Scale-Residual Learning Network for Scene Text Detection

Yuanqiang Cai, Chang Liu, Peirui Cheng, Dawei Du, Libo Zhang, Weiqiang Wang, *Member, IEEE*, and Qixiang Ye, *Senior Member, IEEE*

***Abstract*— Detecting incidentally captured text in the wild remains an open problem due to challenging factors including unconstrained scenarios and large scale variation. In this paper, we establish a large-scale scene text detection dataset (LS-Text), containing 36, 000 images and 270, 783 text instances with various scales and complex scenarios, to promote the research of text detection. We propose a Scale-residual Learning Network (SLN) to deal with the scale variation problem in a progressive optimization manner. Specifically, we integrate both learnable feature concatenation and feature up-sampling operator. It can effectively eliminate the residuals between the outputs of SLN and ground-truth text instances by processing both the Feature Fusion Residuals (FFR) and the Scale Transformation Residuals (STR), simultaneously. By stacking multi-scale feature maps in a deep-to-shallow manner, SLN continuously optimizes feature representation by accumulating strong semantic information and rich texture details in a scale-residual learning way. Extensive experimental results on five challenging datasets demonstrate the state-of-the-art performance of the proposed SLN model, and the challenging aspects related to real-world scenarios of the proposed LS-Text dataset. Both the source code of SLN and the LS-Text dataset are available at https://github.com/SLN-Text-Detection.**

***Index Terms*— Text detection, scale-residual learning, LS-Text dataset.**

I. Introduction

TEXT detection in the wild is one of the fundamental tasks in computer vision. It plays an important role in various practical applications [1]–[4]. While many researchers view that detecting point-and-shoot text objects has been solved initially, detecting incidentally captured text in the wild remains challenging. The incidental text could be captured when the platform is in moving without considering the location of text and the complexity of scenarios. It is of large potential to unmanned vehicles and intelligent robots but greater challenges due to the large scale variation, unpredictable text locations, and clutter scenarios.

In the literature, a considerable number of benchmarks focus on point-and-shoot text (*e.g.*, ICDAR 2011 [5], ICDAR 2013 [6], MSRA-TD500 [7], and COCO-Text [8]). A couple of datasets have been released for incidental text (*e.g.*, SVT [9], [10] and ICDAR 2015 [11]). However, they have very small number of images, which seriously limits the capability about training and evaluating of sophisticated methods in real-world scenarios.

In this paper, we propose a large-scale scene text detection dataset, referred to LS-Text, to promote the development of the text detection community. LS-Text incorporates both point-and-shoot and incidental texts. It spans many challenges, *e.g.*, scale diversity, multiple directions, dense distribution, and unconstrained scenarios. It poses great challenges to state-of-the-art text detection approaches.

To address the challenge of the scale diversity, researchers often pray to the pyramid features. Existing methods can be roughly divided into two categories: separate multi-scale learning [12]–[15] and hybrid multi-scale learning [16]–[22]. The former uses different convolutional feature layers to predict text objects in corresponding scales (see Fig. 1(a)). It limits the ability of fusion and complementarity between feature layers with different scales. The latter uses the feature layers of all scales to predict text objects in each scale (see Fig. 1(b)). It learns similar salient features and objects on each scale layer, and weakens the scale-specific learning ability in each scale layer. Neither of them provides a learnable way for complementary scale matching.

Based on the above analysis, we propose a Scale-residual Learning Network (SLN) to progressively learn and accumulate features with corresponding scales from deep to shallow, just as shown in Fig. 1(c). Specifically, SLN is developed by posing learnable up-sampling and concatenation operators. It can extract discriminative features for text objects in proper scales by pursuing the minimization of scale-residuals between the representations of two operators rather than matching multi-scale text objects. In adjacent scales, by stacking the scale-specific feature maps in a deep-to-shallow manner, SLN

Manuscript received July 11, 2020; revised September 18, 2020; accepted October 2, 2020. Date of publication October 6, 2020; date of current version July 2, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFB1002203; in part by the NSFC projects under Grant 61976201, Grant 61836012, and Grant 61671427; in part by the NSFC Key Projects of International (Regional) Cooperation and Exchanges under Grant 61860206004; and in part by the Ningbo 2025 Key Project of Science and Technology Innovation with under Grant 2018B10071. This article was recommended by Associate Editor L. Zhang. *(Corresponding author: Weiqiang Wang.)*

Yuanqiang Cai, Peirui Cheng, and Weiqiang Wang are with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: caiyuanqiang15@mails.ucas.ac.cn; chengpeirui13@mails.ucas.edu.cn; wqwang@ucas.ac.cn).

Chang Liu and Qixiang Ye are with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: liuchang615@mails.ucas.ac.cn; qxye@ucas.ac.cn).

Dawei Du is with the Computer Science Department, University at Albany, SUNY, Albany, NY 12222 USA (e-mail: ddu@albany.edu).

Libo Zhang is with the Institute of Software, Chinese Academy of Sciences, Beijing 100864, China (e-mail: libo@iscas.ac.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2020.3029167

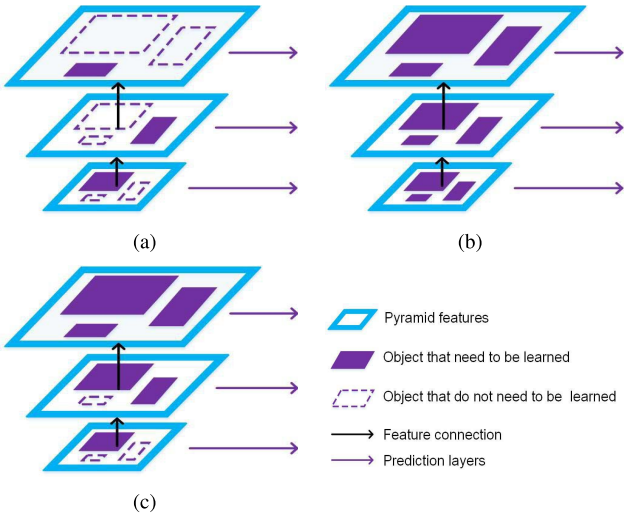

Fig. 1. Comparison between different multi-scale learning strategies. (a)Separate multi-scale learning. (b) Hybrid multi-scale learning. (c) Our scale-residual learning. The pyramid features are marked by the blue boxes, and propagated from deep layers to shallow ones.

can effectively transfer the optimized deep text semantic features to the adjacent shallow layer and continue to optimize. The contributions of this paper include:

- We establish a large-scale scene text detection benchmark, LS-Text, which contains 36,000 images and 270, 783 text instances and spans the challenges of point-and-shoot and incidental scene texts.
- We propose a Scale-residual Learning Network (SLN) to detect text in the natural scenes, where multiple scale-specific feature maps with both the learnable feature up-sampling and feature concatenation operators are stacked in a deep-to-shallow manner based on CNN.
- We propose a novel scale-residual learning strategy to eliminate both the Feature Fusion Residuals (FFR) caused by the concatenation operation and the Scale Transformation Residuals (STR) generated by the up-sampling operation, simultaneously.
- Extensive experiments on five datasets show that SLN achieves the best or competitive performance based on both accuracy and efficiency. Various ablation studies are provided to evaluate the effectiveness of each component of the proposed method.

II. Related Work

A. Text Detection Benchmarks

To evaluate text detection methods, many text detection datasets have been proposed, *e.g.*, ICDAR 2011 [5], ICDAR 2013 [6], MSRA-TD500 [7], and COCO-Text [8]. However, the common shortcoming of these datasets is that images are captured by point-and-shoot based cameras, which is limited in viewing angles, scales and quality in real-world scenarios.

Benefiting from flourishing global hand-held camera terminals (*e.g.*, smart mobile phone and glasses), text detection tasks have been pushed into unconstrained real-world scenarios. Different from point-and-shoot based cameras, text detection with moving camera has several advantages inherently, such as easy to apply, high mobility, changeable views and scales, and approaching the viewing of human vision. Meanwhile, it brings new challenges to existing detection technologies: a) *High density*, unconstrained cameras are flexible to capture scenes at wider view than point-and-shoot camera, leading to larger number of text instances. b) *Scale diversity*, text objects are usually presented randomly in various aspect ratios and sizes due to incidental view, which further increases the difficulty of text detection. c) *Blurring and occlusion*, text objects are motion blurring or partial occlusion due to unconstrained motion shooting. d) *Realtime issues*, the text detector should consider realtime issues and maintain comparable accuracy on real-world scenario application.

To study the above challenges, a few unconstrained text detection datasets are collected such as the Street View Text dataset (SVT) and the ICDAR 2015 incidental text dataset (ICDAR 2015) [11]. However, they only focus on a specific scene with limited images. For example, SVT with 350 images and ICDAR 2015 with 1, 500 images are collected in the street view or shopping mall. The community needs a more comprehensive and large-scale scene text detection in real-world scenarios for further boosting research on the related tasks. To this end, we establish a large scale challenging scene text detection benchmark, named LS-Text. It consists of 36, 000 images and 270, 783 text instances with various scales. To cover the challenges of point-and-shoot and incidental scene text detection, our dataset is captured by smart mobile phones or glasses in an unconstrained or focused manner in complex real-world scenarios.

B. Text Detection Methods

Early text detection methods for images and video frames are mainly based on conventional features and classifier, such as edge [23], Fourier and Laplacian transform [24], [25], slid stroke width transform feature and random forest [7], maximally stable extremal regions [26], and binary transform and SVM [27]. Recently, deep learning based methods refresh all the previous state-of-the-art records in text detection benchmarks. According to the different ways of box prediction, they can be generally grouped into regression-based and segmentation-based.

The regression-based text detection approaches are mainly based on general object detectors [28]–[30], as shown in Fig. 3 (a). They can be divided into single-stage and two-stage methods. The single-stage approaches are mainly based on SSD [29] and DSSD [30], the recent works [12], [14], [31]–[35] improve the design of default boxes, matching strategies, and convolution filter, to adapt to the texts with multiple sizes, orientations and various aspect ratios. To further optimize the box regression, inspired by Faster RCNN [28], many researchers use a two-stage box regression strategy. Jiang *et al.* [31] employ three regions of interests pooling of different sizes to match text objects with various aspects ratios, and merge them for further bounding box regression. Ma *et al.* [32] root in the Faster-RCNN framework, and use rotating region proposals instead of the standard axis-aligned

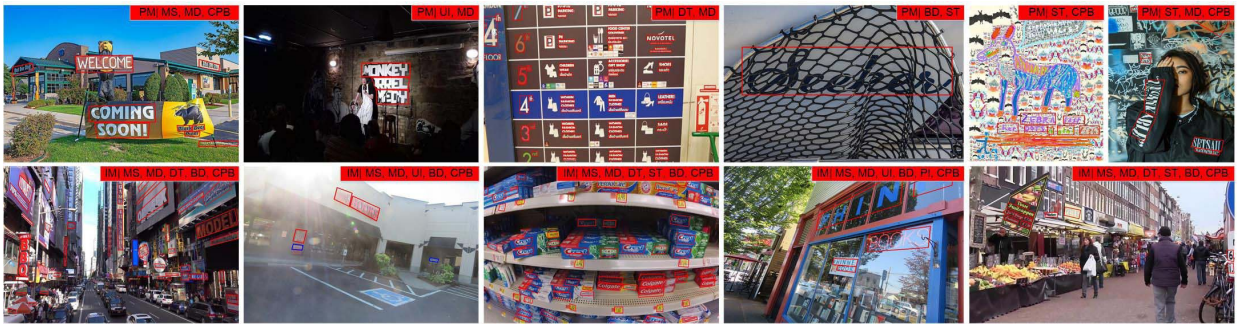

Fig. 2. Examples of the established LS-Text dataset. It spans various challenges of point-and-shoot and incidental shooting modes (shorted by PM and IM) in scene text detection. For instance, multiple scales (MS), multiple directions (MD), dense text distribution (DT), uneven illumination (UI), similar texture (ST), blurring and deformation (BD), perspective interference (PI), complex pose and background (CPB). The red region of the upper right corner describes the shooting mode and the challenges of text detection, (*mode*| *challenges*). Best viewed in color and zoom in.

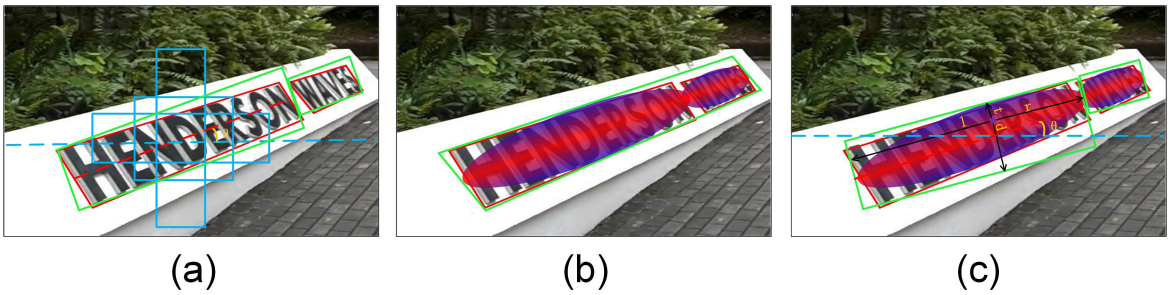

Fig. 3. The illustration of different text detection ways, *i.e.*, regression-based way in (a), post-processing segmentation-based way in (b), and direct prediction segmentation-based way in (c). The red boxes denote the ground-truth, and the green boxes denote the predictions. In (a), the blue boxes denote the hand-design anchors. For multi-direction texts, the prediction head also outputs an angle θ to rotate the axis-aligned detected boxes. In (b), and (c), the red-blue masks denote the heat segmentation map of text regions. Best view in color and zoom in.

bounding boxes to adapt to the texts with arbitrary orientations. Lyu *et al.* [13] use the corner points and position-sensitive regions to effectively locate arbitrary geometry and scale texts. Yang *et al.* [19] design an inception-text module based on Faster-RCNN [28] and introduce a deformable convolution filter to detect multi-orientation texts.

The segmentation-based text detection algorithm divides the pixels in the image into two categories, *i.e.*, background pixels, and text pixels. According to the different utilization ways of segmentation region [36]–[38], this kind of method can be divided two ways, *i.e.*, the post-processing segmentation way and the direct prediction segmentation way. As shown in Fig. 3 (b), the post-processing segmentation way [17], [18], [39] first generate the text heat map in the image, then use the minimum quadrilateral box to represent the high confidence text region. It is noted that this way is robust for text directions. But it is difficult to deal with the situation that the words are close to each other. The text objects that are very close to each other will be regarded as a text object. Therefore, the researchers introduce the box prediction strategy into the segmentation-based way [15], [16], [20]–[22], and further propose the direct prediction segmentation method to discriminate the adjacent text objects, as shown in Fig. 3 (c). Specifically, it outputs two kinds of prediction information at the same time, *i.e.*, the text segmentation map and the prediction boxes. Each pixel in the text segmentation map corresponds to a prediction quadrilateral bounding box. If the score of the pixel is higher than a confidence threshold, the prediction box corresponding to the pixel will be preserved. Finally, these reserved boxes are sent to the NMS operation. It should be noted that for multi-directional texts, as shown in Fig. 3 (c), the algorithm needs to output both the axis-aligned bounding box and the corresponding angle θ, and then rotate the axis-aligned bounding box according to the angle to match the inclined text objects.

Based on the direct prediction segmentation way, we propose a new segmentation-based text detection method, Scale-residual Learning Network (SLN), to detect scene texts. Our SLN integrates multiple scale-specific feature maps with both the learnable feature up-sampling and feature concatenation operators by stacking in a deep-to-shallow manner based on CNN. And a novel multi-scale learning way, scale-residual learning, is designed to eliminate both the Feature Fusion Residuals (FFR) caused by the concatenation operation and the Scale Transformation Residuals (STR) generated by the up-sampling operation, simultaneously.

C. Multi-Scale Processing Strategies

General objects, especially texts, are usually unconstrained presentation in various aspect ratios and sizes due to incidental scene and dynamic capture view. To deal with scale variation problem, two kinds of multi-scale processing strategies have been derived, *i.e.*, separate and hybrid scale processing.

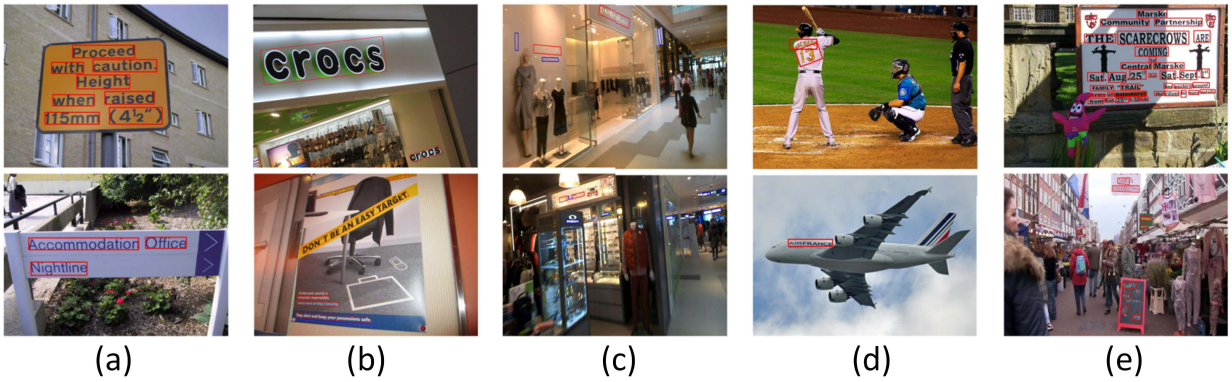

Fig. 4. Visual comparisons of different scene text detection datasets. (a) ICDAR 2013, (b) MSRA-TD500, (c) ICDAR 2015, (d) COCO-Text, and (e) LS-Text. The red or blue boxes are legible or illegible ground-truth text instances respectively. The detection results for illegible text instances are ignored in the final evaluation.

As shown in Fig. 1(a), the separate strategy uses different scale feature layers to capture scale-specific objects respectively. It has been widely applied to general object detection [30], [40], [41] and text detection [13], [15], [33].

As shown in Fig. 1(b), the hybrid strategy uses the feature layers with different scales to detect multiple scale objects simultaneously. It has been successfully applied to salient object detection [42], edge detection [43], [44], semantic segmentation [45], [46], text detection [17], [18].

Different from the previous approaches that only focus on one type of multi-scale process ways, *i.e.*, separate scale processing and hybrid scale processing, we propose a novel multi-scale processing way, *i.e.*, scale-residual learning. It incorporates the advantages of both the learnable feature fusion and scale transformation operations, with the objective to progressively learn and accumulate features with corresponding scales from deep to shallow, as shown in Fig. 1(c).

III. LS-Text Dataset

We establish a large-scale scene text detection dataset, named LS-Text, to further promote the research of text detection in the community. As shown in Fig. 2, the established LS-Text can be divided into two categories based on shooting modes. Point-and-shoot images captured by the point-and-shoot shooting mode (PM), in which most of the text objects appear near in middle position, in the upper line of Fig. 2. Incidental images captured by the incidental shooting mode (IM), the location of the text objects in it is random, in the down line of Fig. 2. Due to the unrestricted scenarios and the diversity of shooting modes, our LS-Text spans various challenges in text detection, *e.g.*, multiple scales, multiple directions, dense text distribution, uneven illumination, similar texture, blurring and deformable, perspective interference, complex pose and background.

A. Data Collection

We collect a large-scale scene text detection dataset with both incidental and point-and-shoot text scenes, to push the detection task to a new florescence. To this end, we invited 10 domain experts to define the standards of dataset collection. Specifically, our dataset contains 36, 000 images, which are selected from the Internet photo and video libraries, *i.e.*, flickr,[1] Google,[2] and YouTube.[3] For the downloaded images, we made three rounds of screening to remove the ordinary and simple images. For the downloaded videos, we first choose the video clips with text, then parse the selected video clips and reserve one frame every 20 frames. Besides, we also perform the same program as the images screening. Finally, the dataset is divided into training set with 24, 000 images, and testing set with 12, 000 images.

B. Data Annotation

For annotation, we invited 300 people to label the dataset for one month. With three rounds of double-check, the errors in annotation are reduced and revised as many as possible. Specifically, we have annotated 270, 783 text instances, including 180, 850 text instances in 24, 000 images of the training dataset and 89, 933 text instances in 12, 000 images of the testing dataset. Each text is labeled in the same quadrilateral way as in the ICDAR 2015 incidental text dataset [11]. Therefore, we use the same criteria as the ICDAR 2015 incidental text dataset [11], *i.e.*, *Recall* (R), *Precision* (P), and *F-score* (F).

C. Dataset Comparison

The qualitative and statistic comparison between the proposed LS-Text and other benchmarks are visualized in Fig. 4 and summarized in Table I. *#Image* denotes the number of images (training images/ testing images/ whole images). *#Text* denotes the number of text instances (training texts/ testing texts/ whole texts). The box-level of text *label* includes: character-level, word-level, line-level. There are two modes of *shoot* text object: point-and-shoot mode (*PM*) and incidental mode (*IM*). The text *density* indicates the average number of legible text instances (annotated by red boxes in Fig. 4) in each image. The *direction* of text consists of horizontal direction box (as shown in Fig. 4 (a) and 4 (d)) and multiple direction

[1]https://www.flickr.com/
[2]https://www.google.com/
[3]https://www.youtube.com/

TABLE I
STATISTIC COMPARISON BETWEEN OUR LS-TEXT AND OTHER BENCHMARKS

Dataset	#Image			#Text			Label			Shoot		Density	Direction
	train	test	all	train	test	all	character	word	line	PM	IM		
ICDAR 2013	229	233	462	848	1,095	1,943	√	√	-	√	-	3.7	Horizontal
MSRA-TD500	300	200	500	1,068	651	1,719	-	-	√	√	-	3.6	Multiple
ICDAR 2015	1,000	500	1,500	11,886	5,230	17,116	-	√	-	-	√	4.5	Multiple
COCO-Text	**43,686**	**20,000**	**63,686**	118,309	27,550	145,859	-	√	-	√	-	1.6	Horizontal
LS-Text (ours)	24,000	12,000	36,000	**180,850**	**89,933**	**270,783**	-	√	-	√	√	**6.9**	Multiple

box (as shown in Fig. 4 (b), 4 (c), and 4 (e)). We highlight the difference between our LS-Text dataset and other datasets as follows.

- **LS-Text *vs.* ICDAR 2013**. ICDAR 2013 only uses a limited number of images (229 training images and 233 testing images) to capture the point-and-shoot and horizontal texts in road signs or billboards (see Fig. 4 (a)). The number of images and text instances in our LS-Text is 77.9 times (*i.e.*, 36, 000 *vs.* 462) and 139.3 times (*i.e.*, 270, 783 *vs.* 1, 943) that in ICDAR 2013, respectively. The text density of our training set is 1.8 times (*i.e.*, 6.9 *vs.* 3.7) that of ICDAR 2013. In summary, LS-Text covers the text and image styles of ICDAR 2013 in many aspects.
- **LS-Text *vs.* MSRA-TD500**. The ground-truth of MSRA-TD500 is line-level annotation, and a bounding box contains one or more text instances (see Fig. 4 (b)). It contains 300 training images and 200 test images with high resolution. Our LS-Text focuses on finding text instances at the word-level (see Fig. 4 (e)), which is conducive to text recognition. Moreover, the number of images and annotation boxes in our LS-Text is 72 times (*i.e.*, 36, 000 *vs.* 500) and 157.5 times (*i.e.*, 270, 783 *vs.* 1, 719) that in MSRA-TD500, respectively. The annotation box density of our training set is 1.9 (*i.e.*, 6.9 *vs.* 3.6) times that of MSRA-TD500. LS-Text is more convenient for text recognition and more challenging for detection than MSRA-TD500.
- **LS-Text *vs.* ICDAR 2015**. The ICDAR 2015 is presented in the Challenge 4 of the 2015 Robust Reading Competition. It consists of a training set with 1000 images, a testing set with 500 images. The number of images and text instances in our LS-Text is 24 times (*i.e.*, 36, 000 *vs.* 1, 500) and 15.8 times (*i.e.*, 270, 783 *vs.* 17, 116) that in ICDAR 2015, respectively. The text density of our training set is 1.5 times (*i.e.*, 6.9 *vs.* 4.5) that of ICDAR 2015. Moreover, The ICDAR 2015 mainly focuses on detecting incidental text in shopping malls and street scenes (see Fig. 4 (c)), and our dataset includes both incidental text and point-and-shoot text in various real-world scenarios, *e.g.*, shopping mall, supermarket, pedestrian street, agricultural market, theme square. Extensive comparisons show that the challenges of both texts and scenes in LS-Text can completely cover that of ICDAR 2015, and push incidental scene text detection to a higher level.
- **LS-Text *vs.* COCO-Text**. The COCO-Text comes from the MS COCO dataset, which consists of 43, 686 training images, and 20, 000 testing images. It mainly focuses on point-and-shoot scene texts, and our LS-Text includes both two point-and-shoot and incidental texts in real-world scenarios. The number of text instances and the text density of our dataset is 1.9 times (*i.e.*, 270, 783 *vs.* 145, 859) and 4.2 times (*i.e.*, 6.9 *vs.* 1.6) that of COCO-Text, respectively. More importantly, the text instances in COCO-Text are mainly labeled with rectangular boxes (see Fig. 4 (d)), while the established LS-Text uses more compact quadrilateral boxes. LS-Text with richer text instances and more precise annotation boxes is more challenging and valuable than COCO-Text.

Based on comparisons between LS-Text and other datasets, we find that the proposed LS-text dataset is the first large-scale scene text detection dataset with point-and-shoot and incidental text objects at the same time.

IV. SCALE-RESIDUAL LEARNING NETWORK

A. Motivation

The separate multi-scale learning way (see Fig. 1(a)) focuses on responding to the text objects matched by its scale, while the hybrid multi-scale learning way (see Fig. 1(b)) focuses on responding to text objects of all scales in each scale feature layer. To compare and analyze the advantages and disadvantages of different multi-scale learning strategies, we implement them based on the same detection framework with the VGG16 backbone network, as shown in Fig. 8, and evaluate them via two ways, *i.e.*, *solution space* and *visual result*. We carry out experiments on two different scales datasets, *i.e.*, the small-scale ICDAR 2015 (1, 000 images for training and 500 images for testing) and the large-scale LS-Text (24, 000 images for training and 12, 000 images for testing).

In Fig. 5, we use the concept of *solution space* to show the learning ability of different scales. Specifically, we first make a statistic histogram on the scale of all texts in the testing subset. The red line is the fitting curve of the vertex of the scale histogram to represent the optimal solution space. Then, the detection results on all scales are represented as a curve. The closer the curve approaches the optimal solution, the better the detection results are. Finally, we visualize and analyze the multi-scale learning process of both separate learning and hybrid learning, Fig. 5(a) and 5(b). In Fig. 5(a), separate multi-scale learning produces scale-specific solution spaces on different scale layers. Because each of its scale layers only responds to the text objects matched by its scale (Fig. 1(a)), it learns the scale-specific salient features and the

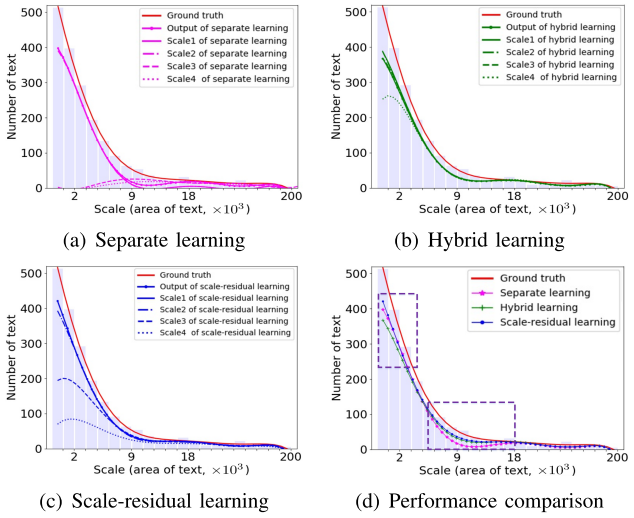

Fig. 5. Comparisons between the solution spaces of the detection result and the ground-truth. The area under each curve represents the *solution space* of the testing results on the ICDAR 2015 dataset. (a), (b) and (c) illustrate the comparisons between the solution spaces of the ground-truth and the detection results for separate learning, hybrid learning, and scale-residual learning, respectively. And (d) denotes the comparisons of detection results among three learning ways. The proposed scale-residual learning way in purple dashed box is obviously better than that of other ways. Best viewed in color and zoom in.

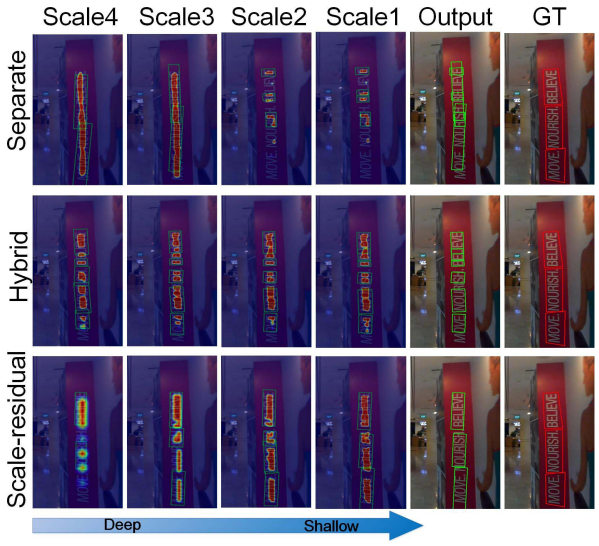

Fig. 6. Visual comparisons of different scale processing strategies. From the first (scale4) to the fourth column (scale1), we show the side-outputs from deep to shallow layers. Our proposed scale-residual learning method can improve the salient feature maps and boxes of text objects gradually. In contrast, other methods have no such characteristic. Best viewed in color.

text objects (see the first row in Fig. 6). Obviously, the separate way limits the ability of fusion and complementarity between feature layers with different scales. In Fig. 5(b), hybrid multi-scale learning generates a similar solution space on each scale feature layer. Since each of its scale layers matches text objects of all scales (in Fig. 1(b)), it learns similar salient features and objects on each scale layer from deep to shallow (see the second row in Fig. 6). Nevertheless, the hybrid way weakens the scale-specific learning ability in each scale layer. To overcome the aforementioned problems, we propose a progressive scale-residual learning strategy to reasonably model the association and complementation among multi-scale representations. More detailed comparison and analysis of three multi-scale learning ways are given below.

By comparing and analyzing the *solution spaces* of the detection results and the ground-truth, as shown in Fig. 5, we have two discoveries. Separate learning has strong scale-specific learning ability on different scales (see Fig. 5(a)), while hybrid learning has good scale fusion ability on each scale (see Fig. 5(b)). Fortunately, as shown in Fig. 5(c), we provide a new multi-scale learning strategy which can approach the optimal solution gradually via learning scale residuals. As shown in two purple dashed boxes of Fig. 5(d), the performance of the separate learning in small-scale text detection is better than that of the hybrid learning, but in medium-scale, the opposite is true. More importantly, the scale-residual learning strategy reaches the best performance on both small-scale and medium-scale text objects.

By observing and comparing the *visual results* between different multi-scale learning ways on each scale output, as shown in Fig. 6, we find that neither of them can detect the adjacent texts very well. In the deep feature layers with low resolution, multiple adjacent words will be treated as a large text because they are too close and they have similar texture. In the shallow feature layers with high resolution, a word can be detected as several objects because of the large spacing between characters. As shown in the third row in Fig. 6, our scale-residual learning strategy can improve and optimize the salient features and boxes of text objects progressively. In contrast, others have no such characteristic.

Based on the above analysis, we notice that multi-scale learning involves two important operations: feature fusion and scale transformation. The feature fusion can make full use of deep semantic features and shallow texture or edge features. The scale transformation provides conditions for the fusion of arbitrary scale features. Effective design and compatibility of two important operations will be the key to deal with the matching problem of multi-scale text objects. The following sections are dedicated to a detailed description of the proposed scale-residual learning strategy and scale-residual learning network.

B. Scale-Residual Learning Strategy

With the deep supervision both on the input and output of feature maps in different scale, Fig. 7 (a), the scale-residual of the ground-truth Y_i is computed. Formally, denoting the input of the i-th scale feature map as r_{i+1} and the additional mapping as $f(s_i)$, the deep supervision is expressed as

$$\begin{cases} r_{i+1} \rightarrow Y_i, \\ r_{i+1} + f(s_i) \rightarrow Y_i, \end{cases} \tag{1}$$

where r_{i+1} and $r_{i+1} + f(s_i)$ are the input and output of the i-th scale feature maps, both of them will be optimized to the same goal Y_i. Since $r_{i+1} + f(s_i) = r_i$, the second item of Eq. (1) can be represented by $r_i \rightarrow Y_i$. $f(s_i)$ is served as scale-residual estimation of s_i. We provide the shortcut connections between

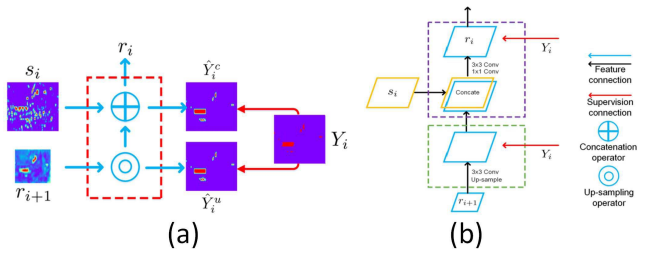

Fig. 7. Illustration of the simplified diagram (a) and the implementation (b) of the i-th scale feature maps. Y_i is the i-th scale ground-truth. $\hat{Y}_i^c$ and $\hat{Y}_i^u$ denote the representation of features in i-th scale layer after the concatenation operator and the up-sampling operator respectively. s_i denotes the low-level features with rich texture details, r_{i+1} and r_i denote the high-level features with strong semantic information.

the ground-truth and outputs on each scale, which implies a functional module for the "flow" of errors among different scales, and thus make it easier to fit complex prediction with higher adaptivity. To the extreme, if the input r_{i+1} is optimal, it will be easier to force $f(s_i)$ to zero than to fit it to ground-truth.

The goal of the scale-residual learning way is to eliminate the scale-residuals within its scale. The scale-residuals come from two aspects, *i.e.*, Feature Fusion Residuals (FFR) generated by the concatenation operation, and Scale Transformation Residuals (STR) produced by the up-sampling operation.

- **Feature fusion residuals**. Since the two groups of features received by the concatenation operation have different representations (*i.e.*, high-level semantic information inherited from r_{i+1} and low-level texture information come from s_i), directly concatenating the two groups of features will bring the network into fusion residuals. Similar to the lateral or skip connection of FPN [40] and DSS [42], we add one 1×1 convolution and one 3×3 convolution layers after the concatenation operation, to make it a learnable operator (see the purple dashed box in Fig. 7 (b)). Then, the feature layers with strong semantics and the feature layers with rich details are adequately fused by the supervision Y_i with the same scale, to eliminate the residuals caused by the concatenation operation.
- **Scale transformation residuals**. Since most of the words appear in groups to express unambiguous information, directly usiing the up-sampling operation will introduce the residuals of text-like features between adjacent text objects, which will lead to adjacent text objects being detected as a object. The previous works [15], [16], [20]–[22] consider little about the residuals caused by scale transformation. Similar to the solution of the concatenation operation, we add one 3×3 convolution layer after the up-sampling operation, to make it a learnable operator, as shown in the green dashed box of Fig. 7 (b). Then, we force the inherited deep semantic features r_{i+1} to present the i-th scale text features as much as possible after the up-sampling operator through the supervision Y_i, thus eliminating the residual caused by the up-sampling operation.

By cascading three scale-specific feature maps in the deep-to-shallow manner of VGG16 network, the scale-residual learning way is extended to common backbone networks. Eq. (1) is reformulated as

$$r_o = r_4 + \sum_{i=1}^{3} f_i(s_i), \tag{2}$$

where r_o is the final output of SLN, and $r_i \in [0, Y_i], i = 1, \ldots, 4$, are the inputs of multi-scale feature maps. Y_i is the i-th scale ground-truth. r_4 is supervised by the 4-th scale ground-truth Y_4 and it is regarded as an initial approximated output. After the Sigmoid operation, the scale-residual is always positive, *i.e.*, $f_i(s_i) \geq 0, i = 1, 2, 3$, where s_i from the pooling layers of VGG16 network. We have $Y_1 \geq r_o \geq r_1 \ldots \geq r_4$, which indicates that the scale-residual monotonically decreases in the deep-to-shallow of the stacking feature maps. It is vividly reflected in the gradual optimization of the solution space from deep layers to shallow ones, as shown in Fig. 5(c).

C. Network Implementation

Our SLN fuses the advantages of both the scale-specific adaptability and complementary based scale-residual learning, the construction process of which is summarized as follows.

The architecture of SLN is shown in Fig. 8. We set the training dataset to $D = \{(X_n, Y_n), n = 1, \ldots, N\}$, where $X_n = \{x_j, j = 1, \ldots, |X_n|\}$ denotes the n-th input image and $Y_n = \{y_j, j = 1, \ldots, |X_n|\}$ denotes the corresponding ground-truth. $y_j = (c_j, G_j)$ includes the confidence $c_j \in \{0, 1\}$ and the geometry information G_j for pixel x_j (if $c_j = 1$). We drop the subscript n for notational simplicity since we consider each sample independently. For simplicity, the collection of all the standard U-shape network layer parameters are denoted as $\mathbf{W}$. We use a backbone of VGG16 [47] as example, and 4 side outputs (s_1, s_2, s_3, and r_4) correspond to *pool2*, *pool3*, *pool4* and *pool5* of VGG16 network [47].

To eliminate the feature fusion residual in each scale feature map, each output of $\{\hat{Y}_i^c\}_{i=1}^3$ and $\hat{Y}_4$ of the SLN network is associated with a classifier for score map and a regressor for geometry. The corresponding weights of them can be denoted by $\mathbf{w_c} = (\mathbf{w}_c^1, \ldots, \mathbf{w}_c^4)$ and $\mathbf{w_r} = (\mathbf{w}_r^1, \ldots, \mathbf{w}_r^4)$. Thus, the objective function of SLN can be written as

$$L(\mathbf{W}, \mathbf{w_c}, \mathbf{w_r}) = \lambda_c \sum_{i=1}^{4} l_c(\mathbf{W}, \mathbf{w}_c^i) + \sum_{i=1}^{4} l_r(\mathbf{W}, \mathbf{w}_r^i), \tag{3}$$

where $l_c(\mathbf{W}, \mathbf{w}_c^i)$ and $l_r(\mathbf{W}, \mathbf{w}_r^i)$ denote the i-th scale loss function of a classifier and a regressor, and λ_c is a hyper-parameter to balance two losses.

To eliminate the scale transformation residual in each scale feature map, the output of each up-sampling operator $\{\hat{Y}_i^u\}_{i=1}^3$ of the SLN network is also associated with a classifier for score map and a regressor for geometry. The corresponding weights of them can be denoted as $\dot{\mathbf{w}}_c = (\dot{\mathbf{w}}_c^1, \dot{\mathbf{w}}_c^2, \dot{\mathbf{w}}_c^3)$ and $\dot{\mathbf{w}}_r = (\dot{\mathbf{w}}_r^1, \dot{\mathbf{w}}_r^2, \dot{\mathbf{w}}_r^3)$. Thus, the final objective function of the proposed SLN is defined as

$$\widetilde{L}(\mathbf{W}, \mathbf{w_c}, \mathbf{w_r}, \dot{\mathbf{w}}_c, \dot{\mathbf{w}}_r) = L(\mathbf{W}, \mathbf{w_c}, \mathbf{w_r}) + L(\mathbf{W}, \dot{\mathbf{w}}_c, \dot{\mathbf{w}}_r), \tag{4}$$

where $L(\mathbf{W}, \dot{\mathbf{w}}_c, \dot{\mathbf{w}}_r)$ has three scale representations including $\hat{Y}_1^u$, $\hat{Y}_2^u$, and $\hat{Y}_3^u$.

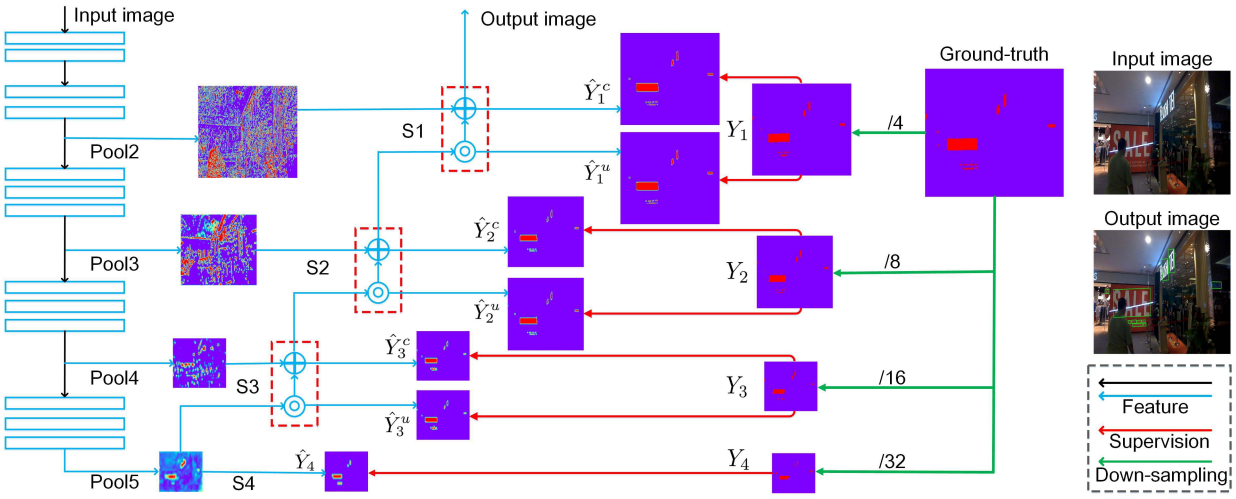

Fig. 8. The architecture of our Scale-residual Learning Network (SLN) that is built on the VGG16 network by stacking multi-scale feature maps in a deep-to-shallow manner. The down-sampling ground-truth Y_1, Y_2, Y_3, and Y_4 are $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$ the size of the ground-truth, respectively.

In the SLN model, for tackling the imbalance problem of positive and negative samples, we adopt the Dice loss to calculate the errors between the ground-truth and the prediction score map. The Dice loss is different from the cross-entropy loss [16], and it is proved to perform well in region segmentation tasks [48], [49] [13]. The $l_c(\mathbf{W}, \mathbf{w}_c^i)$ is abbreviated by l_c for notational simplicity, defined by

$$\text{ł}_c = 1 - \frac{2 * \sum_{i=1}^{|X_n|}(c_i c_i^*)}{\sum_{i=1}^{|X_n|}(c_i) + \sum_{i=1}^{|X_n|}(c_i^*)}, \tag{5}$$

where the sums run over the all $|X_n|$ pixels of the score map. c_i is the confidence value of pixel i in the ground-truth map. c_i^* is the confidence value of pixel i in the predicted score map.

Considering the loss for the geometry map should be scale-invariant and angle-sensitive, the IoU loss [50] is adopted to evaluate the difference between the predicted bounding box with axis alignment and the ground-truth of axis-aligned bounding box, the cosine function [16] is used to calculate the distance between the predicted angle and the ground-truth. l_r can be defined as follows.

$$l_r = \text{IoU}(\mathbf{R}, \mathbf{R}^*) + \lambda_\theta(1 - \cos(\theta, \theta^*)), \tag{6}$$

where $\mathbf{R}$ is the ground-truth of axis-aligned bounding box of the text object, and θ is the ground-truth of rotation angle of the text box, and $G = \{\mathbf{R}, \theta\}$. $\mathbf{R}^*$ and θ^* denote the prediction of text box and angle respectively.

D. Label Generation

The loss function of our SLN is composed of three atomic losses, *i.e.*, the segmentation loss (see Eq. 5), the axis-aligned bounding box loss (see the first half of the Eq. 6) and the text angle loss (see the second half of the Eq. 6). The ground-truth required by the proposed SLN is shown in Fig. 9 (a). Our SLN needs to learn three items [16], *i.e.*, the text core area (see the red region of Fig. 9 (a)), the axis-aligned bounding box generated by combining the point position and the corresponding four distances (see the blue point and line arrow of Fig. 9 (a)), and the corresponding angle of text quadrangle (see the two orange lines of Fig. 9 (a)).

The text core region, *i.e.*, the positive region of the text quadrangle on the feature map is designed to be roughly a shrunk region of the original text region, as shown in the red region of Fig. 9 (a). The generation process is given as follow. For a text quadrangle $Q = \{p_i | i \in 1, 2, 3, 4\}$, where $p_i = \{x_i, y_i\}$ are vertices on the text quadrangle in clockwise order. To shrink Q, we first calculate a *reference edge* e_i for each vertex p_i as

$$e_i = min(d(p_i, p_{(i \ mod \ 4)+1}), d(p_i, p_{((i+2) \ mod \ 4)+1})), \tag{7}$$

where $d(p_i, p_j)$ denotes the L_2 distance between p_i and p_j. We first shrink the two longer edges of a text quadrangle, and then the two shorter ones. For each pair of two opposing edges, we determine the "longer" pair by comparing the mean of their lengths. For each edge $e(p_i, p_{(i \ mod \ 4)+1})$, we shrink it by moving its two endpoints inward along the edge by $0.3e_i$ and $0.3e_{(i \ mod \ 4)+1}$ respectively. The region outside the text quadrangle is a negative samples. The region between the text quadrangle and the shrunk quadrangle will be ignored when calculating the segmentation loss, as shown in the blue region of Fig. 9 (a). It is will not be regarded as positive samples or negative samples.

The axis-aligned bounding box of the text quadrangle, is generated by combining a point position and the corresponding four distances (see the blue point and line with arrow in Fig. 9 (a)). The distance is obtained by calculating the vertical distance from the blue point to the four sides, *i.e.*, $e(p_1, p_2)$, $e(p_2, p_3)$, $e(p_3, p_4)$, and $e(p_4, p_1)$.

The angle of the axis-aligned bounding box, is computed according to the central axis of the text quadrangle and the horizontal line of the text image, see the two orange lines and the angle θ between them of Fig. 9 (a).

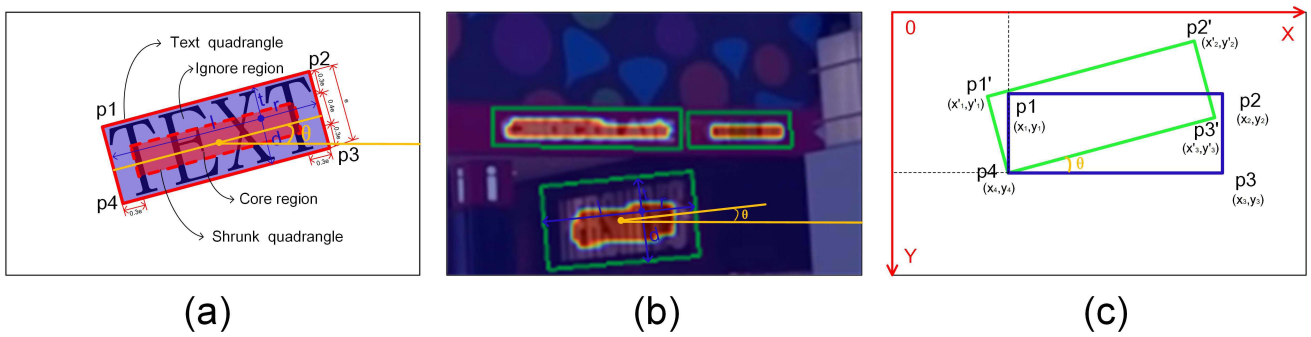

Fig. 9. The illustrations of the ground-truth generation (a), the visual prediction (b), and the axis-aligned bounding box rotation (c). In (a), the text quadrangle denotes the ground-truth annotation, the shrunk quadrangle is generated by shrinking the text quadrangle. In (b), the original image is overlaid by the heat map and the prediction quadrangle. In (c), the prediction quadrangle is obtained by rotating the axis-aligned bounding box with the corresponding angle θ.

E. Prediction

As shown in Fig. 9 (b), the prediction outputs of our SLN consist of three parts: the text segmentation region, the axis-aligned bounding box, and the angle. The latter two are associated with each point in the region. The final text quadrangle is obtained via three steps. First, we use a score threshold filter to delete the candidate pairs of the axis-aligned bounding box and the angle, both associated with low confidence pixels in the text segmentation region, *i.e.*, $c_i^s < \lambda_\mu$. c_i^s denotes the confidence score of pixel p_i in the text segmentation map, and λ_μ is the confidence threshold. Then, the retained candidate pairs will be transformed into a text quadrangle by rotating the axis-aligned bounding box based on angle θ, as shown in Fig. 9 (c). Finally, the Non-Maximum Suppression (NMS) operation is applied to remove the redundant candidate quadrangle boxes.

F. Discussion

U-Net [37] and FPN [40] are the common network structure that has been applied in many computer vision tasks. The detailed discussions about the difference between our SLN and them are given in the following.

SLN vs. U-Net. The essential differences between SLN and U-Net lie in four aspects. 1. Scale-progressive learning strategy. In order to adapt scale variance, SLN utilizes deep supervision with different resolution GT to match the scale of the corresponding outputs, while U-Net uses only one single supervision for all scales. Therefore, SLN provides an effective way to model the association and complementation among scale-progressive representations. 2. Scale-sensitive feature fusion. U-Net ignores the effect of semantic correlation caused by the up-sampling operation, so it is not sensitive to scale-residual transformation. SLN adds convolution with supervision to ease it which further reduces the scale-residual among the multi-scale outputs. 3. Sample unbalance processing. SLN uses dice-loss instead of cross-entropy loss to effectively handle the positive and negative sample unbalance problem. 4. Adjacent objects discrimination. SLN improves the segmentation-based text detection method by introducing the direct prediction strategy, and further proposes the direct prediction segmentation-based text detector to discriminate the adjacent text objects.

SLN vs. FPN. The structure of our SLN and that of the FPN is similar. However, we find that the scale division and learning modes of FPN are not suitable for the text objects. *For scale division*, the text with a large aspect ratio, which is divided based on the area of the text, is large. It should be matched to the deep-level feature layers. However, the short edge of the text may have disappeared in the deep-level feature layers due to the down-sampling operation. And the disappearance of the short edge will lead to the disappearance of the whole text object. It is impossible to learn an object deleted by the down-sampling operation in the deep-level feature layers. Therefore, our SLN adopts adaptive division mode. Specifically, we use the down-sampling operation to gradually reduce the resolution of the ground-truth, and the text ground-truth with different scales will automatically match to the corresponding scale feature layers. So the feature layers $S1$ with high resolution can match all scale objects, and the feature layers $S2$ with medium resolution can match large-scale and medium-scale text objects, and the feature layers $S3$ and $S4$ with low resolution can match large-scale text objects. *For multi-scale learning way*, FPN adopts the separate learning strategy. Concretely, the shallow-level feature layers learn the small scale objects, the middle-level feature layers learn the medium scale objects, and the deep-level feature layers learn the large scale objects. For example, text objects with the same height and different length may be independently supervised at different scale feature layers, it is unreasonable. Therefore, we propose a continuously supervised learning strategy. Specifically, *S3* and *S4* are used to learn large-scale texts. *S2* is used to learn medium-scale texts and continue to optimize large-scale texts. *S1* is used to learn small-scale texts and continue to optimize large-scale and medium-scale texts.

V. Experiments

A. Experimental Settings

The experiments are conducted on a single Titan Xp GPU and an Intel(R) Xeon(R) CPU E5-1603 v4 @ 2.80GHz. We train the network using the Adam Optimizer method. We take the exponential decay learning rate 0.94 after each $10,000$ iterations with the initial value 0.0001. In the loss function of SLN (3), we set the balancing factor as $\lambda_c = 0.01$. In the loss function of geometry map (6), we set the balancing

(a) (b) (c)

Fig. 10. Illustrated results of comparison before (b) and after (c) adding the supervision for scale transform residuals. (a) Images with ground-truth. The red ellipse indicates missed detection, the orange ellipse indicates that the adjacent text objects are not separated, and the red arrow indicates false detection.

factor as $\lambda_\theta = 20$. The batch size is set as 16. The NMS operation is conducted with the threshold 0.2. The confidence threshold λ_μ is set to 0.8. Data augmentation is important for improving the performance of deep neural network model, especially when the training instance is insufficient, as text detection. We crop regions with random scales from the training images and resize them to 512×512 pixels with an invariant aspect ratio.

B. Metrics

To compare the performance of detection methods on the LS-Text and ICDAR 2015 datasets, we use three popular metrics, *i.e.*, *Recall* (R), *Precision* (P), and *F-score* (F) [11]. Since the prediction box will be converted from quadrilateral to rectangular when evaluating the COCO-Text and ICDAR 2013 video datasets, the evaluation metrics of COCO-Text is based on [8], [13], and the evaluation metrics of ICDAR 2013 video is based on [6].

C. Ablation Study

We conduct several experiments to demonstrate the effectiveness of each component of our method. Specifically, we analyze the influence of different multi-scale learning strategies, the supervision of different scale-residual, and each scale output. We select two incidental text datasets, *i.e.*, the LS-Text dataset with large-scale text instances and images and the ICDAR 2015 dataset with small-scale text instances and images, because they can span all challenges.

TABLE II

COMPARATIVE RESULTS FOR DIFFERENT SCALE LEARNING STRATEGIES ON ICDAR 2015 AND LS-TEXT

Dataset	Multi-scale	R	P	F
ICDAR 2015	Separate	0.7357	0.7905	0.7621
	Hybrid	0.7530	0.8324	0.7907
	Scale-residual	**0.8074**	**0.8547**	**0.8304**
LS-Text	Separate	0.4538	0.4585	0.4562
	Hybrid	0.4563	0.5551	0.5009
	Scale-residual	**0.4988**	**0.5620**	**0.5285**

TABLE III

COMPARATIVE RESULTS OF SUPERVISION FOR DIFFERENT SCALE-RESIDUALS (*S-FFR* AND *S-STR*) ON ICDAR 2015 AND LS-TEXT

Dataset	S-FFR	S-STR	R	P	F
ICDAR 2015			0.7814	0.8263	0.8033
	√		0.7910	0.8439	0.8166
	√	√	**0.8074**	**0.8547**	**0.8304**
LS-Text			0.4493	0.5285	0.4857
	√		0.4888	0.5450	0.5154
	√	√	**0.4988**	**0.5620**	**0.5285**

1) Influence of Multi-Scale Learning Strategies: To verify the effectiveness of the proposed progressive multi-scale learning strategy, scale-residual learning, we compare three models (*i.e.*, *separate*, *hybrid*, *scale-residual*) with the VGG16 backbone, as shown in Fig. 8. The comparative results are reported in Table II.

For the large-scale LS-Text, F-score of SLN is 7.23% better than that of the separate model (*i.e.*, 52.85% *vs.* 45.62%), 2.76% better than that of the hybrid model (*i.e.*, 52.85% *vs.* 50.09%). For the small-scale ICDAR 2015, F-score of SLN is 6.83% better than that of the separate model (*i.e.*, 83.04% *vs.* 76.21%), 3.97% better than that of the hybrid model (*i.e.*, 83.04% *vs.* 79.07%). The improvement of all-round performance (recall, precision, and F-score) fully verifies the superiority of the proposed SLN on different scale datasets.

2) Influence of Supervision for Different Scale-Residuals: In order to verify the impact of different scale-residuals, *i.e.*, feature fusion residual (FFR) and scale transformation residual (STR), we compare them on the ICDAR 2015 and LS-Text datasets. The comparative results are reported in Table III. Our SLN with the supervision for both residuals achieves the best performance, in terms of all criteria. By eliminating the scale transformation residuals, SLN obtains an increase of 1.38% F-score on ICDAR 2015 (*i.e.*, 83.04% *vs.* 81.66%) and 1.31% F-score on LS-Text (*i.e.*, 52.51% *vs.* 51.54%). Moreover, in Fig. 10, we visualize the detection results of comparison before and after adding the supervision for STR. We find that it helps to distinguish adjacent words with the same scale and difficult text objects (blurred or vertical). SLN with the supervision for STR leads to the complementary of the same scale representations towards the optimization.

3) Comparison of Each Scale Output: To present the text scales learned at different scales, we evaluate several scale

TABLE IV
EXPERIMENTAL RESULTS FOR DIFFERENT SCALE OUTPUTS

Unit	Output	R	P	F	FPS
S1	$\hat{Y}_1^u$	0.8272	0.8761	0.8509	12.19
	$\hat{Y}_1^c$	0.8262	0.8773	0.8509	12.00
	$(\hat{Y}_1^u, \hat{Y}_1^c)$	**0.8286**	0.8763	**0.8518**	11.24
S2	$\hat{Y}_2^u$	0.7949	0.8749	0.8330	12.41
	$\hat{Y}_2^c$	0.8002	**0.8807**	0.8385	12.40
	$(\hat{Y}_2^u, \hat{Y}_2^c)$	0.7987	0.8704	0.8330	11.63
S3	$\hat{Y}_3^u$	0.5879	0.8653	0.7001	12.57
	$\hat{Y}_3^c$	0.5998	0.8799	0.7134	12.65
	$(\hat{Y}_3^u, \hat{Y}_3^c)$	0.6028	0.8605	0.7089	11.72
S4	$\hat{Y}_4$	0.3168	0.8738	0.4650	**12.89**

output of the model trained on the ICDAR 2015. Table IV reports the experimental results of different scale outputs on the ICDAR 2015 testing dataset. Four group experiments of different scale output (*S1*, *S2*, *S3*, *S4*) are summarized in Table IV. Each group results consist of three kinds of output results *i.e.*, the output after the concatenation operation $\hat{Y}_i^c$, the output after the up-sampling operation $\hat{Y}_i^u$, and the fused outputs of both $(\hat{Y}_i^c, \hat{Y}_i^u)$.

From the performance of different outputs, both recall and F-score of *S1* with higher resolution are better than those of *S3* with lower resolution in terms of $\hat{Y}_i^c$, $\hat{Y}_i^u$, and $(\hat{Y}_i^c, \hat{Y}_i^u)$ respectively. To better illustrate the performance difference of the text detector on different scales, we divide the detection results of four different scales (*i.e.*, *S*1, *S*2, *S*3, and *S*4) into three scale ranges (*i.e.*, *large-scale*, *medium-scale*, and *small-scale*) for quantitative statistics, as shown in Fig. 11. The left coordinates system is used as a statistical comparison of the number of text objects that are obtained by matching the detection results of the algorithm and the ground-truth in different scales. The right coordinates system is to measure the detection performance of different scale layers. The blue line represents the changing trend of the detection performance of the algorithm, and the performance gradually gets better from the *S*4 feature layer to the *S*1 feature layer.

The scale-residual learning ability in adjacent scale layers is demonstrated in Table IV. Each output has high precision. Recall is gradually improved from *S4* to *S1* (*i.e.*, 31.68% → 60.28% → 79.87% → 82.86%), which boosts the F-score from *S4* to *S1* (*i.e.*, 46.50% → 70.89% → 83.30% → 85.18%). The scale-residual learning ability in the same scale layer is also validated in Table IV, the difference between the F-score of $\hat{Y}^u$ and that of $\hat{Y}^c$ became smaller and smaller from *S3* to *S1* (*i.e.*, 1.33% → 0.55% → 0.0%). It also shows that $\hat{Y}^c$ and $\hat{Y}^u$ in each output are compatible. The successive constraints (as shown in Fig. 8) lead to the complementary of multi-scale representations towards the optimization. It can be proved by the cumulative graphs (as shown in Fig. 11) of detection texts with different scales.

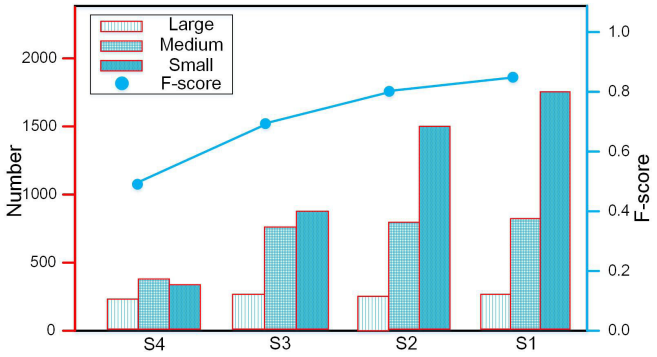

Fig. 11. Illustration of text objects (number of different sizes) and performance (F-score) detected at different scales.

TABLE V
COMPARISON ON THE LS-TEXT DATASET

Method	Backbone	R	P	F	FPS
PSENet [51]	ResNet50	**0.5118**	0.5519	**0.5311**	2.8
EAST [16]	VGG16	0.4338	0.5069	0.4675	**15.4**
R2CNN [31]	VGG16	0.4572	0.5498	0.4992	2.1
PixelLink [39]	VGG16	0.4818	0.5372	0.5080	4.6
SLN (ours)	VGG16	0.4988	**0.5620**	0.5285	13.9

D. Comparison With the State-of-the-Art Methods

1) Performance on LS-Text: To validate the applicability of SLN on the large-scale text detection dataset with point-and-shoot and incidental scene text, we compare SLN and several state-of-the-arts (*e.g.*, EAST [16], R2CNN [31], PixelLink [39], and PSENet [51]) on LS-Text dataset in Table V. As presented in Table V, SLN with a VGG16 network significantly outperforms EAST by 6.1% F-score (*i.e.*, 52.85% *vs.* 46.75%), but has only 1.5 FPS of speed loss (*i.e.*, 13.9 *vs.* 15.4). More importantly, for the methods with VGG16 backbone, SLN achieves the best performance (*i.e.*, 49.88% Recall, 56.20% Precision, and 52.85% F-score). The F-score of the proposed SLN is sightly lower than that of PSENet, but SLN's speed is 4.9 times that of PSENet (*i.e.*, 13.9 *vs.* 2.8). Notably, the excellent algorithms do not perform well on the established LS-Text, implying that the dataset is challenging.

2) Performance on ICDAR 2015: In following experiments of Section V-D, we first use COCO-Text to train the network with 10 epochs, and then real the training data is adopted to fine-tune the model until convergence. Table VI reports the performance of SLN and other state-of-the-arts on the ICDAR 2015 dataset. In terms of VGG16 backbone, SLN achieves the best F-score 85% the fastest speed 11.2 FPS. In terms of other backbones (*i.e.*, ResNet50 and PVANet), the F-score of our SLN is close to that of the state-of-the-art approachs [15], [51], but the speed is 3.1 and 9.2 times faster than that of [15] (*i.e.*, 14.7 *vs.* 4.8) and [51] (*i.e.*, 14.7 *vs.* 1.6) respectively. The speed of SLN is also faster than that of all the other approaches except EAST(a). By using a light weight neural network, PVANET, the speed of EAST(a) with PVANET reaches 16.8 FPS, but its F-score is 9% lower than that of our method (*i.e.*, 76% *vs.* 85%).

3) Performance on COCO-Text: The performance of SLN and a comparison with the state-of-the-art approaches on

TABLE VI

COMPARISON ON THE ICDAR 2015 DATASET. ∗ INDICATES MULTI-SCALE TESTING

Method	Backbone	R	P	F	FPS
EAST(a) [16]	PVANet	0.71	0.81	0.76	**16.8**
EAST(b)* [16]	PVANet2x	0.78	0.83	0.81	-
TextSpotter [21]	PVANet	0.83	0.84	0.83	-
FOTS [20]	ResNet50	0.82	0.89	0.85	7.8
IncepText [19]	ResNet50	0.81	0.91	0.85	-
TextSpotter [15]	ResNet50	0.81	**0.92**	**0.86**	4.8
PSENet [51]	ResNet50	**0.85**	0.87	**0.86**	1.6
SLN (ours)	ResNet50	0.83	0.88	0.85	14.7
Zhang *et al.* [17]	VGG16	0.43	0.71	0.54	-
Yao *et al.* [18]	VGG16	0.59	0.72	0.65	1.6
DMPNet [34]	VGG16	0.68	0.73	0.70	-
SegLink [12]	VGG16	0.73	0.77	0.75	-
WordSup [52]	VGG16	0.77	0.79	0.78	2
R2CNN [31]	VGG16	0.80	0.87	0.83	-
Textboxes++*[33]	VGG16	0.79	0.88	0.83	2.3
PixelLink [39]	VGG16	0.82	0.86	0.84	3.0
TextSnake [22]	VGG16	0.80	0.85	0.83	1.1
RRD* [35]	VGG16	0.80	0.88	0.84	-
Lyu *et al.** [13]	VGG16	0.80	**0.90**	0.84	3.6
SLN (ours)	VGG16	**0.83**	0.88	**0.85**	**11.2**

TABLE VII

COMPARATIVE RESULTS ON THE COCO-TEXT DATASET. ∗ INDICATES MULTI-SCALE TESTING

Method	Backbone	R	P	F	FPS
Yao *et al.* [18]	VGG16	0.271	0.4323	0.3331	-
WordSup [52]	VGG16	0.309	0.452	0.368	1.9
SSTD [53]	VGG16	0.31	0.46	0.37	-
Lyu *et al.* [13]	VGG16	0.262	0.699	0.381	-
EAST [16]	VGG16	0.324	0.5039	0.3945	-
Lyu *et al.** [13]	VGG16	0.324	0.619	0.425	-
SLN (ours)	VGG16	**0.4167**	0.5288	**0.4661**	**16.9**

COCO-Text are listed in Table VII. SLN outperforms the state-of-the-art approach [13] by 4.11% F-score (*i.e.*, 46.61% *vs.* 42.5%). Besides, SLN also surpasses Yao *et al.* [18], WordSup [52], SSTD [53], and EAST [16] by 13.3%, 9.8% 9.6%, and 7.2% F-score, respectively. Meanwhile, our SLN is quite efficient with the running speed of 16.9 FPS. Additionally, the proposed method obtains the best recall and F-score in the comparison experiments, therefore it can reveal that our method holds the strong ability for capturing texts.

4) Performance on ICDAR 2013 Video: To evaluate the transferability of SLN, we run it on the ICDAR 2013 video text dataset [6]. The video texts involve motion blur, extreme aspect ratio, multiple orientations, and low resolution. Comparative results are listed in Table VIII. From the detection results of the spatial information based approaches, we can find that the performances of SLN are higher than that of other approaches in terms of recall, precision, and F-score. Specifically, SLN surpasses Epshtein *et al.* [54], Khare *et al.* [55], Yin *et al.* [26], and EAST [16] by 33.3%, 24.9% 17.6%, and 12.8% F-score, respectively. Compared with all approaches, the F-score of the proposed method is 6.55% higher than that of [56] (*i.e.*, 69.20% *vs.* 62.65%). Moreover, SLN can run at 11.3 FPS and maintain the F-score with 69.20%.

TABLE VIII

COMPARISON ON THE ICDAR 2013 VIDEO DATASET. "S&T" DENOTES THAT THE APPROACH IS BASED ON BOTH SPATIAL AND TEMPORAL INFORMATION

Method	S&T	R	P	F	FPS
Zhao *et al.* [57]	✓	0.4630	0.4702	0.4665	-
Wang *et al.* [58]	✓	0.5174	0.5834	0.5451	-
Wang *et al.* [56]	✓	0.5867	0.7190	0.6265	-
Epshtein *et al.* [54]		0.3253	0.3980	0.3594	-
Khare *et al.* [55]		0.4760	0.4140	0.4430	-
Yin *et al.* [26]		0.5473	0.4862	0.5156	-
EAST [16]		0.5322	0.6413	0.5644	-
SLN (ours)		**0.6055**	**0.8073**	**0.6920**	11.3

TABLE IX

COMPARATIVE RESULTS ON THE ICDAR 2017 MLT DATASET

Method	Backbone	R	P	F	FPS
TH-DL [59]	-	0.35	0.68	0.46	-
He *et al.* [60]	-	0.58	0.77	0.66	-
FOTS [20]	ResNet50	0.58	**0.81**	0.67	-
Border [61]	ResNet50	0.61	0.74	0.67	-
LOMO [62]	ResNet50	0.61	0.79	0.69	-
SPCNET [63]	ResNet50	**0.67**	0.73	**0.70**	-
Lyu *et al.* [13]	VGG16	0.56	**0.84**	0.67	-
SLN (ours)	VGG16	**0.61**	0.76	**0.68**	6.4

5) Performance on ICDAR 2017 MLT: we run SLN on the ICDAR 2017 Multi-lingual scene text (MLT) [59] to further evaluate its transferability from single-English text objects to multi-lingual text objects. MLT consists of 9 languages (*i.e.*, *Arabic, Latin, Chinese, Japanese, Korean, Bangla, Symbols, Mixed, None*). Note that *None* is not one of the other eight script classes. Each text is labeled with a quadrangle box. As present in Table IX, the performance of our SLN is already larger than that of the majority of the existing text detectors. Specifically, SLN surpasses TH-DL [59], He *et al.* [60], FOTS [20], Border [61], Lyu *et al.* [13] by 22% F-score (46% vs. 68%), 2% F-score (66% vs. 68%), 1% F-score (67% vs. 68%), 1% F-score (67% vs. 68%), and 1% F-score (67% vs. 68%) respectively. The proposed SLN is slightly lower than LOMO [62] (68% vs. 69%) and SPCNET [63] (68% vs. 70%). In terms of VGG16 backbone, the proposed method achieves the best performance of 68% F-score compared with Lyu's method [13] (67% vs. 68%). It is noted that the proposed SLN is only one method to release the FPS with 6.4.

Extensive experiments on the above five datasets demonstrate that the proposed Scale-residual Learning Network (SLN) is effective for text detection in real-world scene, and achieves the best or competitive performance based on both accuracy and efficiency. The proposed LS-Text dataset is challenging as an evaluation benchmark of state-of-the-art text detection methods.

VI. CONCLUSION

In this paper, we investigated the problem about multi-scale text detection in the wild. We released a new large-scale scene text detection dataset (*i.e.*, LS-Text) with challenges related to real-world scenarios, which was verified to be a

good touchstone of state-of-the-art methods. We proposed a new segmentation-based text detector, Scale-residual Learning Network (SLN), which can model both the feature fusion residual and scale transformation residual of convolutional feature. SLN demonstrated great potential to scale-related computer vision tasks due to its adaptability to text scales, strong ability for feature transmission, and its simple and effective designed architecture.

REFERENCES

[1] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.

[2] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[3] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 19–36, Feb. 2016.

[4] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," 2018, *arXiv:1811.04256*. [Online]. Available: http://arxiv.org/abs/1811.04256

[5] D. Karatzas, S. R. Mestre, J. Mas, F. Nourbakhsh, and P. P. Roy, "ICDAR 2011 robust reading competition–challenge 1: Reading text in born-digital images (Web and Email)," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1485–1490.

[6] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

[7] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1083–1090.

[8] A. Veit, T. Matera, L. Neumann, J. Matas, and S. J. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," 2016, *arXiv:1601.07140*. [Online]. Available: https://arxiv.org/abs/1601.07140

[9] K. Wang and S. J. Belongie, "Word spotting in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 591–604.

[10] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[11] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.

[12] B. Shi, X. Bai, and S. Belongie, "Detecting oriented text in natural images by linking segments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3482–3490.

[13] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7553–7563.

[14] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 4161–4167.

[15] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 71–88.

[16] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2642–2651.

[17] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai, "Multi-oriented text detection with fully convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4159–4167.

[18] C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao, "Scene text detection via holistic, multi-channel prediction," 2016, *arXiv:1606.09002*. [Online]. Available: https://arxiv.org/abs/1606.09002

[19] Q. Yang, M. Cheng, W. Zhou, Y. Chen, M. Qiu, and W. Lin, "IncepText: A new inception-text module with deformable PSROI pooling for multi-oriented scene text detection," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1071–1077.

[20] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: Fast oriented text spotting with a unified network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5676–5685.

[21] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end TextSpotter with explicit alignment and attention," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5020–5029.

[22] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "TextSnake: A flexible representation for detecting text of arbitrary shapes," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 19–35.

[23] M. R. Lyu, J. Song, and M. Cai, "A comprehensive method for multilingual video text detection, localization, and extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 243–255, Feb. 2005.

[24] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New Fourier-statistical features in RGB space for video text detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1520–1532, Nov. 2010.

[25] P. Shivakumara, R. P. Sreedhar, T. Q. Phan, S. Lu, and C. L. Tan, "Multioriented video scene text detection through Bayesian classification and boundary growing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 8, pp. 1227–1235, Aug. 2012.

[26] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust text detection in natural scene images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 970–983, May 2014.

[27] K. S. Raghunandan, P. Shivakumara, S. Roy, G. H. Kumar, U. Pal, and T. Lu, "Multi-Script-Oriented text detection and recognition in Video/Scene/Born digital images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1145–1162, Apr. 2019.

[28] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[29] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[30] C. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," 2017, *arXiv:1701.06659*. [Online]. Available: https://arxiv.org/abs/1701.06659

[31] Y. Jiang *et al.*, "R2CNN: Rotational region CNN for orientation robust scene text detection," 20147, *arXiv:1706.09579*. [Online]. Available: https://arxiv.org/abs/1706.09579

[32] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.

[33] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[34] Y. Liu and L. Jin, "Deep matching prior network: Toward tighter multi-oriented text detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3454–3461.

[35] M. Liao, Z. Zhu, B. Shi, G.-S. Xia, and X. Bai, "Rotation-sensitive regression for oriented scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5909–5918.

[36] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351. Springer, 2015, pp. 234–241.

[38] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," in *IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[39] D. Deng, H. Liu, X. Li, and D. Cai, "Pixellink: Detecting scene text via instance segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6773–6780.

[40] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[41] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5244–5252.

[42] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5300–5309.

[43] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[44] W. Ke, J. Chen, J. Jiao, G. Zhao, and Q. Ye, "SRN: Side-output residual network for object symmetry detection in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 302–310.

[45] G. Lin, A. Milan, C. Shen, and I. Reid, "RefineNet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5168–5177.
[46] M. A. Islam, M. Rochan, S. Naha, N. D. B. Bruce, and Y. Wang, "Gated feedback refinement network for coarse-to-fine dense semantic image labeling," 2018, *arXiv:1806.11266*. [Online]. Available: https://arxiv.org/abs/1806.11266
[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[48] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 565–571.
[49] J. Zhang, X. Shen, T. Zhuo, and H. Zhou, "Brain tumor segmentation based on refined fully convolutional neural networks with a hierarchical dice loss," 2017, *arXiv:1712.09093*. [Online]. Available: http://arxiv.org/abs/1712.09093
[50] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "UnitBox: An advanced object detection network," in *Proc. ACM Multimedia Conf. MM*, 2016, pp. 516–520.
[51] W. Wang, W. Wang, W. Hou, R.-Z. Liu, T. Lu, and J. Yang, "Shape robust text detection with progressive scale expansion network," 2019, *arXiv:1806.02559*. [Online]. Available: https://arxiv.org/abs/1806.02559
[52] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "WordSup: Exploiting word annotations for character based text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4950–4959.
[53] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3066–3074.
[54] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.
[55] V. Khare, P. Shivakumara, and P. Raveendran, "A new histogram oriented moments descriptor for multi-oriented moving text detection in video," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 7627–7640, Nov. 2015.
[56] Y. Wang, L. Wang, and F. Su, "A robust approach for scene text detection and tracking in video," in *Proc. Pacific-Rim Conf. Multimedia*, 2018, pp. 303–314.
[57] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 790–799, Mar. 2011.
[58] L. Wang, Y. Wang, S. Shan, and F. Su, "Scene text detection and tracking in video with background cues," in *Proc. ACM Int. Conf. Multimedia Retr.*, Jun. 2018, pp. 160–168.
[59] N. Nayef *et al.*, "ICDAR2017 robust reading challenge on multi-lingual scene text detection and script Identification–RRC-MLT," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 1454–1459.
[60] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Multi-oriented and multi-lingual scene text detection with direct regression," *IEEE Trans. Image Process.*, vol. 27, no. 11, pp. 5406–5419, Nov. 2018.
[61] C. Xue, S. Lu, and F. Zhan, "Accurate scene text detection through border semantics awareness and bootstrapping," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11220. Springer, 2018, pp. 370–387.
[62] C. Zhang *et al.*, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10552–10561.
[63] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proc. AAAI*, 2019, pp. 9038–9045.

Yuanqiang Cai received the B.E. and M.E. degrees from the Xi'an University of Science and Technology, Xi'an, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Science, Beijing. His research interests include computer vision, multimedia content analysis, and text localization in images and videos.

Chang Liu received the B.S. degree from Jilin University, Jilin, China, in 2012. He is currently pursuing the Ph.D. degree with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing, China. His research interests include computer vision and machine learning, specifically for neural architecture design, and visual object detection. He has published more than ten papers in referred conference and journals including IEEE CVPR, ICCV, ECCV and NeurIPS.

Peirui Cheng received the B.S. degree from the School of Information Science and Technology, University of Science and Technology of China, Hefei, China, in 2013, and the M.S. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Technology. His research interests mainly include scene text detection and object detection.

Dawei Du received the B.Eng. degree in automation and the M.S. degree in detection technology and automatic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2010 and 2013, respectively, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2018. He is currently a Post-Doctoral Researcher with University at Albany, State University of New York, Albany, NY, USA. His current research interests include object detection, visual tracking, video segmentation, and digital forensics.

Libo Zhang received the Ph.D. degree from the University of Chinese Academy of Sciences in 2017. He is currently an Associate Research Professor with the ISCAS. He wins the ZhuLiyuehua Distinguish Award and is selected as the outstanding doctor of the original innovation program of Chinese Academy of Sciences.

Weiqiang Wang (Member, IEEE) received the B.E. and M.E. degrees in computer science from Harbin Engineering University, in 1995 and 1998, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), China, in 2001. He is currently a Professor with the School of Computer and Control Engineering, University of Chinese Academy of Sciences. His research interests include multimedia content analysis, computer vision, pattern recognition, and human-computer interaction.

Qixiang Ye (Senior Member, IEEE) received the B.S. and M.S. degrees in mechanical and electrical engineering from the Harbin Institute of Technology, China, in 1999 and 2001, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2006. He has been a Professor with the University of Chinese Academy of Sciences, since 2009, and was a Visiting Assistant Professor with the Institute of Advanced Computer Studies (UMIACS), University of Maryland, College Park, until 2013. His research interests include image processing, visual object detection, and machine learning. He has published more than 40 papers in refereed conferences and journals including IEEE CVPR, ICCV, ECCV, NeurIPS and TPAMI, and received the Sony Outstanding Paper Award.