Pattern Recognition Letters 170 (2023) 1–8

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Texts as points: Scene text detection with point supervision

Mengbiao Zhao[a,b], Wei Feng[a,b], Fei Yin[a,b], Cheng-Lin Liu[a,b,*]

[a] State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation of Chinese Academy of Sciences, Beijing 100190, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Article history:
Received 30 March 2022
Revised 18 December 2022
Accepted 16 April 2023
Available online 17 April 2023

Edited by: Prof. S. Sarkar

Keywords:
Scene text detection
Point supervision
Mixed-supervised learning

ABSTRACT

Scene text detection is challenging due to the diverse text appearance, the complex background, and the expensive labeling of training data. For detecting arbitrary-shaped texts, most existing methods require heavy data labeling efforts to produce polygon-level annotations for supervised training. In order to reduce the cost in data labeling, we propose to combine center point annotation into mixed-supervised scene text detection, in which the dataset comprises small number of fully annotated images and large number of weakly annotated images by center points. For better incorporating point supervision, we adopt self-training strategy based on a detector which locates texts by predicting their centers. Besides, in order to weight the pseudo labels generated during self-training, we also propose a novel regression uncertainty estimation module to measure the quality of detection results. Extensive experiments on five benchmark datasets (ICDAR2015, C-SVT, CTW1500, Total-Text and ICDAR-ArT) show that using small amount of polygon annotated data and large amount of center point annotated data, our detector can achieve competitive detection performance.

1. Introduction

Scene text detection [1–3], which is a fundamental step for scene understanding, autonomous driving, and human-computer interaction [4], aims to localize the text instances in scene images. Since text instances in real-world are variable in scale, direction and shape, most scene text detection methods [5–7] use polygon annotation for training robust text detectors. However, the cost of polygon labeling is extremely high, limiting its large-scale extension in real-world applications.

To reduce the cost of data annotation, particularly for arbitrarily-shaped text detection, an alternative is to utilize weak annotations. Some methods [8,9] attempt to annotate texts with bounding boxes, which can roughly distinguish the foreground and background locations. For lowering labeling costs, Wu *et al.* [10] adopted scribble lines to depict the texts. Although these annotations reduce the labeling costs to some extent, they did not lead to competitive performance, while the annotation cost is still considerable. In this paper, we propose to use the very cheap center point annotation in mixed-supervised scene text detection, for achieving competitive performance at low labeling cost. we annotate each text instance in the image by one center point for two main reasons. Firstly, The center point annotation provides strong prior of object location, which is suitable for supervising the training of detector. Secondly, compared with other annotation formats, center point annotation is extremely cheap and suitable for large-scale data scenarios. To compare the costs of four annotation methods, we roughly calculate the average time cost of four annotation methods (polygons, bounding boxes, scribble lines, and the proposed center points) to label 500 images from ICDAR-ArT [11]. As shown in Fig. 1, the average time per image is about 1min for annotating with polygons, 39s with bounding boxes, 25s with scribble lines, and only 17s with center points. We thus aim to utilize the center point annotations to boost the detection performance.

Some researchers adopted weakly-supervised learning [8,10] or semi-supervised learning [12] to scene text detection. However, the big performance gap of these methods with the fully supervised model makes them impractical for real applications. Another promising approach is to utilize mixed-supervised learning, where only a part of data is strongly annotated and the rest is labeled with weak supervision forms. This approach is very practical in real scenarios, where it is easy to acquire a large number of scene images but hard to annotate them in detailed object boundaries. In such cases, mixed-supervised learning enables utilizing weakly annotated data to improve the model performance.

In this paper, we propose a self-training based mixed-supervised method for training scene text detectors combining

* Corresponding author.
E-mail addresses: zhaomengbiao2017@ia.ac.cn (M. Zhao), wei.feng@nlpr.ia.ac.cn (W. Feng), fyin@nlpr.ia.ac.cn (F. Yin), liucl@nlpr.ia.ac.cn (C.-L. Liu).

https://doi.org/10.1016/j.patrec.2023.04.004
0167-8655/

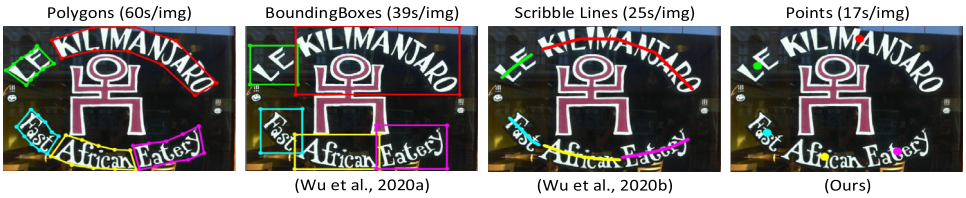

Fig. 1. Examples of four labeling methods and the time cost of using them to label an image.

polygon labels and center point labels. Firstly, for facilitating the incorporation of point labels, we adopt a detector [9], which locates text instances by predicting their centers. Secondly, in order to utilize the weakly annotated data to boost the performance, we adopt a self-training strategy. Specifically, a teacher model is first obtained by training with a small set of strongly annotated data and then used to annotate the weakly annotated data with the guidance of point labels. In practice, there will inevitably be some negative samples in the pseudo labels. To evaluate the quality of pseudo label, we add a novel regression uncertainty estimation module to the detector. Finally, we mix the strongly annotated data with weakly annotated data with pseudo labels, and use them to train a student model. The estimated uncertainties of pseudo labels are used to weight their supervisions, so as to reduce the impact of negative samples. As far as we know, this is the first attempt to accomplish scene text detection using point supervision.

In summary, the contributions of this paper are in three folds:

(1) We propose a novel mixed-supervised scene text detection method by training with small amount of fully annotated data and large amount of weakly point annotated data. It achieves a good trade-off between annotation cost and detection performance.

(2) We adopt a self-training strategy incorporating a novel regression uncertainty estimation module to utilize large amount point annotated data for boosting detection performance.

(3) Extensive experiments on two multi-oriented text datasets (ICDAR2015 [13] and C-SVT [14]) and three arbitrary-shaped text datasets (CTW1500 [15], Total-Text [16] and ICDAR-ArT [11]) show that using only 10% strongly annotated data combined with 90% weakly annotated data, our model yields performance comparable to fully-supervised models.

2. Related works

2.1. Fully supervised scene text detection

The existing methods for scene text detection can be roughly divided into two types: regression-based and segmentation-based. The former is mainly built on generic object detectors [17,18]. To detect multi-oriented (non-horizontal) texts, the Faster-RCNN [17] was adopted with the anchor modified to a rotated form to fit multi-oriented texts [19]. Liao *et al.* [20] proposed TextBoxes, which modified the anchors and kernels of SSD [18] to detect large-aspect-ratio scene texts. SegLink [3] predicts the bounding boxes of character segments and their links. On basis of the Densebox [21], DDR [22] and EAST [1] detect multi-oriented texts by regressing text boundary as quadrangle. Wu *et al.* [23] proposed CE-Text, in which a task-specified hierarchical attention scheme is adopted to enhance feature representation ability on the basis of context information. These methods can only detect oriented texts.

Segmentation based methods take semantic segmentation methods [24] for reference, and utilize convolution operations to extract semantic information form feature maps for pixel-level label prediction. PSENet [6] segments text instance by progressively expanding kernels at different scales. MaskTextSpotter [5] regards arbitrary-shaped text detection as an instance segmentation problem. Wang *et al.* [7] proposed a pixel aggregation network, which is equipped with a low computational-cost segmentation head and a learnable post-processor. Jain *et al.* [25] explored harmonic features to represent the text component shape variations for classifying text and non-text components. These methods can detect arbitrarily-shaped texts, but usually require large amount of strongly annotated data (e.g., in polygonal form) for training.

2.2. Scene text detection with weak labels

To alleviate the requirement of strongly annotated data, some weakly supervised scene text detection methods where proposed. WeText [26] and CRAFT [27] use character-level labels to boost the word detection performance. While to alleviate the cost of character-level labeling, they train a character detector using a small number of character-level annotated text images or synthetic data, and apply rules or threshold to pick the most reliable predicted candidates, to be used as additional supervisions to boost the performance of word detector. Some other methods [8,10,14] use partially annotated data to train the text detector. Specifically, Wu *et al.* [10] use scribble line annotated text images in weakly supervised framework for scene text detection. Sun *et al.* [14] published a large dataset, where each image is only annotated with one dominant text, and proposed an algorithm to combine these partially annotated data and strongly annotated data for joint training. Wu *et al.* [8] proposed to train arbitrarily-shaped text detector with bounding boxes annotated data in dynamic self-training strategy. Further, Liu *et al.* [12] proposed a semi-supervised text detection framework named SemiText, which firstly used fully annotated synthetic dataset for pretraining, then conducted inductive and transductive semi-supervised learning on the unlabeled data.

Although these methods can reduce the cost of annotation significantly, their performance is far inferior to the fully supervised models. In this paper, we propose to train detector using text images annotated as center points, which are much cheaper and easier to annotate compared to polygons, bounding boxes, and lines. To better trade off between annotation cost and performance, we propose a mixed-supervised learning strategy with a center based detector. Using only a small set of strongly annotated images and large amount of weakly annotated images, our mixed-supervised model can yield competitive performance.

3. Method

3.1. Point annotations

We annotate each text instance with a center point (see Fig. 1). As a weak supervision, point labeling has been used to reduce the annotation time for semantic segmentation [28] and generic object detection [29], but it has not been explored well in scene text detection. The existing methods [22,30] usually treat scene text detection as a multi-task learning problem consisting of classification and regression. The classification task is to classify the pixels into text and non-text. Since, the center point annotation contains

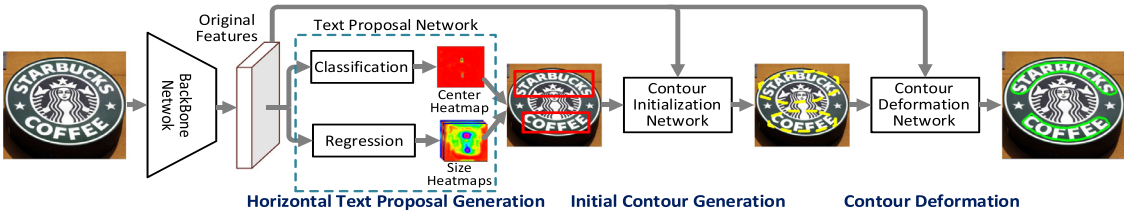

Fig. 2. Illustration of the structure of the text detector.

strong prior of text location, it is suitable for supervising the training of text detector. With the guidance of text center, the text detection degenerates into a simpler task: regressing the offsets from the given points to the text contours.

To evaluate the effectiveness of the mixed-supervised text detection in experiments, the weak labels are generated from the polygon-level labels provided by public datasets. In real applications, the annotator only needs to click within the midpoint of the text centerline without being very accurate. This is a very natural way for people to find text in the image.

3.2. Basic detection model

The detector of [9] is suitable for incorporating point supervision because it localizes text instance by predicting their center points. The whole architecture is illustrated in Fig. 2. The framework consists of four modules: backbone network, text proposal network, contour initialization network, and contour deformation network. Specifically, text proposal network generates horizontal text proposals based on features extracted by the backbone network. Given the original features and text proposals, the contour initialization network are used to get more accurate and suitable initial contour for each text instance. Finally, the contour deformation network takes the initial text contours and original features as input, and perform iterative contour regression to produce outputs.

Text Proposal Network consists of only two branches: (1) The classification branch calculates a heatmap, where the peaks are supposed to be the text centers; (2) The regression branch predicts the offsets from each peak to the upper left and lower right corners of the proposal box.

Contour Initialization Network is essentially a regression model, which could output four extreme points of the text proposal. We extend a line in both directions at each extreme point, and connect their endpoints to obtain a octagon, which can be regarded as the initial contour.

Contour Deformation Network is used to regress the offsets from points on the initial contour to the corresponding points on the ground-truth. And the same regression method is adopted as the contour initialization.

The above three networks are jointly trained in multi-task learning, with loss defined as:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{reg} + \lambda_2 \mathcal{L}_{cin} + \lambda_3 \mathcal{L}_{cdn}. \quad (1)$$

$\mathcal{L}_{cls}$ and $\mathcal{L}_{reg}$, the loss functions of the classification task and regression task in text proposal generation network, adopt the smooth ℓ_1 loss and focal loss [31], respectively. $\mathcal{L}_{cin}$ and $\mathcal{L}_{cdn}$ are the loss functions of contour initialization network and contour deformation network, respectively, and smooth ℓ_1 loss is adopted. λ_1, λ_2 and λ_3 are balancing parameters and are all set to 1 in our experiments for simplicity (observing that fluctuating around 1 does not influence the performance significantly).

3.3. Learning strategy

For mixed supervised learning with a small number of strongly annotated images and a large number of weakly annotated images, we adopt the self-training strategy, which has made considerable progress in semi-supervised learning [32,33]. The main steps are as follows:

(1) Based on the detector in Section 3.2, we firstly train a teacher model using the strongly annotated images.

(2) The trained teacher model is used to generate pseudo-labels for the center point annotated images. Specifically, the peaks on the heatmap of the classification branch output of the teacher model are replaced with the ground truth text centers so as to generate more accurate text proposals. Accordingly, the following contour regression parts can predict better text boundaries, which are used as pseudo labels.

(3) The pseudo labeled images are combined with fully labeled images to train a superior student model.

Inevitably, there are low quality pseudo labeled samples in the mixed dataset used in the third step. Using the confidence of the pseudo label to weight the pseudo supervision can alleviate the deterioration caused by noisy pseudo labels. The existing text detection methods usually use the classification score as the confidence of the output text boundary. However, the pseudo labels in our pipeline are generated based on the ground-truth text centers, the classification scores are not available. Hence, we use the uncertainty of regression to evaluate the pseudo labels, which will be described in detail in the next subsection.

3.4. Regression with uncertainty estimation

A regression uncertainty estimation branch is parallel to the regression branch in the text proposal network, as shown in Fig. 3. It has the same structure with the other two branches. Specifically, the original features are passed through a 3×3 convolution, ReLU and another 1×1 convolution to produce the output of four channels, representing the uncertainty of the horizontal and vertical offset prediction from the center point to upper left and lower right corners of the bounding box, respectively. During training, the regression uncertainty estimation branch is optimized by the KL loss [33].

Specifically, let z denotes an offset, which is the prediction target of the regression task. Assuming the offsets are independent, we use a single variate Gaussian for simplicity, and get the follow-

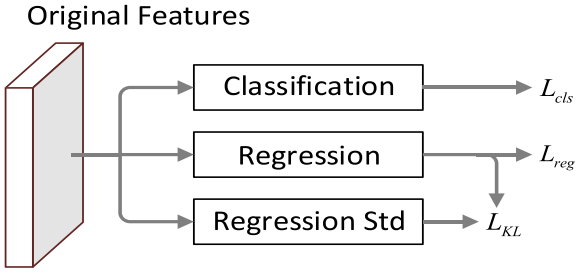

Fig. 3. The architecture of the detection head added with regression uncertainty estimation module.

ing probabilistic model:

$$P_\Theta = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(z-z_e)^2}{2\sigma^2}}, \tag{2}$$

where Θ is parameters of detection model. z_e is the estimated offset. Standard deviation σ measures the uncertainty of the prediction. When $\sigma \to 0$, it means the model is extremely confident about the predicted offset.

The ground-truth offset can also be formulated as a Gaussian distribution, with $\sigma \to 0$, which is a Dirac delta function:

$$P_D = \delta(z - z_g), \tag{3}$$

where z_g is the ground-truth offset.

Here, we take [33] for reference, and try to estimate $\hat{\Theta}$ by minimizing KL-Divergence between $P_\Theta(z)$ and $P_D(z)$:

$$\hat{\Theta} = \arg\min_{\Theta} \frac{1}{N} \sum D_{KL}(P_D(z)||P_\Theta(z)), \tag{4}$$

where N is the number of training samples. And the KL loss $\mathcal{L}_{KL}$ can be formulated as:

$$\begin{aligned} \mathcal{L}_{KL} &= D_{KL}(P_D(z)\|P_\Theta(z)) \\ &= \int P_D(z)\log\left(P_D(z)\right)dz - \int P_D(z)\log\left(P_\Theta(z)\right)\mathrm{d}z \\ &= \frac{(z_g - z_e)^2}{2\sigma^2} + \frac{\log\left(\sigma^2\right)}{2} + \frac{\log\left(2\pi\right)}{2} - H(P_D(z)). \end{aligned} \tag{5}$$

When the offset z_e is predicted inaccurately, the variance σ^2 is expected to be larger so that $\mathcal{L}_{KL}$ will be lower. $log(2\pi)/2$ and $H(P_D(z))$ do not depend on Θ, hence:

$$\mathcal{L}_{KL} \propto \frac{(z_g - z_e)^2}{2\sigma^2} + \frac{log(\sigma^2)}{2}. \tag{6}$$

The loss is differentiable w.r.t offset z_e and offset standard deviation σ:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}z_e}\mathcal{L}_{KL} &= \frac{z_e - z_g}{\sigma^2} \\ \frac{\mathrm{d}}{\mathrm{d}\sigma}\mathcal{L}_{KL} &= -\frac{(z_e - z_g)^2}{\sigma^3} + \frac{1}{\sigma}. \end{aligned} \tag{7}$$

Since σ is in the denominators, the gradient sometimes can explode at the beginning of training. To avoid gradient exploding, our network predicts $\alpha = \log(\sigma^2)$ instead of σ in practice:

$$\mathcal{L}_{KL} \propto \frac{e^{-\alpha}}{2}(z_g - z_e)^2 + \frac{1}{2}\alpha. \tag{8}$$

Meanwhile, considering the influence of outliers, we adopt a smooth L_1 loss:

$$\mathcal{L}_{KL} = e^{-\alpha}(|z_g - z_e| - \frac{1}{2}) + \frac{1}{2}\alpha. \tag{9}$$

Therefore, the loss of the whole detection framework is updated to:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1\mathcal{L}_{reg} + \lambda_2\mathcal{L}_{cin} + \lambda_3\mathcal{L}_{cdn} + \lambda_4\mathcal{L}_{KL}, \tag{10}$$

where λ_4 is set to 1 in our experiments.

For using the estimated uncertainty to measure the quality of output result in inference, we convert the network output α back to σ, and calculate the confidence of each text contour by:

$$s = 1 - \frac{1}{4}\left(\sum_{i=1}^{4} \mathrm{Sigmoid}(\sigma_i)\right), \tag{11}$$

where σ_i is the standard deviations of i-th offset prediction. Finally, to weight the pseudo labels generated during self training (described in Section 3.3), a confidence weighted loss $\hat{\mathcal{L}}$ is proposed as follow:

$$\hat{\mathcal{L}} = s\mathcal{L}, \tag{12}$$

which makes the training focus more on reliable samples.

In short, the learning strategy of the proposed mixed-supervised framework can be summarized as Algorithm 1. Specifically, we have limited polygon labeled images $\{(x_i, y_i)\}_{i=1}^{n}$, where x_i and y_i denote the i-th images and its polygon-level label, and a large set of center point labeled images $\{(\hat{x}_j, p_j)\}_{j=1}^{m}$, where $\hat{x}_j$ and p_j denote the j-th image and its center point label. We firstly use the polygon labeled images to train an teacher detector θ. Then we send the weakly annotated images $\{\hat{x}_j\}_{j=1}^{m}$ to the detector θ to generate pseudo labels. $\mathrm{M}(\cdot)$ in the Algorithm 1 refer to the inference. Specifically, the peaks on the heatmap of the classification branch output of the teacher model are replaced with the ground truth text centers $\{p_j\}_{j=1}^{m}$ so as to generate more accurate text proposals. Accordingly, the following contour regression parts can predict better text boundaries, which are used as pseudo labels $\{\hat{y}_j\}_{j=1}^{m}$. In addition, the regression uncertainty estimation module outputs the confidence $\{s_j\}_{j=1}^{m}$ of the pseudo labels. Finally, we conduct mixed retraining of the model by using both $\{(x_i, y_i)\}_{i=1}^{n}$ and $\{(\hat{x}_j, \hat{y}_j, s_j)\}_{j=1}^{m}$, and confidence s_j is used to weight the supervision of pseudo labels, which makes the training focus more on reliable samples.

Algorithm 1 Self-training based learning strategy.

Require: Polygon labeled images: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$;
Center point labeled images: $(\hat{x}_1, p_1), (\hat{x}_2, p_2), \ldots, (\hat{x}_m, p_m)$.
Ensure: Trained parameters θ.
1: Training the model θ on $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$;
2: Infer pseudo label $\hat{y}_j \leftarrow \mathrm{M}((\hat{x}_j, p_j)), \theta)$;
3: Calculate the confidence s_j of each pseudo label $\hat{y}_j$ by Eq. (11);
4: Mixed retraining model θ on the union of $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ and $(\hat{x}_1, \hat{y}_1, s_1), (\hat{x}_2, \hat{y}_2, s_2), \ldots, (\hat{x}_m, \hat{y}_m, s_m)$.

4. Experiments

4.1. Datasets

To demonstrate the effectiveness of the proposed method, we conduct experiments on five public benchmark datasets.

ICDAR2015 dataset [13] contains 1000 training images and 500 test images. This dataset is focused on incidental scene text, in which each text is labeled as a quadrangle with 4 vertexes in word-level.

C-SVT dataset [14] contains 430,000 training images, of which 30,000 are fully annotated and the remaining are weakly annotated, where only the corresponding text-of-interest in the regions is given as weak annotations. Each text is labeled with adaptive number of vertices. In our experiment, we only use the fully annotate images for the convenience of evaluation.

CTW1500 dataset [15] contains 1000 training images and 500 test images. Besides horizontal and multi-oriented texts, at least one curved text is contained in each images. Each text is labeled as a polygon with 14 vertexes in line-level.

Total-Text dataset [16] has 1255 training images and 300 test images, which contains curved texts, as well as horizontal and multi-oriental texts. Each text is labeled as a polygon with 10 vertexes in word-level.

ICDAR-ArT dataset [11] consists of 5603 training images and 4563 test images, which contains multilingual arbitrary-shaped texts. Each text is labeled with adaptive number of vertices.

Table 1
Detection results on ICDAR2015. The subscript indicates the standard deviation.

Method	Precision	Recall	F-measure	FPS
EAST [1]	83.6	73.5	78.2	13.2
TextSnake [2]	84.9	80.4	82.6	1.1
SegLink+ [36]	80.3	83.7	82.0	-
TextField [37]	84.3	83.9	84.1	1.8
PSENet [6]	86.9	84.5	85.7	1.6
FCENet [38]	90.1	82.6	86.2	-
TextMountain [39]	87.3	84.1	85.7	10.4
PAN+ [7]	**91.4**	83.9	87.5	12.6
MOST [40]	89.1	**87.3**	**88.2**	10.0
100%Poly	89.4	82.4	85.8	
10%Poly&90%Point	$86.2_{\pm 0.36}$	$80.1_{\pm 0.28}$	$83.0_{\pm 0.31}$	
10%Poly&90%Unlabeled	$80.9_{\pm 0.39}$	$77.6_{\pm 0.31}$	$79.2_{\pm 0.36}$	**21.6**
10%Poly	$77.2_{\pm 0.31}$	$76.9_{\pm 0.27}$	$77.0_{\pm 0.29}$	

Table 3
Detection results on CTW1500. The subscript indicates the standard deviation.

Method	Precision	Recall	F-measure	FPS
TextSnake [2]	67.9	85.3	75.6	-
TextRay [42]	82.8	80.4	81.7	-
PSENet-1s [6]	84.8	79.7	82.2	3.9
Wu *et al.*-TAS [10]	83.8	80.8	82.3	9.2
CRAFT [27]	86.0	81.1	83.5	-
TextDragon [30]	84.5	82.8	83.6	8.7
ContourNet [43]	83.7	84.1	83.9	4.5
TextMountain [39]	83.3	83.6	83.4	-
PAN+ [7]	87.1	81.1	84.0	**36.0**
Zhang *et al.* [44]	**87.8**	81.5	84.5	12.2
Dai *et al.* [45]	82.3	**87.2**	**84.7**	11.8
ABCNet v2 [46]	85.6	83.8	**84.7**	-
100%Poly	86.1	82.1	84.1	
10%Poly&90%Point	$83.9_{\pm 0.39}$	$81.8_{\pm 0.27}$	$82.8_{\pm 0.29}$	32.3
10%Poly&90%Unlabeled	$83.4_{\pm 0.67}$	$79.1_{\pm 0.59}$	$81.2_{\pm 0.13}$	
10%Poly	$81.3_{\pm 0.79}$	$79.1_{\pm 1.02}$	$80.2_{\pm 0.26}$	

4.2. Implementation details

We implemented the experiments with Pytorch 1.1 on a workstation with 2.9 GHz 12-core CPU, 256G RAM, GTX Titan X and Ubuntu 64-bit OS. We adopt the DLA-34 [34] as the backbone network. The model is pre-trained with SynthText [35] dataset, and then fine-tuned on the real datasets separately for 200 epochs with batchsize 36. The initial learning rate is set to 1×10^{-4} and is divided by 2 at 80th, 120th, 150th, and 170th epoch. The data is augmented by: (1) rescaling images with ratio from 0.5 to 2.0 randomly, (2) flipping horizontally and rotating in range $[-10^{\circ}, 10^{\circ}]$ randomly, (3) cropping 640×640 random samples from the transformed image. In the inference stage, the short side of the input image is scaled to a fixed length (720 for ICDAR2015, 460 for CTW1500, 960 for ICDAR-ArT, and 640 for C-SVT and Total-Text), with the aspect ratio kept.

4.3. Experimental results

For the three datasets, we randomly select 10% of original training images as strongly annotated data and take the rest 90% as weakly (center point) annotated data, resulting in a 100–900 split for ICDAR2015 and CTW1500, a 125-1,130 split for Total-Text, a 560-5,043 split for ICDAR-ArT, and a 3,000-27,000 split for C-SVT. Based on the data division, we can get the following models:

(1) 100%Poly: Model trained with all images, which are annotated with polygons.

(2) 10%Poly: Model trained with 10% images, which are annotated with polygons.

(3) 10%Poly&90%Point: Model trained with all images, of which 10% are annotated with polygons, and 90% are annotated with center points.

(4) 10%Poly&90%Unlabeled: Model trained with all images of which 10% are annotated with polygons, and 90% are unlabeled. The learning strategy used is similar to that of "10%Poly&90%Point", but there is no point guidance in the generation of pseudo labels. We use a threshold 0.35 to filter low-quality samples.

Considering the impact of data split on the final results, we split the data five times randomly, and report the average results and standard deviations. The results on five benchmarks are given in Tables 1–5, respectively.

Results on Multi-Oriented Text: As shown in Tables 1 and 2, our mixed-supervised model "10%Poly&90%Point" achieves 83.0% and 76.9% of F-measure on ICDAR2015 and C-SVT, respectively. Compared with the two baseline models "10%Poly", the performance gains are 6.0% and 3.7%, respectively. This demonstrates the effectiveness of the proposed mixed-supervised learning in the sense that adding weakly annotate data improves the performance. When comparing with "10%Poly&90%Unlabeled", "10%Poly&90%Point" outperforms by 3.8% and 1.9% on ICDAR2015 and C-SVT, respectively, which proves the significance of point supervisions. Although there are still gaps between fully-supervised models and mixed-supervised models (2.8% and 2% on ICDAR2015 and C-SVT, respectively), considering that the cost of point annotation is much lower, the results are valuable. Some qualitative results are shown in the first and second columns of the Fig. 4, where we can see that our method can handle multi-oriented texts very well.

Results on Arbitrary-Shaped Text: We further conduct experiments on more challenging arbitrary-shaped texts, and the results are shown in Tables 3–5. We can see that our mixed-supervised models "10%Poly&90%Point" have greatly exceeded the baseline models "10%Poly" and achieved 82.8%, 83.2%, and 72.0% of F-measure on CTW1500, Total-Text, and ICDAR-ArT respectively, demonstrating the effectiveness of the proposed method when handling texts with arbitrary shapes. Meanwhile, our mixed-supervised learning strategy does not affect the inference speed, which is still competitive. Some examples of text detection results are shown in the third to fifth columns in Fig. 4, where it is seen that the detection model can accurately locate horizontal, multi-oriented and curved texts.

Table 2
Detection results on C-SVT. The subscript indicates the standard deviation.

Method	Precision	Recall	F-measure	FPS
DB+oCLIP [41]	81.5	70.9	75.8	-
EAST [1]	73.4	**79.3**	76.2	-
PSENet+oCLIP [41]	**90.7**	67.0	77.1	-
Sun *et al.*-Train [14]	80.4	74.6	77.4	-
Sun *et al.*-Train+400K Weak [14]	81.7	75.2	78.3	-
100%Poly	83.5	74.5	**78.9**	
10%Poly&90%Point	$80.6_{\pm 0.27}$	$73.6_{\pm 0.36}$	$76.9_{\pm 0.29}$	**24.2**
10%Poly&90%Unlabeled	$77.8_{\pm 0.29}$	$72.4_{\pm 0.35}$	$75.0_{\pm 0.31}$	
10%Poly	$73.5_{\pm 0.33}$	$72.9_{\pm 0.28}$	$73.2_{\pm 0.32}$	

Fig. 4. Examples of text detection results. First column: ICDAR2015; second column: C-SVT; third column: CTW1500; fourth column: Total-Text; fifth column: ICDAR-ArT.

Table 4
Detection results on Total-Text. The subscript indicates the standard deviation.

Method	Precision	Recall	F-measure	FPS
SemiText-Transductive [12]	78.0	58.3	66.7	2.1
SemiText-Inductive [12]	79.2	59.0	67.6	2.1
Wu *et al.*-TAS [10]	78.5	76.7	77.6	11.2
SelfText [8]	82.5	77.6	80.1	-
TextRay [42]	83.5	77.9	80.6	-
PSENet-1s [6]	84.0	78.0	80.9	3.9
CRAFT [27]	87.6	79.9	83.6	-
Dai *et al.* [45]	82.0	**88.5**	85.2	11.8
PAN+ [7]	89.9	81.0	85.3	**38.3**
FCENet [38]	89.3	82.5	85.8	-
SRSTS [47]	**92.0**	83.0	87.2	18.7
Zhang *et al.* [44]	90.3	84.7	**87.4**	10.3
100%Poly	88.2	83.3	85.6	
10%Poly&90%Point	$84.4_{\pm 0.69}$	$82.1_{\pm 0.47}$	$83.2_{\pm 0.33}$	24.2
10%Poly&90%Unlabeled	$82.9_{\pm 0.30}$	$78.8_{\pm 0.49}$	$80.8_{\pm 0.27}$	
10%Poly	$80.2_{\pm 0.51}$	$78.5_{\pm 0.38}$	$79.4_{\pm 0.23}$	

Table 5
Detection results on ICDAR-ArT. The subscript indicates the standard deviation.

Method	Precision	Recall	F-measure	FPS
TextRay [42]	76.0	58.6	66.2	-
Dai *et al.* [45]	66.1	**84.0**	74.0	11.8
100%Poly	**80.8**	68.7	**74.3**	**16.4**
10%Poly&90%Point	$80.5_{\pm 0.57}$	$65.1_{\pm 0.39}$	$72.0_{\pm 0.42}$	
10%Poly&90%Unlabeled	$77.9_{\pm 0.49}$	$64.2_{\pm 0.55}$	$70.4_{\pm 0.51}$	
10%Poly	$77.2_{\pm 0.63}$	$58.9_{\pm 0.38}$	$66.8_{\pm 0.42}$	

Comparison with Other Methods Using Weak labels: Compared with the method which use character box to boost the performance of word detection, as shown in Tables 3 and 4, the performance of our mixed-supervised model "10%Poly&90%Point" is slightly lower (0.7% and 0.4% on CTW1500 and Total-Text, respectively) than that of CRAFT [27]. However, it should be noted that CRAFT needs complete strong annotations and additional character annotations, while our method only needs 10% strong annotations combined with 90% weak annotations. We achieve the performance close to CRAFT with much lower labeling cost.

Compared to methods which use partial annotations to reduce labeling cost, our method also shows advantage. As shown in Tables 3 and 4, our mixed-supervised model outperforms the corresponding model (TAS) of Wu *et al.* [10], which uses line-level annotations to supervise model training. Compared with SelfText [8], which trains PSENet [6] with pseudo labels generated with the aid of bounding box annotations, as shown in Table 4, our mixed-supervised model also achieves better performance (83.2% vs 80.1%). In addition, Sun *et al.* [14] proposed to use a large partially labeled dataset (each image annotated with one dominant text) to boost performance, and obtained 0.9% improvement by adding 4,000,000 weakly annotated data (as shown in Table 2). In contrast, our mixed-supervised model yields significant improvement compared to the baseline.

Compared to semi-supervised method [12], our method also outperforms by a large margin, as shown in Table 4. Although the method does not use any strong annotation, our method uses only 10% strong annotation to obtain a big advantage, which is very worthwhile.

Table 6
The benefits of the KL loss and confidence weighted loss on "10%Poly&90%Point". "KL" and "CW" indicates KL loss and confidence weighted loss, respectively.

Dataset	KL	CW	Precision	Recall	F-measure
CTW1500	-	-	83.6	80.5	82.0
	✓	-	84.0	80.6	82.3
	-	✓	**85.3**	80.0	82.6
	✓	✓	83.9	**81.8**	**82.8**
Total-Text	-	-	84.1	80.8	82.4
	✓	-	**84.9**	80.6	82.7
	-	✓	83.5	**82.5**	83.0
	✓	✓	84.4	82.1	**83.2**

4.4. Ablation studies

Influence of the Regression Uncertainty Estimation Module. The proposed regression uncertainty is learned by optimizing the KL loss. Therefore, we conduct experiments with or without the KL loss, and the results on CTW1500 and Total-Text are shown in Table 6. It is verified that the models trained with KL loss preform better. Although the improvement brought by adding KL loss is relatively weak, our intention is to estimate the uncertainty of regression, which can be used to weight the pseudo supervision.

Influence of the Confidence Weighted Loss. We evaluate the influence of the confidence weighted loss (see Eq. 12) via experiments with or without it. The results of experiments on CTW1500 and Total-Text are shown in Table 6. We can find that the confidence weighted loss improves the basic models obviously. It attributes to that with confidence weighting, the influence of noise samples on training would be limited, and the model learning tends to focus on reliable samples.

Influence of the Proportion of Strongly Annotated Data. We perform experiments with variable proportions (from 1% to 40%) of strongly annotated data on the ICDAR-ArT dataset. The model trained with only strongly annotated data is regarded as baseline. As shown in Fig. 5, the performance of mixed supervised model improves as the proportion of strongly annotated data increases, and outperforms baseline in all data split settings, which demonstrates that images with point annotations can improve the performance.

Influence of the Point Location. We explored the influence of center point location fluctuation in text line direction and the perpendicular direction of the text line, respectively. Specifically, we

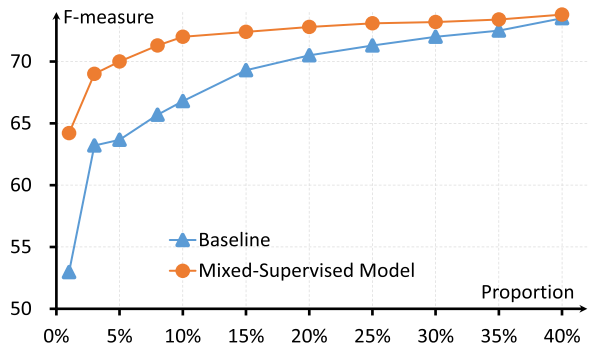

Fig. 5. F-measure versus proportions of strongly annotated data for mixed-supervised models on ICDAR-ArT test set.

Table 7
The influence of point location on "10%Poly&90%Point" on ICDAR2015.

Fluctuation Direction	Ratio Range	Precision	Recall	F-measure
No Fluctuation	-	86.2	80.1	**83.0**
Text Line Direction	[0.00, 0.05]	**86.6**	79.2	82.8
	[0.05, 0.10]	85.1	80.0	82.5
	[0.10, 0.15]	84.2	80.4	82.2
Perpendicular	[0.00, 0.05]	85.9	80.1	82.9
Direction of Text Line	[0.05, 0.10]	85.0	**80.8**	82.8
	[0.10, 0.15]	84.5	80.5	82.4
Both Directions	[0.00, 0.10]	84.8	80.7	82.7

randomly move the center point along the direction of the text line and the perpendicular direction of the text line within a certain range, which is defined as a certain ratio of the length or height of the text line. As shown in Table 7, for two directions, when the fluctuation ratio is lower than 0.1, the performance does not deteriorate. This is because our text detector predicts the region of center point rather than a single center point. When the ratio is higher than 0.1, the performance degrades slightly due to the poor quality of pseudo labels. In addition, we also explore the situation of fluctuations in both line and perpendicular directions, that is, randomly moving the center point within an ellipse with a radius of $(0.1l, 0.1h)$, where l and h are the length and height of the text line, respective. We can find that performance can still be maintained. The stability of performance against slight fluctuation of center point annotation justifies the practicality of the proposed method.

5. Conclusion

In this paper, we verify the effectiveness of center point annotations in mixed-supervised scene text detection task, which greatly reduce the cost of labeling. We adopt a self-training strategy based on the a detector which localizes texts by predicting their centers. For weighting the pseudo labels, we propose a regression uncertainty estimation module to measure the confidence. Extensive experiments show that our mixed-supervised method achieves competitive performance, and adding weakly labeled data can improve the detection performance evidently compared to training with strongly annotated data only.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

This work has been supported by the National Key Research and Development Program under Grant No. 2020AAA0108003, the National Natural Science Foundation of China (NSFC) grants 61733007 and 61721004.

References

[1] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, J. Liang, East: an efficient and accurate scene text detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5551–5560.
[2] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, C. Yao, Textsnake: a flexible representation for detecting text of arbitrary shapes, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 20–36.
[3] B. Shi, X. Bai, S. Belongie, Detecting oriented text in natural images by linking segments, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2550–2558.
[4] Y. Wu, H. Li, et al., Joint intent detection model for task-oriented human-computer dialogue system using asynchronous training, IEEE Trans. Asian Low-Resour. Lang. Inf. Process. (2022).
[5] M. Liao, P. Lyu, M. He, C. Yao, et al., Mask textspotter: an end-to-end trainable neural network for spotting text with arbitrary shapes, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2) (2019) 532–548.
[6] W. Wang, E. Xie, X. Li, Hou, et al., Shape robust text detection with progressive scale expansion network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9336–9345.
[7] W. Wang, E. Xie, X. Li, X. Liu, D. Liang, et al., Pan++: towards efficient and accurate end-to-end spotting of arbitrarily-shaped text, IEEE Trans. Pattern Anal. Mach. Intell. 44 (9) (2021) 5349–5367.
[8] W. Wu, E. Xie, R. Zhang, W. Wang, G. Pang, Z. Li, H. Zhou, P. Luo, Selftext beyond polygon: Unconstrained text detection with box supervision and dynamic self-training, arXiv preprint arXiv:2011.13307 (2020).
[9] M. Zhao, W. Feng, F. Yin, X.-Y. Zhang, C.-L. Liu, Mixed-supervised scene text detection with expectation-maximization algorithm, IEEE Trans. Image Process. 31 (2022) 5513–5528.
[10] W. Wu, J. Xing, C. Yang, et al., Texts as lines: text detection with weak supervision, Math. Probl. Eng. 2020 (2020) 1–12.
[11] C.K. Chng, Y. Liu, Sun, et al., Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art, in: Proceedings of the International Conference on Document Analysis and Recognition, 2019, pp. 1571–1576.
[12] J. Liu, Q. Zhong, Y. Yuan, H. Su, B. Du, Semitext: scene text detection with semi-supervised learning, Neurocomputing 407 (2020) 343–353.
[13] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, et al., Icdar 2015 competition on robust reading, in: Proceedings of the International Conference on Document Analysis and Recognition, 2015, pp. 1156–1160.
[14] Y. Sun, J. Liu, Liu, et al., Chinese street view text: large-scale chinese text reading with partially supervised learning, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9086–9095.
[15] Y. Liu, L. Jin, S. Zhang, S. Zhang, Detecting curve text in the wild: new dataset and new solution, arXiv preprint arXiv:1712.02170 (2017).
[16] C.K. Ch'ng, C.S. Chan, Total-text: a comprehensive dataset for scene text detection and recognition, in: Proceedings of the IAPR International Conference on Document Analysis and Recognition, volume 1, 2017, pp. 935–942.
[17] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2016) 1137–1149.
[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 21–37.
[19] Y. Liu, L. Jin, Deep matching prior network: toward tighter multi-oriented text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1962–1969.
[20] M. Liao, B. Shi, X. Bai, X. Wang, W. Liu, Textboxes: a fast text detector with a single deep neural network, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
[21] L. Huang, Yang, et al., Densebox: unifying landmark localization with end to end object detection, arXiv preprint arXiv:1509.04874 (2015).
[22] W. He, X.-Y. Zhang, F. Yin, C.-L. Liu, Deep direct regression for multi-oriented scene text detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 745–753.
[23] Y. Wu, W. Zhang, et al., Ce-text: a context-aware and embedded text detector in natural scene images, Pattern Recognit. Lett. 159 (2022) 77–83.
[24] G. Shi, Y. Wu, et al., Incremental few-shot semantic segmentation via embedding adaptive-update and hyper-class representation, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 5547–5556.
[25] T. Jain, S. Palaiahnakote, U. Pal, C.-L. Liu, Deformable scene text detection using harmonic features and modified pixel aggregation network, Pattern Recognit. Lett. 152 (2021) 135–142.
[26] S. Tian, S. Lu, C. Li, Wetext: scene text detection under weak supervision, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1492–1500.

[27] Y. Baek, B. Lee, D. Han, S. Yun, H. Lee, Character region awareness for text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 9365–9374.
[28] A. Bearman, O. Russakovsky, V. Ferrari, L. Fei-Fei, Whats the point: semantic segmentation with point supervision, in: Proceedings of the European Conference on Computer Vision, 2016, pp. 549–565.
[29] L. Chen, T. Yang, Zhang, et al., Points as queries: weakly semi-supervised object detection by points, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 8823–8832.
[30] W. Feng, W. He, F. Yin, X.-Y. Zhang, C.-L. Liu, Textdragon: an end-to-end framework for arbitrary shaped text spotting, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 9076–9085.
[31] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[32] Y. Li, C. Guan, H. Li, Z. Chin, A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system, Pattern Recognit. Lett. 29 (9) (2008) 1285–1294.
[33] Y. He, C. Zhu, J. Wang, Savvides, et al., Bounding box regression with uncertainty for accurate object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 2888–2897.
[34] F. Yu, D. Wang, E. Shelhamer, T. Darrell, Deep layer aggregation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2403–2412.
[35] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2315–2324.
[36] J. Tang, Z. Yang, Y. Wang, Q. Zheng, Y. Xu, X. Bai, Seglink++: detecting dense and arbitrary-shaped scene text by instance-aware component grouping, Pattern Recognit. 96 (2019) 106954.
[37] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, X. Bai, Textfield: learning a deep direction field for irregular scene text detection, IEEE Trans. Image Process. 28 (11) (2019) 5566–5579.
[38] Y. Zhu, J. Chen, L. Liang, Z. Kuang, et al., Fourier contour embedding for arbitrary-shaped text detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2021, pp. 3123–3131.
[39] Y. Zhu, J. Du, Textmountain: accurate scene text detection via instance segmentation, Pattern Recognit. 110 (2021) 107336.
[40] M. He, M. Liao, Z. Yang, H. Zhong, et al., Most: A multi-oriented scene text detector with localization refinement, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 8813–8822.
[41] C. Xue, W. Zhang, et al., Language matters: a weakly supervised vision-language pre-training approach for scene text detection and spotting, in: Proceedings of European Conference on Computer Vision, 2022, pp. 284–302.
[42] F. Wang, Y. Chen, F. Wu, X. Li, Textray: contour-based geometric modeling for arbitrary-shaped scene text detection, in: Proceedings of the ACM International Conference on Multimedia, 2020, pp. 111–119.
[43] Y. Wang, H. Xie, Zha, et al., Contournet: taking a further step toward accurate arbitrary-shaped scene text detection, in: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 11753–11762.
[44] S.-X. Zhang, X. Zhu, C. Yang, H. Wang, X.-C. Yin, Adaptive boundary proposal network for arbitrary shape text detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 1305–1314.
[45] P. Dai, S. Zhang, H. Zhang, X. Cao, Progressive contour regression for arbitrary-shape scene text detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 7393–7402.
[46] Y. Liu, C. Shen, L. Jin, T. He, P. Chen, C. Liu, H. Chen, Abcnet v2: adaptive bezier-curve network for real-time end-to-end text spotting, IEEE Trans. Pattern Anal. Mach. Intell. (2021).
[47] J. Wu, P. Lyu, G. Lu, C. Zhang, et al., Decoupling recognition from detection: Single shot self-reliant scene text spotter, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 1319–1328.