RSCA: Real-time Segmentation-based Context-Aware Scene Text Detection

Jiachen Li[1]*, Yuan Lin[3], Rongrong Liu[3], Chiu Man Ho[3], and Humphrey Shi[1,2]
[1]UIUC, [2]University of Oregon, [3]InnoPeak Technology
jiachenl@illinois.edu {yuan.lin,rongrong.liu,chiuman}@innopeaktech.com shihonghui3@gmail.com

Abstract

Segmentation-based scene text detection methods have been widely adopted for arbitrary-shaped text detection recently, since they make accurate pixel-level predictions on curved text instances and can facilitate real-time inference without time-consuming processing on anchors. However, current segmentation-based models are unable to learn the shapes of curved texts and often require complex label assignments or repeated feature aggregations for more accurate detection. In this paper, we propose RSCA: a Real-time Segmentation-based Context-Aware model for arbitrary-shaped scene text detection, which sets a strong baseline for scene text detection with two simple yet effective strategies: Local Context-Aware Upsampling and Dynamic Text-Spine Labeling, which model local spatial transformation and simplify label assignments separately. Based on these strategies, RSCA achieves state-of-the-art performance in both speed and accuracy, without complex label assignments or repeated feature aggregations. We conduct extensive experiments on multiple benchmarks to validate the effectiveness of our method. RSCA-640 reaches 83.9% F-measure at 48.3 FPS on CTW1500 dataset.

1. Introduction

In recent years, scene text detection methods based on deep neural networks have been widely adopted in both academia and industry. Following the development of object detection and segmentation, representations for text instances in scene images rely on instance-level and pixel-level features that are extracted by deep convolutional neural networks. Pixel-level text representation learning, which are also known as segmentation-based methods, starts from EAST [42] that removes anchors and makes multi-oriented text predictions directly from pixels to contours. Then, Textsnake [23] views text instances as sequences of ordered, overlapping disks and makes predictions on curved text. PSENet [36] encodes text spines on multiple scales and uses progressive scale expansion algorithm to reconstruct text instances. PAN [37] further enhances features with repeated feature fusion modules and DBnet [16] proposes differentiable binarization for boundaries of text instances. TextPerception [26] also makes order-aware segmentation with labels on heads, contours and tails for arbitrary-shaped text detection. These segmentation-based methods set a general encoder-decoder prototype for arbitrary-shaped scene text detection and reach state-of-the-art performance on multiple curved scene text detection benchmarks.

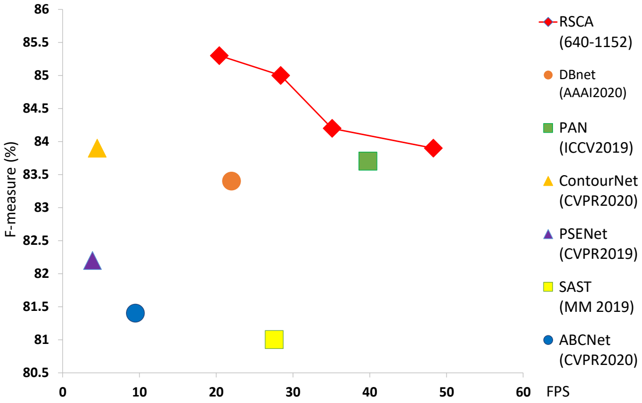

Figure 1: Comparisons between RSCA and other state-of-the-art arbitrary-shaped scene text detection methods on CTW1500 benchmark.

However, among these segmentation-based methods for arbitrary-shaped scene text detection, there are two main problems. Firstly, it lacks modeling of curved shapes of text instances since common convolution and pooling layers only operate with fixed geometric structures, which are designed for predictions of regular bounding boxes. For curved scene text detection, since most instances are irregular polygons, models need ability to learn spatial transformation to reconstruct text polygons from segmentation maps, which is ignored by current state-of-the-art segmentation-based models. Secondly, label generation and assignment rules are complex and exhausting for arbitrary-

[*] Work is done during an internship at InnoPeak Technology

shaped text instances. Different parts of text including heads, tails and boundaries are required to be generated manually and labeled as different classes. Text regions are shrunk with a fixed ratio as foregrounds for training process, which requires many hand-crafted parameters with grid search on different benchmarks to get state-of-the-art performances.

To tackle with these two problems, we propose two corresponding strategies: Local Context-Aware Upsampling (LCAU) and Dynamic Text-Spine Labeling (DTSL). For spatial transformation modeling, previous methods [12] [6] show that self-attention mechanism helps to model global pixel-to-pixel relation but is computationally expensive. Deformable convolution [4] [43] predicts kernel offsets but it shares weights on entire feature maps and is sensitive to parameter initialization. We propose a local context-aware upsampling module that generates an attention weight matrix separately but computes locally on feature maps during upsampling process, which is light-weight compared to global self-attention layer while more effective and efficient according to our experiments. For simplifying label generation and assignment, we propose a simple dynamic text-spine labeling method, which simply shrinks text regions with a gradually increasing ratio during training process. This brings no additional computational burden but learns representations for text regions from easy samples to hard samples. These two strategies help us to build RSCA: a Real-time Segmentation-based Context-aware model for arbitrary-shaped scene text detection, which achieves state-of-the-art performances on multiple benchmarks with real-time inference speed.

To validate effectiveness of our method, we conduct extensive experiments on multiple benchmarks with our RSCA model. In Figure 1, it shows that comparing with other state-of-the-art methods on arbitrary-shaped scene text detection, our RSCA achieves better performance with real-time inference on CTW1500 benchmark. More comparisons and experiments are presented in the following sections.

To summarize, our contributions are as follows:

- We analyze the problems of current segmentation-based models for arbitrary-shaped scene text detection: lack of spatial transformation modeling and complex label assignments.

- We propose RSCA: a real-time segmentation-based context-aware model for arbitrary-shaped scene text detection with local context-aware upsampling and dynamic text-spine labeling, which models local spatial transformation and simplifies labels assignments with dynamically increasing text-spine labels separately.

- We conduct extensive experiments on several benchmarks to validate effectiveness of our RSCA model which achieves state-of-the-art performances with real-time inference speed.

2. Related Works

In this section, we briefly review current scene text detection methods based on deep neural networks, including two main categories: anchor-based methods and segmentation-based methods.

Anchor-based Methods: Anchor-based methods mainly develop based on object detectors, which starts from Faster-RCNN [27] that firstly introduces anchors as pre-defined boxes for accurate regression. Then, one-stage methods SSD [20] employ anchors on feature pyramids and make predictions directly from anchors. Following their design, Textboxes [15] changes anchor scales and follows SSD to detect text instances. Textboxes++ [14] further employs quadrilateral regression for bounding boxes on multi-oriented text detection. RRD [17] decouples classification and regression branches for better multi-oriented text detection. To better handle arbitrary-shaped text detection, Mask TextSpotter [24] adds a segmentation branch for segmenting text instances from bounding boxes, which inherits from Mask RCNN [8]. ContourNet [38] proposes adopted RPN to generate more suitable anchors and segments contours for curved text detection. These anchor-based methods perform well on text detection with regular shapes, but lack robustness to arbitrary-shaped scene text detection, since both shape and aspect-ratio of the pre-defined anchors limit their potential for curved text detection.

Segmentation-based Methods: Segmentation-based methods mainly focus on pixel-level feature representation, which is suitable for arbitrary-shaped scene text detection since most text instances are curved. Following development of semantic segmentation, FCN [22] and U-Net [28] employ an encoder-decoder structure for pixel-level prediction. The encoder part is usually a deep feature extractor like ResNet [9] or VGG [30] and the decoder part is usually feature upsampling by bilinear interpolation or deconvolution layer. EAST [42] firstly removes anchors and make multi-oriented text instances prediction on pixels. Then, more methods [23] [36] [37] [16] [26] come out with focus on improving labeling accuracy and model assign. They push detection accuracy comparable to anchor-based methods on multiple scene text detection benchmarks. Among these segmentation-based methods, they set a baseline with FCN-like structure for pixel-level prediction and generate final detection results with grouping pixels to text instances. To ensure more accurate detection, they adopt repeated feature maps aggregation and complex label assignments, which could slow down the inference speed

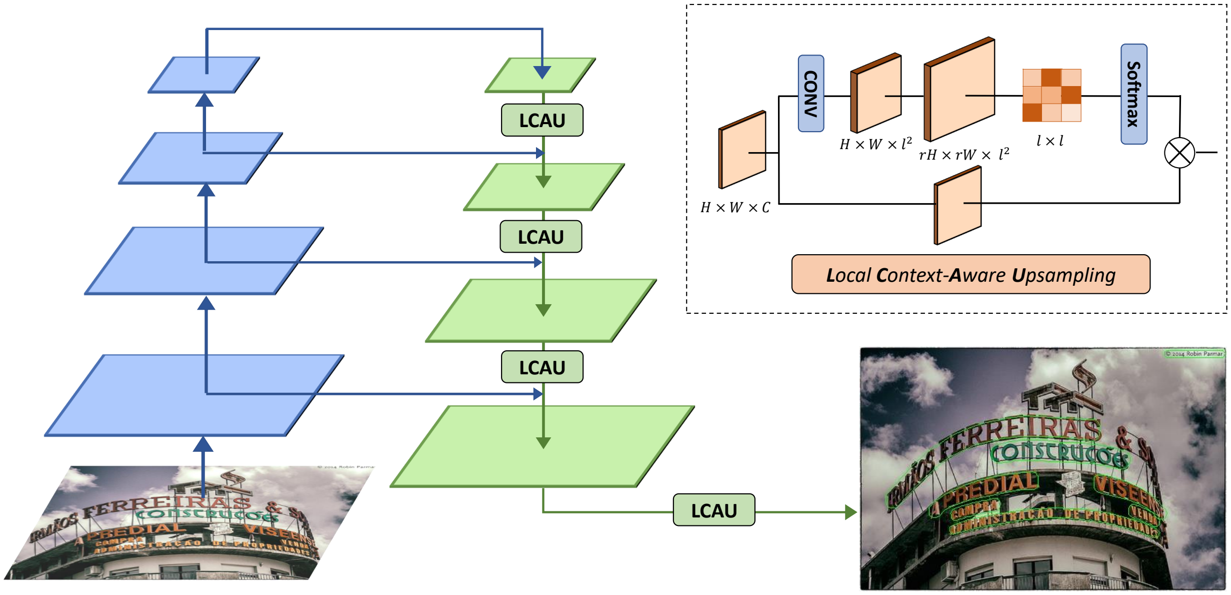

Figure 2: RSCA model architecture with illustrations on Local Context-Aware Upsampling.

and limit their applications in broader scenarios due to specific requirements for labels.

3. RSCA

In this section, we mainly introduce our RSCA pipeline from model architecture to training and inference process, including our two effective strategies: local context-aware upsampling and dynamic text-spine labeling.

3.1. Model Architecture

We show our RSCA model architecture in Figure 2. Specifically, we first adopt ResNet-50 [27] as our backbone for feature extraction with multiple levels of feature maps. It generates 5 stages of feature maps C_1, C_2, C_3, C_4, C_5 and the downsampling rate is 2^l for C_l feature map. Then, following the feature pyramid design from FPN [18], we select C_2, C_3, C_4, C_5 for upsampling and feature aggregation. C_1 is not selected for reducing computational burden. During the feature aggregation stage, we use local context-aware upsamling to model pixel-to-pixel relation in a local range on each feature maps and concatenate augmented C_2, C_3, C_4, C_5 feature maps to C_2 scale with channel aggregation. Finally, we upsample aggregated C_2 to the original scale of the input image to predict text areas and reconstruct text instances.

3.2. Local Context-Aware Upsampling

Upsampling is a common operation in modern deep neural networks for computer vision tasks like object detection and semantic segmentation, since it promotes feature maps from low resolution to high resolution and from semantic level to pixel level. For arbitrary-shaped scene text detection, we propose local context-aware upsampling to model spatial transformation in a local range during upsampling, which improves detection accuracy especially on curved text regions.

Upsampling Operators For previous works in scene text detection, the most common upsampling operators are nearest neighbor and bilinear interpolations, which do not require any additional parameters. In Learning Deconvolution Network [25], it proposes deconvolution layer which is an inverse operator of convolution layer. It is learnable but applies the same kernel across the entire feature maps. In ESPCN [29], it uses pixel shuffle as upsampling module which reshapes feature maps from the depth channel into width and height dimension. Our motivation is to model pixel-to-pixel relation in local range since arbitrary-shaped text are irregular and curved. Global self-attention [6] is a decent solution but it introduces too much additional computational burden on the global spatial attention and channel attention matrices. Motivated by DCN [4] [43], CARAFE [34] and dynamic filter [12], we propose local context-aware upsampling, which models

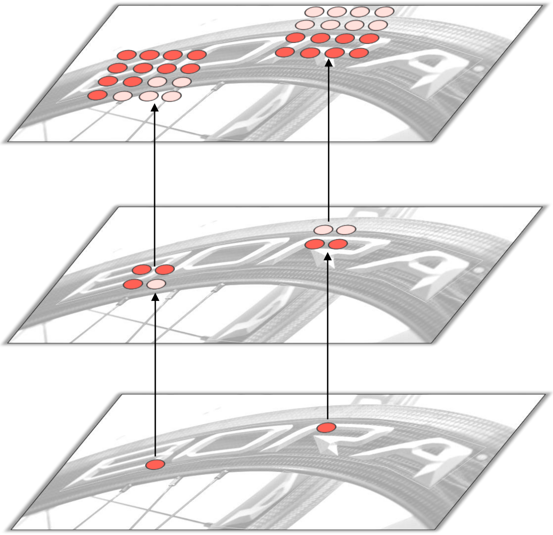

Figure 3: Illustrations with local context-aware upsampling on original images. Shallow red point refers to non-text-region prediction and bright red point refers to text-region prediction.

spatial transformation relation in a local range that does not bring too much computational burden.

Local Context-Aware Upsampling For a feature map $C \times H \times W$, after an upsampling operation, it becomes $C \times rH \times rW$ where r is the upsampling rate. In Figure 2, we display the entire feature processing flow of our local context-aware upsampling operation. On the weight matrix generation branch, we first apply a convolution layer with dimension $3 \times 3 \times C \times C^{'}$ where $C^{'} = l^2$, where l is the receptive field of pixels on feature map $C \times H \times W$ and now the dimension is $C^{'} \times H \times W$. Then, we apply nearest neighbor upsampling operation with scaling rate r and it becomes $l^2 \times rH \times rW$. On the depth channel, l^2 can be viewed as a weight matrix of coordinate (x, y) on feature map $C \times rH \times rW$. Motivated by dynamic filter, we also add a softmax operation to normalize the weight matrix and apply a local context-aware matrix multiplication with original feature map $C \times H \times W$. Finally, the dimension of feature maps becomes $C \times rH \times rW$. Local context-aware upsampling can replace any upsampling operation by adding only a small computational burden. In Figure 3, we make illustrations on original images, for classic nearest neighbor upsampling, predictions on text-region would activate fixed adjunct space on high-resolution feature maps, while local context-aware upsampling generates a local weight matrix that weakens activation of non-text context. More experiments with local context-aware upsampling are shown in ablation studies.

3.3. Dynamic Text-Spine Labeling

Label generation and assignment are important to scene text detection, especially on segmentation-based methods, since they dictate text regions for the model to learn from loss function. For arbitrary-shaped scene text detection, we propose a dynamic text-spine labeling method for label generation and assignment, which dynamically enlarges text-spine as labels during the training process. It provides more positive samples as training process goes from easy ones to hard ones and outperforms previous fixed text-spine labeling methods with cross-entropy loss.

Label Assignment for Segmentation-based Methods For segmentation-based methods like PSENet [36] and DBnet [16], they employ shrunk text instance masks as labels. For example, given a text instance S with a group of vertices $\sum_i^n P_i$, we can compute its perimeter L and area A of the original polygon. Then, the shrunk offset [33] is

$$D = \frac{A}{L}(1 - r^2)$$

where r is the shrink ratio and set to be 0.4 in DBnet but different discrete values in PSENet. For text instance S, it is dilated with the shrunk offset D to be S_d and regions of S_d are considered as text-spine labels for text. In loss function, a binary cross-entropy loss function with a ratio of positive to negative samples as $1 : 3$ is employed:

$$L = \sum_i^n y_i \log x_i + (1 - y_i) \log(1 - x_i)$$

which is similar to salient object detection that views foregrounds of text as labels and predicts text regions during inference process.

Dynamic Text-Spine Labeling Following the label generation from previous methods, there are two main drawbacks. Firstly, parameters of fixed shrink ratios vary with different datasets, which requires hand-crafted fine-tuning empirically. Secondly, splitting original text regions and label shrinking-boundary parts as negatives gives confusing signals to the scene text detector. To tackle with these two problems, we propose a dynamic text-spine labeling approach that enlarges text-spine labels during the training process. For the shrink ratio r, we set an initial value r_a and a final value r_b. As training process goes on, we have

$$r = r_a + (r_b - r_a) \times \frac{epoch}{max_{epoch}}$$

The expansion coefficient $\beta = \frac{epoch}{max_{epoch}}$, which is motivated by the poly learning rate decay policy, decays expo-

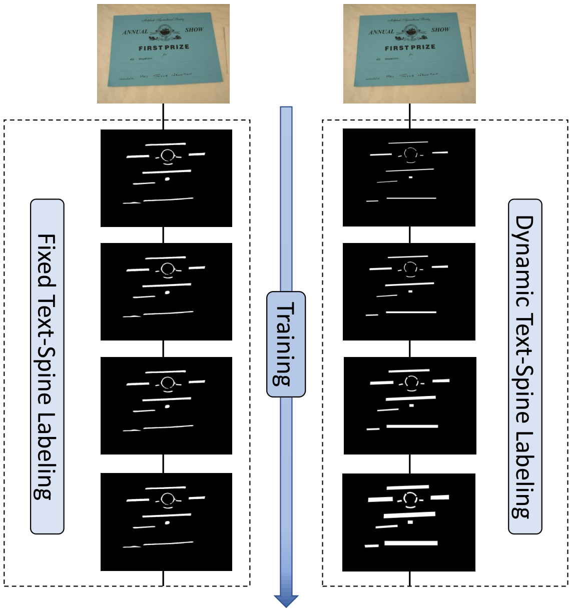

Figure 4: Illustrations between dynamic and fixed text-spine labeling during training process.

nentially as training continues. In Figure 4, we make illustrations of both fixed and dynamic text-spine labeling methods during training process. Under this dynamic text-spine labelling approach, our scene text detection model could learn text regions from easy samples to hard samples, which introduces improvements compared with the setting of best fixed shrink ratio. For loss functions, after experiments with different ones with hard examples mining [19], we choose the basic binary cross-entropy loss with ratio of positive to negative samples as $1:3$. More experiments with dynamic text-spine labeling and loss functions are shown in ablation studies.

3.4. Inference

During inference process, images are first resized to fixed size (l, l), which is similar to the cropped samples during the training process. l is set as 640 and 800 for our RSCA-640 and RSCA-800 model respectively. As shown in Figure 2, the resized image is input to the RSCA model and it outputs a segmentation probability map with text-spine regions after feature extraction and local context-aware upsampling. To reconstruct text instances, we follow same steps used in DBnet [16], which employs polygonal approximation algorithm to get independent text polygons, then dilates each individual text-spine according to its area A_{ts}, perimeter L_{ts} and offset

$$D_{ts} = \frac{A_{ts}}{L_{ts}} * d_{ts}$$

Method	Precision	Recall	F-measure	FPS
TextSnake [23]	85.3	67.9	75.6	-
NASK [2]	82.8	78.3	80.5	12
SAST [35]	85.3	77.1	81.0	27.6
CRAFT [1]	86.0	81.1	83.5	-
DBnet-1024 [16]	86.9	80.2	83.4	22
ABCNet [21]	83.8	79.1	81.4	9.5
PSENet [36]	84.8	79.7	82.2	3.9
ContourNet [38]	84.1	83.7	83.9	4.5
PAN-640 [37]	86.4	81.2	83.7	39.8
RSCA-640	87.2	80.8	83.9	**48.3**
RSCA-800	**87.2**	**82.9**	**85.0**	28.4

Table 1: Detection results on CTW1500 dataset. All results are collected from CTW1500 leaderboard. The number with dash is the height of input images and **bold** indicates best results.

Method	Precision	Recall	F-measure	FPS
CRAFT [1]	87.6	79.9	83.6	-
DBnet-800 [16]	87.1	82.5	84.7	32
ABCNet [21]	85.4	80.1	82.7	9.5
PSENet [36]	84.8	79.7	82.2	3.9
PAN-640 [37]	**89.3**	81.0	85.0	39.6
ContourNet [38]	86.9	**83.9**	**85.4**	3.8
RSCA-640	86.9	78.5	82.5	**40.3**
RSCA-800	86.6	83.3	85.0	30.4

Table 2: Detection results on Total-Text dataset. All results are collected from Total-Text leaderboard. Hyper-parameters of RSCA are adopted directly from CTW1500.

Here d_{ts} is the dilation ratio. After dilating text instances, we reshape both the image and detection results into the original shape and get the final results.

4. Experiments

In this section, we firstly introduce datasets and benchmarks that we use to validate the effectiveness of our method. Then, we show our experimental details including most hyper-parameters and hardware configurations. Furthermore, we compare our methods with other state-of-the-arts and present ablation study mainly on local context-aware upsampling and dynamic text-spine labeling.

4.1. Datasets

CTW1500: CTW1500 [41] is also known as SCUT-CTW1500, which is a text-line based arbitrary-shaped text dataset with both English and Chinese instances. It contains 1000 training images and 500 testing images. Text instances

Method	Precision	Recall	F-measure	FPS
EAST [42]	87.3	67.4	76.1	13.2
TextSnake [23]	83.2	73.9	78.3	1.1
RRD [17]	87.0	73.0	79.0	10.0
CRAFT [1]	88.2	78.2	82.9	8.6
DBnet-736 [16]	91.5	79.2	84.9	32
PAN-640 [37]	84.4	83.8	84.1	30.2
ContourNet [38]	86.9	83.9	85.4	3.8
RSCA-640	**92.8**	80.1	86.0	**52.5**
RSCA-800	91.5	**85.6**	**88.4**	28.9

Table 3: Detection results on MSRA-TD500 dataset. All results are collected from original papers. Hyper-parameters of RSCA are adopted directly from CTW1500.

Method	Precision	Recall	F-measure	FPS
EAST [42]	83.6	73.5	78.2	13.2
TextSnake [23]	84.9	80.4	82.6	1.1
DBnet-1152 [16]	**91.8**	83.2	**87.3**	12
RRD [17]	85.6	79.0	82.2	6.5
PSENet [36]	86.9	84.5	85.7	1.6
PAN-640 [37]	84.0	81.9	82.9	26.1
ContourNet [38]	87.6	**86.1**	86.9	3.5
RSCA-640	85.3	81.3	83.2	**32.9**
RSCA-800	87.2	82.7	84.9	23.3

Table 4: Detection results on ICDAR-2015 dataset. All results are collected both from ICDAR-2015 leaderboard and original papers. Hyper-parameters of RSCA are adopted directly from CTW1500.

are labeled with 14 points as polygons that can be described as arbitrary-shaped curve text.

Total-Text: Similar to CTW1500, Total-Text [3] is an arbitrary-shaped text dataset but with word-level label. It contains 1255 training images and 300 testing images. Word instances are labeled with 10 vertices as polygons for curved text detection.

ICDAR 2015: ICDAR 2015 [13] is commonly used for multi-oriented text detection. It contains 1000 training images and 500 testing images. All text regions are annotated by 4 vertices of quadrangle.

MSRA-TD500: MSRA-TD500 [40] is a multi-language dataset that includes 300 images for training and 200 images for testing with text-line level labels. Following previous methods, we also include HUST-TR400 [39] in the training set with 400 images.

SynthText: SynthText [7] is a synthetic dataset with 800000 images, which are synthesized on scene text with 8000 background images. SynthText is mainly used for pre-training our model.

Method	Precision	Recall	F-measure
Nearest	82.7	78.9	80.8
Bilinear	82.9	79.0	80.9
Deconvolution	82.9	79.5	81.2
Pixel Shuffle	83.2	78.8	80.9
Spatial Attention [6]	84.2	77.5	80.7
Channel Attention [6]	84.9	78.2	81.3
LCAU-FPN	85.8	77.9	81.8
LCAU-All	86.7	78.8	82.6

Table 5: Detection results on CTW1500 dataset with different upsampling operators.

4.2. Experimental Settings

Training and inference setting We build the whole RSCA model illustrated in Figure 2. At first, we pre-train all models on SynthText dataset for 2 epochs. Then, we fine-tune our models on each dataset for 1200 epochs with stochastic gradient descent (SGD). For each dataset, we set batch-size to 16 with synchronized batch normalization. For learning rate policy, we employ a poly learning rate decay in which the initial learning rate is multiplied by $(1 - \frac{epoch}{max_{epoch}})^{power}$, where the initial learning rate is set to be 0.007 and $power$ is set to be 0.9. We also use a weight decay of 0.0001 and a momentum of 0.9. Data augmentation are mainly listed as follows: (1) Images are randomly horizontally flipped and rotated in the range $[-10°, 10°]$; (2) Images are randomly reshaped with ratio $[0.5, 3.0]$ and then cropped by 640 × 640 samples for training efficiency. These training setups mostly follow previous methods in DBnet [16] and PSENet [36] for fair comparisons and quick setting up based on MegReader toolbox.

Hardware and software setting All models are trained based on 4 NVIDIA V100 GPUs and tested on a single V100 GPU in Ubuntu operating system. Our code is based on Pytorch 1.4.0 and CUDA 10.1. The RSCA framework is implemented based on MegReader toolbox. For those solutions with open-sourced codes [36][16][37] [21], we use their open-sourced models for comparisons and test them on a single V100 GPU. For those that are not open-sourced [38][2][35], we simply adopt the numbers of both performance and FPS from their papers.

4.3. Comparisons with State-of-the-arts

We conduct extensive experiments on two curved text detection benchmarks CTW1500 and Total-Text, two multi-oriented text detection benchmarks MSRA-TD500 and ICDAR-2015. All hyper-parameters are tuned based on CTW1500 and directly employed on other datasets.

Backbone	Precision	Recall	H-mean	Size(M)
ResNet-50	86.7	78.8	82.6	28.18
ResNet-101	86.0	79.9	82.8	47.18
Mobilenetv3	81.7	73.8	77.5	6.89
EfficientNet-b0	84.1	75.7	79.7	6.54
EfficientNet-b1	83.8	75.5	79.4	9.04
EfficientNet-b2	84.9	75.5	79.9	10.37

Table 6: Detection results on CTW1500 dataset with different backbones of RSCA. Size refers to model size including backbone and feature pyramids.

Loss function	Precision	Recall	H-mean
BCE loss	82.0	75.7	78.7
BCE loss++	83.2	78.7	80.9
Focal loss[19]	83.9	72.3	77.6

Table 7: Detection results on CTW1500 dataset with different loss functions for RSCA.

Feature Aggregations	F-measure	Model Size(M)
FPN-64c [18]	81.9	26.24
BiFPN-D1-64c [32]	81.8	26.40
BiFPN-D2-88c [32]	81.1	26.60
BiFPN-D3-112c [32]	80.4	26.88

Table 8: Detection results on CTW1500 dataset with different feature aggregations for RSCA.

Component	Time Cost (ms)
Mobilenetv3	18.66
Post-processing	10.65
Total	29.31

Table 9: Inference time of Mobilenetv3-based RSCA.

Curved Text Detection To tackle with curved text detection, we mainly perform experiments on CTW1500 and Total-Text datasets. All experiments with CTW1500 and Total-Text are shown in Table 1 and Table 2.

Multi-Oriented Text Detection To tackle with multi-oriented text detection, we mainly perform experiments on MSRA-TD500 and ICDAR-2015 datasets. All experimental results with MSRA-TD500 and ICDAR-2015 are shown in Table 3 and Table 4.

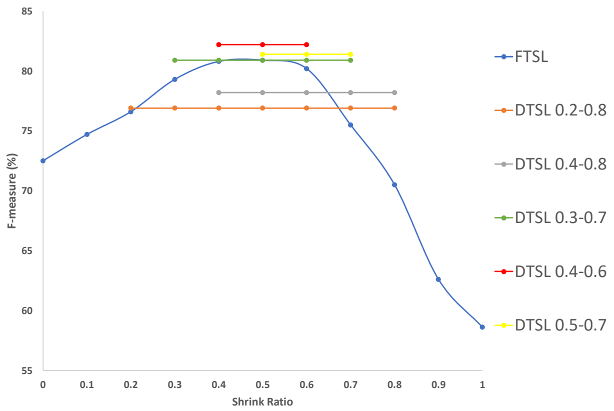

Figure 5: Comparisons between dynamic and fixed text-spine labeling with RSCA on CTW1500 dataset.

4.4. Ablation Study

Local Context-aware Upsampling We study the effectiveness of our local context-aware upsampling module by comparing it with different upsampling operators. We use an Imagenet [5] pre-trained ResNet-50 as backbone and build the whole model in Figure 2. In Table 5, we implement different upsampling methods and it shows that our local context-aware upsampling brings consistent improvements on precision and outperforms previous upsampling methods.

Dynamic Text-Spine Labeling To validate the effectiveness of dynamic text-spine labeling, we make comparisons with same models under different constant text-spine shrink ratios in Figure 5. FTSL is fixed text-spine labeling that uses a constant shrink ratio with RSCA model, which achieves best performances around 80.5% F-measure when shrink ratio is 0.4 or 0.5 on CTW1500 dataset. For dynamic text-spine labeling that shrinks from r_a to r_b, it shows that the best setting with $r_a = 0.4$ and $r_b = 0.6$ could achieve 82.1% F-measure with the same model.

Different Backbones We evaluate our RSCA with different backbones in Table 6. It includes ResNet with different depth of layers, EfficientNet [31] with different scales and MobileNetv3 [10]. All backbones are pre-trained on ImageNet [5] and it shows that our RSCA is compatible with these state-of-the-art light-weight backbones, which can be deployed on mobile devices.

Different Loss Functions We evaluate our RSCA with different loss functions in Table 7. It includes basic binary cross entropy loss (BCE loss), binary cross entropy loss with hard negative mining (BCE loss++) and focal loss [19]. It shows that binary cross entropy loss with hard

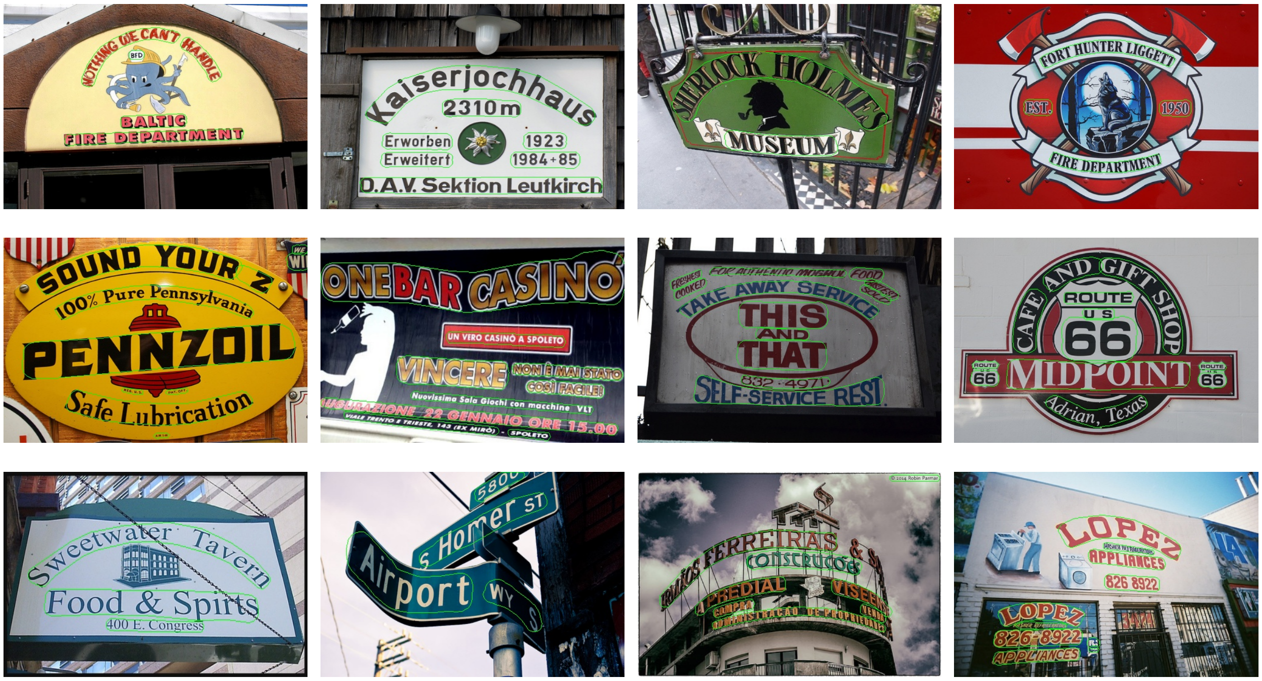

Figure 6: Visualization of detection results. Images are selected from test set of CTW1500 dataset.

negative mining outperforms others. Setting different loss functions usually require many hand-crafted fine-tuning parameters like binary thresholds and shrink ratios. The way to find the most suitable loss function for scene text detection remains an open problem for the community.

Different Feature Aggregations We evaluate our RSCA with different feature aggregations in Table 8. We mainly implement basic FPN [18] and BiFPN [32] with different repeated times. FPN-64c means the channels of all feature maps are 64 during the feature aggregation stage and the same to BiPFN. It shows that repeated feature aggregations with more scalings are not helpful for improving performance of scene text detection because the bottleneck now is the lack of modeling spatial transformation on local ranges.

Mobile Device Inference To analyze the performance of our model on mobile devices, we use the RSCA model based on Mobilenetv3 [10] backbone, which can achieve 34.11 FPS on a single V100 GPU and shown in Table 9. Since the inference time of Mobilenetv3 is 192ms on a Snapdragon 660 CPU, according to the AI-Benchmark [11]. An corresponding estimation of inference time of RSCA model on a Snapdragon 660 CPU based mobile phone like Redmi Note 7, would be around 302 ms, which is around 3 FPS. For a more powerful mobile CPU like Snapdragon 855, which takes 47ms for Mobilenetv3 inference according to the AI-Benchmark [11], our RSCA can reach an estimation inference time at 74ms, which is around 13.5 FPS. More accelerations and optimizations like model pruning and quantization can be further employed for mobile device deployment.

4.5. Visualization

We visualize some detection results of our RSCA model on test set of CTW1500 dataset in Figure 6. It shows that our model can accurately detect most of curved text instances in different scenes, including text on billboards, traffic signs and slogans.

5. Conclusion

In this paper, we first analyze the problems and bottlenecks of current segmentation-based models for arbitrary-shaped scene text detection, mainly on the lack of geometrical modeling and complex label assignments or repeated feature aggregations. To tackle these problems, we propose RSCA: a Real-time Segmentation-based Context-Aware model for arbitrary-shaped scene text detection with local context-aware upsampling and dynamic text-spine labeling, which models local spatial transformation and simplifies label assignments separately. Our experiments show that RSCA achieves state-of-the-art performances with real-time inference speed on multiple arbitrary-shaped scene text detection benchmarks.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9365–9374, 2019.

[2] Meng Cao and Yuexian Zou. All you need is a second look: Towards tighter arbitrary shape text detection. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2228–2232. IEEE, 2020.

[3] 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)*, volume 1, pages 935–942. IEEE, 2017.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[7] 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*, pages 2315–2324, 2016.

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1314–1324, 2019.

[11] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[12] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in neural information processing systems*, pages 667–675, 2016.

[13] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[14] 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.

[15] 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*, pages 4161–4167, 2017.

[16] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020.

[17] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[18] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[21] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2020.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[23] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 20–36, 2018.

[24] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 67–83, 2018.

[25] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.

[26] Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Text perceptron: Towards end-to-end arbitrary-shaped text spotting, 2020.

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[29] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[31] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019.

[32] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020.

[33] Bala R Vatti. A generic solution to polygon clipping. *Communications of the ACM*, 35(7):56–63, 1992.

[34] Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. Carafe: Content-aware reassembly of features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3007–3016, 2019.

[35] Pengfei Wang, Chengquan Zhang, Fei Qi, Zuming Huang, Mengyi En, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. A single-shot arbitrarily-shaped text detector based on context attended multi-task learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1277–1285, 2019.

[36] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[37] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[38] 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*, pages 11753–11762, 2020.

[39] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Transactions on Image Processing*, 23(11):4737–4749, 2014.

[40] 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. IEEE, 2012.

[41] Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017.

[42] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[43] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.