GAPF: Curve Text Detection based on Generative Adversarial Networks and Pixel Fluctuations

Jun Yang[1], Zhaogong Zhang[1] and Xuexia Wang[2]
[1]*School of Computer Science and Technology, Heilongjiang University, Harbin, China*
[2]*Department of Mathematics, University of North Texas, Denton, U.S.A.*

Keywords: Deep Learning, Pattern Recognition, Curved Text Detection, Generative Adversarial Networks, Pixel Fluctuations Numbers.

Abstract: Scene text detection has witnessed rapid progress especially with the recent development of convolutional neural networks. However, curved text detection is still a difficult problem that has not been addressed sufficiently. Presently, the most advanced method is based on segmentation to detect curved text. However, most segmentation algorithms based on convolutional neural networks have the problem of inaccurate segmentation results. In order to improve the effect of image segmentation, we propose a semantic segmentation network model based on generative adversarial networks and pixel fluctuations, denoted as GAPF; which is able to effectively improve the accuracy of text segmentation. The model consists of two parts: the generative model and the discriminative model. The main function of the generative model is to generate semantic segmentation graph, and then the discriminative model and generative model perform adversarial learning, which optimize the generative model to make the generated image closer to the ground truth. In this paper, the information about pixel fluctuations numbers is input into the generative network as the segmentation condition to enhance the invariance of translation and rotation. Finally, a text boundary generation algorithm for text is designed, and the final detection result is obtained from the segmentation result. Experimental results on CTW1500, Total-Text, ICDAR 2015 and MSRA-TD500 demonstrate the effectiveness of our work.

1 INTRODUCTION

Scene text is one of the most common objects in images, which usually appears on license plates, product packages, billboards, and carries rich semantic information. Scene text detection is to identify text regions of the given scene text images, which is also an important prerequisite for many multimedia tasks, such as image understanding and video analysis. Compared to common objects, scene text is born with multiple orientations, large aspect ratios, arbitrary shapes or layouts, and complex backgrounds; which creates difficulties for detection and recognition. With the development of convolutional neural networks(Kaiming,2016), there have been many attempts on text detection in natural scenes and great progress has been achieved in recent years. The early attempts to detect text are with annotations of horizontal texts and the approaches for arbitrary-oriented scene text detection have been also proposed. However, in real-world scenarios, there are many text regions with irregular shapes, such as curve words or logos. It is very challenging to detect these regions with different shapes.

The detection of curve text and arbitrary shape text is almost always based on semantic segmentation, and the final detection results are obtained from the segmentation graph through the post-processing algorithm. From this perspective, accurate segmentation results are an important prerequisite for improving the accuracy of text detection. However, most semantic segmentation algorithms based on convolutional neural networks have inaccurate segmentation. Generative adversarial networks(GAN)(Goodfellow,2014) have been proven to effectively improve network performance. Based on the idea of GAN, this paper designs a semantic segmentation network model based on generative adversarial learning to generate accurate segmentation results for text images. Finally, the final text detection results are obtained by the text boundary generation algorithm.

The contributions of this paper can be summarized as follows.

Yang, J., Zhang, Z. and Wang, X.
GAPF: Curve Text Detection based on Generative Adversarial Networks and Pixel Fluctuations.
DOI: 10.5220/0010298905450552
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 545-552
ISBN: 978-989-758-488-6

- The generative adversarial network is introduced into the field of text detection. Combining with the idea of conditional generative adversarial networks, the original image is used as the input of the generator to generate the desired semantic segmentation results. By training the generator so that the discriminator cannot distinguish between the image generated by the generator and the results of manual annotation, the generator can generate satisfactory image segmentation results.
- Information on the pixel fluctuations numbers is proposed. The pixel fluctuations numbers as a condition are input of the generator network. This paper first calculates the pixel fluctuations numbers of each pixel in its pixel interval, then stacks the calculation result with the original picture into the generator.
- Lastly, a text boundary generation algorithm is designed. The bounding box of the text is generated from the segmentation result, and the final output is obtained.

2 RELATED WORK

Scene text detection based on deep learning methods has achieved remarkable results over the past few years. Modern methods are mostly based on deep neural networks, which can be coarsely classified into two categories: regression-based methods and segmentation-based methods.

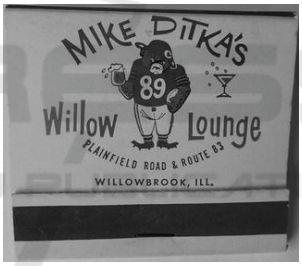
Regression-based methods mainly draw inspiration from general object detection frameworks. Based on Faster-RCNN(Ren,2017), Ma et al. (Ma,2018) devised Rotation Region Proposal Networks (RRPN) to detect arbitrary Oriented text in natural images. Textboxes (Liao,2017)adopted SSD (Liu,2016) and added long default boxes and filters to handle the significant variation of aspect ratios of text instances. EAST (Zhou,2017) uses a single neural network to directly predict score map, rotation angle and text boxes for each pixel. Tian et al. introduced CTPN (Zhi,2016) using LSTM (Alex,2005) to link several text proposals. These methods can handle multi-directional text but may have shortcomings when dealing with curved text, which widely exist in real-world scenarios.

Segmentation-based methods are mainly inspired by fully convolutional networks (FCN) (Jonathan,2015). Zhang et al. (Zheng,2016) first adopted FCN to extract text blocks and detect character candidates from those text blocks via MSER. PixelLink (Dan,2018) separated texts lying close to each other by predicting pixel connections between different text. The framework of TextSnake (Long,2018) considered a text instance as a sequence of ordered disks. To deal with the problem of separation of the close text instances, Li et al. (Li,2019) designed the PSENet, a progressive scale algorithm to gradually expand the predefined kernels. The methods reviewed above have made significant improvements on various benchmarks in this field. However, most segmentation algorithms fail to produce accurate segmentation results, resulting in low accuracy of the final detection results.

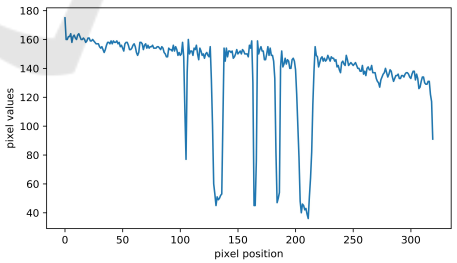
3 PROPOSED METHOD

3.1 Pixel Fluctuations of Grayscale Image

We know that images are made up of pixels. For example, Figure 1(a) is a grayscale image. If we were to draw the pixel values of the twentieth line into a curve, we would get the following Figure 1(b).



(a)



(b)

Figure 1: Visualization of pixel fluctuations. (a) is the grayscale image of the original image. (b) is the fluctuation curve of the twentieth line of the grayscale image.

It can be seen that the curve fluctuates up and down continuously, some areas have relatively small fluctuations, and some areas suddenly show large fluctuations. By comparing the images, we can see that the curve fluctuates wildly in the text line area, while the fluctuation is relatively smooth in the background area. This shows that the fluctuation is closely

related to the image. Large fluctuations represent sharp changes in color; small fluctuations represent smooth color transitions. But then the question becomes how to quantify this fluctuation. We propose to use pixel fluctuations numbers of the image, which can find out whether there is a fluctuation in the pixel interval. First, we need to define the total variation.

3.1.1 Total Variation

Let $f(x)$ be the function defined on [a,b], with a partition, $D: a = x_0 < x_1 < \cdots < x_n = b$. Let:

$$V_a^b(f,D) = \sum_{i=1}^{n} |f(x_i) - f(x_{i-1})| \tag{1}$$

Where $V_a^b(f,D)$ the variation of $f(x)$ with respect to partition D. If $\exists M > 0$, to divide everything D, $V_a^b(f,D) \leq M$. Then, $f(x)$ is called the bounded variation function on [a, b]. Denote:

$$V_a^b(f) = supV_a^b(f,D) \tag{2}$$

Let $V_a^b(f)$ be the total variation of $f(x)$ on [a, b], the sup here is the upper bound which is the smallest upper bound. Based on the total variation, we will define pixel fluctuations numbers.

3.1.2 Pixel Fluctuations Numbers

Let $f(x)$ be a function defined on [a, b] when the distribution is constant:

$$V_n = \frac{V_a^b(f)}{|P_{\max} - P_{\min}|} \tag{3}$$

Where V_n is called pixel fluctuations numbers of $f(x)$. $V_a^b(f)$ is the total variation of $f(x)$ on the selected pixel interval, $P_{\max}$ is the maximum pixel value of $f(x)$ on the selected pixel interval, $P_{\min}$ is the minimum pixel value of $f(x)$ on the selected pixel interval.

In this paper, the pixel interval we choose is the length of 20 pixels. A pixel interval of a pixel includes the 10 pixels to the left and right of that pixel. The pixel fluctuations numbers of that pixel are calculated within this interval. If there are less than 10 pixels to the left or right of a pixel, we supplement it with the pixel value of that pixel, so that the pixel interval is also 20 pixels long. Pixel fluctuations numbers of each pixel in its pixel interval is calculated line by line, in order to obtain a pixel fluctuation map of the same size as the image, which is sent into the generative network together with the original image.

Pixel fluctuations numbers are calculated which is the invariant under the affine transformation group, that is, the translation invariance and the invariance of the rotation. By adding this feature, more information can be provided to the image. We provide a brief proof:

The affine transformation group contains two basic transformations:

(1)Scale transformation: $T_1(f) = af(x), \quad a \neq 0$

(2)Translation transformation: $T_2(f) = f(x) + b$

It is easy to know that $T1 \circ T1, T1 \circ T2, T2 \circ T1, T2 \circ T2$ are still affine transformation. We only need to prove that the number of fluctuations remains the same for the two basic transformations.

Let the existence group $T_1(f) = af(x)$, bring it into the total variation formula:

$$\begin{aligned} V_n(T_1(f)) &= \frac{V_a^b(T_1(f(x)))}{\left|P_{\max(T_1(f(x)))} - P_{\min(T_1(f(x)))}\right|} \\ &= \frac{supV_a^b(af(x),D)}{\left|P_{\max(a(f(x)))} - P_{\min(a(f(x)))}\right|} \end{aligned} \tag{4}$$

When $a > 0$, there are:

$$\begin{aligned} V_n(T_1(f)) &= \frac{asupV_a^b(f(x),D)}{a\left(P_{\max(f(x))} - P_{\min(f(x))}\right)} \\ &= \frac{supV_a^b(f(x),D)}{P_{\max(f(x))} - P_{\min(f(x))}} = V_n(f(x)) \end{aligned} \tag{5}$$

When $a < 0$, there are:

$$\begin{aligned} V_n(T_1(f)) &= \frac{-asupV_a^b(f(x),D)}{-a\left(P_{\max(f(x))} - P_{\min(f(x))}\right)} \\ &= \frac{supV_a^b(f(x),D)}{P_{\max(f(x))} - P_{\min(f(x))}} = V_n(f(x)) \end{aligned} \tag{6}$$

Similarly, when $T_2(f) = f(x) + b$, it is easy to get: $V_n(T_2(f)) = V_n(f(x))$.

3.2 Generative Adversarial Networks

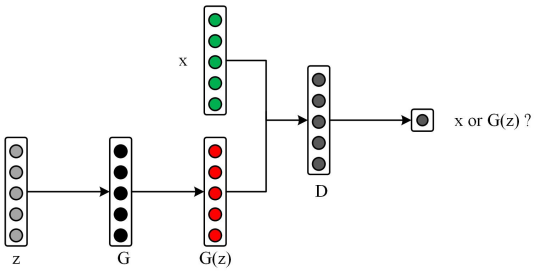


Figure 2: The training process of the generative adversarial network.

The Generative Adversarial Network(GAN) was proposed by Goodfellow et al. in 2014. The idea of GAN is derived from game theory. It proposes the idea of designing two different networks, one of which is a generative network to generate a target with a specific

meaning image; another network is used as a discriminative network to determine whether the input image is a network-generated image or a real image. In this way, the two networks produce adversarial training.

The discriminative network is a binary classification network: if the input image is a real image, its output is 1; if it is a target image generated by the generation network, its output is 0. During training, the generative network constantly adjusts its parameters to make the generated image as similar to the real image as possible, which results in the discriminative network not being able to correctly distinguish whether it is the generated image or the real image. In contrast, the purpose of the discriminative network is to distinguish between the two as much as possible. In this way, after a long period of training, the GAN finally reaches the state of Nash equilibrium, and the results generated by the generative network can achieve the real effect of falsehood. Assume that the network has completed the training. The generated target images (such as segmentation results, coloring results, etc.) can be used as correct results.

Its network training process is shown in Figure 2. Differentiable functions G and D in the figure represent generative networks and discriminative networks, respectively, and z and x represent random variables and real data, respectively. G(z) refers to the data generated by the generative network.

3.3 Methodology

3.3.1 Pipeline

The pipeline of the proposed GAPF is illustrated in Figure 3. In the stage of pre-processing the pictures, we generate the pixel fluctuations map of each image. Two entries are defined: the original image of the input image and the pixel fluctuations map of the original image. At the same time, we also add a picture entry for discriminating the network to load pixel fluctuations information.

In the figure, G represents the generative network we use to generate the image semantic segmentation results, x represents the real picture input to the generator, and f represents the pixel fluctuations map of the corresponding image. D represents the discriminative network in the generative adversarial network framework. The input samples of the discriminator include two types:

(1) A sample is composed of the original image x, the pixel fluctuations map f of the original image, and the semantic segmentation result G (x) generated by the generator, we call it a negative sample pair. This is shown in the left part of Figure 3.

(2) A sample is composed of the original image x, the pixel fluctuations map f of the original image, and the artificially labeled semantic segmentation result y of the image, which is called a positive sample pair. As shown on the right side of Figure 3, we define the loss function as follows:

$$Loss = E_{x+f,y}[\log D(x+f,y)] + E_{x+f,G(x)}[\log (1-D(x+f,G(x))] \quad (7)$$

The training process of D is to maximize the accuracy, that is, to minimize the loss function. The output of positive samples tends to 1 and the output of negative samples tends to 0, so that the overall loss of D tends to 0. The purpose of G training is to minimize the accuracy of D, that is, the output of negative samples approaches 1.

3.3.2 Generative Network

The whole generative network is shown in Figure 4. Inspired by FPN (Lin,2017) and U-net (Ronneberger,2015), we adopt a scheme that gradually merges features from different levels of the stem network. We choose ResNet-50 (He,2016) as our stem network for the sake of direct and fair comparison with other methods.

As for the feature merging network, several stages are stacked sequentially, each consisting of a merging

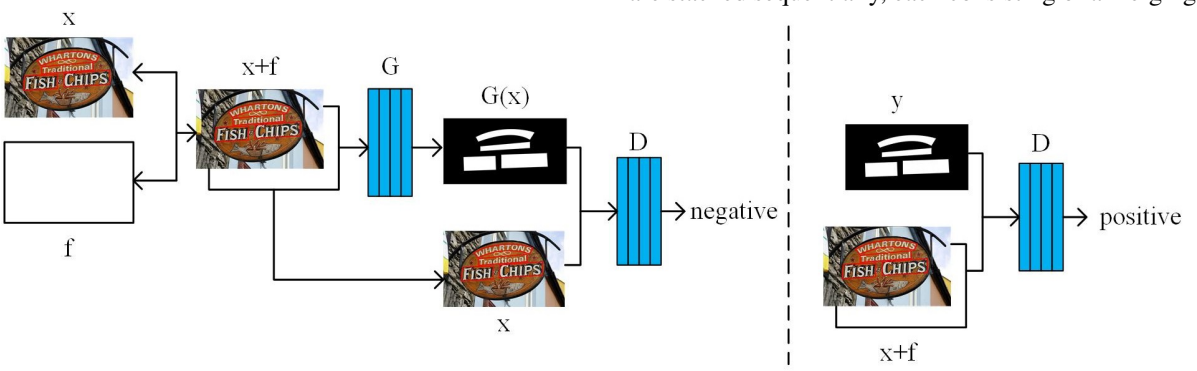


Figure 3: Illustration of our overall pipeline.

unit that takes feature maps from the last stage and corresponding stem network layer. Merging unit is defined by the following equations:

$$h_1 = f_1 \tag{8}$$

$$h_i = conv_{3\times3}\left(conv_{1\times1}[f_i; UpSampling_{\times2}(h_{i-1})]\right), \quad for\ i \geq 2 \tag{9}$$

where f_i denotes the feature maps of the i-th stage in the stem network and h_i is the feature maps of the corresponding merging units. In our experiments, upsampling is implemented as a deconvolutional layer as proposed in (Zeiler,2010). After the merging, we obtain a feature map whose size is 1/2 of the input images. We apply an additional upsampling layer and two convolutional layers to produce dense predictions:

$$h_{final} = UpSampling_{\times2}(h_5) \tag{10}$$

$$P = conv_{1\times1}(conv_{3\times3}(h_{final})) \tag{11}$$

where $P \in R^{h\times w\times2}$, these two channels represent text or non-text areas. As a result of the additional upsampling layer, P has the same size as the input image. The final predictions are obtained by taking the softmax algorithm.

3.3.3 Discriminative Network

Compared to the generative network, the discriminative network is simpler. The purpose of the discriminative network is to distinguish between positive and negative sample pairs, which is a binary classification problem. For the discriminator, we use a simple convolutional network structure, as shown in Figure 5:

The discriminator combines two inputs: the real image and the tag image corresponding to the real image (the generator generates the image or manually annotated image). Using convolution to continuously extract high-dimensional features from the input. Finally, sigmoid is used to classify the results.

3.4 Label Generation

Figure 6 shows the label generation process. Figure 6(a) is an example of manually annotated text, shown in a red border-box. The sample points in the solid red box are defined as positive samples with a value of 1, while the other sample points are defined as negative samples with a value of 0. The final result is shown in Figure 6(b).

(a) (b)

Figure 6: The illustration of label generation. (a) shows the original text instances. (b) shows the segmentation masks.

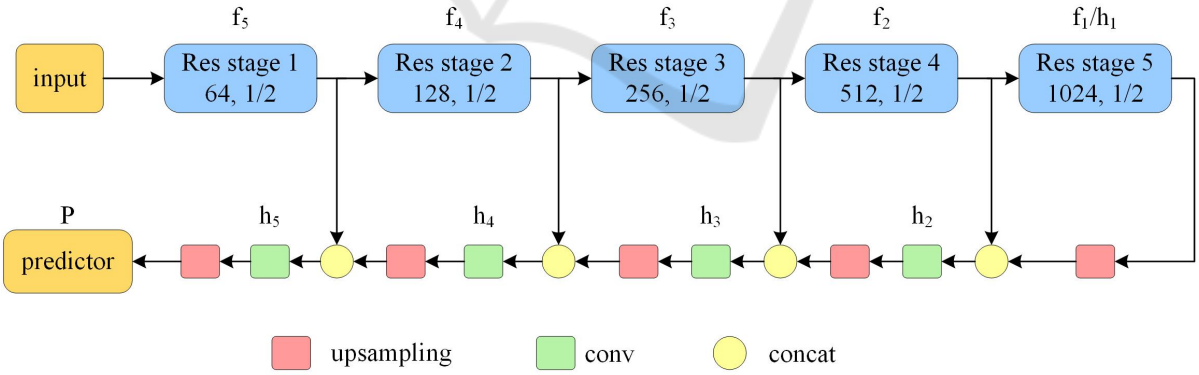


Figure 4: The architecture of generative Network.

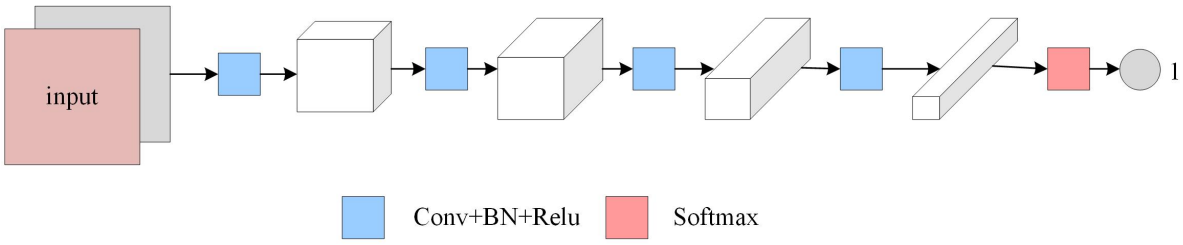


Figure 5: The architecture of discriminative Network.

3.5 Text Boundary Generation

After adversarial learning of the generative network and the discriminative network, the generative network generates accurate segmentation results. And our goal is to get the bounding box that surrounds the text area. The specific implementation details are shown in Algorithm 1. We chose n positive pixels in each text region and use the principal curve (Trevor,1989) to regress the curve centerline. Seven points are chosen from the centerline. For each pair of points that are adjacent in the centerline, we use the center point of two points as a rectangle center and generate a circumscribed rectangle of the area where are the text region between the pare center points. We will get the boundary of the text region by repeating these steps.

Algorithm 1: Text Boundary Generation.

Input: Segmentation result set S for a given image
Output: Text boundary set B for the image

$B = \{\}$
for each segmentation result $s \in S$ **do**
  Boundary $b = [\ ]$
  choose n positive pixels
  use principal curve to find curve center line
  choose 7 point p_i ($i = 0 \cdots 6$) to represent center line
  for $i \in [1,6]$ **do**
    generate circumscribed rectangle of the area between b_i and b_{i+1}
    regard rectangle points as boundary points
    insert the left two points into b
  end for
  insert the b into B
end for
return B

4 EXPERIMENT

4.1 Datasets

CTW1500 (Liu,2017) consists of 1000 training and 500 testing images. Each text instance annotation is a polygon with 14 vertexes to define the text region at the level of the text line. The text instances include both inclined texts as well as the horizontal texts.

Total-Text(Chee,2017) is a newly-released dataset for curve text detection. Horizontal, multi-Oriented and curve text instances are contained in Total-Text. The benchmark consists of 1255 training images and 300 testing images. The images are annotated at the level of the word by a polygon with 2N vertices (N $\in 2, ..., 15$).

MSRA-TD500 (Karatzas,2015) contains 500 natural images, which are split into 300 training images and 200 testing images, collected both indoors and outdoors using a pocket camera. The images contain English and Chinese scripts. Text regions are annotated by rotated rectangles.

ICDAR2015 (Yao,2012) was introduced in the ICDAR 2015 Robust Reading Competition for incidental scene text detection, consisting of 1000 training images and 500 testing images, both with texts in English. The annotations are at the word level using quadrilateral boxes.

4.2 Implementation Details

The generator and discriminator perform adversarial training. First, the generator is fixed to train the discriminator, then the discriminator is fixed to train the generator, and then the loop training is continued. Through adversarial learning, the capabilities of the generator's discriminator have been enhanced. In the end, the discriminator cannot distinguish between the segmentation result generated by the generator and the real segmentation result. At this point, the training is over. When a new image is input, the segmentation result generated by the generator can be used as the correct text segmentation result.

Our method is implemented in Pytorch1.1. Specifically, our network is trained with stochastic gradient descent (SGD) for 100K iterations with the initial learning rate being 0.0001 and a minibatch of 6 images. Weight decay and momentum are set as 0.0005 and 0.9. In terms of initial assignment, gaussian distribution with a mean of 0 and a variance of 0.01 is used for random initialization.

4.3 Results and Comparison

To test the ability of curve text detection, we evaluate our method on CTW1500 and Total-Text, which mainly contains the curve texts.

Table 1: Quantitative results of different methods evaluated on CTW1500.

Method	Precision	Recall	F-measure
CTD (Liu,2017)	74.3	65.2	69.5
CTD+TLOC (Liu,2017)	77.4	69.8	73.4
SLPR (Zhu,2018)	80.1	70.1	74.8
TextSnake (Long,2018)	67.9	**85.3**	75.6
PSENet (Li,2019)	82.09	77.84	79.9
LSAE (Tian,2019)	82.7	77.8	80.1
GAPF	**83.2**	79.6	**81.3**

Table 2: Quantitative results of different methods evaluated on Total-Text.

Method	Precision	Recall	F-measure
MaskSpotter (Lyu,2018)	69.0	55.0	61.3
TextSnake (Long,2018)	82.7	74.5	78.4
PSENet (Li,2019)	**84.54**	75.23	79.61
GAPF	83.6	**76.8**	**80.05**

Besides, to prove the universality of the proposed GAPF in this paper, we evaluate the proposed method on the ICDAR2015 and MSRA-TD500 to test its ability for oriented text detection. Table 3 and Table 4 show the comparison results with advanced methods. The results show that the method in this paper has an accurate detection effect on the horizontal and inclined text.

Table 3: Quantitative results of different methods evaluated on Total-Text.

Method	Precision	Recall	F-measure
CCNF (Yao,2016)	72.86	58.69	64.77
RRPN (Ma.2018)	82.17	73.23	77.44
EAST (Zhou,2017)	83.27	78.33	80.72
TextSnake (Long,2018)	84.90	80.40	82.60
PixelLink (Dan,2018)	85.50	**82.00**	83.70
GAPF	**86.42**	81.83	**84.06**

Table 4: Quantitative results of different methods evaluated on MSRA-TD500.

Method	Precision	Recall	F-measure
RRPN (Ma.2018)	82	69	75
EAST (Zhou,2017)	83.56	67.13	74.45
TextSnake (Long,2018)	83.2	**73.9**	78.3
PixelLink (Dan,2018)	83	73.2	77.8
GAPF	**84.2**	73.6	**78.54**

Figure 7 depicts several detection examples by the proposed GAPF. The solid red line box in the figure represents the output text detection box. It can be seen that the detection results of the method GAPF are better on these curved texts and can also effectively support horizontal and multi-direction text detection.

Figure 7: Qualitative results by the proposed method. From top to bottom in row: image from CTW1500, Total-Text, ICDAR 2015, and MSRA-TD500.

5 CONCLUSION

We propose a text detection framework (GAPF) for the arbitrary shape scene text. The text segmentation map is generated by fusing pixel fluctuation information into the generation adversarial network. Then use the boundary generation algorithm to get the bounding box of the text area. Our method is robust to shapes and can easily detect text instances of arbitrary shapes. Experimental comparisons with the state-of-the-art approaches on multiple datasets show the effectiveness of the proposed GAPF for the text detection task.

REFERENCES

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 770–778(2016)

Goodfellow I J, Pougetabadie J, Mirza M, et al. Generative Adversarial Networks[J]. Advances in Neural Information Processing Systems, 2672–2680(2014)

Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39(6), 1137–1149 (2017)

Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-Oriented Scene Text Detection via Rotation Proposals. IEEE Trans. Multimedia 20(11), 3111–3122 (2018)

Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: TextBoxes: A Fast Text Detector with a Single Deep Neural Network. In Proc. AAAI, 4161–4167(2017)

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., Berg, A.: SSD: Single Shot MultiBox Detector. In Proc.ECCV, 21–37(2016)

Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: EAST: An Efficient and Accurate Scene Text Detector. In Proc.CVPR, 2642–2651(2017)

Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao, "Detecting text in natural image with connectionist text proposal network," in ECCV, 56–72(2016)

Alex Graves and Jürgen Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures", Neural Networks, vol.18, no.5-6, pp.602–610 (2005)

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 3431–3440(2015)

Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In CVPR, 4159–4167(2016)

Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In AAAI, 6773–6780(2018)

Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In ECCV,19–35(2018)

Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, and Jian Yang, "Shape robust text detection with progressive scale expansion network," In CVPR, 9336–9345(2019).

Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: The IEEE Conference on Computer Vision and Pattern Recognition. In CVPR, 936–944(2017)

Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer International Publishing. In MICCAI, 234–24 (2015)

He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In CVPR, 770–778(2016)

Zeiler, M.D., Krishnan, D., Taylor, G.W., Fergus, R.: Deconvolutional networks. In CVPR, 2528–2535 (2010)

Trevor Hastie and Werner Stuetzle, "Principal curves", Journal of the American Statistical Association, vol. 84, no. 406, pp. 502–516(1989)

Yuliang Liu, Lianwen Jin, Shuaitao Zhang, and Sheng Zhang, "Detecting curve text in the wild: New dataset and new solution", arXiv:1712.02170, (2017)

Chee Kheng Ch'ng and Chee Seng Chan, "Total-text: A comprehensive dataset for scene text detection and recognition", in ICDAR, 935–942 (2017)

Karatzas, D., Gomez, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V., Lu, S., Shafait, F., Uchida, S., Valveny, E.: ICDAR 2015 competition on robust reading. In Proc.ICDAR, 1156–1160(2015)

Yao, C., Bai, X., Liu, W., Ma, Y., Tu, Z.: Detecting texts of arbitrary orientations in natural images. In CVPR, 1083–1090(2012)

Yuliang Liu, Lianwen Jin, Shuaitao Zhang, and Sheng Zhang, "Detecting curve text in the wild: New dataset and new solution", arXiv:1712.02170, (2017)

Yixing Zhu and Jun Du, "Sliding line point regression for shape robust scene text detection", arXiv:1801.09969, (2018)

Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, Jiaya Jia:Learning Shape-Aware Embedding for Scene Text Detection. In CVPR, 4234–4243(2019)

Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes",in ECCV,71–88(2018)

Yao, C., Bai, X., Sang, N., Zhou, X., Zhou, S., Cao, Z.: Scene Text Detection via Holistic, Multi-Channel Prediction. CoRR abs/1606.09002 (2016)