Mixed-Supervised Scene Text Detection With Expectation-Maximization Algorithm

Mengbiao Zhao, Wei Feng, Fei Yin, Xu-Yao Zhang, *Senior Member, IEEE*, and Cheng-Lin Liu, *Fellow, IEEE*

***Abstract*— Scene text detection is an important and challenging task in computer vision. For detecting arbitrarily-shaped texts, most existing methods require heavy data labeling efforts to produce polygon-level text region labels for supervised training. In order to reduce the cost in data labeling, we study mixed-supervised arbitrarily-shaped text detection by combining various weak supervision forms (e.g., image-level tags, coarse, loose and tight bounding boxes), which are far easier to annotate. Whereas the existing weakly-supervised learning methods (such as multiple instance learning) do not promote full object coverage, to approximate the performance of fully-supervised detection, we propose an Expectation-Maximization (EM) based mixed-supervised learning framework to train scene text detector using only a small amount of polygon-level annotated data combined with a large amount of weakly annotated data. The polygon-level labels are treated as latent variables and recovered from the weak labels by the EM algorithm. A new contour-based scene text detector is also proposed to facilitate the use of weak labels in our mixed-supervised learning framework. Extensive experiments on six scene text benchmarks show that (1) using only 10% strongly annotated data and 90% weakly annotated data, our method yields comparable performance to that of fully supervised methods, (2) with 100% strongly annotated data, our method achieves state-of-the-art performance on five scene text benchmarks (CTW1500, Total-Text, ICDAR-ArT, MSRA-TD500, and C-SVT), and competitive results on the ICDAR2015 Dataset. We will make our weakly annotated datasets publicly available.**

***Index Terms*— Mixed-supervised learning, scene text detection, weak supervision forms, expectation-maximization algorithm.**

I. Introduction

SCENE text detection has received increasing attention in recent years due to its wide applications in document analysis and scene understanding. Benefited from deep neural networks, many previous methods have made great progress. In the literature, different methods have been proposed to detect not only horizontal texts [1], but also multi-oriented texts [2], [3] and even more challenging arbitrarily-shaped texts [4], [5].

However, deep learning based methods usually require large scale strongly annotated data during training. Since texts in real-world are highly variable in size, direction and shape, we cannot describe scene texts as rectangles like generic objects. Instead, most scene text detection methods [6], [7], [8] utilize polygon annotation to train robust text detectors. According to [9], labeling one curved-shape text with polygon consumes approximately triple time as that for labeling with quadrangle (13s vs 4s). Therefore, acquiring such a strongly polygon annotated dataset costs a large amount of human labor and financial resources, and has impeded its application in large-scale real problems.

To reduce data annotation costs, an alternative is to utilize weak annotations. In this paper, we introduce four forms of weak supervision for arbitrarily-shaped texts, which are far easier to label compared with polygons. The first form of weak supervision is *tight bounding box*, which is defined by the rectangle enclosing the text instance tightly. The second one is *loose bounding box*, which is broader than the tight bounding box, and is more easier to annotate. The third one is *coarse bounding box* which encloses a cluster of texts roughly. The fourth one is *image-level tag*, which indicates whether an image contains text or not. Fig. 1 shows some examples of four weak supervision forms as well as the traditional strong supervision of polygons.

Actually, different weak supervision forms in Fig. 1 have different labeling complexities. To compare quantitatively the time costs of labeling with different annotation policies, we randomly selected 500 images, and annotated them with these annotation policies by ourselves. The average time cost of each annotation policy is also shown in Fig. 1. We can see that it takes about 1 minute for annotating an image with polygons, 39s with tight bounding boxes, 28s with loose bounding boxes, 15s with coarse bounding boxes, and only 2s with image-level tags. Based on this, we estimated the time costs for labeling the whole ICDAR-ArT [10] dataset with different annotation policies. As shown in Fig. 2, the time cost can be largely reduced by combining weak labels, which demonstrates the effectiveness and low costs of the proposed weak supervision forms. However, how to use these weak labels to train text detectors still remains a problem.

Some researchers adopted weakly-supervised learning based [11], [12] or semi-supervised learning based [13]

Manuscript received 27 March 2021; revised 16 March 2022 and 5 June 2022; accepted 27 July 2022. Date of publication 17 August 2022; date of current version 22 August 2022. This work was supported in part by the National Key Research and Development Program under Grant 2020AAA0108003 and in part by the National Natural Science Foundation of China (NSFC) under Grant 61733007 and Grant 61721004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Senem Velipasalar. *(Corresponding author: Cheng-Lin Liu.)*

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: zhaomengbiao2017@ia.ac.cn; wei.feng@nlpr.ia.ac.cn; fyin@nlpr.ia.ac.cn; xyz@nlpr.ia.ac.cn; liucl@nlpr.ia.ac.cn).

Digital Object Identifier 10.1109/TIP.2022.3197987

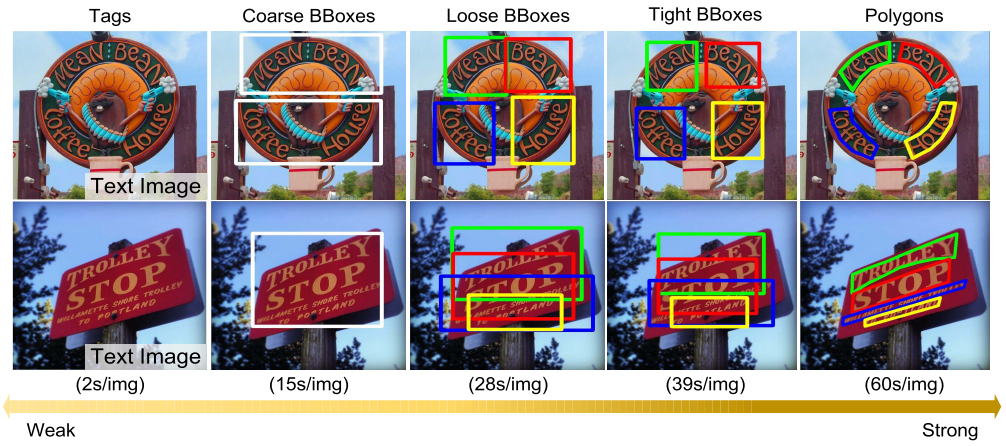

Fig. 1. Examples of five supervision forms, and the time costs of labeling an image with them.

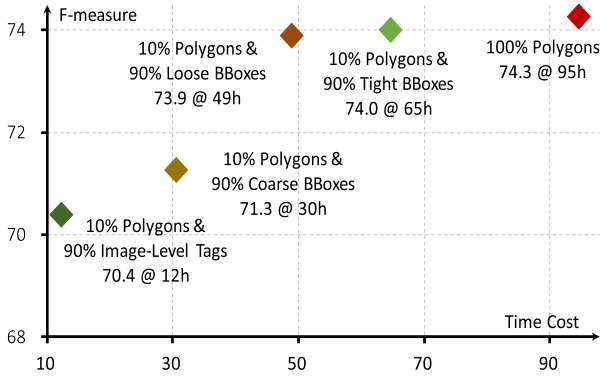

Fig. 2. Comparison of several annotation policies on the ICDAR-ArT dataset, in terms of accuracy (F-measure) and annotation time cost. The combination of weak annotation policies and strong annotation policy achieves the ideal tradeoff between effectiveness and time consumption.

methods to solve this problem. Although great progress has been made, there still exists some gap in performance compared to the models trained with strongly annotated data. This makes it impractical for the real application scenarios, where accurate detection results are required for the following text recognition. 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. For example, for autopilot cars, every tour will get a large amount of scene images, captured by the camera mounted on the car. To use these data to train better scene understanding models, annotating the object boundaries accurately would be very expensive, but people can annotate course bounding boxes at low costs. In such cases, mixed-supervised learning enables utilizing such weakly annotated data to improve the model performance.

Most existing approaches [14], [15], [16], [17], [18] for training detection models from such kind of very weak labels are based on the multiple instance learning (MIL) framework. They tend to focus on the most unique part of the object instead of capturing the entire object, thus fall in the dilemma of local object coverage and deteriorate the detection performance compared to their fully-supervised counterparts. To approximate the performance of fully-supervised detection, we propose an Expectation-Maximization (EM) based mixed-supervised framework for training scene text detectors combining strong polygon-level labels and weak labels. Firstly, for facilitating the incorporation of weak labels, we propose a contour-based scene text detector, which regresses the contours of texts based on instance-level text proposals. Due to the relevance of the proposal mechanism to the proposed weak supervision forms (coarse, loose, and tight bounding boxes), the detector is suitable for incorporating mixed-supervised learning. Secondly, in order to utilize the weakly-annotated data to boost the performance, we treat polygon-level labels as latent variables for weakly-annotated data, and use an EM-like optimization algorithm (see Fig. 6) to solve the mixed-supervised learning problem. Specifically, the algorithm alternates between two steps: (1) E-step: estimate a probability distribution over all possible latent polygons of text instances; (2) M-step: update the weights of detection model using estimated polygon-level pseudo labels from the last E-step. In practice, the quality of the learned model depends heavily on the initialization, since the whole optimization problem is highly non-convex. Therefore, we initialize the model by pre-training with a small set of strongly annotated data. The feasibility and the effectiveness of our method have been demonstrated in experiments on six scene text benchmarks.

The main contributions of this work are in the following aspects: (1) We propose an EM-based framework for mixed-supervised scene text detection, which can accommodate various forms of weak supervision. (2) To facilitate the incorporation of weak supervision, we propose a contour-based text detector which performs detection by text contour regression. (3) Our experiments show that using only 10% strongly annotated data combined with 90% weakly annotated data, our model yields the performance comparable to fully-supervised models, which demonstrates the superiority of our method. Meanwhile, under full supervision, our method achieves state-of-the-art performance on five public

benchmarks (CTW1500 [9], Total-Text [19], ICDAR-ArT [10], MSRA-TD500 [20], and C-SVT [21]) and competitive results on ICDAR2015 [22].

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the details of the proposed method. Section IV presents experimental results, and Section V draws concluding remarks.

II. Related Work

In this section, we review the most relevant works on scene text detection, weakly-, semi- and mixed-supervised text detection. Recent related reviews can be found in Ye *et al.* [23], Zhu *et al.* [24] and Long *et al.* [25].

A. Scene Text Detection

Traditional methods of scene text detection can be grouped into sliding window based (like generic object detection) methods and component based methods (aggregating character components into words or lines). Component based methods with components detected by SWT [26] and MSER [27], showed superiority in the non-deep learning era. Their performance is limited by the inaccuracy of component extraction based on hand-crafted features. Deep learning based methods have been proposed to locate texts at word/line level or character level [28]. Word/line level methods locate texts by regressing the boundary of whole word or text line directly. To detect multi-oriented (non-horizontal) texts, the Faster-RCNN [29] was adopted with the anchor modified to a rotated form to fit multi-oriented texts [30]. Liao *et al.* [1] proposed TextBoxes, which modified the anchors and kernels of SSD [31] to detect large-aspect-ratio scene texts. On basis of the Densebox [32], DDR [2] and EAST [3] detect multi-oriented texts by regressing text boundary as quadrangle. Character level methods first detects characters or components by sliding window or pixel based classification, then group components into words or lines [33].

Recently, detecting texts with arbitrary shapes has gradually drawn the attention of researchers. TextSnake [34] describes the curved text as a series of ordered, overlapping disks centered at symmetric axes. MaskTextSpotter [8] regards arbitrarily-shaped text detection as an instance segmentation problem. CRAFT [35] detects individual characters, and connects them to get text instances. PSENet [6] adopts a progressive scale algorithm to gradually expand the predefined kernels. LOMO [7] regresses text geometry in a coarse-to-fine manner. TextMountain [36] generates text polygons after predicting text score, text center border probability, and text center direction. Wang *et al.* [37] proposed to localize the key points on the contours of arbitrary-shape texts within the proposals. However, these methods usually require large amount of strongly annotated data (with text boundary labeled as polygon, e.g.) for training. It is desired to use weakly annotated data (labeled with coarse bounding boxes, e.g.) to replace strongly annotated data in training, for saving annotation costs while preserving the detection performance.

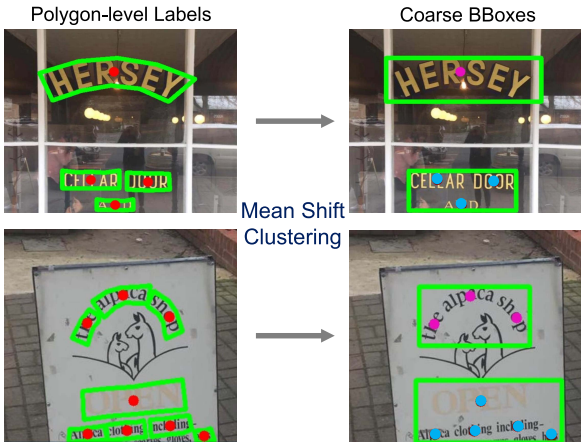

Fig. 3. Generation of coarse bounding box. The first column is the original polygon-level labels of images, where the red dot represents the center of each text instance. Then, the Mean Shift algorithm is adopted to cluster the centers of text instances. Finally, as shown in the second column, we get the coarse bounding box of each text cluster, where different color centers represent different text clusters.

B. Weakly- and Semi-Supervised Scene Text Detection

Weakly-supervised methods attempt to learn scene text detector using only weak labels. Wu *et al.* [11] introduced a form of weak label named coarse mask, which is defined by the line across the text region. And they proposed a segmentation-based network combined with a modified crossentropy loss function to use the coarse mask for training. Zhang *et al.* [12] just used the image-level tag as weak label, and proposed a classification-based framework for weakly-supervised text localization, where a MSER based method was utilized to get text regions from the last convolutional layer of the classification network. Semi-supervised based methods usually use a pretrained supervised model and the unlabeled data to train a text detector. Liu *et al.* [13] 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. These methods can really reduce the annotation cost, however, there is a large gap between their performance and those trained with strongly annotated data.

C. Mixed-Supervised Scene Text Detection

To learn well-performing models with weak labels, mixed-supervised methods have been used in object detection [38], semantic segmentation [39] and other fields. In the field of scene text detection, there are also some methods utilizing strong annotated data combined with weakly annotated data for training. WeText [40] trains scene text detection models on a small number of character-level annotated text images, followed by boosting the performance with a much larger number of weakly annotated images at the word/text line level. CRAFT [35] proposes a text detection method by using characters and the affinity between characters, and it uses word-level annotations of real data to generate character pseudo labels for fine-tuning. Sun *et al.* [21] published a large

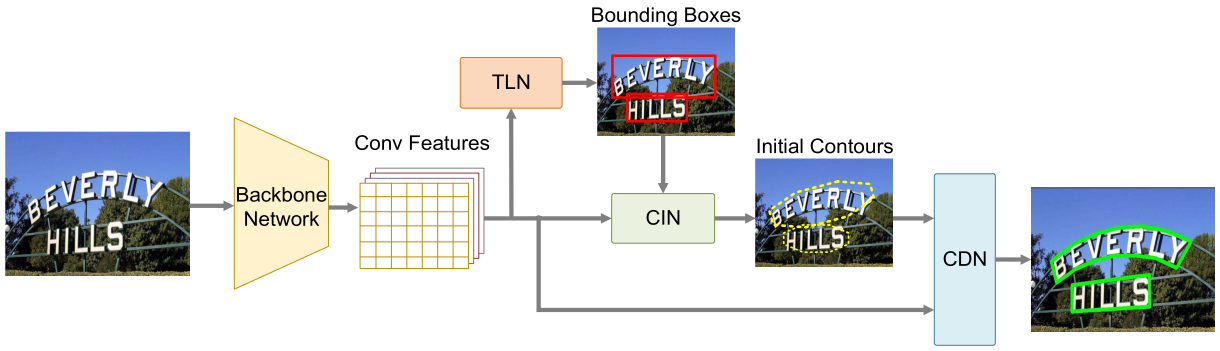

Fig. 4. Illustration of the structure of the proposed text detector."TLN", "CIN", and "CDN" represent text localization network, contour initialization network and contour deformation network, respectively.

partially annotated dataset (each image is only annotated with one dominant text) and proposed an algorithm to use these partially annotated data and strongly annotated data for joint training.

In this paper, we propose a mixed-supervised scene text detection framework that can utilize weakly annotated data to boost the detection performance. Our framework is flexible in incorporating various forms of weak labels (tight/loose/coarse bounding boxes, image-level tags), and when using a small amount of strongly annotated data combined with weakly annotated data, it can yield competitive performance to the detector training with large amount of strongly annotated data.

III. Methodology

The proposed mixed-supervised framework consists of two major parts: a contour-based scene text detector and an EM-based learning algorithm. The framework can utilize any of the four weak supervision forms introduced in this paper. In the following, we first introduce the methods for generating four forms of weak labels for public datasets, then describe two major parts of the framework in detail.

A. Four Forms of Weak Supervision

As analyzed above, labeling scene text with polygons is laborious and tedious. Therefore, we propose four forms of weak supervision (see Fig. 1) for scene texts as alternatives from the perspective of annotators. The *tight bounding box* is defined as the rectangle enclosing the text instance tightly, so the annotator needs to find the four extreme points of text contour, which is still time-consuming. In order to speed up annotation, we then propose the *loose bounding box*, which could be broader than the *tight bounding box*. Later, based on the observation of the text distribution in the scene, we further simplify the annotation process, using *coarse bounding box* to roughly locate the position of a cluster of texts (multiple instances). However, when the number of images is very large, the cost of any box-level annotation is also very large, and the *image-level tag* becomes the easiest choice. As shown in Fig. 1, with the decrease of annotation complexity, the time costs of labeling with different annotation policies are gradually decreasing.

Public datasets of arbitrarily-shaped text detection are mostly annotated with polygon-level labels, from which the proposed four forms of weak labels can be generated. The *tight bounding box* can be obtained from the external bounding box of the polygon. The *loose bounding box* can be obtained by expanding the height and width of the tight bounding box by 0.1 to 0.2 times respectively. The *coarse bounding box* can be generated from the external bounding box of the text cluster. We adopt Mean Shift algorithm [41] to cluster the centers of text regions, where the clustering radius is set to 0.3 times of the short side length of the image (see Fig. 3). The *image-level tag* indicates whether an image contains text or not, which can be easily obtained.

Our purpose of generating weak labels from polygon-level labels is to evaluate the effectiveness of the mixed-supervised text detection. This is also meaningful for practical applications, where it is much easier (cheaper) to annotate texts in box-like weak labels than in polygon-level labels. For instance, we compare several annotation policies on a public dataset (ICDAR-ArT). As shown in Fig. 2, combing the proposed weak supervisions with polygon, the time cost of labeling is greatly reduced.

B. Contour-Based Text Detection

Inspired by [42], we propose a contour based scene text detector for better facilitating mixed-supervised learning. An overview of the detector is illustrated in Fig. 4. After extracting original features by the backbone network, a text localization network is used to generate bounding-box-level text proposals. After that, we adopt a contour initialization network to produce the initial text contour for each text proposal. Finally, initial text contours together with original features are sent to the contour deformation network, which performs iterative contour regression to obtain the text instance boundary. Since the proposal mechanism in this pipeline is very close to the proposed weak supervision forms (coarse, loose, and tight bounding boxes), the proposed method can fully utilize the weak labels to boost the detection performance.

1) Text Localization Network: We adopt the CenterNet [43] to generate text proposals, which reformulates the detection task as a keypoint detection problem. The detection head

has two branches: (1) The classification branch calculates a heatmap, where the peaks are supposed to be the text instance centers; (2) The regression branch predicts the height and width of the proposal bounding box for each peak.

For the generation of bounding boxes, at inference time, we first extract the peaks in the heatmap, and detect all responses whose value is greater or equal to its 8-connected neighbors and keep the top 100 peaks. Let $\hat{\mathcal{P}}$ indicates the set of n detected center points $\hat{\mathcal{P}} = \{(\hat{u}_i, \hat{v}_i)\}_{i=1}^{n}$, where $(\hat{u}_i, \hat{v}_i))$ is the coordinates of each keypoint location. Then, a bounding box at a keypoint location can be produced by:

$$(\hat{u}_i - \hat{w}_i/2,\ \hat{v}_i - \hat{h}_i/2,\ \hat{u}_i + \hat{w}_i/2,\ \hat{v}_i + \hat{h}_i/2), \qquad (1)$$

where $(\hat{w}_i, \hat{h}_i)$ is the width and height prediction. Finally, a IoU based non-maxima suppression is conducted to remove redundant samples.

2) Contour Initialization Network: Since the initial input has influence on the contour deformation and the detected text proposals usually have some offsets or errors, we propose this network to produce more accurate and suitable initial contours for text instances. In [44] and [42], octagon encloses the arbitrarily-shaped object much tighter than the rectangle. Therefore, we also choose it as the initial contour. In fact, the octagon could be formed by four extreme points, which are the top, leftmost, bottom, rightmost pixels in an object, denoted by $\{z_i^{ex} | i = 1, 2, 3, 4\}$. Therefore, the problem is how to get the extreme points from the bounding box.

Given a bounding box, we could extract the four center points at the top, left, bottom, right box edges, denoted by $\{z_i^{bb} | i = 1, 2, 3, 4\}$, and then connect them to get a diamond contour. After that, we adopt the Deep Snake, a contour regression model proposed by [42]. It takes the diamond contour as input and outputs four offsets that point from each z_i^{bb} to the extreme point z_i^{ex}, namely $z_i^{ex} - z_i^{bb}$. Finally, we extend a line in both directions at each extreme point, whose length is 1/4 to the corresponding edge, and connect their endpoints to get the octagon.

The network architecture of the contour regression model is shown in Fig. 5, which consists of three parts: a backbone, a fusion block, and a prediction head. The backbone contains 8 CirConv-Bn-ReLU layers and uses residual skip connections for all layers, where CirConv means circular convolution. The fusion block is used to fuse the information across all contour points at multiple scales. Specifically, it concatenates features from all layers in the backbone and forwards them through a 1×1 convolution layers to the vertex features and output vertex-wise offsets.

3) Contour Deformation Network: Contour deformation network is used to regress the offsets from points on the initial contour to the corresponding points on the ground-truth. We adopt the same regression method as the contour initialization. To make the output contour smoother, we sample N points along the octagon contour. Similarly, the ground-truth is generated by uniformly sampling N vertices along the polygon. In addition, in order to simplify the difficulty of regression, an iterative optimization strategy is adopted. Specifically, the output contour of the previous iteration is used as the initial contour of the next iteration. In our experiments, we set the number of iterations to 3.

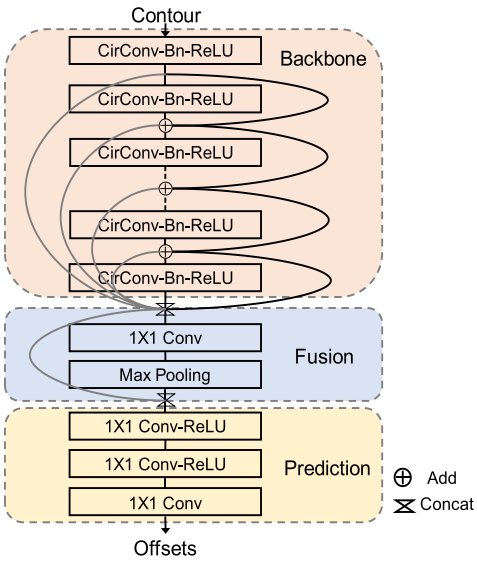

Fig. 5. The network architecture of the contour regression model, which contains three parts: a backbone, a fusion block, and a prediction head. The black lines and grey lines indicate residual learning and feature fusion, respectively.

C. EM-Based Learning Algorithm

Most existing methods [14], [15], [16], [17], [18] formulate the weakly-supervised localization task as a multiple instance learning (MIL) problem. The method in [17] defines an MIL classification objective based on the per-class spatial maximum of the local label distributions, and [18] adopts a softmax function. Methods in [14], [15], [16] propose to leverage convolutional neural networks (CNN) to extract discriminative appearance features. While this kind of approach has worked well for image classification tasks [45], [46], it is less suitable for text detection as it does not promote full object coverage: The model becomes tuned to focus on the most distinctive object parts instead of capturing the whole object. In order to avoid the dilemma of local object coverage, in this paper, we propose to employ the EM algorithm for the task of mixed-supervised scene text detection. The basic idea behind our EM algorithm is the alternating optimization: in the E-step, pseudo strong supervision information is recovered from the weak annotations; while in the M-step, the parameters of the whole network are optimized with recovered supervised information. Specifically, let x denotes the image values, y denotes the polygon-level labels of the image, and t denotes the weak labels of the image. As for the weakly annotated image, we can observe the image values x and the weak labels t, but the text instances polygons are latent variable. We have the following probabilistic graphical model:

$$P(x, y, t; \theta) = P(x)P(y|x; \theta)P(t|y). \qquad (2)$$

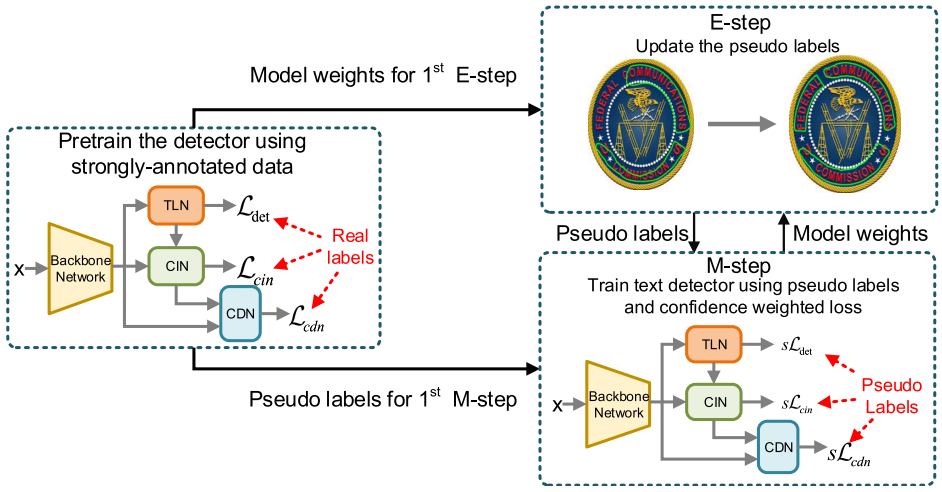

Fig. 6. The pipeline of the EM-based learning algorithm. The algorithm is firstly initialized by the model trained with a small amount of strongly annotated data. Then the algorithm alternates between updating the pseudo labels and optimizing the model weights with the pseudo label learning, where the confidence weighted loss is adopted.

Then, in order to learn the model parameters θ from the weakly annotated data, we adopt an EM-based learning strategy as follows:

1) E-Step: The purpose of E-step is to estimate the complete-data log likelihood. Given the previously estimated parameter θ', the expected complete-data log likelihood for weakly annotated image x and it's label t is given by

$$\begin{aligned} Q(\theta;\theta') &= \sum_{y} P(y|x,t;\theta')\log P(y|x;\theta) \\ &\approx \log P(\hat{y}|x;\theta), \end{aligned} \tag{3}$$

where we adopt a hard-EM approximation, estimating the latent variable by

$$\hat{y} = \arg\max_{y} P(y|x,t;\theta'). \tag{4}$$

2) M-Step: The M-step is to maximize the $Q(\theta;\theta')$ with respect to θ. According to Eq. 3, the key to maximize $Q(\theta;\theta')$ is maximizing $\log P(\hat{y}|x;\theta)$. Here, we treat $\hat{y}$ as ground truth polygons, and optimize $\log P(\hat{y}|x;\theta)$ by the mini-batch SGD algorithm.

We integrate the detector in Section III-B into the learning algorithm, and obtain a pipeline for mixed-supervised text detection, which is shown in Fig. 6. The parameter θ is equivalent to the weights of the detection model. We use a small amount of strongly annotated data to train a model, which provide the initial state for the 1st M-step. And the estimated latent polygon-level label $\hat{y}$ is given by the output of contour deformation network in the detection model. As shown in Eq. 4, $\hat{y}$ is related to weak labels t. Different weak supervisions will provide different information, so there are different approaches to estimate the latent variables. We introduce them separately as follows:

3) Learning With Image-Level Tags: The whole pipeline is shown in Fig. 7(a). We send the text images to the detection model of the previous M-step, and obtain a candidate pseudo annotation set $D_I = \{(b_1, p_1, s_1), (b_2, p_2, s_2), \ldots, (b_n, p_n, s_n)\}$, where (b_j, p_j, s_j) corresponds to the bounding box, polygon, and confidence score of the j-th detected instance, respectively. For the issue of false positives, we adopt a filter, which use the confidence threshold to select the more reliable samples. This process can be formulated as:

$$D'_I = \{(b_j, p_j, s_j)|s_j > S, (b_j, p_j, s_j) \in D_I\}, \tag{5}$$

where D'_I is the final pseudo annotation set, and S is the confidence threshold.

4) Learning With Coarse Bounding Boxes: Similarly, the detection model of previous M-step is applied to the weakly annotated images (see Fig. 7(a)), and a candidate pseudo annotation set D_C is obtained. In the filtering process, in addition to the confidence threshold, we can utilize the coarse bounding box which can perform like a text filter. Specifically, we use the IoU between detection results and ground truths as a metric to filter out the false positives. This process can be formulated as:

$$\begin{aligned} D'_C = \{&(b_j, p_j, s_j)|s_j > S, (b_j, p_j, s_j) \in D_C, \\ &\max_k \text{IoU}(b_j, g_k) > H, g_k \in G_C\}, \end{aligned} \tag{6}$$

where D'_C is the final pseudo annotation set, G_C is the ground-truth set, S and H is the confidence threshold and IoU threshold respectively.

5) Learning With Tight Bounding Boxes: Given the tight bounding boxes, the text proposal localization in the detection pipeline is unnecessary. We replace the bounding boxes generated by text localization network with the bounding boxes from weak labels (see Fig. 7(b)). With the ground truth bounding boxes, the more accurate initial contour are generated. Accordingly, the contour deformation network can predict better final contours. Therefore, all the generated results can be taken into final candidate pseudo annotation set D'_T, and their confidence scores are all set to 1, which can be formulated as:

$$D'_T = \{(g_k, p_k, s_k)|g_k \in G_T\}, \tag{7}$$

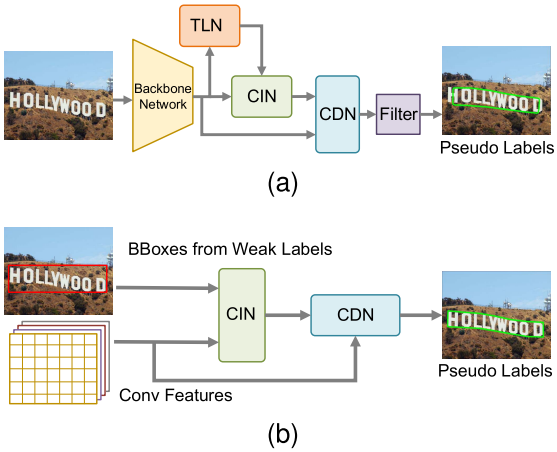

Fig. 7. Pseudo labels generation. (a) Pipeline for images labeled with image-level tags or coarse bounding boxes, which includes an inference step and a filtering process. (b) Pipeline for images labeled with tight or loose bounding boxes, where bounding boxes generated by text localization network are replaced with the bounding boxes from weak labels.

where D'_T is the final pseudo annotation set, and G_T is the ground truth set.

6) Learning With Loose Bounding Boxes: Almost the same approach as the tight bounding box is utilized in this case (see Fig. 7(b)). However, the loose bounding boxes contain more background noises than tight bounding boxes, which might deteriorate the performance of contour regression. Therefore, care must be taken during pre-training in this case, the generated bounding boxes of text proposals should be expanded randomly before sent to the contour initialization network, which can weaken the sensitivity of contour initialization network to the looseness of text bounding box. Similar to tight bounding boxes, all the generated results are taken into final candidate pseudo annotation set D'_L and their confidence scores are all set to 1. The final candidate pseudo annotation can be written as:

$$D'_L = \{(g_k, p_k, s_k) | g_k \in G_L\}, \tag{8}$$

where D'_L is the final pseudo annotation set, and G_L is the ground truth set.

D. Loss Function

During network training, the same loss function as CenterNet [43] is adopted in text localization, which can be denoted as $\mathcal{L}_{det}$. And the smooth ℓ_1 loss is used to learn contour initialization and contour deformation. The loss function for contour initialization is defined as

$$\mathcal{L}_{cin} = \frac{1}{4}\sum_{i=1}^{4} \ell_1(\hat{z}_i^{ex} - z_i^{ex}), \tag{9}$$

where $\hat{z}_i^{ex}$ is the predicted extreme points. And the loss function for iterative contour deformation is defined as

$$\mathcal{L}_{cdn} = \frac{1}{N}\sum_{i=1}^{N} \ell_1(\hat{z}_i - z_i^{gt}), \tag{10}$$

where N, $\hat{z}_i$ and z_i^{gt} are the number of sampling points on the contour, the deformed contour point, and the ground-truth boundary point, respectively. For the end-to-end training of the three tasks, the whole loss function can be formulated as:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda_1 \mathcal{L}_{cin} + \lambda_2 \mathcal{L}_{cdn}, \tag{11}$$

where the weights constants λ_1 and λ_2 are all set to 1 in our experiments.

Inevitably, even if we filter the candidate pseudo annotation set generated in E-step, there are still negative samples in the final pseudo annotation set, especially for D'_I and D'_C. Once using these noisy labels to update the parameters in M-step, the performance of the model tends to deteriorate iteratively. To make the training focus more on reliable samples, a confidence weighted loss $\hat{\mathcal{L}}$ is proposed as follows:

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

where s is the confidence score of a sample obtained in the E-step (see Eq. 5-8), which ensures the stability of the training process.

IV. Experiments

A. Datasets

We evaluated the performance of the proposed method on six public datasets, including three datasets of arbitrary shaped text and three of oriented text.

1) CTW1500: The CTW1500 dataset [9] contains 1000 training images and 500 test images. Besides horizontal and multi-oriented texts, at least one curved text is contained in each image. Each text is labeled as a polygon with 14 vertexes in line-level.

2) Total-Text: The Total-Text dataset [19] 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.

3) ICDAR-ArT: The ICDAR-ArT dataset [10] consits of 5603 training images and 4563 test images, which contains multi-lingual arbitrarily-shaped texts. Each text is labeled with adaptive number of vertices.

4) ICDAR2015: The ICDAR2015 dataset [22] contains 1000 training images and 500 texting images. This dataset is focused on incidental scene text, in which each text is labeled as a quadrangle (a special case of polygon) with 4 vertexes in word-level.

5) MSRA-TD500: The MSRA-TD500 dataset [20] contains 300 training images and 200 test images, where there are many multi-oriented text lines. Texts in this dataset are stably captured with high resolution and bi-lingual of both English and Chinese. Each text is labeled as a quadrangle with 4 vertexes in line-level.

6) C-SVT: The C-SVT dataset [21] 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 annotated images for the convenience of evaluation.

B. 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 use the DLA-34 [47] as the backbone network. For all the models, we first pre-train them with the SynthText [48] dataset, then fine-tune the models on the corresponding real-world datasets. The number of sampling points on contour N is set to 128, which can fit well most arbitrary shapes. For different experiments, the hyperparameters setting is consistent: confidence threshold $S = 0.35$, and IoU threshold $H = 0.7$. Each M-step consists of 200 epochs. And the batchsize is set to 36. We adopt the "multistep" strategy to adjust the learning rate. The initial learning rate is set to 1×10^{-4} and is divided by 2 at 80th, 120th, 150th, and 170th epoch. For data labeled with coarse bounding boxes or image-level tags, we stop training after 3 M-steps. For data labeled with tight or loose bounding boxes, since the initial pseudo annotations are of high quality, it just needs 1 M-step. Data augmentation is important for training, especially when there is only a small amount of strongly annotated data: (1) rescaling images with ratio from 0.5 to 2.0 randomly, (2) flipping horizontally and rotating in range $[-10°, 10°]$ 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 (460 for CTW1500, 960 for ICDAR-ArT, 720 for ICDAR2015, and 640 for Total-Text, MSRA-TD500 and C-SVT), with the aspect ratio kept. Since the evaluation tool of ICDAR2015 and MSRA-TD500 only supports quadrangle results, minAreaRect in OpenCV is used to obtain the bounding boxes of the text contours.

C. Mixed-Supervised Experimental Results

For all six datasets, we randomly select 10% of original training images as strongly annotated data and take other 90% as weakly annotated data, resulting in a 100-900 split for CTW1500 and ICDAR2015, a 125-1130 split for Total-Text, a 560-5043 split for ICDAR-ArT, a 30-270 split for MSRA-TD500, and 3000-27000 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%XXX*: Model trained with all images, of which 10% are annotated with polygons, and 90% are annotated with a kind of weak annotation.

Considering the impact of data split on the final results, for each mixed-supervised experiment, we run data split five times randomly, and report the average of the five results. The results on six benchmarks are given in Table I-VI, respectively.

*1) Results of the "*10%Poly & 90%Tag*":* As shown in Table I-VI, the "*10%Poly & 90%Tag*" outperforms the baseline model "*10%Poly*" obviously (by 1%, 1.4%, 3.6%, 2.2%, 2.9%, and 1.8% on CTW1500, Total-Text, ICDAR-ArT, ICDAR2015, MSRA-TD500, and C-SVT, respectively). Although the image-level tag contains little supervision information, it still improves the performance evidently due to the expansion of data. Therefore, this kind of weak label is suitable for the scene where a large number of cost-free images can be obtained, such as in the field of automatic driving, where the camera on the car can capture a large amount of data. If we discard these data, it will lead to a waste of data. Using our method, these data can be made useful by indicating annotate presence/absence of texts only.

*2) Results of the "*10%Poly & 90%Coarse*":* The performance of the model trained with coarse bounding boxes on six datasets is shown in Table I-VI. With the help of coarse bounding box, the performance of "*10%Poly & 90%Coarse*" outperforms "*10% & 90%Tag*" by about 1% to 2%, which verifies the importance of location information for detection

TABLE I

DETECTION RESULTS ON CTW1500. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Type	Model	P	R	F	FPS
Segmentation-based	TextSnake [34]	67.9	**85.3**	75.6	-
	LOMO [7]	**89.2**	69.6	78.4	3.0
	SAE [49]	82.7	77.8	80.1	-
	SAST [50]	85.3	77.1	81.0	27.6
	TextField [51]	83.0	79.8	81.4	-
	PSENet-1s [6]	84.8	79.7	82.2	3.9
	Wu *et al.*-TAS [11]	83.8	80.8	82.3	9.2
	Wu *et al.*-FM [11]	86.2	80.5	83.2	9.0
	DB-ResNet-50 [5]	86.9	80.2	83.4	22.0
	CRAFT [35]	86.0	81.1	83.5	-
	PAN-640 [52]	86.4	81.2	83.7	**39.8**
	CRNet [53]	87.0	80.9	83.8	-
Hybrid	CSE [54]	81.1	76.0	78.4	-
	ContourNet [55]	83.7	84.1	83.9	4.5
	Mask-TTD [56]	79.7	79.0	79.4	-
	SD [57]	85.8	82.3	84.0	-
Regression-based	CTD-CLOC [58]	77.4	69.8	73.4	-
	ATRR [59]	80.1	80.2	80.1	-
	TextRay [60]	82.8	80.4	81.6	-
	100%Poly	86.1	82.1	**84.1**	32.3
	10%Poly & 90%Tight	$86.0_{\pm 0.52}$	$81.2_{\pm 0.36}$	$83.6_{\pm 0.12}$	
	10%Poly & 90%Loose	$86.3_{\pm 0.85}$	$80.1_{\pm 0.74}$	$83.1_{\pm 0.11}$	
	10%Poly & 90%Coarse	$84.0_{\pm 0.55}$	$80.6_{\pm 0.62}$	$82.3_{\pm 0.09}$	
	10%Poly & 90%Tag	$83.4_{\pm 0.67}$	$79.1_{\pm 0.59}$	$81.2_{\pm 0.13}$	
	10%Poly	$81.3_{\pm 0.79}$	$79.1_{\pm 1.02}$	$80.2_{\pm 0.26}$	

TABLE II

DETECTION RESULTS ON TOTAL-TEXT. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Type	Model	P	R	F	FPS
Segmentation-based	TextSnake [34]	82.7	74.5	78.4	-
	TextDragon [4]	85.6	75.7	80.3	-
	TextField [51]	81.2	79.9	80.6	-
	PSENet-1s [6]	84.0	78.0	80.9	3.9
	LOMO [7]	88.6	75.7	81.6	3.0
	Wu *et al.*-TAS [11]	78.5	76.7	77.6	11.2
	Wu *et al.*-FM [11]	83.1	81.6	82.4	10.4
	CRAFT [35]	87.6	79.9	83.6	-
	DB-ResNet-50 [5]	87.1	82.5	84.7	32.0
	PAN-640 [52]	**89.3**	81.1	85.0	**39.6**
Hybrid	SemiText-Transductive [13]	78.0	58.3	66.7	2.1
	SemiText-Inductive [13]	79.2	59.0	67.6	2.1
	SemiText-GT [13]	84.5	**84.7**	84.6	2.1
	Mask-TextSpotter-v2 [8]	81.8	75.4	78.5	-
	SPCNet [61]	83.0	82.8	82.9	-
	ContourNet [55]	86.9	83.9	85.4	3.8
Regression-based	TextRay [60]	82.8	80.4	81.6	-
	ReLaText [62]	84.8	83.1	84.0	-
	Wang *et al.* [37]	85.2	83.5	84.3	-
	100%Poly	88.2	83.3	**85.6**	24.2
	10%Poly & 90%Tight	$85.4_{\pm 0.79}$	$83.8_{\pm 0.62}$	$84.6_{\pm 0.12}$	
	10%Poly & 90%Loose	$86.6_{\pm 0.54}$	$82.1_{\pm 0.49}$	$84.3_{\pm 0.18}$	
	10%Poly & 90%Coarse	$84.7_{\pm 0.39}$	$79.6_{\pm 0.23}$	$82.0_{\pm 0.21}$	
	10%Poly & 90%Tag	$82.9_{\pm 0.30}$	$78.8_{\pm 0.49}$	$80.8_{\pm 0.27}$	
	10%Poly	$80.2_{\pm 0.51}$	$78.5_{\pm 0.38}$	$79.4_{\pm 0.23}$	

TABLE III

DETECTION RESULTS ON MSRA-TD500. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Type	Model	P	R	F	FPS
Segmentation-based	DDR [2]	77.0	70.0	74.0	-
	EAST [3]	87.3	67.4	76.1	13.2
	SegLink [33]	86.0	70.0	77.0	8.9
	PixelLink [63]	83.0	73.2	77.8	3.0
	Wu *et al.*-TAS [11]	80.6	74.1	77.2	12.0
	Wu *et al.*-FM [11]	83.2	76.6	79.8	11.6
	TextField [51]	87.4	75.9	81.3	-
	CRAFT [35]	88.2	78.2	82.9	-
	PAN-640 [52]	84.4	**83.8**	84.1	**30.2**
	CRNet [53]	86.0	82.0	84.0	-
Hybrid	Mask-TextSpotter-v2 [8]	80.8	68.6	74.2	-
	Lyu *et al.* [64]	87.6	76.2	81.5	-
	Mask-TextSpotter-v3 [65]	**90.7**	77.5	83.5	-
Regression-based	RRPN [66]	82.0	68.0	74.0	-
	RRD [67]	87.0	73.0	79.0	10.0
	100%Poly	88.7	81.1	**84.7**	24.2
	10%Poly & 90%Tight	$87.5_{\pm 0.74}$	$80.6_{\pm 0.61}$	$83.9_{\pm 0.11}$	
	10%Poly & 90%Loose	$88.4_{\pm 0.53}$	$78.9_{\pm 0.34}$	$83.4_{\pm 0.15}$	
	10%Poly & 90%Coarse	$83.9_{\pm 0.44}$	$74.5_{\pm 0.46}$	$78.9_{\pm 0.31}$	
	10%Poly & 90%Tag	$83.5_{\pm 0.39}$	$71.1_{\pm 0.23}$	$76.8_{\pm 0.29}$	
	10%Poly	$81.2_{\pm 0.30}$	$67.7_{\pm 0.39}$	$73.9_{\pm 0.27}$	

TABLE IV

DETECTION RESULTS ON ICDAR-2015. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Type	Model	P	R	F	FPS
Segmentation-based	SegLink [33]	73.1	76.8	75.0	-
	EAST [3]	83.6	73.5	78.2	13.2
	DDR [2]	82.0	80.0	81.0	-
	PixelLink [63]	82.9	81.7	82.3	7.3
	TextSnake [34]	84.9	80.4	82.6	1.1
	PAN-640 [52]	84.9	81.9	82.9	**26.1**
	TextFild [51]	84.3	83.9	84.1	1.8
	PSENet-1s [6]	86.9	84.5	85.7	1.6
	CRAFT [35]	85.3	**89.0**	**87.1**	**-**
Hybrid	Lyu *et al.* [64]	**89.5**	79.7	84.3	-
	ContourNet [55]	86.1	87.6	86.9	3.5
	Mask-TextSpotter-v2 [8]	86.6	87.3	87.0	-
Regression-based	RRPN [66]	82.0	73.0	77.0	-
	RRD [67]	85.6	79.0	82.2	6.5
	Wang *et al.* [37]	88.1	82.2	85.0	
	100%Poly	89.4	82.4	85.8	21.6
	10%Poly & 90%Tight	$86.0_{\pm 0.23}$	$82.5_{\pm 0.39}$	$84.2_{\pm 0.27}$	
	10%Poly & 90%Loose	$84.4_{\pm 0.33}$	$83.0_{\pm 0.49}$	$83.7_{\pm 0.17}$	
	10%Poly & 90%Coarse	$84.8_{\pm 0.64}$	$78.8_{\pm 0.52}$	$81.7_{\pm 0.32}$	
	10%Poly & 90%Tag	$80.9_{\pm 0.47}$	$77.6_{\pm 0.35}$	$79.2_{\pm 0.29}$	
	10%Poly	$77.2_{\pm 0.26}$	$76.9_{\pm 0.33}$	$77.0_{\pm 0.22}$	

TABLE V

DETECTION RESULTS ON ICDAR-ART. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Model	P	R	F	FPS
TextRay [60]	76.0	58.6	66.2	-
Dai *et al.* [68]	**84.0**	66.1	74.0	-
100%Poly	80.8	**68.7**	**74.3**	**16.4**
10%Poly & 90%Tight	$81.9_{\pm 0.27}$	$67.5_{\pm 0.36}$	$74.0_{\pm 0.27}$	
10%Poly & 90%Loose	$82.0_{\pm 0.29}$	$67.3_{\pm 0.38}$	$73.9_{\pm 0.39}$	
10%Poly & 90%Coarse	$77.7_{\pm 0.54}$	$65.8_{\pm 0.66}$	$71.3_{\pm 0.44}$	
10%Poly & 90%Tag	$77.9_{\pm 0.49}$	$64.2_{\pm 0.55}$	$70.4_{\pm 0.51}$	
10%Poly	$77.2_{\pm 0.63}$	$58.9_{\pm 0.38}$	$66.8_{\pm 0.42}$	

model. Compared with "*10% & 90%Tag*", the IoU based filter process is added here, which can ensure that lots of false positive samples are reduced in the pseudo annotation sets used in the iterative training process. It is the improvement of quality of training samples that leads to the improvement of performance. It is noteworthy that the F-measure of "*10%Poly & 90%Coarse*" is already comparatively close to the performance of the other methods.

TABLE VI

DETECTION RESULTS ON C-SVT. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Model	P	R	F	FPS
EAST [3]	73.4	**79.3**	76.2	-
Sun *et al.*-Train [21]	80.4	74.6	77.4	-
Sun *et al.*-Train+400K Weak [21]	81.7	75.2	78.3	-
100%Poly	**83.5**	74.5	**78.9**	**24.2**
10%Poly & 90%Tight	$81.9_{\pm 0.53}$	$74.9_{\pm 0.41}$	$78.2_{\pm 0.31}$	
10%Poly & 90%Loose	$80.4_{\pm 0.47}$	$75.5_{\pm 0.63}$	$77.9_{\pm 0.41}$	
10%Poly & 90%Coarse	$79.3_{\pm 0.73}$	$72.1_{\pm 0.53}$	$75.5_{\pm 0.43}$	
10%Poly & 90%Tag	$77.8_{\pm 0.63}$	$72.4_{\pm 0.44}$	$75.0_{\pm 0.48}$	
10%Poly	$73.5_{\pm 0.61}$	$72.9_{\pm 0.38}$	$73.2_{\pm 0.39}$	

*3) Results of the "*10%Poly & 90%Loose*":* In the experiments, loose bounding boxes are used to guide the generation of pseudo annotation sets so as to capture all the text instance in the images. In addition, the quality of pseudo labels has been significantly improved. Inspired by [69], we have used the TIoU-metric to measure the compactness of predicted polygons against the ground truth in this case, and have achieved 77.0% and 77.2% (F-measure) on CTW1500 and Total-Text, respectively, which verifies that our pseudo labels are very close to the standard annotations. Therefore, The performance of the "*10%Poly & 90%Loose*" is close to that of the fully supervised model, despite that slight inferiority to that of "*10%Poly & 90%Tight*".

*4) Results of the "*10%Poly & 90%Tight*":* Compared with the other three forms of weak supervision, tight bounding box contains the least noise or background area. Therefore, the quality of pseudo labels generated under its guidance is also the best. We also used the TIoU metric to measure the compactness of predicted polygons in this case, and have achieved 77.5% and 77.8% (F-measure) on CTW1500 and Total-Text, respectively. These are very close to the performance of training with full strong annotation data (100%Poly). As shown in Table I-VI, the "*10%Poly & 90%Tight*" achieves the best performance among the four mixed-supervised models. Using only 10% strongly-supervised data, we can reach the performance close to the state of the art, which demonstrates the effectiveness of our mixed-supervised framework.

5) Qualitative Results: The visualization of the pseudo labels generated under different forms of weak supervision are illustrated in Fig. 8. We can see that the pseudo label generated under the guidance of tight bounding box can tightly wrap the text, while in the case of loose bounding box, the pseudo label covers more background area. In addition, in the case of image-level tag, there are some false positive samples in the pseudo labels sets (see the first column of the Fig. 8), which cannot be eliminated based on the confidence threshold. However, in the case of coarse bounding box, we can mask them out using the IoU based filter (comparing the first and second columns in Fig. 8).

6) Comparison with Other Methods Using Weak Labels: Compared to methods which use character-box to boost the performance of word detection, our mixed-supervised models "10%Poly & 90%Tight" and "10%Poly & 90%Tight" have surpassed CRAFT [35] obviously, as shown in Tables I - III. In addition, it should be noted that CRAFT requires both

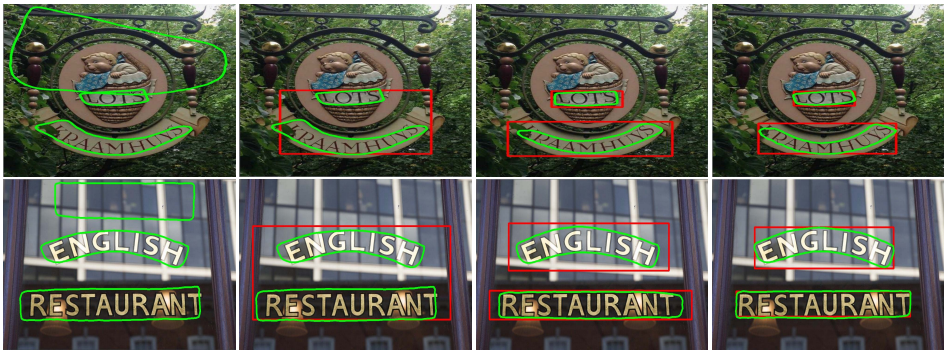

Fig. 8. Comparison of pseudo labels generated under different forms of weak supervision. The red boxes indicate weak labels provided, and the green polygons indicate pseudo labels generated. From left to right, there are image-level tag, coarse bounding box, loose bounding box, and tight bounding box.

full polygon-level annotations and pseudo character-box-level annotations for training, while our method only needs 10% polygon-level annotations and 90% weak annotations. Our method yields higher performance at lower labeling cost. Besides, WeText [40] also uses character-box to supervise the training of word detector. Because it has been tested only on ICDAR-2013 [70], we also evaluated our method on this dataset for comparison. Under full supervision, our method yield F-measure 1.1% higher (88.0% vs 86.9%) than WeText which uses additional character supervision information.

Compared to methods which use partial annotations to reduce labeling cost, our method also shows advantage. As shown in Tables I - III, our fully-supervised model and mixed-supervised model both outperform the corresponding models (FM and TAS) of Wu *et al.* [11], which uses line-level annotations to supervise model training. In addition, Sun *et al.* [21] 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 4000,000 weakly annotated data. In contrast, our mixed-supervised model yields significant improvement compared with baseline.

Compared to semi-supervised method [13], our method also outperforms by a large margin, as shown in Table II. 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.

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

1) Evaluation on CTW1500: As shown in Table I, our fully-supervised model "*100%Poly*" achieves state-of-the-art performance. Specifically, our method outperforms the regression-based methods (*e.g.* CTD-CLOC [54], ATRR [59] and TextRay [60]) by a large margin. When comparing with PSENet [6] which segments text regions using a progressive scale expansion, our method has promoted F-measure by 1.9%. Besides, compared with LOMO [7] that also localizes texts progressively, the performance of our method increases by 5.7% in F-measure. When fixing the short side length of input image as 460, and our method achieves a speed of 32.3 FPS, which is also competitive. The qualitative examples are shown in the first row of Fig. 9, where we can see that our method can handle long curved texts.

2) Evaluation on Total-Text: As shown in Table II, our fully-supervised model "*100%Poly*" is obviously superior to the regression-based methods (*e.g.* TextRay [60], ReLaText [62] and Wang *et al.* [37]). Compared with the same contour based detection method [37], our method outperforms by 1.3% in F-measure. Besides, our method also outperforms the hybrid-based methods (*e.g.* Mask-TextSpotter-v2 [8], SPCNet [61], and ContourNet [55]). For instance, our method significantly boost the F-measure by 7.1%, compared with the well-known model Mask-TextSpotter-v2. When fixing the short side length of input image as 640, our method achieves a speed of 24.2 FPS. The qualitative examples are shown in the second row of Fig. 9.

3) Evaluation on MSRA-TD500: As shown in Table III, our fully-supervised model "*100%Poly*" achieves 84.7% in F-measure, outperforming all the other existing methods. Especially compared with well known method EAST [3] for detecting multi-oriented scene texts, out method outperforms by 1.4%, 13.7%, and 8.6% in Recall, Precision, and F-measure, respectively. Besides, when comparing with the PAN [52], which uses pixel aggregation to predict similarity vectors between text kernels and surrounding pixels, our method also promotes the F-measure from 84.1% to 84.7%. When fixing the short of input image as 640, our method achieves a speed of 24.2 FPS. Some text detection results in the second row of Fig. 10 show that our method can handle multi-oriented texts very well.

4) Evaluation on ICDAR-2015: This dataset contains oriented texts, though our method is designed for arbitrary shaped texts. Compared with hybrid methods (Mask-TextSpotter-v2 [8] and ContourNet [55]), our method still achieves competitive performance as shown in Table IV. However, in terms of speed, our method outperforms them by a large margin (21.6 FPS vs 3.5 FPS). This demonstrates the advantage of our method in the trade-off between speed and accuracy. Some text detection results in the first row of Fig. 10 show that our method can works well even in perspective distortion scenario.

5) Evaluation on ICDAR-ArT: This new large dataset has been evaluated for existing methods TextRay [60] and Dai *et al.* [68]. As shown in Table V, our method achieves the best performance both in accuracy and speed. Some text detection results are shown in the third row of Fig 9, where

Fig. 9. Examples of curved text detection results. First row: CTW1500; second row: Total-Text; third row: ICDAR-ArT.

we can see that our method maintains good performance in a bilingual (English and Chinese) environment.

6) Evaluation on C-SVT: The results on C-SVT dataset are shown in Table VI. Compared to previous methods, the proposed method achieves state-of-the-art performance. Although the scale of Chinese text varies dramatically, the proposed contour iterative regression mechanism can make the framework robust to scale variance. Some text detection results in the third row of Fig. 10 show that the proposed method can detect text of various scales and orientations.

E. Ablation Studies

1) Influence of Contour Initialization and Circular Convolution: We conduct ablation studies on the CTW1500 dataset to evaluate the main components of our proposed text detector, including the contour initialization network and circular convolution for boundary deformation. Table VII summarizes the results. The row "Baseline" lists the result of a direct combination of text localization network and contour deformation network. Specifically, the text localization network produces bounding-box-level text proposals. Then the bounding boxes are deformed towards text boundaries through graph convolutional network (GCN). Note that, this baseline method represents the contour as a graph and uses a graph convolutional network for contour deformation. To evaluate the influence of the contour initialization, we add the contour initialization network before the contour deformation. Instead of directly using the bounding box, the proposal step generates an octagon initialization by predicting four text extreme points, which not only compensates for the detection errors but also encloses the text more tightly. The comparison between the first and the second row shows a 1.4% improvement in F-measure by contour initialization. Finally, the graph convolution is replaced with the circular convolution, which achieves 1% improvement in F-measure. The results of different convolution operators and different iterations of contour deformation are shown in Table VIII. It is evident that circular convolution outperforms graph convolution across all

TABLE VII
THE INFLUENCE OF CONTOUR INITIALIZATION AND CIRCULAR CONVOLUTION ON CTW1500 TEST SET. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Model	P	R	F
Baseline	83.8	79.7	81.7
+ Contour Initialization	85.1	81.2	83.1
+ Circular Convolution	**86.1**	**82.1**	**84.1**

TABLE VIII
RESULTS OF MODELS WITH DIFFERENT CONVOLUTION OPERATORS AND DIFFERENT ITERATIONS OF CONTOUR DEFORMATION ON CTW1500 TEST SET IN TERMS OF THE F-MEASURE

Operator	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5
Graph Conv	80.1	82.5	83.1	82.9	82.7
Circular Conv	80.6	83.5	**84.1**	83.8	83.2

Fig. 10. : Examples of multi-oriented text detection results. First row: ICDAR2015; second row: MSRA-TD500; third row: C-SVT.

TABLE IX

DETECTION RESULTS OF FIVE GROUPS OF DATA SPLIT ON CTW1500. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Exps	10%Poly			10%Poly+10%Tag			10%Poly+90%Coarse			10%Poly+90%Loose			10%Poly+90%Tight		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
1	82.0	78.7	80.3	82.9	79.3	81.0	83.6	80.7	82.1	85.5	80.6	83.0	86.0	81.2	83.6
2	82.1	78.8	80.4	84.9	77.7	81.1	83.0	81.3	82.2	84.1	81.9	83.0	86.0	81.7	83.8
3	83.1	77.8	80.4	83.7	79.1	81.4	82.7	81.9	82.3	85.3	81.4	83.3	86.6	81.4	83.9
4	81.3	79.1	80.2	83.4	79.1	81.2	84.0	80.6	82.3	86.3	80.1	83.1	86.4	81.1	83.7
5	83.5	76.3	79.7	84.0	78.5	81.2	84.1	80.1	82.1	86.5	80.0	83.1	85.1	82.1	83.6

inference iterations, which justifies that circular convolution is more suitable for deforming contours.

2) Influence of the Data Split: The above experiments used 10% strong labeled data in training the basic model, and for each experiment, the results were averaged over five random splits of strongly/weakly labeled data. To justify the stability of performance over different splits of data, we show the detailed results of five splits for five supervision policies on the CTW1500 dataset in Table IX. Although the performance of the "10%Poly" fluctuates between 79.7% and 80.4%, the differences among the four groups of mixed-supervised models are relatively small. This is because although the performance of the basic models is different, the quality of the pseudo labels generated by them is very similar. The stable quality of pseudo labels leads to stable promotion of performance of the basic model.

3) Influence of the Proportion of Strongly Annotated Data: The above experiments show that the mixed-supervised model with only 10% strongly annotated data can match the performance of the model trained with the full set of strongly annotated data. To evaluate the influence of the proportion of strongly annotated data, we perform experiments with variable proportion (from 1% to 40%) of strongly annotated data on the ICDAR-ArT dataset. The results are shown in Fig. 11. We can see that the performance of mixed-supervised models improves as the proportion of strongly annotated data increases. For "Tight BBoxes" and "Loose BBoxes", its performance quickly approaches the performance of the fully-supervised model. However, for "Tags" and "Coarse BBoxes", with the increase of strongly labeled data, the performance increases slowly. This is because the performance of the baseline model has been relatively high, and the pseudo labels derived from very weak supervision are noisy.

4) Influence of the Confidence Weighted Loss: We evaluate the influence of the weighted confidence loss (see Eq. 12) via experiments with or without it. The results of two groups

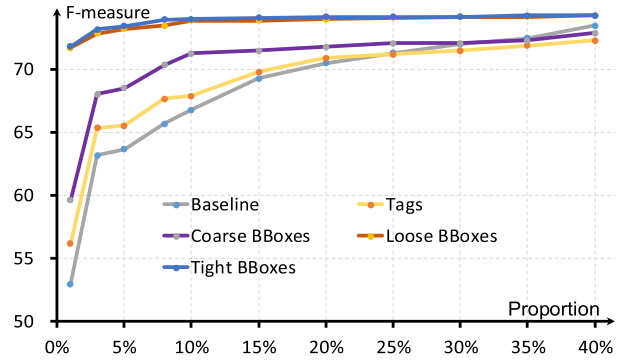

Fig. 11. F-measure versus proportions of strongly annotated data for four mixed-supervised models on ICDAR-ArT test set. The baseline model is trained with only a certain proportion of strongly annotated data. The other four mixed-supervised models are trained with a certain proportion of strongly annotated data and the remaining weakly annotated data (the legends represent the weak annotation policies).

TABLE X

THE BENEFITS OF THE CONFIDENCE WEIGHTED LOSS ON ICDAR-ART. "WL" INDICATES CONFIDENCE WEIGHTED LOSS. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY

Method	WL	P	R	F
10%Poly & 90%Coarse	-	**80.4**	60.8	69.2
	✓	77.7	**65.8**	**71.3**
10%Poly & 90%Tag	-	76.5	61.0	67.9
	✓	**78.0**	**64.2**	**70.4**

TABLE XI

THE QUALITY OF PSEUDO LABELS VERSUS TRAINING ROUNDS FOR TWO MIXED-SUPERVISED MODELS ON WEAKLY-ANNOTATED DATA ON CTW1500

Model	Rnd. 0	Rnd. 1	Rnd. 2	Rnd. 3
10%Poly & 90%Tag	69.8	71.9	74.9	74.5
10%Poly & 90%Coarse	71.1	73.2	75.7	75.5

of experiments on ICDAR-ArT are shown in Table X. It is verified that the models trained with confidence weighted loss yield better results. The confidence weighted loss improves the two basic models by about 2.1% and 2.5% of F-measure, respectively. It attributes to that with confidence weighting, the influence of noise samples on training can be limited, and the network learning tends to focus on reliable samples.

5) Influence of the Number of Training Rounds: The results of the mixed-supervised training within three rounds are shown in Fig. 12. The result of the 0 training round is the performance of the initial model. The performance of mixed-supervised models has improved in the first two rounds, which attributes to the improvement of pseudo label quality. Then the performance saturates at the third round, due to the error accumulation in more rounds. For better explaining the trend of model performance, we use the F-measure to evaluate the quality of pseudo labels generated in each round. As shown in Table XI, the quality of pseudo labels has improved in the first two rounds, and saturates at the third round.

6) Budget-Aware Omni-Supervised Text Detection: The proposed method is applicable to training with multiple forms of

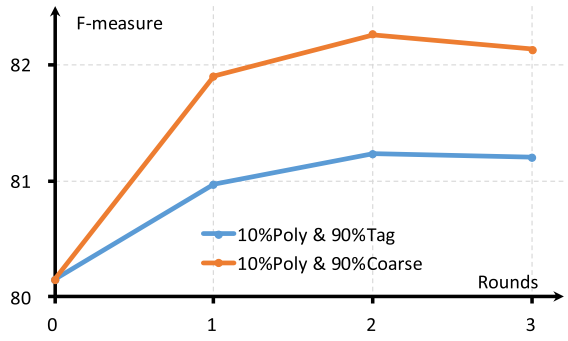

Fig. 12. F-measure versus training rounds for two mixed-supervised models on CTW1500 test set.

weak supervision, which we call omni-supervised learning. To evaluate the performance of this setting, we compare different models trained with the same budget of data annotation. Considering that the budget consumption is directly proportional to the time cost, we just use the time cost as the budget consumption.

The total budget is set to 43,200 seconds (12 hours). Then, we evaluate three protocols of budget assignment: (1) **Strong**: all the budget is used to annotate polygons. (2) **Equal Time**: in addition to 560 strongly labeled images, the remaining time is equally divided into four weak supervision forms. (3) **Equal Number**: in addition to 560 strongly labeled images, the remaining time is used for the four weak supervision forms, each with equal number of images under the budget. In addition, in order to evaluate the cost-effectiveness of the four weak supervision forms, we conduct another four comparative experiments, in which 80% budget is spent on polygons and the rest is spent on a form of weak supervision.

The results on two datasets ICDAR-ArT and C-SVT (Table XII) show that the performance of different models varies in a similar trend. Specifically, "Equal Time" and "Equal Number" protocol perform better than the "Strong" protocol, and the "Equal Time" protocol performs best. These results suggest that spending a certain amount of budget to annotate more images with weak labels is better than adopting fully strong labels. In the "80% Poly" experiments, the four weak supervision forms have achieved comparable results, and "Loose Bounding Box" is slightly superior. Nevertheless, all the results of "80% Poly" are inferior to that of "Equal Time", which exploits more images in training.

In order to verify the impact of different budgets on the performance of different annotation policies, we have conducted experiments with budget of 10h and 11h on ICDAR-ArT. As shown in Table XIII, with the budget of 10h and 11h, the three annotation policies perform comparably. Under the same budget, though cheap weak labeling allows more images, the benefit is offset by the inaccuracy of weak supervision. In particular, when the budget is 10h, "Equal Number" is even inferior to "Strong", which shows that the added weakly-annotated data brings more noise than supervisions. While, for the budget of 12h, the advantage of weakly labeled data exceeds the noisy pseudo labels, so "Equal Time" and "Equal Number" both outperform "Strong" obviously.

TABLE XII

DETECTION RESULTS OF DIFFERENT ANNOTATION POLICIES ON ICDAR-ART AND C-SVT. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY. IN THE "LABELS" COLUMN, THE "I", "C", "L", "T" AND "POLY" REPRESENT THE IMAGE-LEVEL TAGS, COARSE BOUNDING BOXES, LOOSE BOUNDING BOXES, TIGHT BOUNDING BOXES AND POLYGONS, RESPECTIVELY

Policy	Image amount	Labels	ICDAR-ArT			C-SVT		
			P	R	F	P	R	F
Strong	710	Poly	79.5	59.7	68.2	73.0	61.9	67.0
Equal Time	560+58+81+152+1143	Poly+T+L+C+I	**80.3**	**65.3**	**72.1**	74.4	**69.0**	**71.6**
Equal Number	560+108×4	Poly+T+L+C+I	78.7	63.7	70.4	75.8	64.4	69.6
80% Poly	568+219	Poly+T	77.1	63.8	69.8	70.0	67.1	68.5
	568+307	Poly+L	78.2	64.1	70.5	73.0	65.2	68.9
	568+576	Poly+C	77.1	63.2	69.5	69.0	66.9	67.9
	568+4320	Poly+I	75.8	64.4	69.6	**76.6**	62.5	68.8

TABLE XIII

DETECTION RESULTS OF DIFFERENT ANNOTATION POLICIES UNDER DIFFERENT BUDGETS ON ICDAR-ART. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY. IN THE "LABELS" COLUMN, THE "I", "C", "L", "T" AND "POLY" REPRESENT THE IMAGE-LEVEL TAGS, COARSE BOUNDING BOXES, LOOSE BOUNDING BOXES, TIGHT BOUNDING BOXES AND POLYGONS, RESPECTIVELY

Budeget	Policy	Image amount	Labels	P	R	F
10h	Strong	592	Poly	73.1	61.9	67.0
	Equal Time	560+12+17+32+243	Poly+T+L+C+I	72.1	62.9	67.2
	Equal Number	560+23×4	Poly+T+L+C+I	72.1	61.8	66.5
11h	Strong	651	Poly	73.9	62.1	67.5
	Equal Time	560+35+49+92+693	Poly+T+L+C+I	73.7	62.8	67.8
	Equal Number	560+66×4	Poly+T+L+C+I	73.8	62.5	67.7
12h	Strong	710	Poly	79.5	59.7	68.2
	Equal Time	560+58+81+152+1143	Poly+T+L+C+I	**80.3**	**65.3**	**72.1**
	Equal Number	560+108×4	Poly+T+L+C+I	78.7	63.7	70.4

TABLE XIV

THE EFFECT OF COMBINATION OF TWO WEAK LABELS ON ICDAR-ART. "P", "R", AND "F" REPRESENT THE PRECISION, RECALL AND F-MEASURE, RESPECTIVELY. IN THE "LABELS" COLUMN, THE "I", "C", "L", "T" AND "POLY" REPRESENT THE IMAGE-LEVEL TAGS, COARSE BOUNDING BOXES, LOOSE BOUNDING BOXES, TIGHT BOUNDING BOXES AND POLYGONS, RESPECTIVELY

Combination	Image amount	P	R	F
Poly+T+L	560+116+162	73.5	64.1	68.5
Poly+T+C	560+116+304	73.6	62.6	67.7
Poly+T+I	560+116+2285	75.3	64.1	69.2
Poly+L+C	560+162+304	72.8	63.3	67.7
Poly+L+I	560+162+2285	**76.2**	63.7	**69.4**
Poly+C+I	560+304+2285	73.6	**64.3**	68.6

Finally, to verify the effect of combination of two different weak labels, we conduct experiment by assigning the budget (12h) in this way: in addition to 560 strongly labeled images, the remaining time is equally divided into any two weak supervision forms. The results in Table XIV show that "L+I" (Loose BBoxes + Image-Level Tags) achieves the best performance, and "T+I" (Tight BBoxes + Image-Level Tags) also achieves good performance. This shows that when the performance of the basic model is low, a large amount of increased data is very helpful to the performance.

V. CONCLUSION

In this paper, we propose an EM-based framework for mixed-supervised scene text detection, so as to take advantage of various forms of weak annotations in addition to polygon-level strong annotation. The proposed framework consists of a contour-based arbitrarily-shaped text detector and an EM-based learning strategy. Extensive experiments have been conducted to demonstrate the effectiveness of the proposed framework. Using 10% strongly annotated data, our mixed-supervised model can match the performance of the model trained with 100% strongly annotated data. Meanwhile, when training the contour-based detector with 100% strongly annotated data, our method achieves state-of-the-art performance. In the future we will consider other weak annotation methods, and mixed-supervised learning for end-to-end scene text spotting.

REFERENCES

[1] 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.

[2] W. He, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Deep direct regression for multi-oriented scene text detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 745–753.

[3] X. Zhou *et al.*, "EAST: An efficient and accurate scene text detector," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5551–5560.

[4] W. Feng, W. He, F. Yin, X.-Y. Zhang, and C.-L. Liu, "TextDragon: An end-to-end framework for arbitrary shaped text spotting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9076–9085.

[5] M. Liao, Z. Wan, C. Yao, K. Chen, and X. Bai, "Real-time scene text detection with differentiable binarization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11474–11481.

[6] W. Wang *et al.*, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9336–9345.

[7] 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.

[8] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 532–548, Feb. 2021.

[9] L. Yuliang, J. Lianwen, Z. Shuaitao, and Z. Sheng, "Detecting curve text in the wild: New dataset and new solution," 2017, *arXiv:1712.02170*.

[10] C. K. Chng *et al.*, "ICDAR2019 robust reading challenge on arbitrary-shaped text—RRC-ArT," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1571–1576.

[11] W. Wu, J. Xing, C. Yang, Y. Wang, and H. Zhou, "Texts as lines: Text detection with weak supervision," *Math. Problems Eng.*, vol. 2020, pp. 1–12, Jun. 2020.

[12] J. Zhang, C. Du, Z. Feng, Y. Wang, and C. Wang, "A text localization method based on weak supervision," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 775–780.

[13] J. Liu, Q. Zhong, Y. Yuan, H. Su, and B. Du, "SemiText: Scene text detection with semi-supervised learning," *Neurocomputing*, vol. 407, pp. 343–353, Sep. 2020.

[14] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell, "On learning to localize objects with minimal supervision," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1611–1619.

[15] C. Wang, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 431–445.

[16] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1637–1645.

[17] D. Pathak, E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional multi-class multiple instance learning," 2014, *arXiv:1412.7144*.

[18] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1713–1721.

[19] C. K. Ch'ng and C. S. Chan, "Total-text: A comprehensive dataset for scene text detection and recognition," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, Nov. 2017, pp. 935–942.

[20] 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.

[21] Y. Sun, J. Liu, W. Liu, J. Han, E. Ding, and J. Liu, "Chinese street view text: Large-scale Chinese text reading with partially supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9086–9095.

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

[23] 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.

[24] 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, 2016.

[25] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 161–184, 2021.

[26] 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.

[27] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. Asian Conf. Comput. Vis.*, 2011, pp. 770–783.

[28] 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.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[30] 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. 1962–1969.

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

[32] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "DenseBox: Unifying landmark localization with end to end object detection," 2015, *arXiv:1509.04874*.

[33] 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. 2550–2558.

[34] 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.*, 2018, pp. 20–36.

[35] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9365–9374.

[36] Y. Zhu and J. Du, "TextMountain: Accurate scene text detection via instance segmentation," *Pattern Recognit.*, vol. 110, 2021, Art. no. 107336.

[37] H. Wang *et al.*, "All you need is boundary: Toward arbitrary-shaped text spotting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12160–12167.

[38] Y. Li, J. Zhang, K. Huang, and J. Zhang, "Mixed supervised object detection with robust objectness transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 3, pp. 639–653, Mar. 2019.

[39] D. Wang *et al.*, "Mixed-supervised dual-network for medical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, 2019, pp. 192–200.

[40] S. Tian, S. Lu, and C. Li, "WeText: Scene text detection under weak supervision," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1492–1500.

[41] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[42] S. Peng, W. Jiang, H. Pi, X. Li, H. Bao, and X. Zhou, "Deep snake for real-time instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8533–8542.

[43] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," 2019, *arXiv:1904.07850*.

[44] X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 850–859.

[45] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Weakly supervised object recognition with convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1545–5963.

[46] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Untangling local and global deformations in deep convolutional networks for image classification and sliding window detection," 2014, *arXiv:1412.0296*.

[47] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2403–2412.

[48] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.

[49] Z. Tian *et al.*, "Learning shape-aware embedding for scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4234–4243.

[50] P. Wang *et al.*, "A single-shot arbitrarily-shaped text detector based on context attended multi-task learning," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 1277–1285.

[51] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai, "TextField: Learning a deep direction field for irregular scene text detection," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5566–5579, Nov. 2019.

[52] W. Wang *et al.*, "Efficient and accurate arbitrary-shaped text detection with pixel aggregation network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8440–8449.

[53] Y. Zhou, H. Xie, S. Fang, Y. Li, and Y. Zhang, "CRNet: A center-aware representation for detecting text of arbitrary shapes," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 2571–2580.

[54] Z. Liu, G. Lin, S. Yang, F. Liu, W. Lin, and W. L. Goh, "Towards robust curve text detection with conditional spatial expansion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7269–7278.

[55] Y. Wang, H. Xie, Z.-J. Zha, M. Xing, Z. Fu, and Y. Zhang, "ContourNet: Taking a further step toward accurate arbitrary-shaped scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11753–11762.

[56] Y. Liu, L. Jin, and C. Fang, "Arbitrarily shaped scene text detection with a mask tightness text detector," *IEEE Trans. Image Process.*, vol. 29, pp. 2918–2930, 2019.
[57] S. Xiao, L. Peng, R. Yan, K. An, G. Yao, and J. Min, "Sequential deformation for accurate scene text detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 108–124.
[58] Y. Liu, L. Jin, S. Zhang, C. Luo, and S. Zhang, "Curved scene text detection via transverse and longitudinal sequence connection," *Pattern Recognit.*, vol. 90, pp. 337–345, Jun. 2019.
[59] X. Wang, Y. Jiang, Z. Luo, C.-L. Liu, H. Choi, and S. Kim, "Arbitrary shape scene text detection with adaptive text region representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6449–6458.
[60] F. Wang, Y. Chen, F. Wu, and X. Li, "TextRay: Contour-based geometric modeling for arbitrary-shaped scene text detection," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 111–119.
[61] E. Xie, Y. Zang, S. Shao, G. Yu, C. Yao, and G. Li, "Scene text detection with supervised pyramid context network," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 9038–9045.
[62] C. Ma, L. Sun, Z. Zhong, and Q. Huo, "ReLaText: Exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks," *Pattern Recognit.*, vol. 111, Mar. 2021, Art. no. 107684.
[63] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting scene text via instance segmentation," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.
[64] 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.
[65] M. Liao, G. Pang, J. Huang, T. Hassner, and X. Bai, "Mask textspotter v3: Segmentation proposal network for robust scene text spotting," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 706–722.
[66] J. Ma *et al.*, "Arbitrary-oriented scene text detection via rotation proposals," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 3111–3122, Nov. 2018.
[67] 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.
[68] P. Dai, S. Zhang, H. Zhang, and X. Cao, "Progressive contour regression for arbitrary-shape scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7393–7402.
[69] Y. Liu, L. Jin, Z. Xie, C. Luo, S. Zhang, and L. Xie, "Tightness-aware evaluation protocol for scene text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9612–9620.
[70] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.

Mengbiao Zhao received the B.S. degree in automation from Nankai University, Tianjin, China, in 2017. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with the Institute of Automation, Chinese Academy of Sciences, Beijing. His research interests include computer vision, and scene text detection and recognition.

Wei Feng received the B.S. degree in automation from Tianjin University, Tianjin, China, in 2016, and the Ph.D. degree in pattern recognition and intelligent systems from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2021. His research interests include computer vision, and scene text detection and recognition.

Fei Yin received the B.S. degree in computer science from the Xidian University of Posts and Telecommunications, Xi'an, China, in 1999, the M.E. degree in pattern recognition and intelligent systems from the Huazhong University of Science and Technology, Wuhan, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He has published over 100 papers in international journals and conferences. His research interests include pattern recognition, document image analysis, and handwriting recognition.

Xu-Yao Zhang (Senior Member, IEEE) received the B.S. degree in computational mathematics from Wuhan University, Wuhan, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013. He was a Visiting Researcher with the Centre for Pattern Recognition and Machine Intelligence (CENPARMI), Concordia University, in 2012. From March 2015 to March 2016, he was a Visiting Scholar with the Montreal Institute for Learning Algorithms (MILA), University of Montreal. He is currently an Associate Professor with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. His research interests include pattern recognition, machine learning, and handwriting recognition.

Cheng-Lin Liu (Fellow, IEEE) received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, in 1989, the M.E. degree in electronic engineering from the Beijing University of Technology, Beijing, China, in 1992, and the Ph.D. degree in pattern recognition and intelligent control from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1995. He was a Postdoctoral Fellow at the Korea Advanced Institute of Science and Technology (KAIST) and later at the Tokyo University of Agriculture and Technology from March 1996 to March 1999. From 1999 to 2004, he was a Research Staff Member and later a Senior Researcher at the Central Research Laboratory, Hitachi Ltd., Tokyo, Japan. Since 2005, he has been a Professor at the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, and is currently the Director of the NLPR. He has published over 300 technical papers in prestigious international journals and conferences. His research interests include pattern recognition, machine learning, and the applications to character recognition and document analysis. He is a fellow of the IAPR, the CAA, and the CAAI. He is an Associate Editor-in-Chief of *Pattern Recognition* journal and *Acta Automatica Sinica* and is on the editorial board of several international and domestic journals.