Kernel Adaptive Convolution for Scene Text Detection via Distance Map Prediction

Jinzhi Zheng[1,2], Heng Fan[3], Libo Zhang[1,2*]
[1]Institute of Software Chinese Academy of Sciences, Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
libo@iscas.ac.cn
[3]Department of Computer Science and Engineering, University of North Texas, Denton, USA
heng.fan@unt.edu

Abstract

Segmentation-based scene text detection algorithms that are accurate to the pixel level can satisfy the detection of arbitrary shape scene text and have received widespread attention. On the one hand, due to the complexity and diversity of the scene text, the convolution with a fixed kernel size has some limitations in extracting the visual features of the scene text. On the other hand, most of the existing segmentation-based algorithms only segment the center of the text, losing information such as the edges and directions of the text, with limited detection accuracy. There are also some improved algorithms that use iterative corrections or introduce other multiple information to improve text detection accuracy but at the expense of efficiency. To address these issues, this paper proposes a simple and effective scene text detection method, the Kernel Adaptive Convolution, which is designed with a Kernel Adaptive Convolution Module for scene text detection via predicting the distance map. Specifically, first, we design an extensible kernel adaptive convolution module (KACM) to extract visual features from multiple convolutions with different kernel sizes in an adaptive manner. Secondly, our method predicts the text distance map under the supervision of a priori information (including direction map, and foreground segmentation map) and completes the text detection from the predicted distance map. Experiments on four publicly available datasets prove the effectiveness of our algorithm, in which the accuracy and efficiency of both the Total-Text and TD500 outperform the state-of-the-art algorithm. The algorithm efficiency is improved while the accuracy is competitive on ArT and CTW1500.

*Corresponding author: Libo Zhang (libo@iscas.ac.cn)

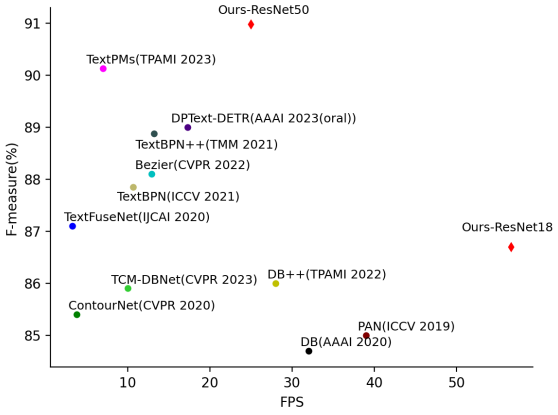

Figure 1. Performance comparison of several recent scene text detection methods on the Total-Text dataset. Our method achieves a new ideal tradeoff between effectiveness and efficiency compared to current state-of-the-art methods.

1. Introduction

Scene text detection, as an important research of computer vision, has a wide range of applications in fields such as scene parsing, blind assistance, automatic driving, and product recommendation [14, 25, 27, 44]. Therefore, it has received extensive attention for a long time. With the development of artificial intelligence technology, some excellent scene text algorithms [19, 20, 37, 41, 42] have achieved impressive progress. However, the complexity and diversity of the scene text [45], such as the variability of the size, the uncertainty of the direction, the arbitrariness of the shape, the randomness of the fonts, and the complexity of the background, make the scene text detection still challenging.

Segmentation-based algorithms are able to achieve pixel-level accuracy and are suitable for arbitrary shape text detection, and thus have received increasing attention [15, 19,

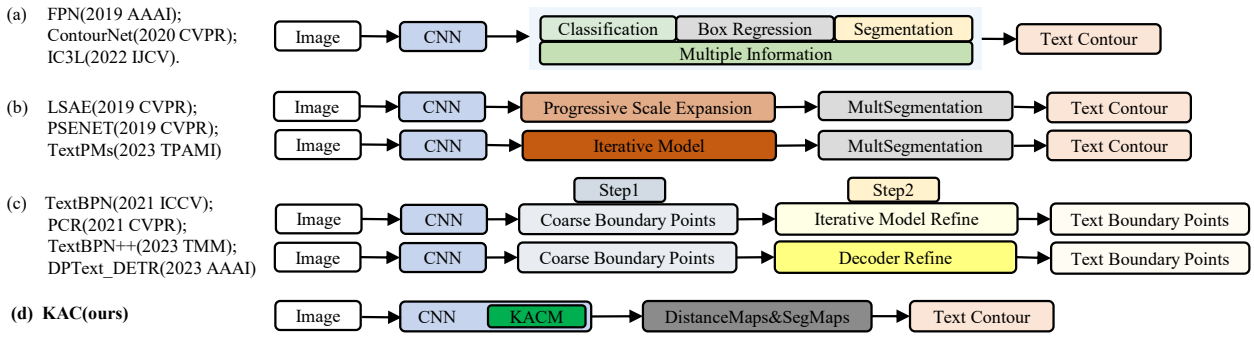

Figure 2. Comparison of the inference detection process of some scene text detection algorithms that are most relevant to ours. Our kernel adaptive convolution via distance map detects scene text and has a more concise structure.

20]. To separate adjacent text, early segmentation-based algorithms usually shrink the text region to obtain the text center, which is segmented. The text center region obtained by segmentation in the inference stage is expanded during post-processing to obtain the complete text region. The advantage is that arbitrary shape text can be detected, but the disadvantage is that only the central region of the text is used in the detection, while other effective information including text edge information is lost. In order to alleviate this problem, some algorithms [7, 32, 33] that learn multiple text information are gradually proposed. These algorithms usually incorporate multiple ideas such as classification of text, regression, etc., to complete text detection, as shown in Fig.2(a). To enhance the suppression of noise in complex backgrounds, some algorithms [16, 29, 43] have been proposed to construct different variants of the same type of information and correct the segmentation map or text border points by iterative or progressive, as shown in Fig.2(b).To further improve the scene text detection accuracy, two-stage scene text detection algorithms have been gradually proposed [6, 37, 41, 42, 46]. The first stage of such algorithms generates rough boundary proposal points, and in the second stage, these boundary proposal points are optimized by iteration or decoding, as shown in Fig.2(c). This type of two-stage algorithm achieves the current relatively superior detection results, but at the same time sacrifices the efficiency of the algorithm due to the increase in model complexity. The requirements for performance and real-time cannot be met simultaneously. In conclusion, the main challenges of current scene text detection algorithms remain in the areas of performance and efficiency.

In this paper, we propose a simple and effective scene text detection algorithm from two aspects. On the one hand, a standard Convolution Neural Net (CNN) uses a fixed-size convolution kernel, and the receptive fields of each convolution layer are fixed [1]. The size of the text in the scene is variable, and the receptive field should be able to adapt itself according to the specifics of the text. Based on this idea, we design a plug-and-play kernel adaptive convolution module from the perspective of optimizing the convolution kernel and enhancing the adaptive ability of the convolution receptive field. On the other hand, the existing segmentation-based algorithms only segment the central region of text without edge information, and the algorithm accuracy is limited. Or it is necessary to predict multiple other information, such as boundary regression, foreground segmentation, etc., at the expense of efficiency. Inspired by [41–43], this paper propose a method for text detection based on text distance map prediction. As shown in Fig.1 and 2, the KAC algorithm proposed in this paper has a more concise structure and also achieves better performance than SOTA. The main contributions of this paper are as follows:

- We propose a simple and effective scene text detection network, called Kernel Adaptive Convolutional (KAC), which has a simple model structure, achieves more than state-of-the-art performance by means of distance map prediction, and is competitive in terms of efficiency.
- We propose an extensible Kernel Adaptive Convolution Module (KACM) that sets up a set of convolutions with different kernel sizes, and then adaptively provides a set of weights corresponding to this set of convolutions for each position in the feature space, thus enhancing the feature representation at each position.
- Experiments on four publicly available datasets validate the effectiveness of our method. Specifically, our algorithm outperforms the state-of-the-art algorithms in terms of accuracy on Total-Text and TD500, with efficiency improvements of 127.3% and 115.0%, respectively. On ArT and CTW1500, the efficiency is improved by 142.1% and 114.9%, while the detection accuracy is also competitive.

2. Related Work

2.1. Regression-Based Methods

The scene text detection methods [17, 18, 24, 48] based on regression treat scene text detection as a general ob-

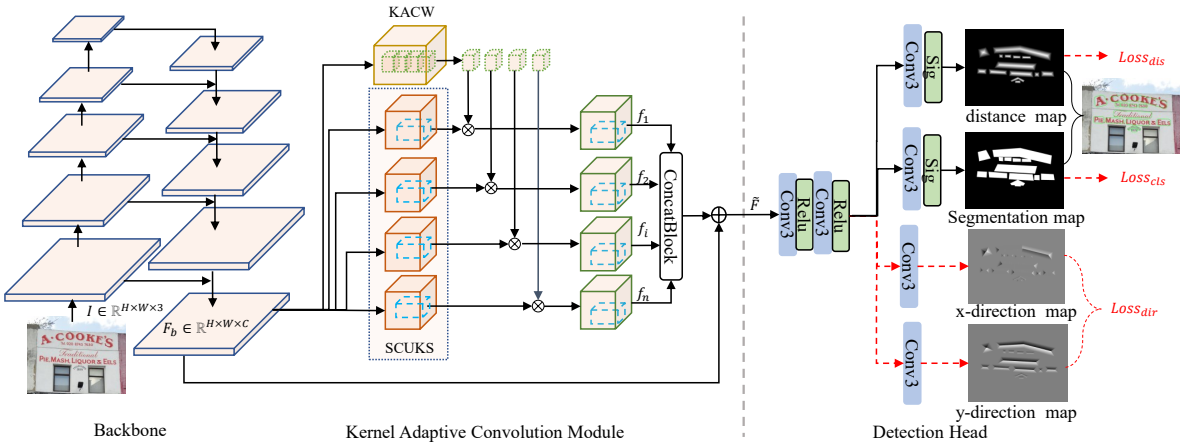

Figure 3. The overall framework of KAC is mainly composed of a Backbone, a kernel adaptive convolution module (KACM), and a detection head. Backbone and kernel adaptive convolution module extract visual features $\widetilde{F}$ from image $I \in \mathbb{R}^{H\times W\times 3}$. The detection head predicts the distance map and segmentation map from the visual features and generates the text outline to complete the text detection. In the detection head, the red dashed exists only during the training phase, indicating the supervision of prior information.

ject detection task and perform regression on text region bounding boxes. For example, TextBoxes [17] sets anchor boxes on visual features, returning anchor boxes that surround the text. EAST [48] directly predicts the rotation angle of the text box. To improve the accuracy, there are also some improved methods. For example, MOST[10] designs a Position-Aware Non-Maximum Suppression(PA-NMS). HAM [11] proposed a regression method to hide anchor frames. This kind of method is usually only applicable to horizontal, vertical, or a certain angle of the scene text.

2.2. Connected Component-based Methods

In order to be able to detect curved text, Connected Component-based Methods [2, 28, 40] split the text into a series of components, and then complete the detection task by linking the text components. For example, CTPN [28] uses BLSTM to generate fine-grained text proposal blocks, CRAFT [2] divides the text into characters and creates affiliation attributes between the characters, and DDRG [40] divides the text into different text components, constructs geometric attributes such as size, angle, etc. for each text component, and then groups the text components based on the set attributes. However, such algorithms are challenging for arbitrary shape text detection.

2.3. Segmentation-based Methods

Segmentation-based methods [16, 19, 20, 29, 33] can be accurate to the pixel level, so they have a wide range of applications in arbitrary shape text detection. For example, DB [19] and DB++ [20] directly segment the central region of the text. LSAE [29] and PAN [30] segment the center region and the foreground region of the text and determine the text region by combining the results of the center and foreground segmentation in post-processing. Literature [43] constructs multiple probabilistic maps for text regions, trains the probabilistic maps in an iterative manner, and then recovers the regions of text from the probabilistic maps. Segmentation-based scene text detection methods are able to detect arbitrary shape text, but their accuracy has certain limitations.

2.4. Boundary Points-Based Methods

Boundary points-based methods [2, 27, 31, 46] perform text detection by predicting or generating key points on the text boundary. For example, CTD [39] uses RNN decoding to predict 14 boundary points to depict the text region. Literature [31] uses BLSTM to adaptively predict text boundary points, including the number of boundary points. BPN [41], BPN++ [42], and DPText-DETR [37] use a two-stage detection scheme. The first stage generates coarse boundary proposal points. The second stage corrects the boundary proposal points of the first stage by iterative boundary deformation or transformer decoder. These two-stage algorithms improve detection accuracy but do not offer significant efficiency advantages due to the use of iterative corrections or complex decoding and encoding methods.

3. Methodology

As shown in Fig.3, the KAC proposed in this paper mainly consists of a Backbone, a plug-and-play kernel adaptive convolution module KACM, and a detection head. Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the backbone extracts the image visual feature F_b. Then, this visual feature is fed to the kernel adaptive convolution module. Finally, the detection head, supervised by the a priori information, predicts

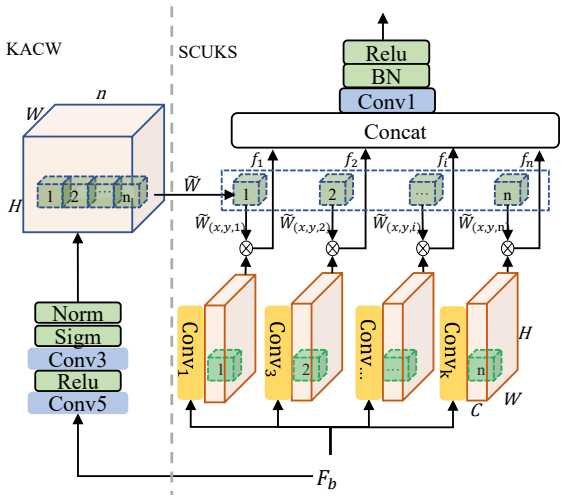

Figure 4. The structure of the kernel adaptive convolution layer, which consists of kernel adaptive convolutional weights (KACW) and a set of convolutions with unequal kernel sizes (SCUKS).

the text distance map and segmentation map from the features provided by the kernel adaptive convolution module to complete the text detection. In the training phase, the detection head sets up supervised learning of the text distance map, segmentation map, and direction map. Considering the detection efficiency, the prediction of the direction map is removed in the forward inference stage. Thresholding the predicted distance map, the text center region is extracted. Then, the Vatti clipping algorithm [1] is used to expand the center region and output the text detection results.

3.1. Backbone

As shown in Fig.3, in order to make the extracted features as adaptable as possible to scene text instances of varying scales, and to combine high-level semantic information with low-level context information, we use pyramid network and ResNet structure, which are similar or identical to algorithms [41–43], as the backbone. The visual features extracted by the backbone network from the image $I \in \mathbb{R}^{H\times W\times 3}$ are formulaically expressed as

$$F_b = R(I) \in \mathbb{R}^{H\times W\times C}, \tag{1}$$

where H and W represent the height and width of the image, and C represents the feature channels. R represents the backbone network structure shown in Fig.3.

3.2. Kernel Adaptive Convolution Module

Standard convolution uses a fixed-size convolution kernel, and the receptive field is fixed and cannot be adjusted adaptively to accommodate the variable text scales in complex scenes. To enable the convolution to adjust the receptive field adaptively, we designed the kernel adaptive convolution module (KACM). As shown in Fig.3, the kernel adaptive convolution module consists of kernel adaptive convolution layer (KACL) and residual linkage.

KPN(Kernel Prediction Network)[23] predicts the same filter for different channels at each position of the spatial resolution. FAC(Filter Adaptive Convolutional)[47] predicts different filters for different channels at each position of the spatial resolution. KPN [23], and FAC [47] generate different filters for different positions of the spatial resolution, which changes the way the same filter is shared by all the locations and enhances flexibility. Dynamic Convolution [3] uses the same weights at all spatial positions and the operation objects are the kernel of the standard convolution. Inspired by KPN [23], FAC [47], Dynamic Convolution [3], SENet [12], CondConv [34], we propose the Kernel Adaptive Convolution Layer (KACL) as shown in Fig.4.

To further enhance the flexibility of the convolution kernels, the Kernel Adaptive Convolution Layer provides weights for a set of convolution kernels of different scales at each position of the spatial resolution. Similar to Dynamic Convolution, Kernel Adaptive Convolution has n convolution kernels. The Kernel Adaptive Convolution Layer consists mainly of Kernel Adaptive Convolution Weights and a set of convolutions with unequal kernel sizes.

Kernel Adaptive Convolution Weights. In this paper, we set up a branch consisting of a 5×5 convolution, a $Relu$ activation function, a 3×3 convolution, a $sigmoid$ activation function, and a normalization layer to generate its adaptive weights from the input feature F_b. The purpose of the normalization layer is to make the sum of the weights 1. This process can be formulated as

$$\begin{cases} \widetilde{W}' = \sigma(Conv_3(\gamma(Conv_5(F_b)))) \\ \widetilde{W}_{(x,y,i)} = \frac{W'_{(x,y,i)}}{\sum_{j=1}^{n} W'_{(x,y,j)}}, i \in [1,n], \end{cases} \tag{2}$$

where $Conv_3$, $Conv_5$ represent convolutions with kernel size of 3×3 and 5×5. γ, σ represent activation functions $Relu$ and $Sigmoid$. n is the maximum number of convolution. $\widetilde{W}_{(x,y,i)}$ and $\widetilde{W}'_{(x,y,i)}$ denote the values of $\widetilde{W} \in \mathbb{R}^{H\times W\times n}$ and $\widetilde{W}' \in \mathbb{R}^{H\times W\times n}$ on (x,y,i), respectively.

Set of Convolution with Unequal Kernel Sizes. Generate a set of convolution kernels with sizes ranging from small to large. The size of the $i-th$ convolution kernel is defined as $k_i \times k_i$. The formula for k_i is expressed as follows:

$$k_i = 2 \times (i-1) + 1; i \in [1,n], \tag{3}$$

where n is the maximum number of convolutions.

Kernel Adaptive Convolution Module computing features. The features F_b input a set of convolutions in parallel, and the size of each convolution kernel is determined by the above formula (3). Then, the output is activated by the $Relu$ function. Finally, the output features of each convolution are broadcast multiplied with the corresponding kernel

adaptive convolution weight. The adaptive feature formulation of the $i-th$ convolution kernel can be expressed as

$$f_i = \gamma(Conv_{k_i}(F_b)) * \widetilde{W}_{(:,:,:,i)}, i \in [1, n], \quad (4)$$

where $f_i \in \mathbb{R}^{H \times W \times C}$, and $*$ denotes the broadcast multiplication of the matrix. The adaptive feature of this set of convolution can be expressed as

$$T_C = Concat(f_1, \ldots, f_n). \quad (5)$$

In addition, we set residual links, and the kernel adaptive convolution module transform feature can be formulated as

$$\widetilde{F} = \gamma(Bn(Conv_1(T_C))) + F_b, \quad (6)$$

where $\widetilde{F} \in \mathbb{R}^{H \times W \times C}$, Bn denotes the batch normal operation. The size of different text instances in the scene varies greatly, and adaptive convolution kernels to provide different receptive fields are necessary. Dynamic Convolution [3], SENet[12] and CondConv [34] adjust convolution operations based on the unit of the channel or the whole feature of an instance, while the kernel adaptive convolution module takes spatial position as the unit, making the convolution operations at each position adaptive.

3.3. Detection Head

Unlike TextPMs [43], which split multiple probability maps, KAC only needs to predict one distance map and one segmentation graph when inference is performed, thus providing higher efficiency. And unlike DB [19] which segmented the text center dropping the text edge information, we predict the complete text region distance map and thus have higher accuracy.

We used a detection head similar to BPN++ [42], as shown in Fig.3. Inspired by [41–43], during training, distance maps, segmentation maps, and direction maps in x and y directions are set as prior information to supervise training. The detection head consists of a 3 × 3 convolution with an expansion rate of 2, a $relu$ activation function, a 3 × 3 convolution with an expansion rate of 2, a $relu$ activation function, and four output modules. Two of the four output modules are composed of convolution with kernel 3×3 and $sigmoid$ activation function, and the other two 3×3 convolutions output direction maps in the x and y directions.

The segmentation map contains the confidence score of whether each pixel is text or not. The distance map defined as the distance of the internal pixels of the text from the text edges and the distance value between pixels outside the text region and the text boundary is 0. The distance map contains richer information about the geometry of the text compared to the text center segmentation map and provides more flexibility in extracting the text center. The direction map represents the direction of a pixel within a text region to its nearest point on the text boundary. Inference stage, the distance map and segmentation map only need to be predicted. The center region of the text is intercepted from the distance map, and the number and position of the text are preliminarily determined. The pixel confidence score of the segmentation map is used as auxiliary information to remove the fake text center region and preserve the true text center region.

3.4. Loss Function

Four types of supervised information are set up in the training phase, where the supervision in the x, y direction can be considered as a two-dimensional vector. Therefore, the multi-objective loss function contains three components: distance map, segmentation map, and direction map. We adopted the boundary proposal loss of [41, 42]. The formulae are expressed as

$$Loss = \alpha_1 Loss_{cls} + \alpha_2 Loss_{dis} + \alpha_3 Loss_{dir}, \quad (7)$$

where $Loss_{cls}$ denotes the cross-entropy loss function (cross-entropy loss) between the predicted segmentation map and the segmentation map ground truth, and $Loss_{dis}$ denotes the L_2 loss between the predicted distance map and the distance map ground truth. $Loss_{dir}$ is the $L_2 - Norm$ distance and angular distance of the direction map. α_1, α_2, α_3 are equilibrium factors, similar to [41, 42], taking values of 1,3,1 respectively.

4. Experiments

4.1. Datasets

SynthText [8] collects about 800k synthetic scene images, and the text in the scene images is synthesized from rendering. The use of SynthText in the training stage can reduce the cost of manual annotation and alleviate the problem of insufficient training samples to some extent.

Total-Text [4] contains horizontal text, multi-directional text, and curved text, where the training set contains 1255 images and the test set contains 300 images. The dataset provides annotations of text instances at the word level.

MSRA-TD500(TD500) [35] contains a training set of 300 scene images and a test set of 200 images. The dataset contains arbitrarily shaped Chinese and English text.

ICDAR19 ArT(ArT) [5] is a large-scale multi-lingual arbitrary shape scene text dataset with a variable number of boundary point annotations for each text. The training set contains 5603 images and the test set contains 4563 images.

CTW1500 [21] consists of a training subset of 1000 images and a testing set of 500 images. The dataset contains a large number of curved texts, and each text in the training subset is annotated with 14 edge points.

4.2. Implementation details

In order to make a fair comparison, this paper adopts experimental strategies that are as identical or similar as possible to the SOTA algorithm [37, 41, 42]. The experiment uses the ResNet [9] in the backbone. Experiments are implemented by Pytorch 1.7.0 and Python 3.7.13. The image resized to 640×640 for training, and the original image size is maintained for forward inference. Training and inference were performed on a single NVIDIA TITAN RTX with 24G. Similar to the training strategy of M3 TextSpotter [13], since KAC shares a backbone with the Boundary Proposal Module of BPN++ [42], we download its model from the official website of BPN++ and extract the weights to initialize KAC. The training process is divided into two phases: pre-train and finetune. Pretraining was performed on SynthText for one epoch, with the learning rate fixed at 0.0005 and batch size set to 8. Finetune was performed on two corresponding real datasets with 600 epochs (TD500 is 1200 epochs), the initial learning rate was set to 0.0005, the learning rate decayed by 0.9 for every additional 50 epochs, and the batch size was set to 6. The optimizer Adam is used for training. Data preprocessing operations include random clipping, random flipping, and random rotation according to a Gaussian distribution with angles in the range (-30, 30), among others.

4.3. Ablation Studies

Methods	KACM	P(%)	R(%)	F(%)
DB [19]	×	91.5	79.2	84.9
DB [19]	√	92.2	80.8	86.1
KAC(Ours)	×	93.31	85.91	89.76
KAC(Ours)	√	93.87	88.14	90.79

Table 1. Detection results of KACM embedding into DB and KAC on TD500.

Kernel Adaptive Convolution Module (KACM). To validate the effectiveness of KACM as a plug-and-play module, we embed the module into DB [19] for ablation studies, as shown in Table 1. It can be seen that embedding the Kernel Adaptive Convolution Module improves the DB algorithm by 0.7%, 1.2%, and 1.6% in the Precision, F-measure, and Recall, respectively. KACM enables KAC to improve the Precision, F-measure, and Recall by 0.56%, 1.03%, and 2.23%, respectively.

Distance Map prediction. In order to validate the effectiveness of the text distance map prediction (TDMP) strategy proposed in this paper, we compared the performance of the algorithm when distance map and text region-centered segmentation (TRCS) [16, 19, 29] were used as the training targets, and the statistical results were shown in Table 2. It can be seen that the distance prediction strategy proposed in this paper outperforms the strategy with text region-centered segmentation. Specifically, the distance map prediction strategy instead of the text region-centered segmentation strategy improves the Precision, F-measure, and Recall by 1.9%, 2.56%, and 3.81%, respectively.

Methods	TDMP	TRCS	P(%)	R(%)	F(%)
KAC(Ours)	×	√	93.01	84.26	88.42
KAC(Ours)	√	×	94.10	88.07	90.98

Table 2. Comparison of detection results between traditional text regions center segmentation (TRCS) and our text distance map prediction(TDMP) strategy on Total-Text.

(a) dis_thres=0.9 (b) dis_thres=0.9 (c) dis_thres=0.9
(d) dis_thres=0.6 (e) dis_thres=0.6 (f) dis_thres=0.6
(g) dis_thres=0.3 (h) dis_thres=0.3 (i) dis_thres=0.3

Figure 5. Visualization of the text center regions under different distance thresholds. (a,d,g) CTW1500. (b,e,h) Total-Text. (c, f, i) ICDAR19 ArT.

Method	DirMap	SegMap	P(%)	R(%)	F(%)
Baseline	×	×	91.12	89.36	87.67
Baseline	√	×	92.45	90.21	88.09
Baseline	√	√	92.50	89.53	90.99

Table 3. Effectiveness of prior information supervision on Total-Text. DirMap and SegMap represent the prior information on the direction map and segmentation map, respectively.

Visualization of the Flexibility of Distance Map prediction. Different distance thresholds can separate different text center regions from the distance map. We visualized the

obtained text center regions under different distance thresholds, as shown in Fig.5. The distance threshold takes values in the range (0, 1). The larger the threshold, the smaller the obtained text region, and the greater the confidence that the text region is true. And vice versa. This proves that the strategy of extracting the text center from the prediction distance map is more flexible than the traditional text center region segmentation strategy. Moreover, the distance map contains information about the edges of the text, pixel distribution rules, etc., compared to the center map that predicts the shrink text regions, so it is more accurate to predict the text center area based on the distance map. As shown in Fig.5, the extracted text is relatively complete when dis_thres=0.3.

Effectiveness of Prior information supervision learning. As shown in Table 3, the use of prior information for supervised learning can improve the performance of the algorithm. Specifically, the direction map improves Precision, Recall, and F-measure by 1.33%, 0.85%, and 0.42%. By using the distance map and segmentation map simultaneously, the Precision and Recall of the algorithm can be improved by 1.38%, and 3.32%, respectively.

Number	P(%)	F(%)	R(%)	FPS
$n = 3$	92.67	90.54	87.49	29.5
$n = 4$	94.10	90.98	88.07	30.0
$n = 5$	93.31	90.78	88.36	30.0
$n = 6$	93.34	90.82	88.34	30.3

Table 4. Detection results of KACM with different Kernel numbers on Total-Text. n is the number of kernels.

Effectiveness of KACM with different Kernel numbers. The performance statistics of the kernel adaptive convolution module with different numbers of kernels are shown in Table 4. The algorithm has the highest detection accuracy when the number of kernels is 4, and the Precision reaches 94.10%. KACM with more kernel did not bring significant performance gains. For example, when the number of kernels in KACM is 5, the Precision is 93.31%. Therefore, the number of kernels in KACM in our experiment is fixed at 4.

4.4. Comparison with State-of-the-Art Methods

We compare our method with other algorithms on four benchmarks, including two curved texts (CTW1500, Total-Text) and two multi-lingual texts (TD500, ICDAR19).

Curved text detection. To verify the robustness of Our method to text shapes, we conducted experiments on the curved text Total-Text and CTW1500. On the Total-Text and CTW1500, the binarization threshold dic_thres of the distance map is 0.325 and 0.3, respectively. As shown in Table 5, the performance and efficiency of our algorithm

Methods	P(%)	R(%)	F(%)	FPS
TextSnake [22]	82.7	74.5	78.4	-
CRAFT [2]	87.6	79.9	83.6	-
PAN [30]	89.3	81.0	85.0	<u>39.6</u>
PSENET [16]	84.5	75.2	79.6	8.4
DRRG [40]	86.5	84.9	85.7	-
DB [19]	87.1	82.5	84.7	32.0
ContourNet [32]	86.9	83.9	85.4	3.8
TextFuseNet [36]	89.0	85.3	87.1	3.3
FCENet [49]	85.1	82.7	83.9	-
PCR [6]	88.5	82.0	85.2	-
TextBPN [41]	90.67	85.19	87.85	10.7
I3CL [7]	89.8	84.2	86.9	-
Bezier [27]	90.7	85.7	88.1	12.9
TextPMs [43]	89.95	87.67	88.79	7.0
TCM-DBNet [38]	-	-	85.9	-
DB++ [20]	88.9	83.2	86.0	28.0
CT-Net [26]	90.8	85.0	87.8	10.1
TextBPN++ [42]	<u>92.44</u>	<u>87.93</u>	<u>90.13</u>	13.2
DPText-DETR [37]	91.8	86.4	89.0	17.3
KAC-ResNet18(Ours)	90.24	83.43	86.70	**56.6**
KAC-ResNet50(Ours)	**94.10**	**88.07**	**90.98**	25.0

Table 5. Detection results on Total-Text. Bold and underline indicate optimal and suboptimal, respectively.

on the Total-Text outperform current state-of-the-art methods. Specifically, the proposed algorithm exceeds the current state-of-the-art TextBPN++ [42] by 1.66%, 0.85%, and 0.14% in Precision, F-measure, and Recall, respectively. FPS increased from 13.2 to 25.0, increasing efficiency by 89.4% and significantly improving detection efficiency. As can be seen from Table 6, the performance of DPText-DETR [37] on CTW1500 is better than our algorithm, but as can be seen from Table 5, its efficiency is lower than ours. It can be concluded that our method can achieve better efficiency when the Scene text detection performance is comparable on CTW1500 dataset.

Multi-language text detection. In order to verify the effectiveness of our method for multi-lingual scene text, we conducted experiments on the ICDAR19 ArT and TD500. On the ICDAR19 ArT and TD500, the binarization threshold dic_thres of the distance map is 0.475 and 0.3, respectively. As can be seen from Table 7, efficiency is improved by 142.1% while remaining performance competitive on the ICDAR19 ArT. As shown in Table 8, our method achieves performance and efficiency over current state-of-the-art algorithms on the TD500 dataset. Specifically, compared to the current optimal TextBPN++ [42] algorithm our algorithm improves Precision, F-measure, and Recall by 0.18%, 0.69%, and 1.37%, while improving efficiency by 18.3%.

Experiments show that compared to other methods

Methods	P(%)	R(%)	F(%)	FPS
TextSnake [22]	67.9	85.3	75.6	-
LSAE [29]	82.7	77.8	80.1	-
CRAFT [2]	86.0	81.1	83.5	-
PAN [30]	86.4	81.2	83.7	39.8
PSENET [16]	82.1	77.8	79.9	8.4
DRRG [40]	85.9	83.0	84.5	-
DB [19]	80.2	83.4	86.9	22.0
ContourNet [32]	83.7	84.1	83.9	4.5
TextFuseNet [36]	87.8	85.4	86.6	3.7
FCENet [49]	87.6	83.4	85.5	-
PCR [6]	87.2	82.3	84.7	11.8
TextBPN [41]	86.45	83.60	85.00	12.2
I3CL [7]	88.4	84.6	86.5	7.6
Bezier [27]	88.1	82.4	85.2	12.9
TextPMs [43]	87.75	83.83	85.75	9.1
TCM-DBNet [38]	-	-	84.9	-
DB++ [20]	87.9	82.8	85.3	26.0
CT-Net [26]	88.5	83.8	86.1	11.2
TextBPN++ [42]	88.34	84.71	86.49	16.5
DPText-DETR [37]	**91.7**	**86.2**	**88.8**	-
KAC-ResNet18(Ours)	87.52	80.90	84.08	**82.9**
KAC-ResNet50(Ours)	88.60	85.43	86.84	19.2

Table 6. Detection results on CTW1500. Bold and underline indicate optimal and suboptimal, respectively.

Methods	P(%)	R(%)	F(%)	FPS*
CRAFT* [2]	77.2	68.9	72.9	-
TextFuseNet* [36]	82.6	69.4	75.4	-
PCR [6]	84.0	66.1	74.0	-
I3CL [7]	82.7	71.3	76.6	-
TextBPN++ [42]	**84.48**	77.05	**80.59**	1.9
DPText-DETR [37]	83.0	73.7	78.1	-
KAC-ReNet50(Ours)	83.0	**77.3**	80.0	**4.6**

Table 7. Detection results on ICDAR19 ArT. '*' represents the data reproduced in our experimental environment.

on datasets CTW1500 and ICDAR19 ArT, our method achieves comparable performance and higher efficiency. On Total-Text and TD500, our approach outperforms the state-of-the-art method in both performance and efficiency.

4.5. Qualitative and Efficiency analysis

To further qualitatively analyze the effectiveness of our method, we show some of the detection results in Fig. 6. As shown in Fig. 6, our algorithm can detect arbitrary shape scene text. Our Kernel Adaptive Convolution can handle situations where scene text is adjacent, text regions partially overlap, and the large text contains the small text.

Methods	P(%)	R(%)	F(%)	FPS
TextSnake [22]	83.2	73.9	78.3	1.1
LSAE [29]	84.2	81.7	82.9	-
CRAFT [2]	88.2	78.2	82.9	8.6
PAN [30]	84.4	83.8	84.1	30.2
DRRG [40]	88.1	82.3	85.1	-
DB [19]	91.5	79.2	84.9	32.0
PCR [6]	90.8	83.5	87.0	-
TextBPN [41]	86.62	84.54	85.57	12.3
TextPMs [43]	91.01	86.94	88.93	10.6
TCM-DBNet [38]	-	-	88.8	10.0
DB++ [20]	91.5	83.3	87.2	29.0
CT-Net [26]	90.8	84.4	87.5	11.6
TextBPN++ [42]	93.69	86.77	90.10	15.3
KAC-ResNet18(Ours)	88.39	82.47	85.33	**90.8**
KAC-ResNet50(Ours)	**93.87**	**88.14**	**90.79**	18.1

Table 8. Detection results on TD500. Bold indicates that the corresponding value is optimal.

(a) Total-Text (b) CTW1500 (c) ArT (d) TD500

Figure 6. Visual results of our algorithm on four datasets. The green irregular polygon is the detected text contours.

5. Conclusion

In this paper, we propose a simple yet effective scene text detection network, Kernel Adaptive Convolution, which is embedded with a kernel adaptive convolution module for text detection by predicting the text region distance map. The kernel adaptive convolution module makes the representation of extracted visual features more effective, and the prediction of distance maps makes text center prediction more accurate. Experiments on benchmark datasets show that the algorithm in this paper achieves better performance than the state-of-the-art algorithm. In the future, we will further explore how to build a simple and efficient end-to-end scene text spotting model to meet the accuracy and real-time requirements of real-world applications.

Acknowledgement. Libo Zhang was supported by Youth Innovation Promotion Association, CAS (2020111). Heng Fan and his employer receive no financial support for this work.

References

[1] R. D. Andreev. Algorithm for clipping arbitrary polygons. *Computer Graphics Forum*, 8:183–191, 1989. 2, 4

[2] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9357–9366, 2019. 3, 7, 8

[3] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11027–11036, 2020. 4, 5

[4] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 935–942, 2017. 5

[5] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, Chee Seng Chan, and Lianwen Jin. Icdar2019 robust reading challenge on arbitrary-shaped text - rrc-art. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1571–1576, 2019. 5

[6] Pengwen Dai, Sanyi Zhang, Hua Zhang, and Xiaochun Cao. Progressive contour regression for arbitrary-shape scene text detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7389–7398, 2021. 2, 7, 8

[7] Bo Du, Jian Ye, Jing Zhang, Juhua Liu, and Dacheng Tao. I3cl: Intra- and inter-instance collaborative learning for arbitrary-shaped scene text detection. *International Journal of Computer Vision*, 130:1961–1977, 2022. 2, 7, 8

[8] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[10] Minghang He, Minghui Liao, Zhibo Yang, Humen Zhong, Jun Tang, Wenqing Cheng, Cong Yao, Yongpan Wang, and Xiang Bai. Most: A multi-oriented scene text detector with localization refinement. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8809–8818, 2021. 3

[11] Jie-Bo Hou, Xiaobin Zhu, Chang Liu, Kekai Sheng, Long-Huang Wu, Hongfa Wang, and Xu-Cheng Yin. Ham: Hidden anchor mechanism for scene text detection. *IEEE Transactions on Image Processing*, 29:7904–7916, 2020. 3

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 4, 5

[13] Jing Huang, Guan Pang, Rama Kovvuri, Mandy Toh, Kevin J. Liang, Praveen Krishnan, Xi Yin, and Tal Hassner. A multiplexed network for end-to-end, multilingual OCR. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[14] Chulmoo Kang, Gunhee Kim, and Suk I. Yoo. Detection and recognition of text embedded in online images via neural context models. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 4103–4110. AAAI Press, 2017. 1

[15] Jiachen Li, Yuan Lin, Rongrong Liu, Chiu Man Ho, and Humphrey Shi. Rsca: Real-time segmentation-based context-aware scene text detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2349–2358, 2021. 1

[16] Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, and Jian Yang. Shape robust text detection with progressive scale expansion network. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9328–9337, 2019. 2, 3, 6, 7, 8

[17] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, page 4161–4167. AAAI Press, 2017. 2, 3

[18] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690, 2018. 2

[19] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*. AAAI Press, 2020. 1, 3, 5, 6, 7, 8

[20] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):919–931, 2023. 1, 2, 3, 7, 8

[21] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recogn.*, 90(C): 337–345, 2019. 5

[22] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Computer Vision – ECCV 2018*, pages 19–35, Cham, 2018. Springer International Publishing. 7, 8

[23] Ben Mildenhall, Jonathan T. Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018. 4

[24] Zobeir Raisi, Mohamed A. Naiel, Georges Younes, Steven Wardell, and John S. Zelek. Transformer-based text detection in the wild. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3156–3165, 2021. 2

[25] Xuejian Rong, Chucai Yi, and Yingli Tian. Recognizing text-based traffic guide panels with cascaded localization network. In *Computer Vision – ECCV 2016 Workshops*, pages 109–121, Cham, 2016. Springer International Publishing. 1

[26] Zhiwen Shao, Yuchen Su, Yong Zhou, Fanrong Meng, Hancheng Zhu, B. Liu, and Rui Yao. Ct-net: Arbitrary-shaped text detection via contour transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023. 7, 8

[27] Jingqun Tang, Wenqing Zhang, Hongye Liu, MingKun Yang, Bo Jiang, Guanglong Hu, and Xiang Bai. Few could be better than all: Feature sampling and grouping for scene text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4563–4572, 2022. 1, 3, 7, 8

[28] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *Computer Vision – ECCV 2016*, pages 56–72, Cham, 2016. Springer International Publishing. 3

[29] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4229–4238, 2019. 2, 3, 6, 8

[30] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8439–8448, 2019. 3, 7, 8

[31] Xiaobing Wang, Yingying Jiang, Zhenbo Luo, Cheng-Lin Liu, Hyunsoo Choi, and Sungjin Kim. Arbitrary shape scene text detection with adaptive text region representation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6442–6451, 2019. 3

[32] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 7, 8

[33] Enze Xie, Yuhang Zang, Shuai Shao, Gang Yu, Cong Yao, and Guangyao Li. Scene text detection with supervised pyramid context network. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 2019. 2, 3

[34] Brandon Yang, Gabriel Bender, Quoc V. Le, and Jiquan Ngiam. Condconv: Conditionally parameterized convolutions for efficient inference. In *Neural Information Processing Systems*, 2019. 4, 5

[35] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1083–1090, 2012. 5

[36] Jian Ye, Zhe Chen, Juhua Liu, and Bo Du. Textfusenet: Scene text detection with richer fused features. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 516–522, 2020. 7, 8

[37] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Bo Du, and Dacheng Tao. Dptext-detr: Towards better scene text detection with dynamic points in transformer. In *AAAI*. AAAI Press, 2023. 1, 2, 3, 6, 7, 8

[38] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6978–6988, 2023. 7, 8

[39] Liu Yuliang, Lianwen Jin, Zhang Shuaitao, and Zhang Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv: Computer Vision and Pattern Recognition*, 2017. 3

[40] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9696–9705, 2020. 3, 7, 8

[41] Shi-Xue Zhang, Xiaobin Zhu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Adaptive boundary proposal network for arbitrary shape text detection. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1285–1294, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[42] Shi-Xue Zhang, Chun Yang, Xiaobin Zhu, and Xu-Cheng Yin. Arbitrary shape text detection via boundary transformer. *IEEE Transactions on Multimedia*, pages 1–14, 2023. 1, 2, 3, 5, 6, 7, 8

[43] Shi-Xue Zhang, Xiaobin Zhu, Lei Chen, Jie-Bo Hou, and Xu-Cheng Yin. Arbitrary shape text detection via segmentation with probability maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2736–2750, 2023. 2, 3, 4, 5, 7, 8

[44] Xiang Zhang, Yongwen Su, Subarna Tripathi, and Zhuowen Tu. Text spotting transformers. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9509–9518, 2022. 1

[45] Jinzhi Zheng, Ruyi Ji, Libo Zhang, Yanjun Wu, and Chen Zhao. Cmfn: Cross-modal fusion network for irregular scene text recognition. In *Neural Information Processing*, pages 421–433, Singapore, 2024. Springer Nature Singapore. 1

[46] Jinzhi Zheng, Libo Zhang, Yanjun Wu, and Chen Zhao. Bpdo: Boundary points dynamic optimization for arbitrary shape scene text detection. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5345–5349, 2024. 2, 3

[47] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Wangmeng Zuo, Haozhe Xie, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2482–2491, 2019. 4

[48] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: An efficient and accurate scene text detector. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2651, 2017. 2, 3

[49] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3122–3130, 2021. 7, 8