comparison results on MLT.

Methods	R	P	F
He et al. [48]	57.90%	76.70%	66.00%
Border [49]	60.60%	73.90%	66.60%
FOTS [43]	57.50%	79.50%	66.70%
DRRG [44]	61.04%	74.99%	67.31%
LOMO [45]	60.60%	78.80%	68.50%
Corner [18]	55.60%	83.80%	66.80%
PSENet [10]	68.40%	77.01%	72.45%
CRAFT [15]	68.20%	80.60%	73.90%
Pixel-Anchor [46]	59.54%	79.54%	68.10%
DB [42]	63.8%	81.90%	71.70%
SPCNet [11]	68.60%	80.60%	74.10%
Huang et al. [47]	69.80%	80.00%	74.30%
Ours	**73.93%**	**82.13%**	**77.81%**

4.5 Failure cases

Our method may fail to detect scene text when: (1) objects that appear like texts (e.g., telephone pole,

Fig.12. Examples of failure cases.

leaves, window) occupy most of the image; (2) only one word-art character exists and occupies most of the image; (3) character pixels are largely split. Examples of failure cases are given in Fig. 12.

5 Conclusion and Future Work

In this paper, we present an effective multi-oriented scene text detection approach, which can detect individual characters as well as character flows. We separate the characters regardless of their directions through a feature pyramid network. Then the connection relationship of adjacent characters is predicted by a joint network, and thus constructing character flows. In summary, our approach is able to extract character regions without handcrafted anchor boxes, and derive text lines without heuristics grouping rules.

Challenges still remain in the research of natural scene text detection, especially for the distorted, long, and multi-lingual texts. There are several interesting directions we would like to expand upon, such as arbitrary-shaped text detection by mask segmentation, text dataset augmentation, and text recognition.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant 61902435, in part by the National Science and Technology Major Project under Grant 2018AAA0102102, in part by the 111 Project under Grant B18059, and in part by Hunan Provincial Natural Science Foundation of China under Grant 2019JJ50808.

References

[1] Liao M, Shi B, Bai X, Wang X, Liu W. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017, pp. 4161–4167.

[2] Liao M, Shi B, Bai X. Textboxes++: A single-shot oriented scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3676–3690.

[3] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C. SSD: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, 2016, pp. 21–37.

[4] Liu Y, Jin L. 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.

[5] Ma J, Shao W, Ye H, Wang L, Wang H, Zheng Y, Xiangyang. Arbitrary-oriented scene text detection via rotation proposals. *Journal of IEEE Transactions on Multimedia*, 2018, 20(11):3111–3122.

[6] Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, Liang J. EAST: an efficient and accurate scene text detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5551–5560.

[7] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[8] Lyu P, Liao M, Yao C, Wu W, Bai X. Mask text spotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision*, 2018, pp. 67–83.

[9] He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2980–2988.

[10] Li X, Wang W, Hou W, Liu R Z, Lu T, Yang J. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9336–9345.

[11] Xie E, Zang Y, Shao S, Yu G, Yao C, Li G. Scene text detection with supervised pyramid context network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 2019, pp. 9038–9045.

[12] Baoguang S, Bai X, Belongie S. 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.

[13] Deng D, Liu H, Li X, Cai D. Pixellink: Detecting scene text via instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, pp. 6773–6780.

[14] Lin T Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 936–944.

[15] Baek Y, Lee B, Han D, Yun S, Lee H. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9365–9374.

[16] Tian Z, Huang W, He T, He P, Qiao Y. Detecting text in natural image with connectionist text proposal network. In *Proceedings of the European Conference on Computer Vision*, 2016, pp. 56–72.

[17] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Journal of Neural Networks*, 2005, 18(5-6):602–610.

[18] Lyu P, Yao C, Wu W, Yan S, Bai X. 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.

[19] Epshtein B, Ofek E, Wexler Y. Detecting text in natural scenes with stroke width transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2963–2970.

[20] Wu H, Zou B, Zhao Y, Guo J. Scene text detection using adaptive color reduction, adjacent character model and hybrid verification strategy. *Journal of The Visual Computer*, 2017, 33(1):113–126.

[21] HChen, SSTsai, GSchroth, DMChen, RGrzeszczuk, BGirod. Robusttext detection in natural images with edge-enhanced maximally stable extremal regions. In *Proceedings of the IEEE International Conference on Image Processing*, 2011, pp. 2609–2612.

[22] Matas J, Chum O, Urban M, Pajdla T. Robust wide-baseline stereo from maximally stable extremal regions. *Journal of Image and Vision Computing*, 2004, 22(10):761–767.

[23] Yin X C, Yin X, Huang K, Hao H W. Robust text detection in natural scene images. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 2014, 36(5):970–983.

[24] Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, 39(11):2298–2304.

[25] Liao M, Zhu Z, Shi B, Xia G, Bai X. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5909–5918.

[26] Zhang Z, Zhang C, Shen W, Yao C, Liu W, Bai X. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4159–4167.

[27] Long S, Ruan J, Zhang W, He X, Wu W, Yao C. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European Conference on Computer Vision*, 2018, pp. 20–36.

[28] Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991, 13(6):583–598.

[29] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[30] Gupta A, Vedaldi A, Zisserman A. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2315–2324.

[31] Tian S, Pan Y, Huang C, Lu S, Yu K, Tan C L. 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.

[32] Gers F A, Schraudolph N N, Schmidhuber J. Learning precise timing with lstm recurrent networks. *Journal of Machine Learning Research*, 2002, Aug(3):115–143.

[33] Karatzas D, Shafait F, Uchida S, Iwamura M, Bigorda L G, Mestre S R, Mas J, Mota D F, Almazàn J A, Heras L P. ICDAR 2013 robust reading competition. In *Proceedings of International Conference on Document Analysis and Recognition*, 2013, pp. 1484–1493.

[34] Karatzas D, Gomez-Bigorda L, Nicolaou A, Ghosh S, Bagdanov A, Iwamura M, Matas J, Neumann L, Chandrasekhar V R, Lu S, Shafait F, Uchida S, Valveny E. ICDAR 2015 competition on robust reading. In *Proceedings of International Conference on Document Analysis and Recognition*, 2015, pp. 1156–1160.

[35] Yao C, Bai X, Liu W, Ma Y, Tu Z. Detecting texts of arbitrary orientations in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1083?–1090.

[36] Nayef N, Yin F, Bizid I, Choi H, Feng Y, Karatzas D, Luo Z, Pal U, Rigaud C, Chazalon J, Khlif W, Luqman M M, Burie J C, Liu C, Ogier J M. ICDAR 2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *Proceedings of the IEEE Conference on International Conference on Document Analysis and Recognition*, 2017, pp. 1454–1459.

[37] Kingma D P, Ba J. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations*, 2015.

[38] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations*, 2015.

[39] Jiang Y, Zhu X, Wang X, Yang S, Li W, Wang H, Fu P, Luo Z. R2CNN: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[40] He P, Huang W, He T, Zhu Q, Qiao Y, Li X. Single shot text detector with regional attention. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3047–3055.

[41] Tian Z, Shu M, Lyu P, Li R, Zhou C, Shen X, Jia J. Learning shape-aware embedding for scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4234–4243.

[42] Liao M, Wan Z, Yao C, Chen K, Bai X. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, pp. 11474–11481.

[43] Liu X, Liang D, Yan S, Chen D, Qiao Y, Yan J. 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.

[44] Zhang S X, Zhu X, Hou J B, Liu C, Yang C, Wang H, Yin X C. Deep relational reasoning graph network for arbitrary shape text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9699–9708.

[45] Zhang C, Liang B, Huang Z, En M, Han J, Ding E, Ding X. Look more than once: An accurate detector for text of arbitrary shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10552–10561.

[46] Li Y, Yu Y, Li Z, Lin Y, Xu M, Li J, Zhou X. Pixel-anchor: A fast oriented scene text detector with combined networks. *arXiv preprint arXiv:1811.07432*, 2018.

[47] Huang Z, Zhong Z, Sun L, Huo Q. Mask r-cnn with pyramid attention network for scene text detection. In *Proceedings of 2019 IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 764–772.

[48] He W, Zhang X Y, Yin F, Liu C L. Multi-oriented and multi-lingual scene text detection with direct regression. *Journal of IEEE Transactions on Image Processing*, 2018, 27(11):5406–5419.

[49] Xue C, Lu S, Zhan F. Accurate scene text detection through border semantics awareness and bootstrapping. In *Proceedings of the European Conference on Computer Vision*, 2018, pp. 355–372.