2020 19th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES)

LFTD: A Light and Fast Text Detector

1st Guanghao Hu
School of Internet of things engineering
Jiangnan University
Wuxi,China
huguanghao520@gmail.com

2nd Silu Chen
School of Internet of things engineering
Jiangnan University
Wuxi,China
794258229@qq.com

3rd Jun Sun
School of Internet of things engineering
Jiangnan University
Wuxi,China
sunjun_wx@hotmail.com

***Abstract*—Aiming at the problem that the existing text detection technology runs slowly on edge devices and terminal devices with limited storage space and low computing capacity, this paper proposes a method based on A Light and Fast Face Detector for Edge Devices (LFFD) and Connectionist Text Proposal Network. (CTPN) A Light and Fast Text Detector (LFTD). First of all, distinguishing from the current situation of large number of parameters and complex model structure of previous text detectors, this paper is based on the LFFD face detection model. It introduces the characteristics that it does not need to preset a large number of anchor boxes with different sizes and proportions, which makes the detection box network in this paper. The frame is lighter. Secondly, for the problem that the detection range of text and image detection frames is different, this paper improves the label part of LFFD and combines the CTPN method to divide the detection frame of this article into several detection frames according to the font size. The proposed method can theoretically detect Large continuous text scale with 100% coverage. Experiments were performed on popular benchmark datasets (ICDAR11, ICDAR13 and ICDAR15). The proposed method can obtain fast inference speed (NVIDIA TITAN 1080Ti: 131.45 FPS at 640 × 480; NVIDIA 2080Ti at 640 × 480: 136.99 FPS; 640 × 480 Raspberry Pi 3 Type B +: 8.44 FPS), the parameter amount of this model is 8 MB. (*Abstract*)**

Keywords—text detection; convolutional network ;anchor free(*key words****)***

I. INTRODUCTION (*HEADING 1*)

Traditional text detection algorithms rely on manual feature extraction [9,10, 11, 12] to capture the attributes of the text area. Although a large amount of data is not needed for training and it runs fast, it is not reliable due to the influence of light, paper wrinkles, watermarks, and other uncertain factors. Recently, detection algorithms based on convolutional neural networks [13,14,15,16] rely on finding effective feature methods from training data to make great progress in text detection.

However, text detectors are sometimes deployed on some terminal computing devices and edge computing devices, such as mobile phones, scanners, and so on. The salient features of these devices are limited storage space and low computing power. Aiming at this problem, this paper proposes a high-precision and fast text detector. The previous text detection methods [18,14,15,16] have achieved good performance in various benchmark tests in this field. Most of them use VGG16, Resnet50, Dense net and other models as the backbone network. We extensively surveyed the top 5 best performing methods and show their accuracy in Table I. Although these methods have similar accuracy, they all have complex model structures and a large number of training parameters, resulting in longer detection and recognition time on devices without a GPU. It is difficult and unrealistic to further improve accuracy by using backbone networks with more complex structures and more parameters. In our opinion, in order to better apply to different scenarios, it is necessary to balance model accuracy and algorithm complexity.

TABLE I. COMPARISON WITH OTHER RESULTS ON ICDAR 2015

Method	IC15		
	R	***P***	***H***
EAA[30]	83	84	83
PSENet[2]	85.2	89.3	87.2
PixelLink[26]	82.0	85.5	83.7
FOTS[28]	82.0	88.8	85.3
CRAFT[5]	84.3	89.8	86.9

In response to the urgent needs of Internet food sales business license detection and recognition, and the existing models have slow recognition speed, and the license watermark has a large impact on the recognition effect. This paper presents a light weight fast text detector LFTD.

II. RELATED WORK

Scene text detection and recognition has always been a research hotspot in the field of computer vision. There have been many effective methods [2,3,1,4,7,15,16] published. This section will focus on work related to this algorithm.

Traditional methods rely on the characteristics of manual extraction. Stroke Width Transform (SWT) [18] and Maximally Stable Extremal Regions (MSER) [19,20] use edge detection or extreme region extraction to find candidate characters. FAS Text [17] is a fast text detection system, which is improved on the basis of a fast key point detector for stroke extraction. However, these methods lag behind those based on deep neural networks in accuracy and adaptability, especially when dealing with challenging scenes, such as low resolution and geometric distortion.

Text detection algorithm based on target detection. Unlike general object detection, text has the characteristics of uncertain shape and large length and width. To solve this problem, Textbox [22] modified the aspect ratio of the convolution kernel and anchor to effectively capture various text shapes. DMPNet [21] attempts to further solve this problem by merging quadrilateral sliding windows. In recent years, the rotation-supported regression detector (RSDD) [23] takes full advantage of rotation invariance through active rotation conv. However, in these methods, the limitation of detecting regular text of quadrilaterals has not been shaken out, and text in real scenes may There are various shapes.

2473-3636/20/$31.00 ©2020 IEEE
DOI 10.1109/DCABES50732.2020.00073

Another commonly used method is based on segmentation detection algorithms, segmentation-based Multi-scale FCN [24], Holistic-prediction [25] and Pixel Link [26] .Recently, some new segmentation-based text detection algorithms have appeared for detecting irregular text. LOMO [1] solves irregularly shaped text detection by using Iterative Refinement Module and Shape Expression Module. LSAE [m] uses embedded clustering to predict text instances of arbitrary shapes. PSE Net [2] uses a forward progressive scaling method to distinguish adjacent text instances. PMTD [3] uses a method similar to Mask R-CNN. It uses the form of pixel-wise regression under the supervision of Pyramid Label to obtain a soft text mask with more information in each text area.

Text detection algorithm combining detection and segmentation. The method based on regression generally has the problem of insensitive text size, and the method based on segmentation is not effective for small text targets. Then combining the advantages of regression and segmentation can effectively solve their problems. SSTD [27] uses an attention mechanism that enhances text-related areas by reducing background interference on feature layers. Pixel Anchor [4] efficiently integrates pixel-level image semantic segmentation and anchor-based detection regression methods through feature sharing, and uses pixel-level image semantic segmentation results as an attention mechanism for supervising anchor detection regression , While effectively guaranteeing the text detection rate, the accuracy of text detection is improved.

Existing text detection models can achieve better detection results, but in order to improve the accuracy and recall of these models, they all have problems with large parameters and complex model structures, which leads to their complexity and storage space. With high requirements, they cannot effectively balance the detection effect and algorithm complexity, and thus cannot be used on some edge computing devices. Therefore, there is still huge room for improvement in text detection on low-powered devices. Exploring lightweight text detection on the premise of ensuring accuracy.

This work was inspired by LFFD[6], which uses a lightweight and fast face detection algorithm to solve the problem of face detection on low computing power devices. LFFD is designed for face detection. According to the characteristics of face size, the detection frame is defined as a square. However, the shape of the text is more variable, so you cannot directly use LFFD for text detection. In addition, we refer to the annotation method of CTPN[7] to solve the problem of indefinite text aspect ratio.

III. Light and Fast Text Detector

In this section, we first review the concept of RF and its relationship to text detection in Section A. Then, Section B gives the network design details. How to generate labels is described in Section C. Section D explains how Loss is calculated.

A. *Rethinking receptive fields in the context of text detection*

Text detection is a branch of general object detection, and it has its own unique characteristics. First of all, for short texts, we only need the receptive field related to the size of the text to determine the position of the text. Secondly, for long texts, sufficient receptive fields are needed to cover the entire text area. However, if large receptive field background information is used, it will interfere with the detection to a certain extent and affect the detection effect. Step size, the text cannot be covered with matching receptive fields for texts with different aspect ratios. However, when the human eye looks at the long text, the entire text is not completely seen, but the text area that is currently being read is annotated. Therefore, the text is detected by stitching adjacent text to form a long text. Large receptive fields cover the entire area.

The first-stage detection algorithm mostly uses a predefined box anchor box. In order to detect different objects, the anchor frame has a variety of aspect ratios and sizes. These anchors always have a lot of redundancy when they are defined. In text detection, it is reasonable to use a larger aspect ratio anchor box, because most text is rectangular, which is also mentioned in [Textbox]. In face detection applications, LFFD analyzes the effect of effective receptive fields on detection, and designs anchors for different face sizes, but it is not suitable for rectangles and irregular quadrilaterals. A similar ingenuity is also designed in EAST Matching strategy, but its effect is not good for long texts, and it requires a large receptive field as mentioned above, so we refer to CTPN to block and form a small rectangle, then the shape of RFs is also rectangular. Different from CTPN, instead of using a fixed small rectangle with a wideband of 16 pixels, we use different proportions of the feature map to reduce the size of the text.

The width of the box. As for the matching strategy, this method uses a straightforward and concise approach-matching RF to ground truth-box instead of threshold IOU if and only if the center is in ground truth-box.

B. *Network structure*

Based on the above analysis, we can design a specialized backbone network for text detection. The size of the receptive field and the pace determine the setting of the Loss branch. The size of the receptive field ensures that the features of the text are robust and distinguishable, while the stride ensures 100% coverage. A schematic diagram of the network proposed by the overall architecture is shown in Figure 2. This method can detect text with a width greater than 10 pixels. It can be seen that the backbone is a single-phase model consisting of four parts. For specific information about the Loss branch, see Table II.

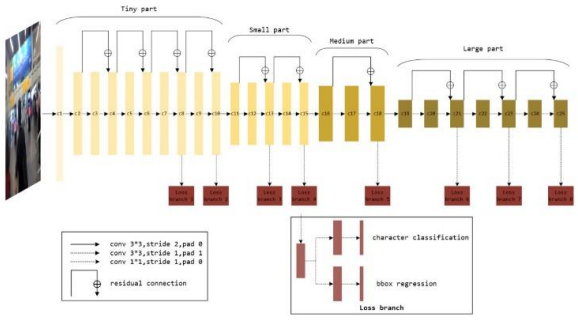

Figure. 2 Network structure

The feature extraction section has 10 convolutional layers. The first two down sampling steps are 2. Therefore, the RFs at other convolutional layers in this section are 4 steps. A key principle is: take input samples as fast as possible, while maintaining 100% coverage text. There are two branches in this section. Loss branch 1 comes from c8. For text height 10-15, its RF size is 55. Similarly, for text with a height of 15-20

pixels, the second Loss branch is responsible, and its receptive field is 71. Obviously, these two branches can ensure that small text can be detected in the center of our receptive field, thereby achieving 100% coverage. As we discussed in Section 3.1, microtext requires more contextual information and smaller receptive fields than large text. Therefore, the RFs we use are much larger than the average text score. The ratios of RFs and average text for branches 1 and 2 are 4.4 and 4.0, respectively. In Table 2, this ratio gradually decreases from 4.4 to 1.3 because larger texts require less contextual information. In the backbone, the kernel size of all convolutional layers is 3 * 3. However, the kernel size of the convolutional layer in the branch is 1 * 1, which does not change the size of the RFs. Each branch has two sub-branches, one for text classification and the other for box regression.

TABLE II. DETAILS ABOUT NETWORK DEFINITION

	No.ilter	*branch*	*Location*	*RF size*	*RF stride*
Tiny part	64	Loss branch1	c8	55	4
		Loss branch2	c10	71	4
Small part	64	Loss branch3	c13	111	8
		Loss branch4	c15	143	8
Mediu m part	128	Loss branch5	c18	223	16
Large part	128	Loss branch6	c21	383	32
		Loss branch7	c23	511	32
		Loss branch8	c25	639	32

This method can cover all text from 10 pixels to 560 pixels in height. The entire backbone network consists only of conv 3 × 3, conv 1 × 1, ReLu, and residual connections that are deeply optimized by each neural network framework. They are also widely used in other models. Although BN [8] has become the standard configuration of many models, due to its slow inference speed, we have not adopted BN [8] as a network component in the model. It has been shown in LFFD that the use of BN has little effect on the detection effect of the model. As shown in Figure 2, in each part, in order to facilitate the training of deeper networks, the residual connections are placed side by side. The number of filters for all convolutional layers in the first two parts is 64. We did not add a filter, because the first two parts have larger feature maps, which are more expensive to calculate. However, the number of filters in the last two sections can be increased to 128, without much extra effort. See Table 2 for more details.

C. Label Generation

1) Score Map Generation

Since LFCD is based on horizontal text detection, we only consider horizontal text and text with a small tilt angle. Figure * shows the generated score map area, which is the reduction of the original label box.

For a quadrilateral $Q = \{p_i | i < \{1,2,3,4\}\}, p_i = \{x_i, y_i\}$is a vertex of the quadrilateral, and the four vertices are arranged clockwise. To reduce Q, we first calculate each side length r_i

$$r_i = \min\left(D(p_i, p_{(imod4)+1}), \left(p_i, p_{((i+2)mod4+1)}\right)\right) \quad (1)$$

Where $D(p_i, p_j)$represents the Euclidean distance from p_i to p_j.

We first shrink the two long sides of the quad, and then the two short sides. For each pair of edges, we determine the "long" edge by comparing the average of their lengths. For each edge $(p_i, p_{(imod4)+1})$, we move its two endpoints inward by 0.3r to reduce it, and r is the length of the short edge (this is different from EAST).

2) Geometry Map Generation

As discussed in Section 3.1, we generate different score maps for different sizes of text, and then divide into different detection branches and divide them into boxes of different widths. The box generation process is shown in Figure *.

We first generate a reduced quadrilateral, and then for each point in the quadrilateral, the ground truth is defined as:

$$T = \frac{RF_y - b_y^t}{RF_s/2} \quad (2)$$

$$B = \frac{RF_y - b_y^b}{RF_s/2} \quad (3)$$

RF_y represents the center point of the y-axis of RF, b_y^t represents the top y coordinate, b_y^b represents the bottom y coordinate, and RFs represents the size of the branch RF. The calculation results are put into 2 channels of Box ground truth.

The width of the box and the output size of the text segmentation box of different sizes are consistent with the reduction ratio of the original image, as shown in Table II.

D. Model Outputs and Loss Functions

Each loss branch contains two sub-branches for text classification and Box regression. The classification branch loss function uses two types of cross-entropy loss functions. Box regression For the regression of the upper and lower distances of the RF anchor, the height of the text is continuous. If the height of the text is close to the interval we set, it is easy to be classified into another type of branch. Therefore, we set an interval gray. In grayscale intervals, the text is ignored by the corresponding branch. For example, the detection height in branch 3 is 20-40 pixels, so the corresponding upper and lower gray interval is [18,20] and [40,44]. For Box regression, use the L2 loss function directly. Box regression L2 loss is calculated only at positive RF anchor points, other points are ignored. In the final loss function, both losses have the same weight.

Loss can be expressed by the formula:

$$L = \sum_{i=0}^{8}(L_{si} + \lambda_g L_{gi}) \quad (4)$$

$$L_{gi} = \sqrt{(T_{ti} - T_{li})^2} + \sqrt{(B_{ti} - B_{li})^2} \quad (5)$$

Among them, L_{si} and L_{gi} represent the loss of the i-th branch classification and box regression, respectively, the weight coefficient between the two losses of λ_g weight. In our experiments, we set λ_{gi} . T_{ti}, B_{ti} represents the distance to the upper and lower borders, and the calculation method is 3.4, and T_{li}, B_{li} represents the network output.

IV. Experimental Results and Discussions

Our LFTD was evaluated on three standard benchmarks: ICDAR 2015, ICDAR 2013, ICDAR 2011 Total-Text. ICDAR 2015 includes 1,500 images collected using Google Glass. This training set has 1,000 images, and the remaining 500 are used for estimation. This dataset is challenging due to the multi-directional and very small text examples. Total-Text contains 1555 pictures with different text types, including horizontal, multi-directional, and curved text examples. There are 1,255 and 300 images for training segmentation and test segmentation, respectively.

Table III shows the results and operating speed of different models on the ICDAR 2015 dataset. Figure 3 shows the results of LFTD on invoice inspection and business license inspection.

TABLE III. Comparison with other results on ICDAR 2015

Method	IC15			
	R	***P***	***H***	***FPS***
EAA[30]	83	84	83	3.5
PSENet[2]	85.2	89.3	87.2	1.6
PixelLink[26]	82.0	85.5	83.7	8.8
FOTS[28]	82.0	88.8	85.3	7.1
CRAFT[5]	84.3	89.8	86.9	7.6
LFTD	75.2	52.4	62.3	121.4

Figure. 3 Test results on the left invoice, test results on the right business license

V. Conclution

This paper aims at the problem that the existing text detection algorithms run slowly on some terminal computing devices and edge computing devices. A new lightweight fast text detection method is proposed. This method improves the label part of the text detection model and realizes the theoretical detection of a large range of continuous text scales with 100% coverage. A large number of experiments have shown that LFTD is more lightweight and has fewer parameters when the accuracy rate is similar to the existing text detection models.

Possible research directions in the future, adding LFTD to the recognition branch and using CTC Loss, which will be combined with text recognition to achieve end-to-end text recognition.

Acknowledgment

The work described in this paper was fully supported by a grant from the National Key R&D Program of China (No. 2018YFC1603303).

References

[1] Zhang, C.; Liang, B.; Huang, Z.; En, M.; Han, J.; Ding,E.; and Ding, X. Look More Than Once: An Accurate Detector for Text of Arbitrary Shapes. In CVPR,2019.

[2] X. Li, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang.Shape robust text detection with progressive scale expansion network. arXiv preprint arXiv:1806.02559, 2018.

[3] J. Liu, X. Liu, J. Sheng, D. Liang, X. Li, and Q. Liu, "Pyramid mask text detector," arXiv preprint arXiv:1903.11800, 2019.

[4] Y. Li, Y. Yu, Z. Li, Y. Lin, M. Xu, J. Li, and X. Zhou. Pixelanchor: A fast oriented scene text detector with combined networks. arXiv preprint arXiv:1811.07432, 2018.

[5] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in CVPR, 2019.

[6] Y. He, D. Xu, L. Wu, M. Jian, S. Xiang and C. Pan. LFFD: A Light and Fast Face Detector for Edge Devices, arXiv:1904.10633, 2019.

[7] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network.In European Conference on Computer Vision, pages 56–72.Springer, 2016.

[8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.arXiv:1502.03167, 2015.

[9] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In Proc. of ACCV, 2010.

[10] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In Proc. of CVPR,2012.

[11] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In Proc. of ICCV, 2013.

[12] X. C. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. IEEE Trans. on PAMI, 36(5):970–983, 2014.

[13] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In Proc. of ECCV, 2014.

[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman.Reading Text in the Wild with Convolutional Neural Networks. International Journal of Computer Vision, 116(1):1–20, jan 2016.

[15] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. arXiv preprint arXiv:1604.06646, 2016.

[16] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai.Multi-oriented text detection with fully convolutional networks. In Proc. of CVPR, 2015.

[17] M. Busta, L. Neumann, and J. Matas. Fastext: Efficient unconstrained scene text detector. In Proc. of ICCV, 2015.

[18] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In Proc. of CVPR, 2010.

[19] L. Neumann and J. Matas. A method for text localization and recognition in real-world images. In Proc. of ACCV, 2010.

[20] L. Neumann and J. Matas. Real-time scene text localization and recognition. In Proc. of CVPR, 2012.

[21] Y. Liu and L. Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In CVPR, pages 3454–3461, 2017.

[22] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In AAAI,pages 4161–4167, 2017.

[23] M. Liao, Z. Zhu, B. Shi, G.-s. Xia, and X. Bai. Rotationsensitive regression for oriented scene text detection. In CVPR, pages 5909–5918, 2018.

[24] D. He, X. Yang, C. Liang, Z. Zhou, G. Alexander, I. Ororbia,D. Kifer, and C. L. Giles. Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In CVPR, pages 474–483, 2017.

[25] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao.Scene text detection via holistic, multi-channel prediction. arXiv preprint arXiv:1606.09002, 2016.

[26] D. Deng, H. Liu, X. Li, and D. Cai. Pixellink: Detecting scene text via instance segmentation. In AAAI, 2018.

[27] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In ICCV, volume 6,2017

[28] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan. Fots: Fast oriented text spotting with a unified network. In CVPR, pages 5676–5685, 2018