You Can even Annotate Text with Voice: Transcription-only-Supervised Text Spotting

Jingqun Tang*
Ant Group
Hangzhou, Zhejiang, China
jingquntang@163.com

Su Qiao*
Zhejiang Gongshang University
Hangzhou, Zhejiang, China
qiaosu98@outlook.com

Benlei Cui*
Alibaba Group
Hangzhou, Zhejiang, China
a457435687@126.com

Yuhang Ma†
University College London
London, United Kingdom
yuhang_ma0307@163.com

Sheng Zhang
Zhejiang University
Hangzhou, Zhejiang, China
zsheng@zju.edu.cn

Dimitrios Kanoulas
University College London
London, United Kingdom
d.kanoulas@ucl.ac.uk

ABSTRACT

End-to-end scene text spotting has recently gained great attention in the research community. The majority of existing methods rely heavily on the location annotations of text instances (e.g., word-level boxes, word-level masks, and char-level boxes). We demonstrate that scene text spotting can be accomplished solely via text transcription, significantly reducing the need for costly location annotations. We propose a query-based paradigm to learn implicit location features via the interaction of text queries and image embeddings. These features are then made explicit during the text recognition stage via an attention activation map. Due to the difficulty of training the weakly-supervised model from scratch, we address the issue of model convergence via a circular curriculum learning strategy. Additionally, we propose a coarse-to-fine cross-attention localization mechanism for more precisely locating text instances. Notably, we provide a solution for text spotting via audio annotation, which further reduces the time required for annotation. Moreover, it establishes a link between audio, text, and image modalities in scene text spotting. Using only transcription annotations as supervision on both real and synthetic data, we achieve competitive results on several popular scene text benchmarks. The proposed method offers a reasonable trade-off between model accuracy and annotation time, allowing simplification of large-scale text spotting applications.

CCS CONCEPTS

• **Applied computing** → **Optical character recognition**; • **Computing methodologies** → *Scene understanding*; • **Human-centered computing** → *Human computer interaction (HCI).*

KEYWORDS

Weakly-Supervised Text Spotting, Audio Annotation, Coarse-to-fine Cross Attention

ACM Reference Format:
Jingqun Tang[1], Su Qiao[1], Benlei Cui[1], Yuhang Ma[2], Sheng Zhang, and Dimitrios Kanoulas. 2022. You Can even Annotate Text with Voice: Transcription-only-Supervised Text Spotting. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), Oct. 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547787

(a) Rectangle. (b) Quadrilateral. (c) Polygon.
(d) Character. (e) Single-Point. (f) Trasncription-Only.

Figure 1: Different annotation styles for text spotting. The blue point and lines are the position annotation for the text. The transcription annotation "Hills" is displayed in the top left corner of each image.

1 INTRODUCTION

For a long time, scene text spotting has been an active research field due to its broad practical applications, e.g., text translation in images, text information extraction from videos and images for multi-modal content understanding, and subtitle recognition in TV shows and music videos. The focus of research in the Optical Character Recognition (OCR) community has recently moved on from horizontal and multi-oriented text to arbitrary-shaped text, as well as from annotation formats based on horizontal rectangles and quadrilaterals to polygons (see Fig. 1).

Classical arbitrary-shaped text spotting approaches [22, 26, 52] in the deep learning era leap forward toward a unified end-to-end architecture by sharing a convolutional feature backbone and employing a feature cropping mechanism to extract the relevant area of interest for the recognition branch. These architectures have

*Equal Contribution
†Corresponding Author

achieved noticeable improvements in scene text spotting, particularly for arbitrarily-shaped text. Apart from the mutual optimization of the backbone, text detection and recognition are distinct tasks that require transcription annotations for recognition, as well as polygon or bounding box annotations for detection, as shown in Fig. 1(a-d). Relying heavily on the text location as the supervision information to train the network is costly and consumes a large amount of annotation manpower and resources. More sophisticated methods have recently been proposed to directly locate and classify characters in text [2, 32, 49]. Although, those require additional character-level annotations and thus higher annotation costs, as shown in Fig. 1(c). Time-consuming character-level or word-level text annotation leads to difficulties in the study of super large-scale text spotting on real data. Since the location annotation of text is expensive, some work started investigating how to alleviate the reliance on text annotation. An example is SPTS [31]'s proposal for single-point annotation, which uses a single point to represent the text position. It is a promising and leading attempt, but there is still room for improvement. As illustrated in Fig. 2, a large proportion of the annotation time is spent on labelling the detection-related ground truth. The transcription annotation alone requires less than half the time compared to the commonly used polygon annotation style. Compared to the single-point annotation style, the transcription-only annotation style reduces annotation time by 25%. The audio annotation style requires minimal time, i.e., 55% less than the single-point annotation.

Transformers have shown great potential for many vision tasks [4, 23, 24, 38, 40, 47]. In the field of image captioning, it is possible to automatically learn which caption corresponds to which portion of an image, only with the supervision of a text description. Inspired by the rapid development of weakly-supervised image captioning [6, 9], we figure out that it is sufficient to identify text instances and approximate their location by visual attention without the supervision of text position information. Here, we rethink scene text spotting as an image captioning problem, i.e., reading the textual content of a rough text region of interest directly, without text location supervision. Specifically, our **T**ranscription-**o**nly-**S**upervised Text **S**potter (TOSS) approach consists of a backbone for feature extraction, an interaction module between text queries and image features for learning implicit features of text location, and a coarse-to-fine cross attention localization mechanism for improving text localization accuracy. Additionally, we propose an audio-based method for annotating scene text that increases annotation speed and does not require the use of hands, enabling people with hand disabilities to work on scene text annotation. It can be achieved by simply converting audio annotation to text annotation using an Automatic Speech Recognition (ASR) model and then applying it in conjunction with our proposed TOSS model.

The main contributions of this work are summarized as follows:

- We introduce an effective paradigm based on the interaction of text queries and image embeddings for scene text spotting, requiring only text transcription annotations to be trained from scratch on both synthetic and real data. To our knowledge, it is the first successful attempt without any location information as supervision.
- To train such a weakly-supervised model, we propose a circular curriculum learning strategy for fast convergence.
- We propose a coarse-to-fine cross-attention localization mechanism to obtain approximate locations of text instances in the absence of text location annotation.
- An audio annotation solution for text spotting is offered, which reduces annotation time by 40% with only about a 1.0% loss of accuracy compared to transcription annotation.

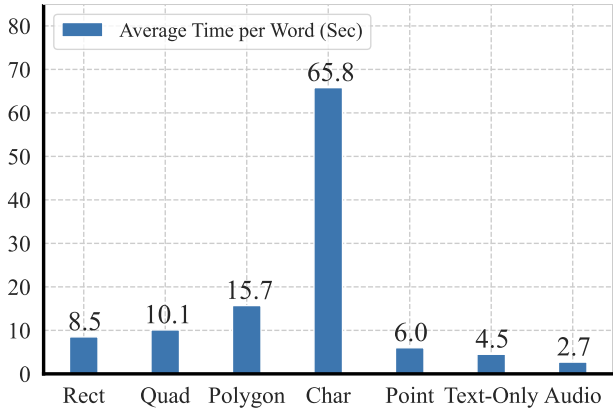

Figure 2: The effect of annotation styles on labelling time. The X-axis represents the different annotation styles, while the Y-axis represents the average annotation time they consume. The annotation time is experimentally derived from the average time spent on ICDAR2013 dataset by ten annotators. Audio annotation styles consume the least time.

2 RELATED WORK

2.1 Fully-Supervised Text Spotting

Many early studies of scene text spotting followed the mainstream generic object detection approaches using a two-stage framework. They build a detector to locate text and then extract the region of interest (RoI) features for text recognition and location refinement. For example, Li et al. [20] proposed an end-to-end text spotting framework that includes a CNN-based detector and an RNN-based recognition branch. The method applies RoI pooling to crop text regions and then recognizes them by a sequence-to-sequence head. However, this method is only capable of horizontal text spotting. To accommodate multi-oriented instances, He et al. [13] proposed an extended RoIAlign method to extract quadrilateral region features of arbitrary orientation, while Sun et al. [39] introduced a perspective RoI transform layer, which can align quadrangle proposals into small feature maps. Liu et al. [25] proposed the RoIRotate module to obtain axis-aligned feature maps, which works well for straight text but fails when dealing with curved text.

Recently, methods that can handle curved text have attracted much academic attention. Liao et al. [22] proposed Mask TextSpotter, which performs character-level semantic segmentation for text recognition. The method can detect text of arbitrary shapes. However, it requires character location for training, which leads to a high annotation cost. Another option, in addition to methods based on

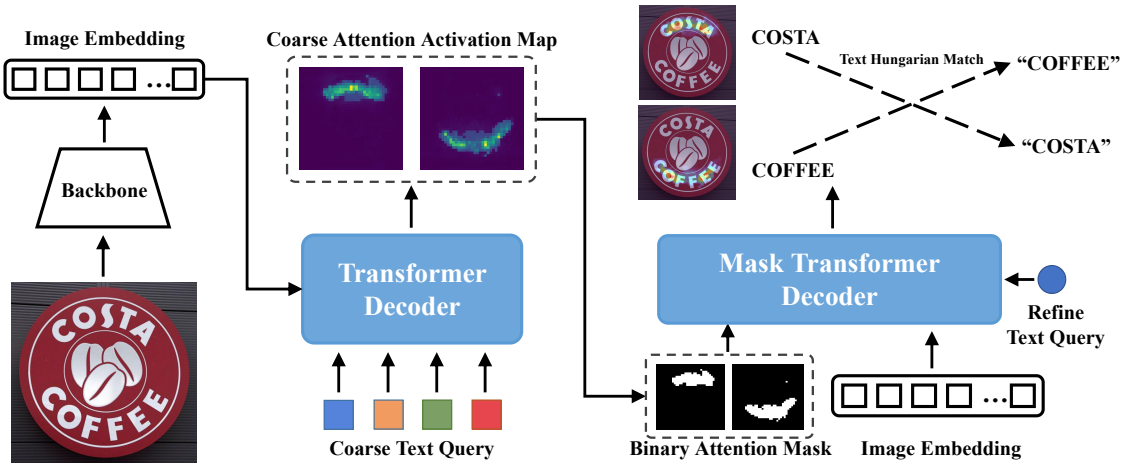

Figure 3: Overview of the proposed TOSS model architecture. The visual and contextual embeddings are first extracted by a backbone network. The embeddings are then decoded to focus on the relevant text regions using a query-based cross attention module. Next, the refine-stage text query, the mask generated by the attention activation map, and image embeddings are decoded together to obtain a refined text position.

segmentation networks, is to parameterize the text polygons. One of the representative works, ABCNet [26], fits arbitrarily-shaped text with a parameterized Bezier curve. However, in order to generate annotations, not only the polygons but also the distinction between the upper and lower edges of the text need to be annotated.

High annotation cost limits the prospect of scene text spotting. Our proposed TOSS adaptively learns text representations with the help of an attention mechanism and can roughly locate text with only transcription annotation, which significantly reduces the annotation cost.

2.2 Weakly-Supervised Text Spotting

Currently, weakly-supervised text spotting has not been extensively studied, with most research focusing on text detection tasks. Wu et al. [48] simplified text polygons into line segments and proposed a text detection method based on a fully supervised pre-trained semantic segmentation network. Tian et al. [41] used a pre-trained character detector to generate annotations for the training of the final model. Qin et al. [35] proposed a weakly-supervised curved text detection framework, which uses a large number of pseudo mask annotations generated by a pre-trained text detector. The aforementioned methods reduce the cost of annotation for text detection tasks to a certain extent, but still rely on a pre-trained model obtained by fully supervised training. In contrast to this, Rong et al. [36] used a class activation map to approximate text localization. The method only needs to annotate whether the image contains text or not, but it does not achieve satisfactory results.

In terms of weakly-supervised text spotting, there are few works in the published literature. However, some intriguing work can be found on arXiv. For instance, SPTS [31] demonstrated the high data annotation cost of fully-supervised methods and proposed a single point representation of text location. Xue et al. [50] proposed a weakly supervised pre-training approach for text spotting. They employ a contrastive learning-based approach to assist image encoders in perceiving text, but lack the ability to perform end-to-end text spotting. Kittenplon et al. [17] proposed an end-to-end method for weakly-supervised text spotting based on the modified DETR [5]. They train the model in a fully supervised manner on synthetic data and then fine-tune it in a weakly supervised manner on real data (i.e., not a completely weakly-supervised method). Thus, we propose a fully transcription-based supervised approach, both on synthetic data and real data.

3 METHODOLOGY

The overall architecture consists of a hybrid encoder-decoder, followed by a recognition branch and a classification branch. There are two critical components in this framework, which we will describe in detail below: a query-based attention module for generating text embeddings; and a coarse-to-fine cross-attention localization module based on the query-based attention module for improving the accuracy of text localization. In the training phase, the recognition branch generates text sequences, and a text-based Hungarian Matching algorithm is employed to match the prediction results with the ground truth in order to minimize the overall loss. Subsequently, the "text" or "non-text" label is assigned to each query according to the matching results. The entire training process follows a circular curriculum learning strategy. Lastly, we present how to use audio annotation to supervise text spotting.

3.1 Query-based Cross Attention

The Query-based Cross Attention module follows the standard transformer decoder block [42], consisting of stacked multiple layers of self-attention and cross-attention modules. Starting from the randomly initialized text queries $Q \in \mathbb{R}^{N_q \times d}$, the self-attention module generates N_q query-based embeddings h of size d, where N_q and d denote the number of queries and the representation dimensionality, respectively. Next, the cross attention module decodes these embeddings in parallel. Specifically, the input of this module

consists of h and the image embeddings $f \in \mathbb{R}^{L\times d}$ generated by the backbone, where L denotes the length of the flattened feature map. Using the cross attention mechanism between the h and f, the model can extract the corresponding local features of each text instance from the global image embedding. These local features are then fed into a classification branch and a recognition branch for classification and recognition, respectively. During model optimization, the text queries gradually learn to perceive the area of interest, i.e., where the text is located.

3.2 Coarse-to-Fine Cross Attention Localization Mechanism

The Coarse-to-Fine Cross Attention Localization Mechanism consists of a two-stage decoder that achieves coarse-to-fine text localization. The first stage utilizes the query-based cross attention module mentioned in Sec. 3.1 for text localization and performs filtering of irrelevant features. The second stage utilizes a parameter-sharing text query to filter irrelevant text areas further and fine-tune the text regions. Specifically, by using the query-based cross attention module, we can compute the attention activation map as follows:

$$AAM = softmax(\frac{W_q(Q+h)W_k(f)^T}{\sqrt{d}}), \tag{1}$$

where $Q \in \mathbb{R}^{N_q\times d}$ denotes the learnable coarse text queries, $h \in \mathbb{R}^{N_q\times d}$ denotes the query-based embeddings generated by the previous module (initialized from 0) and $f \in \mathbb{R}^{L\times d}$ denotes the image embeddings generated by the backbone. Equation 1 calculates the correlation between the coarse text queries and the image embeddings. By analyzing the attention activation map, we can acquire where the text query is capturing the feature and thus the text location, as illustrated in Fig. 3. According to the attention activation map, irrelevant tokens are filtered out of image embeddings for each query before proceeding to the second stage. By passing the relevant tokens through the cross-attention mechanism once, a rough text region can be formed.

Next, the filtered tokens are fed into the mask transformer decoder. This module is trained to learn a refined text query $Q_r \in \mathbb{R}^{1\times d}$ to apply query-based cross attention to obtain a more precise region of interest:

$$RefinedMap = \left(M_1, M_2 \ldots, M_{N_q}\right), \tag{2}$$

where M_i denotes the refined attention map of the ith coarse text query, as shown in Equation 3,

$$M_i = softmax(\frac{W_q(Q_r+h_i)W_k(\hat{f}_i)^T}{\sqrt{d}}), \tag{3}$$

where $\hat{f}_i$ denotes the filtered tokens generated by the ith coarse text query. In contrast to the first stage text queries, the second stage text query does not require the extraction of local text-related features from global features. Instead, it is used to refine the text region by further filtering text-independent information.

3.3 Text-based Hungarian Matching Loss

The Text-based Hungarian Matching loss implicitly supervises the text localization by matching predicted text to the corresponding ground truth, enabling transcription-only supervision. Specifically, the Text-based Hungarian Matching loss consists of two components: the loss of the text recognition branch, which is based on the Hungarian algorithm, and the loss of the classification branch, which is based on the text matching results.

The loss for text recognition starts from the text embeddings $x \in \mathbb{R}^{N_q\times d}$ generated by the decoder block, where x is decoded by the recognition branch as N_q text sequences. The Hungarian algorithm [18] is employed to find a one-to-one matching function σ between the text annotation and predicted text sequences:

$$\hat{\sigma} = \underset{\xi\in N_{gt}}{argmin} \sum_{i=1}^{N_q} H(t_i, \hat{t}_{\sigma(i)}), \tag{4}$$

where H denotes the criteria used to perform the matching, $\hat{t}$ denotes the text annotation, t denotes the predicted text sequences, N_q denotes the number of predictions or object queries, and N_{gt} denotes the number of the text annotations. The text recognition $L_{text}(t, \hat{t})$ loss is formulated based on the matching function $\hat{\sigma}$:

$$L_{text}(t, \hat{t}) = \sum_{i=1}^{N} L(t_i, \hat{t}_{\sigma(i)}). \tag{5}$$

The loss of the classification branch is calculated based on the results of text matching. As mentioned before, the matching process is done by the Hungarian algorithm. If there is matching ground truth for the predicted text, we assign a "text" label to the corresponding prediction, otherwise, a "non-text" label is assigned. Then, the loss of the classification branch is calculated from the assigned labels:

$$\mathrm{L}_{cls}(p_i, \hat{t}_{\sigma(i)}) = \begin{cases} -\alpha \log p_i & \text{if } \hat{t}_{\sigma(i)} \neq \emptyset \\ -(1-\alpha)\log(1-p_i) & \text{otherwise,} \end{cases} \tag{6}$$

where p_i denotes the confidence scores generated by the classification branch, and α denotes class weights for addressing class imbalance.

3.4 Circular Curriculum Learning Strategy

For the tasks with transcription-only supervision, training from scratch with complex data could cause model convergence issues. We design a circular curriculum learning paradigm in which the model is trained in three stages: easy, semi-hard, and hard, with gradually increasing data complexity, followed by multiple rounds of circular training.

During the easy stage of training, the model is fed with simple data to help it learn simple scenes quickly. Easy samples are those that contain legible text in a straightforward context, as shown in Fig. 4 (Easy). During the semi-hard stage of training, as the model can already perceive the approximate text area and identify simple text, the model can progress to more difficult samples. Semi-hard samples are those synthesized on complex backgrounds, as shown in Fig. 4 (Semi-hard). Due to the limited number of fixed backgrounds, some learning difficulties are alleviated. During the

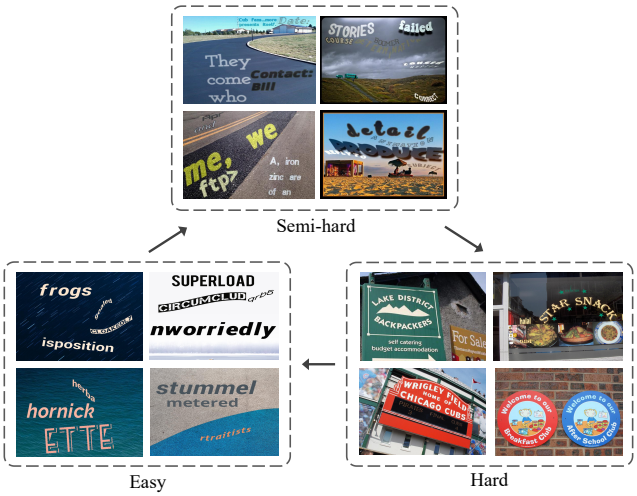

Figure 4: Illustration of circular curriculum learning strategy. "Easy", "Semi-hard", and "Hard" denote the training datasets of different difficulty levels.

hard stage of training, we use scene text in the wild for training, as shown in Fig. 4 (Hard). The model gradually acquires the ability to transfer knowledge from the first two stages to real-world scenes, completing transcription-only-supervised text spotting.

In addition, we incorporate a circular mechanism that is distinct from the original curriculum learning strategy [21]. Many studies have demonstrated that deep learning algorithms are susceptible to catastrophic forgetting [11, 19, 37]: when the network acquires new knowledge, it may forget what it previously learned. Therefore, we extend the curriculum learning framework and propose a circular curriculum learning strategy. As illustrated in Fig. 4, the model is trained cyclically on data of varying difficulty, which accelerates convergence and improves generalizability.

3.5 Audio Annotation

Apart from text annotation, we provide a solution to annotate text with voice. Audio annotation is more convenient than text annotation, which frees up one's hands and accelerates the annotation process. In practice, as shown in Fig. 5, we use an ASR model [1, 3] to convert annotations from audio modal to text modal. Two forms of annotation are available: character-by-character and word-by-word. Both of these forms are accomplished through a long pause or a specific word to create a division of words. Clearly, the former is more accurate, as individual characters' pronunciations are more easily recognized. While annotating word-by-word is faster and easier, recognizing the pronunciations of words is more likely to be influenced by the annotator's pronunciation, making recognition more difficult for the ASR model.

3.6 Inference

We use the intermediate results of cross attention to generate text location. The text content is decoded by the recognition branch, and the corresponding confidence score for each text is obtained by the classification branch. Specifically, the processes of inference are as follows: first, the hybrid-backbone generates a lower-resolution feature map of the initial image. The feature map is flattened as a sequence and then fed into the transformer decoder. By performing the query-based cross attention, we can obtain a coarse text region for each query. Based on those regions, the useless tokens are filtered, and the remaining ones are fed into the mask transformer decoder for further refinement. Finally, we locate the text using the method described in Section 3.2. Simultaneously, this module generates embeddings for each text region, which are then fed into the recognition and classification branches for the final prediction.

4 EXPERIMENTS

4.1 Datasets

Extensive experiments are conducted on several popular benchmarks, i.e., ICDAR-2013 [16], ICDAR-2015 [15], Total-Text [7], and SCUT-CTW1500 [51], to demonstrate the effectiveness of our method. Following ABCNet [26], SynthText 150k [26], COCO-Text [43], and ICDAR-MLT 2019 [30] are invovled in training. A detailed description of the datasets can be found in the supplementary material.

4.2 Evaluation Protocol

Polygon Metric. Due to the reliance on text transcription for training and the absence of location information, our model can only provide approximate positions. For polygon-based evaluation, we follow the same evaluation metric as Mango [32], using an IoU threshold of 0.1 for a successful match. Many previous algorithms under this metric would exhibit lower precision.

Single-point Metric. SPTS [31] encounters a similar issue in which it is unable to generate precise text bounding boxes and proposes an evaluation metric based on single-point positions, which we adopt for single-point evaluation. In practice, our method produces the center of gravity of each text instance. The formula for the center of gravity is given by the equation:

$$\hat{X} = \frac{\sum_{i \in \mathbb{A}} w_i X_i}{\sum_{i \in \mathbb{A}} w_i}, \tag{7}$$

where $\mathbb{A}$ denotes the positive area of the mask, w_i denotes the confidence score of each token, and $\hat{X}$ denotes the centre of gravity. The matching method is identical to that proposed in SPTS, and it contributes to accuracy only when the predicted text is identical to the text annotation.

4.3 Implementation Details

Network Architecture. The network described in this paper consists of a backbone, a two-stage decoder, a recognition branch, a classification branch, and two sets of text queries. Resnet50 [12], followed by two DCN [8] blocks, is used as the backbone. The query-based decoder section is divided into two stages, from coarse to fine. The first stage of the decoder consists of 4 layers of stacked 8-head transformers, while the second stage consists of 2 layers with 8 heads. The first stage contains 80 text queries, and the second stage shares one query. All the queries are randomly initialized and then optimized during training. The recognition branch consists of a 2-layer Bi-LSTM [14] and produces 512-dimensional features for text decoding. The classification branch employs an MLP layer

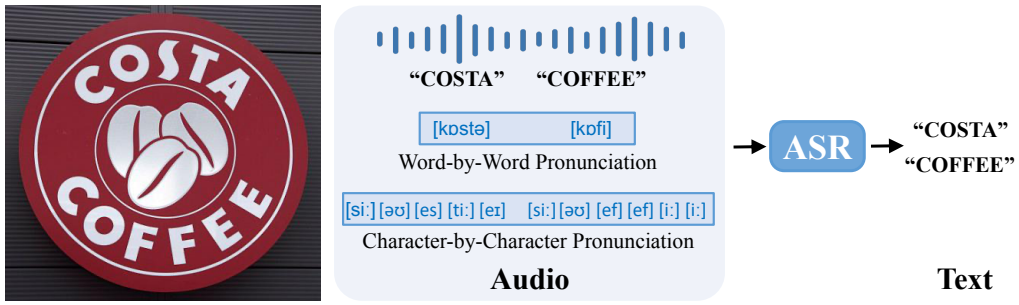

Figure 5: Illustration of the workflow for audio annotation. We provide two types of annotations for English, word-by-word pronunciation and character-by-character pronunciation, respectively. Audio annotation can only handle case-insensitive text spotting tasks.

Figure 6: Qualitative results. Images are selected from ICDAR 2013 (first col.), SCUT-CTW1500 (second col.), Total-Text (third col.), and ICDAR 2015 (fourth col.). The first row contains visualizations of single-point, while the second row contains visualizations of masks. As shown in the figure, our method is robust against various text types, including long text, large text, small text, curved text, perspective text, and fuzzy text.

directly for binary classification to distinguish whether it is text or not.

Training details. All experiments are implemented with 32x40 GB Tesla-A40 GPUs. Following ABCNet [26], the training data is collected from publicly available English datasets, including SynthText-150k [26], 15k images filtered from the original COCO-Text [43], and 7k MLT19 data [30]. First, the data is divided into three parts: easy, semi-hard, and hard. Easy represents 10k simple data, filtered from SynthText-150k, semi-hard represents the remaining 140k, and hard represents all the real data. The training procedure is then performed according to the strategy described in Sec. 3.3, which involves training ten epochs each on the easy, semi-hard, and hard datasets as a single cycle, for a total of ten cycles. In the training phase, the short side of the input images is randomly resized from 672 to 1280. Random contrast, brightness, and rotating are employed for data augmentation. We utilize AdamW [28] as the optimizer, with an initial learning rate of $1e^{-4}$ and a decaying learning rate of $1e^{-5}$ after the 8th cycle. The batch size on each GPU is set to 4, resulting in a total batch size of 128. The entire training process lasts 4 days.

Inference details. In the inference phase, the short side of the images is set to 768, 1280, 960, and 960 on ICDAR-2013, ICDAR-2015, Total-Text, and SCUT-CTW-1500, respectively, while keeping the long side shorter than 2048 pixels. The inference FPS is 5.06, 2.31, 6.30, and 7.90 on ICDAR-2013, ICDAR-2015, Total-Text, and SCUT-CTW-1500, respectively. Furthermore, in order to preserve as many positive text features as possible in the first stage of the coarse-to-fine cross-attention module, we set the threshold of the attention activation map to 0.1.

4.4 Ablation Study

Number of the coarse-to-fine cross attention localization mechanism. To explore the effect of the number of refinement stages in the coarse-to-fine cross attention localization mechanism on the model's accuracy, we conduct experiments to compare the number of refinement stages from 0 to 2. As illustrated in Tab. 1, omitting the refinement stage results in a significant loss of accuracy (65.1 vs 58.5). When the refinement stage is employed twice, the improvement in accuracy is not remarkable (65.1 vs 66.3). Considering the memory and speed of the model, we choose to employ the refinement stage once.

Ablation Study on Circular Curriculum Learning strategy. To demonstrate the effectiveness of the proposed Circular Curriculum Learning strategy (CCL), we compare three different training strategies for our TOSS model: the commonly used pretrain-finetune (PF), curriculum learning (CL), and ours, respectively. For PF, we simply pretrain on SynthText-150k and then finetune on Total-Text. For CL, we first train on simple data (10k filtered from SynthText-150k), followed by semi-hard data (the remaining 140k SynthText-150k data), and finally hard data (Total-Text), in an acyclic manner. For CCL, we follow the training process described in Sec. 4.2. As shown in Tab. 2, our proposed CCL strategy outperforms PF and CL in terms of accuracy while maintaining a significantly higher rate of convergence (3 times more than PF and 1.5 times more than CL).

4.5 Experiment Results on Scene Text Benchmarks

Horizontal text. Tab. 3 compares the performance of the proposed TOSS with state-of-the-art methods on ICDAR-2013. It's worth noting that our method is trained without using any location information, whereas SPTS uses single-point positions and the other methods use more expensive bounding boxes. In terms of the polygon-based evaluation metric, the weakly-supervised TOSS gives competitive results compared to the fully-supervised state-of-the-art methods, with a slight loss of accuracy. In comparison to SPTS, the proposed method achieves comparable results with all three lexicons, e.g., 82.2 *vs.* 82.9 with "Generic" lexicon.

Multi-oriented text. The quantitative results of ICDAR-2015 dataset are shown in Tab. 4. There is a performance gap between the proposed method and fully-supervised methods. Specifically, since almost all text instances are small, it is difficult to locate small text by text queries alone without location supervision. In addition, many text instances are ambiguous and arranged in more varied directions. All of these bring major challenges to weakly-supervised methods. Compared to SPTS, our method performs slightly better with "Strong" and "Weak" lexicons, which demonstrates the potential of our approach for small text.

Arbitrary-shaped text. Experiments are conducted on Total-Text and SCUT-CTW-1500 to demonstrate the robustness of our method in spotting arbitrary-shaped text. As shown in Tab. 5, the proposed method achieves comparable results to SPTS, demonstrating our method's effectiveness for curved text. Additionally, the gap between TOSS and fully-supervised methods is still visible, and the proposed method's limitation is its inability to locate text accurately. Similar conclusions can be drawn from Tab. 6 on SCUT-CTW1500.

In summary, our transcription-only-supervised approach(TOSS) offers an excellent trade-off between annotation time and recognition accuracy. It can produce a single point of text or a polygon of text as a result. On all four benchmarks, TOSS achieves comparable

Number of Coarse-to-Fine Attention	Total-Text End-to-End	
	None	Full
0	58.5	67.4
1	65.1	74.8
2	66.3	75.2

Table 1: End-to-end recognition results on Total-Text w.r.t number of times of coarse-to-fine cross attention localization mechanism. "None" denotes lexicon-free. "Full" denotes that we use all the words appearing in the test set. All the results are obtained under the single-point metric.

Training Strategy	Total-Text End-to-End		Convergence Epochs
	None	Full	
PF	60.3	68.5	300
CL	64.4	73.6	150
CCL	**65.1**	**74.8**	100

Table 2: End-to-end recognition results on Total-Text w.r.t. training strategy. "PF", "CL", and "CCL" denote pretrain-finetune, curriculum learning, and circular curriculum learning, respectively. All the results are obtained under the single-point metric.

Method	IC13 End-to-End		
	S	W	G
Fully-supervised methods			
FOTS [25]	88.8	87.1	80.8
TextNet [39]	89.8	88.9	83.0
Mask TextSpotter [29]	**93.3**	**91.3**	**88.2**
Boundary [44]	88.2	87.7	84.1
Text Perceptron [33]	91.4	90.7	85.8
Point-based methods			
SPTS (Single-Point) [31]	87.6	85.6	82.9
Transcription-only methods			
TOSS (Single-Point) (Ours)	86.4	85.1	82.2
TOSS (Polygon) (Ours)	77.7	76.8	73.3

Table 3: End-to-end recognition results on ICDAR 2013. "S", "W", and "G" denote recognition with "Strong", "Weak", and "Generic" lexicon, respectively. "Single Point" and "Polygon" denote the two metrics mentioned in Sec. 4.1.

Method	IC15 End-to-End		
	S	W	G
Fully-Supervised methods			
FOTS [25]	81.1	75.9	60.8
Mask TextSpotter [29]	83.0	77.7	73.5
CharNet [49]	83.1	**79.2**	69.1
TextDragon [10]	82.5	78.3	65.2
Mask TextSpotter v3 [22]	**83.3**	78.1	**74.2**
MANGO [32]	81.8	78.9	67.3
ABCNetV2 [27]	82.7	78.5	73.0
PAN++ [46]	82.7	78.2	69.2
Point-based methods			
SPTS (Single-Point) [31]	64.6	58.8	54.9
Transcription-only methods			
TOSS (Single-Point) (Ours)	65.9	59.6	52.4
TOSS (Polygon) (Ours)	60.2	54.5	47.1

Table 4: End-to-end recognition results on ICDAR 2015.

results to SPTS in the absence of single-point supervision of text. The proposed method even outperforms several fully-supervised algorithms with bounding boxes, indicating the superior text spotting performance of our method. On the polygon metric, the proposed method still falls short of the fully-supervised algorithm, highlighting the method's limitations in that it can only approximate text position. Numerous factors contribute to our method's exceptional performance when supervised exclusively by text: (1) Text queries can extract local features associated with text instances from the global image using a cross attention mechanism. Local features eliminate the interference of redundant backgrounds, ensuring text recognition performance. (2) The proposed method employs a text-based Hungarian Matching algorithm to supervise the position of text queries in an implicit manner, compensating for the absence of text position annotations. Further, the circular curriculum learning strategy enables the model to rapidly converge on such a difficult task. (3) The coarse-to-fine two-stage attention mechanism improves the accuracy of text location by acting as a refinement for coarse masks. Thus, we can obtain text masks, but they are still inferior to state-of-the-art fully-supervised methods.

Method	Total-Text End-to-End	
	None	Full
Fully-Supervised methods		
CharNet [49]	66.2	-
ABCNet [26]	64.2	75.7
PGNet [45]	63.1	-
Mask TextSpotter [29]	65.3	77.4
Qin et al [34]	67.8	-
Mask TextSpotter v3 [22]	71.2	78.4
MANGO [32]	**72.9**	**83.6**
PAN++ [46]	68.6	78.6
ABCNet v2 [27]	70.4	78.1
Point-based methods		
SPTS (Single-Point) [31]	67.9	74.1
Transcription-only methods		
TOSS (Single-Point) (Ours)	65.1	74.8
TOSS (Polygon) (Ours)	61.5	73.0

Table 5: End-to-end recognition results on Total-Text. "None" denotes lexicon-free. "Full" denotes that we use all the words appearing in the test set.

Method	SCUT-CTW1500 End-to-End	
	None	Full
Fully-Supervised methods		
TextDragon [10]	39.7	72.4
ABCNet [26]	45.2	74.1
MANGO [32]	**58.9**	**78.7**
ABCNet v2 [27]	57.5	77.2
Point-based methods		
SPTS (Single-Point) [31]	56.3	67.2
Transcription-only methods		
TOSS (Single-Point) (Ours)	54.2	65.3
TOSS (Polygon) (Ours)	51.4	61.7

Table 6: End-to-end recognition results on SCUT-CTW1500.

4.6 Experiments on Audio Annotation

To verify the feasibility of using audio to annotate scene text, we conduct experiments on Total-Text. First, since there is no ready-made audio annotation on Total-Text, we use a TTS (Text To Speech) model in DeepSpeech [1] to convert the original text annotation to audio annotation. Then, an ASR model [1] is used to convert audio to text according to Fig. 5. The converted text transcription annotations are utilized for training our TOSS model. Finally, the model trained using the audio-converted annotations is applied for inference to validate the performance. As shown in Tab. 7, text spotting with word-by-word audio annotation performs slightly worse on Total-Text (64.3 *vs.* 65.1), owing to a decrease in the annotation accuracy during the audio-to-text conversion process. The character-by-character annotation has no significant loss of accuracy but is slower than the word-by-word annotation. Overall, audio annotation achieves a favourable trade-off between annotation time and model accuracy, saving 40% of annotation time while maintaining model accuracy within a 1% loss.

Annotation Style	Total-Text End-to-End	
	None	Full
Text Transcription	65.1	74.8
Audio (Word)	64.3	73.9
Audio (Character)	64.9	74.5

Table 7: End-to-end recognition results w.r.t annotation styles on Total-Text. "Word" and "Character" denote word-by-word and character-by-character pronunciation, respectively.

5 CONCLUSION

In this paper, we present a scene text spotting framework, termed TOSS, that can be trained in a transcription-only supervised manner. By using the interaction between text queries and image embeddings to learn a joint representation of recognition and detection tasks, we can forego the expensive annotations required by other approaches and achieve a reasonable trade-off between model accuracy and annotation time. To enable training from scratch with weak supervision, a circular curriculum learning method is proposed that enables the model to converge rapidly. In addition, to mitigate the inaccuracy of weakly-supervised localization, we propose a coarse-to-fine cross-attention mechanism to locate text instances and attempt to obtain coarse text masks. Remarkably, we provide a solution for annotating scene text with voice, which requires minimal annotation time and builds a bridge between the three modes of speech, text, and image in text spotting. In comparison to fully-supervised approaches, our approach achieves competitive results on several benchmarks with minimal supervision. Without the need for costly text location annotations, we believe there is a good opportunity to build a much larger dataset that will advance the state of the art in text spotting. We hope this work will lead to new research directions in scene text spotting and provide new insights into what annotations are truly necessary for this task.

REFERENCES

[1] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, Jie Chen, Jingdong Chen, Zhijie Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Ke Ding, Niandong Du, Erich Elsen, Jesse Engel, Weiwei Fang, Linxi Fan, Christopher Fougner, Liang Gao, Caixia Gong, Awni Hannun, Tony X. Han, Lappi Vaino Johannes, Bing Jiang, Cai Ju, Billy Jun, Patrick LeGresley, Libby Lin, Junjie Liu, Yang Liu, Weigao Li, Xiangang Li, Dongpeng Ma, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Yiping Peng, Ryan Prenger, Sheng Qian, Zongfeng Quan, Jonathan Raiman, Vinay Rao, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Kavya Srinet, Anuroop Sriram, Haiyuan Tang, Liliang Tang, Chong Wang, Jidong Wang, Kaifu Wang, Yi Wang, Zhijian Wang, Zhiqian Wang, Shuang Wu, Likai Wei, Bo Xiao, Wen Xie, Yan Xie, Dani Yogatama, Bin Yuan, Jun Zhan, and Zhenyao Zhu. 2016. Deep speech 2: end-to-end speech recognition in English and mandarin. In *International Conference on Machine Learning*.

[2] Young Min Baek, Seung Shin, Jeonghun Baek, Sungrae Park, Junyeop Lee, Daehyun Nam, and Hwalsuk Lee. 2020. Character Region Attention for Text Spotting. In *European Conference on Computer Vision*.

[3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Neural Information Processing Systems*.

[4] Haoyu Cao, Jiefeng Ma, Antai Guo, Yiqing Hu, Hao Liu, Deqiang Jiang, Yinsong Liu, and Bo Ren. 2022. GMN: Generative Multi-modal Network for Practical Document Information Extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 3768–3778. https://aclanthology.org/2022.naacl-main.276

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*. Springer, 213–229.

[6] Nenglun Chen, Xingjia Pan, Runnan Chen, Lei Yang, Zhiwen Lin, Ren Yuqiang, Haolei Yuan, Xiaowei Guo, Feiyue Huang, and Wenping Wang. 2021. Distributed Attention for Grounded Image Captioning. In *ACM Multimedia*.

[7] Chee Kheng Ch'ng and Chee Seng Chan. 2017. Total-text: A comprehensive dataset for scene text detection and recognition. In *Proc. ICDAR*, Vol. 1. 935–942.

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *Proc. ICCV*. 764–773.

[9] Samyak Datta, Karan Sikka, Anirban Roy, Karuna Ahuja, Devi Parikh, and Ajay Divakaran. 2019. Align2Ground: Weakly Supervised Phrase Grounding Guided by Image-Caption Alignment. In *International Conference on Computer Vision*.

[10] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. 2019. TextDragon: An End-to-End Framework for Arbitrary Shaped Text Spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[11] Robert M French. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences* 3, 4 (1999), 128–135.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. CVPR*. 770–778.

[13] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. 2018. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5020–5029.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. 2015. ICDAR 2015 competition on robust reading. In *ICDAR*. 1156–1160.

[16] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. 2013. ICDAR 2013 robust reading competition. In *Proc. ICDAR*. 1484–1493.

[17] Yair Kittenplon, Inbal Lavi, Sharon Fogel, Yarin Bar, R Manmatha, and Pietro Perona. 2022. Towards Weakly-Supervised Text Spotting using a Multi-Task Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4604–4613.

[18] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval research logistics quarterly* 2, 1-2 (1955), 83–97.

[19] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Overcoming catastrophic forgetting by incremental moment matching. *Advances in neural information processing systems* 30 (2017).

[20] Hui Li, Peng Wang, and Chunhua Shen. 2017. Towards End-To-End Text Spotting With Convolutional Recurrent Neural Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[21] Hui Li, Peng Wang, and Chunhua Shen. 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. (2017), 5238–5246.

[22] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. 2020. Mask TextSpotter v3: Segmentation Proposal Network for Robust Scene Text Spotting. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 706–722.

[23] Hao Liu, Xinghua Jiang, Xin Li, Zhimin Bao, Deqiang Jiang, and Bo Ren. 2022. NomMer: Nominate Synergistic Context in Vision Transformer for Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12073–12082.

[24] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, Bo Ren, and Rongrong Ji. 2021. Show, read and reason: Table structure recognition with flexible context aggregator. In *Proceedings of the 29th ACM International Conference on Multimedia*. 1084–1092.

[25] X. Liu, L. Ding, Y. Shi, D. Chen, and J. Yan. 2018. FOTS: Fast Oriented Text Spotting with a Unified Network. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] Y. Liu, H. Chen, C. Shen, T. He, and L. Wang. 2020. ABCNet: Real-Time Scene Text Spotting With Adaptive Bezier-Curve Network. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[27] Y. Liu, C. Shen, L. Jin, T. He, P. Chen, C. Liu, and H. Chen. 2021. ABCNet v2: Adaptive Bezier-Curve Network for Real-time End-to-end Text Spotting. (2021).

[28] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*.

[29] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. 2018. Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

[30] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. 2019. ICDAR2019 robust reading challenge on multi-lingual scene text detection and recognition—RRC-MLT-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 1582–1587.

[31] Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Shenggao Zhu, Jing Li, Dahua Lin, Chunhua Shen, et al. 2021. SPTS: Single-Point Text Spotting. *arXiv preprint arXiv:2112.07917* (2021).

[32] Liang Qiao, Ying Chen, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. 2021. MANGO: A Mask Attention Guided One-Stage Scene Text Spotter. In *National Conference on Artificial Intelligence*.

[33] Liang Qiao, Sanli Tang, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. 2020. Text perceptron: Towards end-to-end arbitrary-shaped text spotting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11899–11907.

[34] S. Qin, A. BiSsAcco, M. Raptis, Y. Fujii, and Y. Xiao. 2019. Towards Unconstrained End-to-End Text Spotting. *IEEE* (2019).

[35] Xugong Qin, Yu Zhou, Dongbao Yang, and Weiping Wang. 2019. Curved text detection in natural scene images with semi-and weakly-supervised learning. In *Proc. ICDAR*. 559–564.

[36] Li Rong, En MengYi, Li JianQiang, and Zhang HaiBin. 2017. Weakly supervised text attention network for generating text proposals in scene images. In *Proc. ICDAR*, Vol. 1. 324–330.

[37] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. 2017. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*. 3400–3409.

[38] Luchuan Song, Bin Liu, Guojun Yin, Xiaoyi Dong, Yufei Zhang, and Jia-Xuan Bai. 2021. TACR-Net: Editing on Deep Video and Voice Portraits. In *Proceedings of the 29th ACM International Conference on Multimedia*. 478–486.

[39] Yipeng Sun, Chengquan Zhang, Zuming Huang, Jiaming Liu, Junyu Han, and Errui Ding. 2018. Textnet: Irregular text reading from images with an end-to-end trainable network. In *Asian Conference on Computer Vision*. Springer, 83–99.

[40] Jingqun Tang, Wenqing Zhang, Hongye Liu, MingKun Yang, Bo Jiang, Guanglong Hu, and Xiang Bai. 2022. Few Could Be Better Than All: Feature Sampling and Grouping for Scene Text Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4563–4572.

[41] Shangxuan Tian, Shijian Lu, and Chongshou Li. 2017. Wetext: Scene text detection under weak supervision. In *Proc. ICCV*. 1492–1500.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[43] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. 2016. COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. *arXiv: Computer Vision and Pattern Recognition* (2016).

[44] Hao Wang, Pu Lu, Hui Zhang, Mingkun Yang, Xiang Bai, Yongchao Xu, Mengchao He, Yongpan Wang, and Wenyu Liu. 2020. All You Need Is Boundary: Toward Arbitrary-Shaped Text Spotting. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 07 (Apr. 2020), 12160–12167. https://doi.org/10.1609/aaai.v34i07.6896

[45] Pengfei Wang, Chengquan Zhang, Fei Qi, Shanshan Liu, Xiaoqiang Zhang, Pengyuan Lyu, Junyu Han, Jingtuo Liu, Errui Ding, and Guangming Shi. 2021.

PGNet: Real-time Arbitrarily-Shaped Text Spotting with Point Gathering Network. *AAAI. AAAI* (2021), 2782–2790.

[46] Wenhai Wang, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Yang Zhibo, Tong Lu, and Chunhua Shen. 2021. PAN++: Towards Efficient and Accurate End-to-End Spotting of Arbitrarily-Shaped Text. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021), 1–1. https://doi.org/10.1109/TPAMI.2021.3077555

[47] Weijia Wu, Yuanqiang Cai, Debing Zhang, Sibo Wang, Zhuang Li, Jiahong Li, Yejun Tang, and Hong Zhou. 2021. A bilingual, OpenWorld video text dataset and end-to-end video text spotter with transformer. *arXiv preprint arXiv:2112.04888* (2021).

[48] Weijia Wu, Jici Xing, Cheng Yang, Yuxing Wang, and Hong Zhou. 2020. Texts as Lines: Text Detection with Weak Supervision. *Mathematical Problems in Engineering* 2020 (2020).

[49] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. 2019. Convolutional character networks. In *Proc. ICCV*. 9126–9136.

[50] Chuhui Xue, Yu Hao, Shijian Lu, Philip Torr, and Song Bai. 2022. Language Matters: A Weakly Supervised Pre-training Approach for Scene Text Detection and Spotting. *arXiv preprint arXiv:2203.03911* (2022).

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

[52] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, and Wayne Zhang. 2021. Fourier Contour Embedding for Arbitrary-Shaped Text Detection. In *Computer Vision and Pattern Recognition*.