Towards Weakly-Supervised Text Spotting using a Multi-Task Transformer

Yair Kittenplon Inbal Lavi Sharon Fogel Yarin Bar
R. Manmatha Pietro Perona

AWS AI Labs

{yairk, ilavi, shafog, yarinbar, manmatha, peronapp}@amazon.com

Abstract

Text spotting end-to-end methods have recently gained attention in the literature due to the benefits of jointly optimizing the text detection and recognition components. Existing methods usually have a distinct separation between the detection and recognition branches, requiring exact annotations for the two tasks. We introduce TextTranSpotter (TTS), a transformer-based approach for text spotting and the first text spotting framework which may be trained with both fully- and weakly-supervised settings. By learning a single latent representation per word detection, and using a novel loss function based on the Hungarian loss, our method alleviates the need for expensive localization annotations. Trained with only text transcription annotations on real data, our weakly-supervised method achieves competitive performance with previous state-of-the-art fully-supervised methods. When trained in a fully-supervised manner, TextTranSpotter shows state-of-the-art results on multiple benchmarks.

1. Introduction

Text spotting, *i.e.*, detecting and reading text in images, is a key capability for machines to operate in the real world. Applications include vehicle navigation in buildings and cities, indexing of image collections and video, automated handling of packages, and prosthetics for blind and visually impaired people. This challenge was recognized early in the computer vision literature [6, 19, 40] and is currently undergoing a deep learning revival [14, 20], with most researchers focusing on two issues: architectures and data.

Early systems [3, 13] use separate architectures for text detection and recognition, without sharing any component. More recent approaches take a leap forward towards a unified end-to-end architecture by sharing a convolutional feature backbone [5, 20] and employing a feature cropping mechanism to extract the relevant area of interest for the recognition head. Such architectures are still not ideal, since the recognition head is usually trained using the detection ground-truth and thus it is not optimized for the predictions of the detection head. Furthermore, the detection head is trained as a standard object detection model, without regard to the additional supervision given by the text transcription or to the downstream recognition task. Other than

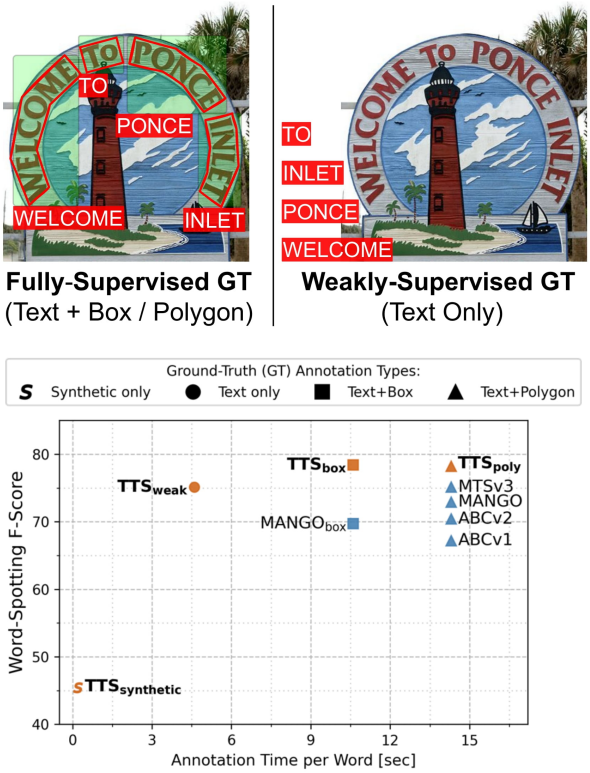

Figure 1. **Weakly-supervised text spotting.** Top: Visualization of 'fully' (left) and 'weakly' (right) supervised ground-truth (GT) annotations. Bottom: Text spotting methods results on the Total-Text dataset (higher is better) vs. the time cost per word to annotate the datasets used for training (Sec. 4.5, lower is better). Even when using weaker annotations only, our method surpasses state-of-the-art fully-supervised methods.

mutually optimizing the backbone, the tasks are separate, requiring both transcription annotations for the recognition head, and polygons or bounding box annotations for the detection head. Recently, more sophisticated methods forgo the two-stage approach by directly localizing and classifying the characters in the text [1, 30], which further requires character-level annotations.

The datasets in the field of text spotting consist of synthetic and real data. Real data annotation is an expensive task, however relying solely on synthetic data leads to poor results. Most of the annotation time is dedicated to the detection ground-truth, while the transcription annotation alone requires less than half of the time, as discussed in Sec. 4.5. State-of-the-art-methods explicitly segment the text area, allowing the recognizer to cope with rotated, curved, or densely located text and ignore background noise [5, 22]. A disadvantage of such methods is that they require expensive polygonal annotations [7, 16].

In this work, we suggest a new text spotting approach, TextTranSpotter (TTS), which can forgo the expensive spatial annotations and use only transcript annotations for real data. This setting is *weakly supervised*, in the sense that only partial information about the text in the image is used for training. At inference time, the model outputs both the detection and the transcription of the text in the image. The weakly supervised setting has many use-cases, especially in situations where annotation resources are limited or there is an existing dataset with only text transcription annotations [15]. Furthermore, TTS can be trained in both a fully- or a weakly-supervised manner, thus allowing a trade-off between model performance and annotation cost (Fig. 1).

To allow the weakly-supervised setting, we depart from existing text spotting methods which treat text detection and recognition as related but independent tasks. Our approach includes a novel architecture and loss function which better entangle the two tasks, taking a step further towards a unified end-to-end system. TextTranSpotter takes advantage of recent developments in transformers [4, 10, 38] to create a multitask network (see Fig. 2), learning a single object query embedding for both detection and recognition heads. The task heads are very simple and lean; the detection head is a linear feed-forward network and the recognition head is a Recurrent Neural Network (RNN) [34]. This indicates that the majority of the computation is performed in the shared transformer, unlike most approaches which use more intricate recognition and detection networks (see supplementary for comparison of methods). The input to the recognition head is the transformer output, which allows it to learn the relevant areas of interest for the given query instead of being given this area explicitly as input. Therefore, it does not require accurate segmentation of the text to perform even in challenging scenarios, such as rotated text, arbitrary-shaped text, or text with overlapping bounding boxes (Sec. 4.4). If the segmentation output is desired, a mask head can be added similarly to the detection and recognition head using a simple deconvolutional decoder.

Our weakly-supervised training scheme is obtained by introducing a new loss function based on the Hungarian matching loss [4] that simultaneously optimizes the detection and recognition tasks. The Hungarian loss, which has shown promise in the field of object detection [4, 9, 29, 44], is meaningful in our setting, where the matching explicitly uses the text content for the detection optimization. Our Hungarian loss, which we call Text Hungarian Loss, replaces the detection cost with a recognition cost in the matching criteria. The shared embedding that is optimized in this manner allows for a significant benefit compared to training on synthetic data only, without using any spatial information about the real data. Our weakly-supervised model reaches results comparable to existing fully-supervised methods.

Our main contributions are:

1. A weakly-supervised training scheme using only the text annotations without any spatial ground-truth for real data, utilizing a novel text-based Hungarian matching loss.

2. The first multi-task transformer-based approach for text spotting, in which a single representation is being learned per word for both detection and recognition predictions.

3. Extensive quantitative benchmarks showing our fully-supervised method achieves state-of-the-art results on common text spotting benchmarks, and our weakly-supervised method achieves results competitive with previous fully-supervised methods.

4. The first text spotting framework to offer both a fully-supervised training scheme and a weakly-supervised one for the same architecture, presenting a trade-off between model accuracy and annotation cost.

2. Related Work

Text Spotting. Li *et al*. [20] may be the first to integrate deep detection and recognition modules into a unified end-to-end system, by using a shared backbone encoder and RoIPooling [33] to feed the detected features to the recognition head. Liu *et al*. [25] suggest using RoIRotate to enable feature extraction from rotated rectangle detection results. Liao *et al*. [28] introduce Mask TextSpotter, which takes advantage of character-level annotations to detect and recognize characters and instance masks, in order to handle arbitrary-shaped scene text. Xing *et al*. [41] detect and

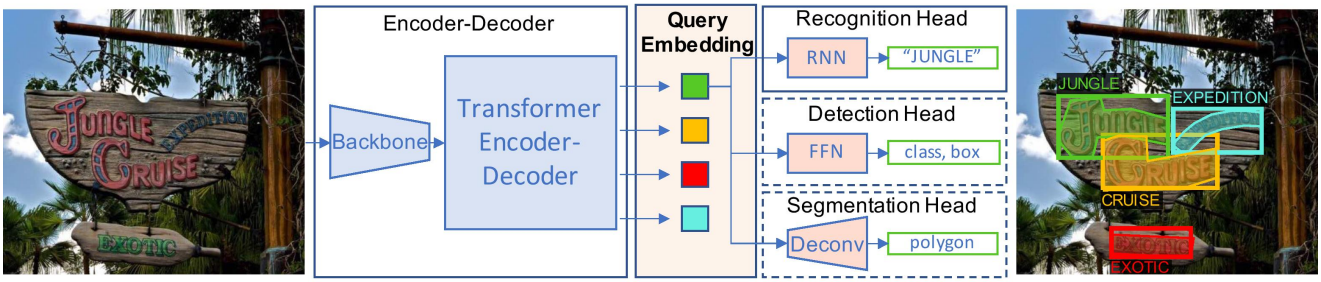

Figure 2. **TextTranSpotter.** An overview of our end-to-end architecture. Unlike previous approaches which share only the backbone, in TextTranSpotter the transformer encoder-decoder computes a joint query embedding for each detection (colored square). This embedding is shared for both recognition, detection and segmentation heads, which consist of a recurrent neural network (RNN), a linear feed-forward network (FFN) and a deconvolutional decoder (Deconv), respectively. Our weakly-supervised setting works by training only the recognition and classification heads on the real data, using the detection head at inference time for box prediction. An illustration comparing our architecture with previous text-spotting approaches can be found in the supplementary.

recognize individual characters, using the text instance detection results to group them. Liu *et al*. [26] fit parameterized Bezier curves to the text contour, and design a Bezier-Align layer for curved text feature extraction. Qin *et al*. [31] propose RoiMask, focusing on the arbitrary-shaped text region. Feng *et al*. [11] suggest using RoISlide, a sampling method which fuses features from the predicted segments of the text, allowing robustness to long arbitrary-shaped text. Liao *et al*. [22] improve Mask TextSpotter [28] by adding a Segmentation Proposal Network (SPN) to generate proposals represented by accurate polygons. Qiao *et al*. [30] remove the RoI operations and design a position-aware attention module to coarsely localize the text sequences. However, character-level and polygon annotations are required. Baek *et al*. [1] also learn character-level masks, which are fed into an attention-based recognizer.

We adopt the idea of an end-to-end system, and further suggest a unified encoding-decoding mechanism, based on a multi-task transformer. Learning a mutual feature embedding per query frees us from the need to design a hand-crafted feature pooling operation. Furthermore, the multi-task nature of our method alleviates the need for exact annotations such as polygons or character-level annotations.

Weakly Supervised Approaches. Zhao *et al*. [42] suggest a weakly-supervised approach for arbitrary text detection, by using an Expectation-Maximization based method, and provide an extensive study of the annotation time under different supervision levels. Janouskova *et al*. [15] generate a large dataset for text recognition out of weakly-annotated existing data by using a pre-trained localization module as its annotator. In order to create pseudo ground truth labels, they use Levenshtein distance to match predicted transcriptions to a weakly annotated ground-truth set. Bartz *et al*. [2] suggest training an actual end-to-end text spotting system in a weakly supervised manner, by using a fixed resolution grid as a differentiable localization pooling mechanism. Qiao *et al*. [30] take a step in the direction of weakly-supervised text spotting by training with bounding boxes instead of polygons, but this results in a significant reduction in performance.

Motivated by the study of Zhao *et al*. [42], showing the high cost of polygonal or segmentation masks annotations, we suggest an end-to-end recognition method in which bounding boxes are sufficient for the task. Moreover, we introduce a weakly-supervised framework, in which text transcriptions are the only real-data annotations needed for training, and provide a study of annotation times for both detection and text transcription.

Hungarian Matching. Throughout the past decade, learning based approaches for object detection [24, 32, 33, 37] have been used to learn engineered dense predictions, and filter near-duplicate predictions using hand-crafted rules. Recently, Carion *et al*. [4] presented a new object detection method, DETR, that formulates the problem as a direct set prediction problem. It uses a bipartite matching loss based on the Hungarian algorithm [18] to perform a one-to-one matching between ground-truth and predicted detections, unlike dense approaches in which the matching is one-to-many. This sparse detection paradigm has become popular in the object detection literature [35, 36, 44] and has advanced the field. Zhu *et al*. [44] mitigate some of the issues in DETR, namely the slow convergence and low performance on small objects, by incorporating deformable attention and a multi-scale architecture.

Following this line of research, we find the sparse detection approach suitable for multi-task loss formulation, where a given object query can be optimized for additional tasks besides detection. We use a Hungarian matching based loss, by adding a recognition cost term to the matching criteria.

3. Method

We suggest an end-to-end text spotting approach, named TextTranSpotter (TTS). A description of its architecture is

presented in Sec. 3.1, a novel variation of the Hungarian matching loss for text spotting is described in Sec. 3.2, and an adaptation of this method into a weakly-supervised setting is described in Sec. 3.3.

3.1. Architecture

TTS consists of a transformer-based Encoder-Decoder, followed by parallel detection, recognition, and segmentation heads, as illustrated in Fig. 2.

Joint Query Embedding. Our Encoding-Decoding module is shared between the detection and recognition branches. Following Carion *et al*. [4], our architecture uses a predetermined number of learned positional embeddings as input to the decoder, called object queries. The decoder learns a latent representation per object query, $q_{emb} \in \mathbb{R}^{d_{emb}}$, which is used as an input to all the task-specific heads in our model. Both detection and recognition heads are designed in a light-weight manner, meaning that the majority of the computation is done by the transformer which is optimized jointly for both tasks. This setting improves the embedding for the detection task not only through the detection loss but through the recognition loss optimization, as we show in the ablation study in Sec. 4.6. If a polygon output is desired, the optimized query embedding can be used as an input to a segmentation head, as described below.

The network is based on the Deformable-DETR architecture for object detection [44]. It consists of a conventional CNN backbone that generates a multi-scale feature map, followed by a deformable transformer encoder-decoder, in which the offsets of the attention heads are learned in addition to the attention maps themselves. The dynamic structure of this attention mechanism enables the recognition of rotated, curved, and even upside-down text, without any special treatment as described in Sec. 4.4. In fact, our network is able to achieve this even though it is trained using only axis-aligned box annotations, which are significantly less costly than polygon annotations.

Detection Head. We follow recent object detection methods [4, 44], and use a 3-layer feed-forward network (FFN) to regress the normalized parameters of the query word box w.r.t. the input image, and a linear projection layer to predict the query score, *i.e*., classify whether or not a query contains a word.

Recognition Head. To the best of our knowledge, all previous recognition models, including ones used in text spotting approaches, use a spatial signal as input to the recognition head (*e.g*., an image, or a cropped output of the backbone). In our approach, only the one-dimensional joint query embedding, computed by the transformer encoder-decoder, is used. To extract the text transcription we use a sequential LSTM-based decoder, with a one-to-many mapping where the input is the joint query embedding q_{emb} and the output for each time-step k is the character probabilities $t^k \in \mathbb{R}^l$ where l is the length of the alphabet.

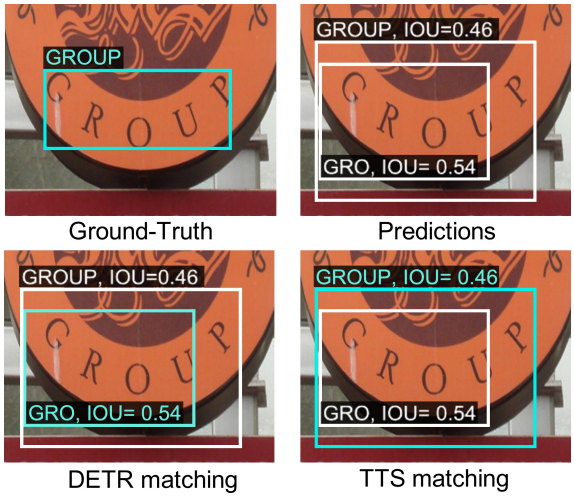

Figure 3. **Matching operation.** Top: GT and model predictions during training. Bottom: Matching operation using the original DETR criteria [4] (left), and using our suggested criteria (right). The prediction matched with the GT is marked in blue. It can be seen that TTS matches the prediction with the best predicted transcription, even though its box IOU score is lower.

Segmentation Head. TTS is trained in its fully-supervised setting using the text bounding boxes and recognition transcriptions, without any polygon annotations. However, if a polygon output is desired, a segmentation head can be trained separately based on the frozen TTS model weights. Given the pre-trained query embedding, a light-weight segmentation head, built with 4 linear layers and 3 deconvolution layers, may be used to extract a binary mask, describing the text in the detected bounding box. A polygonal output is then computed from the binary mask.

3.2. Text Hungarian Loss

Inspired by recent object detection approaches [4,35,44], we adopt the bipartite matching loss approach, using the Hungarian algorithm [18] to find a one-to-one matching $\hat{\sigma}$, between the ground-truth and predicted detections:

$$\hat{\sigma} = \underset{\sigma \in \theta_N}{\operatorname{argmin}} \sum_{i=1}^{N} C(y_i, \hat{y}_{\sigma(i)}), \tag{1}$$

where C is the criteria used to perform the matching, y is the ground truth set, $\hat{y}$ is the predicted set, N is the number of predictions, or object queries, and θ_N is the set of possible matches. The Hungarian loss function is formulated based on the matching $\hat{\sigma}$:

$$L_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} L(y_i, \hat{y}_{\hat{\sigma}(i)}). \tag{2}$$

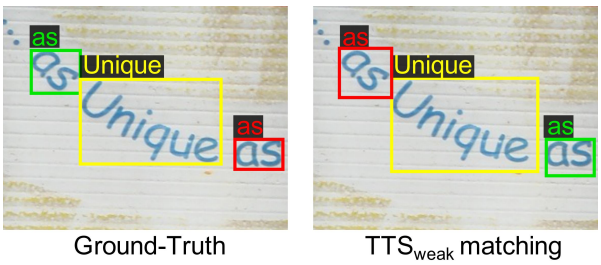

Figure 4. **Weakly-supervised matching swap.** GT instances are shown on the left. Predictions during the training of TTS$_{\text{weak}}$ are shown on the right, where each prediction is matched with the GT of the same color. Although the matching of the two occurrences of the word "as" are swapped, the weakly-supervised loss remains the same (Eq. 7), hence the matching is correct.

To better leverage the transcription annotations, we take into account not only the detection and classification criteria as in [4], but also add a recognition-based criteria C_{rec}, and loss L_{rec}, into the matching cost C and loss L, introducing a novel Text Hungarian Loss.

The fully-supervised matching criteria is:

$$C(y, \hat{y}_{\sigma(i)}) = -\alpha_c \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \alpha_{\text{box}} C_{\text{box}}(b_i, \hat{b}_{\sigma(i)}) + \mathbb{1}_{\{c_i \neq \emptyset\}} \alpha_{\text{rec}} C_{\text{rec}}(t_i, \hat{t}_{\sigma(i)}), \quad (3)$$

where c_i, b_i and t_i are the ground truth class, bounding box and transcription respectively, $\hat{p}_{\sigma(i)}(c_i)$ is the predicted probability for class c_i, and α_c, α_{box} and α_{rec} are the weights for the classification, bounding box, and transcription criteria. The fully-supervised loss term is:

$$L(y_i, \hat{y}_{\hat{\sigma}(i)}) = -\beta_c log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \beta_{\text{box}} L_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i \neq \emptyset\}} \beta_{\text{rec}} L_{\text{rec}}(t_i, \hat{t}_{\hat{\sigma}(i)}). \quad (4)$$

Where L_{box} is the bounding box loss, defined as in DETR [4], and β_c, β_{box} and β_{rec} are the weights for the classification, bounding box, and transcription losses.

We use a cross entropy loss for both the recognition criteria and loss terms:

$$C_{\text{rec}}(t_i, \hat{t}_{\sigma(i)}) = L_{\text{rec}}(t_i, \hat{t}_{\sigma(i)}) = \sum_j -log \hat{p}_{\sigma(i)}(t_i^j) \quad (5)$$

where j is the character index in the word t_i.

Fig. 3 shows examples of matching between the ground-truth and the model's predictions using different criteria. Using only the detection and classification scores for the matching, as in DETR [4], may lead the model to match a box query with a higher intersection-over-union (IOU) but worse recognition results.

We experiment with the new loss and various settings for the new matching term, as described in Sec. 4, and show that the addition of the recognition term contributes to better recognition performance in the end-to-end results.

3.3. Weakly Supervised Text Spotting

Our Text Hungarian Loss finds a matching between ground-truth and predictions based not only on the detected box but also on the recognition output. This opens up the possibility to match the ground-truth and predicted words based only on the recognition and classification criteria. As a result, the model can be optimized using only the transcription annotations, *i.e.*, a list of words that appear in the image, without any spatial annotations. At inference time the model still outputs bounding boxes for the predicted words, similarly to the fully-supervised models. We train the models in this setting with fully-supervised synthetic data, and weakly-supervised real (non-synthetic) data.

The criteria used in the weakly supervised training is therefore:

$$C_{\text{weak}}(y, \hat{y}_{\sigma(i)}) = -\alpha_c \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \alpha_{\text{rec}} C_{\text{rec}}(t_i, \hat{t}_{\sigma(i)}) \quad (6)$$

and the loss term is:

$$L_{\text{weak}}(y_i, \hat{y}_{\hat{\sigma}(i)}) = - \beta_c log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \beta_{\text{rec}} L_{\text{rec}}(t_i, \hat{t}_{\hat{\sigma}(i)}). \quad (7)$$

Note that in this setting, if there are multiple words with the same transcription, it is possible that there is more than a single correct match. An example of this case is shown in Fig. 4, where the word "as" repeats twice in the image causing the queries to be mismatched. Since the training is performed only on the recognition head and not the bounding-box regression, the supervision for each of the transcriptions remains the same and does not affect the training process.

4. Experiments

We evaluate TextTranSpotter on common benchmarks using both the fully- and weakly-supervised settings. We further test the model performance on rotated and curved text, and conduct ablation studies regarding its architecture and matching criteria.

4.1. Implementation Details

Following Liao *et al.* [22], the model is first trained on SynthText [12], a large synthetic dataset with over 850k images, designed for both detection and recognition of text in images to obtain TTS$_{\text{synthetic}}$. We then train our model on a mix of SynthText together with real datasets; Total-Text [7], about 1k images including mainly curved text in various orientations and shapes, ICDAR 2015 [16], 1k images containing mostly small text instances, ICDAR 2013 [17], 229 training images with mostly near-horizontal text, COCOText [39], 43k train images taken from the MS-COCO

Method	ICDAR 2015						Total-Text			
	Word Spotting			End-to-End			Word Spotting		End-to-End	
	S	W	G	S	W	G	None	Full	None	Full
MTS-V1 [28]	79.3	74.5	64.2	79.3	73.0	62.4	-	-	52.9	71.8
MTS-V2 [21]	82.4	78.1	73.6	83.0	77.7	73.5	-	-	65.3	77.4
TextDragon [11]	**86.2**	**81.6**	68.0	82.5	78.3	65.2	-	-	48.8	74.8
ABCNet-V1 [26]	-	-	-	-	-	-	67.2	76.4	63.7	76.6
MTS-V3 [22]	83.1	79.1	75.1	83.3	78.1	74.2	75.1	81.8	71.2	78.4
ABCNet-V2 [27]	-	-	-	82.7	78.5	73.0	70.4	78.1	-	-
CRAFTS [1]	-	-	-	83.1	**82.1**	74.9	-	-	**78.7**	-
MANGO* [30]	85.2	81.1	74.6	**85.4**	80.1	73.9	72.9	83.6	68.9	78.9
TTS$_{\text{poly}}$	85.0	81.5	**77.3**	85.2	81.7	**77.4**	**78.2**	**86.3**	75.6	**84.4**

Table 1. **Evaluation results on ICDAR 2015 and Total-Text datasets.** Word spotting and end-to-end f-score using strong (S), weak (W), generic (G), none and full lexicons. * MANGO [30] evaluated with IOU 0.1. Our method shows the best results using generic lexicons.

Method	Annotations (real)			ICDAR 2015						Total-Text			
				Word Spotting			End-to-End			Word Spotting		End-to-End	
	Text	Box	Poly	S	W	G	S	W	G	None	Full	None	Full
TTS$_{\text{synthetic}}$				53.1	46.9	42.9	53.2	47.0	43.0	45.4	60.9	46.3	58.8
TTS$_{\text{weak}}$	✓			78.6	75.1	70.2	78.7	75.2	70.1	75.1	83.5	71.5	80.1
TTS$_{\text{box}}$	✓	✓		**84.9**	**81.3**	**77.1**	**85.0**	**81.5**	**77.1**	**78.4**	**86.6**	**75.8**	**84.5**
MANGO$_{box}$ [30]	✓	✓		-	-	-	-	-	-	69.7	80.6	-	-
MTS-V3 † [22]	✓	✓	✓	82.7	78.5	74.7	82.5	77.4	73.5	74.8	81.2	70.5	77.7

Table 2. **Results under limited training data annotations.** Word spotting and end-to-end f-score using strong (S), weak (W), generic (G), none and full lexicons. "Text", "Box" and "Poly" denotes the model trained on real data using annotations of text, bounding boxes and polygons, respectively. Axis-aligned evaluation was used. Results are improved dramatically when training on real data with text-only annotations (TTS$_{\text{weak}}$) compared to training only with fully-supervised synthetic data (TTS$_{\text{synthetic}}$). Using box annotations (TTS$_{\text{box}}$) improves results even further. † We show the results of MTS-V3 [22] using axis-aligned evaluation as a reference point.

dataset [23], and SCUT [43], 1k training images containing varied text. Both our weakly- (TTS$_{\text{weak}}$) and fully-supervised (TTS$_{\text{box}}$) models are obtained using this setup, where for TTS$_{\text{weak}}$ we use fully-annotated synthetic data, and weakly-annotated real data. To produce a polygonal output, we freeze TTS$_{\text{box}}$ weights, and train only a segmentation head, using the same mix of real datasets, and a subset of the SynthText dataet, with polygonal annotations. We call this model TTS$_{\text{poly}}$.

We use Total-Text and ICDAR 2015 test data to evaluate both our fully-supervised and weakly-supervised models. To test our method's robustness to rotations, we use the Rotated ICDAR 2013 dataset similarly to Liao *et al*. [22].

4.2. Comparison to Previous Methods

The evaluation results of TextTranSpotter, compared to previous approaches on Total-Text and ICDAR 2015, are shown in Table 1. Evaluation was done using the standard polygonal evaluation protocol with IOU threshold of 0.5.

Both word spotting and end-to-end results are presented. For ICDAR 2015 we use "strong", "weak" and "generic" dictionaries, and for Total-Text, we show the results without a lexicon and using a "full" lexicon. On the Total-Text dataset, our method outperforms previous approaches with and without using a lexicon in the word spotting setting and using a "full" lexicon in the end-to-end setting. On ICDAR 2015, our method shows the best results using "generic" lexicon, the most common and challenging use-case.

4.3. Weakly-Supervised Results

In Table 2 we show results using different supervision types. Unlike most of the previous methods, TTS$_{\text{box}}$, TTS$_{\text{synthetic}}$ and TTS$_{\text{weak}}$ output axis-aligned bounding boxes and not polygons, and are therefore evaluated by matching the bounding boxes of the ground truth polygons with our method's bounding boxes output, using a matching threshold for the axis-aligned IOU of 0.5.

We show that this change has only a minor affect on the evaluation by demonstrating on a previous method (MTS-V3 [22]). Using the published model, we compute bounding boxes for the polygonal outputs and evaluate with our axis-aligned evaluation (Table 2). We compare the results of

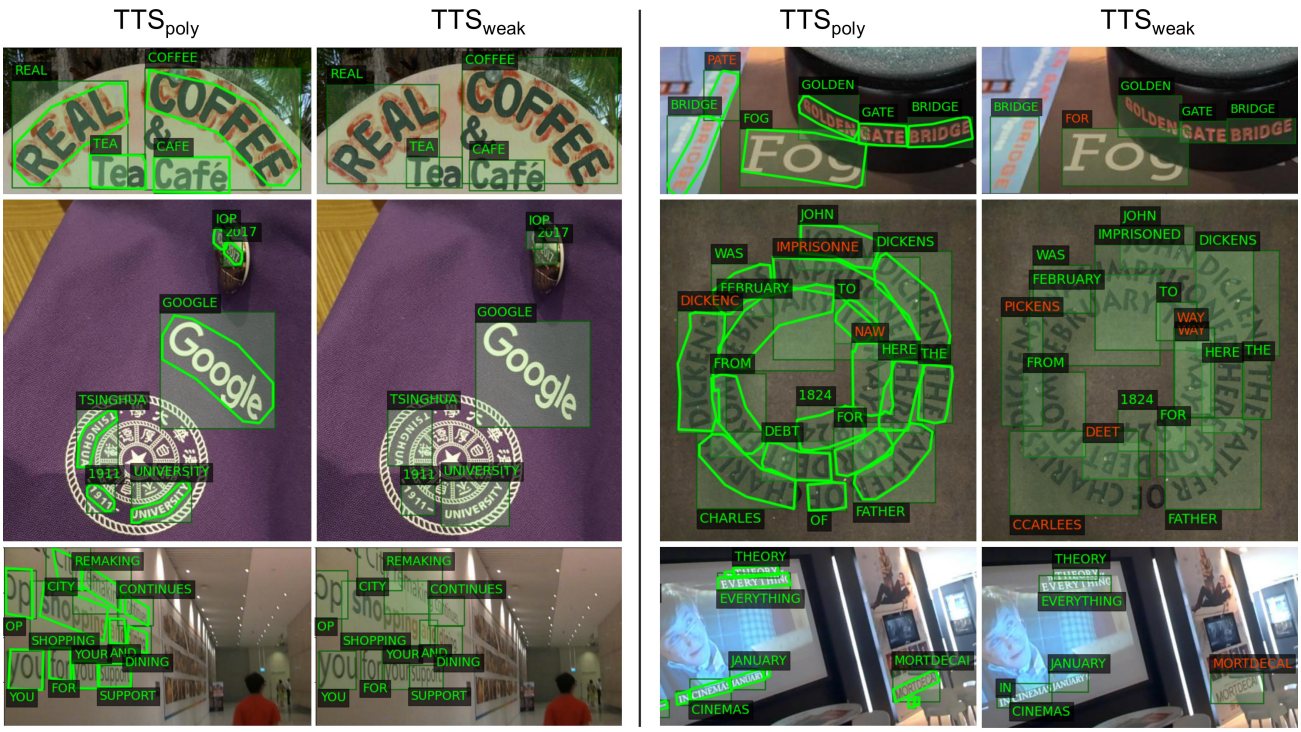

Figure 5. **Qualitative results.** Prediction examples of our weakly-supervised (TTS$_{\text{weak}}$) and fully-supervised (TTS$_{\text{poly}}$) models, on Total-Text and ICDAR 2015 samples. TTS can handle rotated, curved, and even upside-down text instances, effectively distinguishing between overlapping boxes by extracting only the relevant text from the bounding box. TTS$_{\text{weak}}$ has lower performance than TTS$_{\text{poly}}$, which is expected given the reduction in supervision, however it manages to output high quality results. Fail cases are presented on the right.

Method	45°		60°	
	Det.	E2E	Det.	E2E
CharNet R-50 [41]	57.2	33.9	58.8	9.3
MTS-V2 [28]	62.2	54.2	65.5	56.6
MTS-V3 [22]	84.2	76.1	84.7	76.6
TTS$_{\text{poly}}$	**88.8**	**80.4**	**87.6**	**80.1**
MTS-V3† [22]	82.9	75.4	81.2	75.3
TTS$_{\text{box}}$†	**89.9**	**80.1**	**89.7**	**81.0**

Table 3. **Results on Rotated ICDAR 2013 dataset.** F-measure of detection (Det.) and end-to-end (E2E) recognition, under different rotation angles. † means that axis-aligned evaluation was used. TTS outperforms existing methods.

the axis-aligned evaluation to the method's official polygonal evaluation results (Table 1). The axis-aligned evaluation slightly lowers the results in comparison with the polygonal evaluation, so the results in Table 2 can be compared to the polygonal evaluation results in Table 1. Training using only the synthetic data (TTS$_{\text{synthetic}}$) results in very low performance. In comparison, using our weakly-supervised training scheme (TTS$_{\text{weak}}$) improves the results significantly, reaching competitive results to fully-supervised state-of-the-art methods while only using the transcription supervision on the real datasets. Using bounding box supervision (TTS$_{\text{box}}$) improves results even further and reaches state-of-the-art results. When directly comparing TTS$_{\text{box}}$ and TTS$_{\text{poly}}$ (Table 1), we see that the polygonal annotations do not improve the results, and sometimes even degrade them. This is due to the fact that the model is trained without polygons, and the segmentation head is trained afterwards, with the rest of the model weights frozen. We believe further optimization of this head can improve the results, but this is not the focus of this work.

4.4. Robustness to Rotation and Curvature

We evaluate TextTranSpotter's robustness to rotation by testing it on the Rotated ICDAR 2013 dataset. Table 3 shows our model improves performance on this dataset compared to previous approaches, even though our models are trained using only bounding boxes (the segmentation head is trained separately, as described in Sec. 3.1). Using bounding boxes causes more background text and noise to enter the recognition head, and there is no explicit information about the orientation of the text. However, since TextTranSpotter uses the transformer output as an input to both recognition and detection heads, it is able to ignore the irrelevant information and produce the correct transcript. Fig. 5 shows TTS$_{\text{weak}}$and TTS$_{\text{poly}}$ performance on challenging

TTS Architecture	Detection		
	P	R	F
EncDec. + det.	88.4	82.8	85.5
EncDec. + det. + recog.	**90.9**	**84.4**	**87.6**

Table 4. **Detection ablation.** Detection precision, recall, and F-measure of TTS, with and without the recognition head, on Total-Text dataset. It can be seen that optimizing the query embedding for the recognition task, improves the detection task.

Matching Criter.	Recog. Head (params)	End-to-End	
		None	Full
det. + cls.	linear (3.4M)	73.6	83.6
det. + cls.	RNN (2.8M)	74.0	84.5
det. + cls. + recog.	**RNN (2.8M)**	**75.8**	**84.5**

Table 5. **Recognition head and matching ablation.** End-to-end results of models on the Total-Text dataset, trained in a fully-supervised manner, using linear and RNN recognition heads, and with and without the recognition criterion. It can be seen that using the recognition matching criterion improves performance, and that the RNN recognition head is preferable to the linear head.

examples. Our method is able to handle large variations in rotation and font scale as well as cases with large overlaps between the bounding boxes.

4.5. Annotation Cost Study

To estimate the annotation time required for each labeling method, we conducted a user study on 100 images out of the TotalText dataset [7] with 9 annotators. Each user was asked to annotate different images with polygons and transcriptions, bounding boxes and transcriptions, or only transcriptions. The results as presented in Fig. 1 show that the average annotation time per instance is 14.3, 10.6 and 4.6 seconds for polygon, bounding box and transcription only annotations respectively. This is consistent with the results by Zhao *et al*. [42] which show that the average annotation time per image on the ICDAR-ArT dataset [8] is 60 and 39 seconds using polygons and bounding boxes respectively (without transcriptions).

4.6. Ablation Study

We test the detection performance of the original Deformable DETR [44] compared to the fully-supervised TextTranSpotter, shown in Table 4. We train both models in the same manner, with the significant differences being the Text Hungarian loss for TTS and the recognition head. The models are evaluated on the Total-Text dataset using the standard text detection metrics. TTS_{box} outperforms the vanilla Deformable DETR model, improving both recall and precision of the model. This experiment highlights the benefit of mutually optimizing the detection and recognition tasks, in comparison to training separate standalone models for each task.

Next, we study the impact of the Text Hungarian Loss proposed in Sec. 3.2. We train our fully supervised model using two different matching criteria for the Hungarian matching algorithm; detection and classification, as presented in DETR [4], versus detection, classification and recognition, as in our Text Hungarian Loss (Sec. 3.2). We evaluate the models for both end-to-end and detection on Total-Text, and show our results in Table 5. The text matching criterion improves results, mainly for recognition. Therefore, using it improves performance for the end-to-end setting without a lexicon. When using a lexicon, the improvement to the recognition performance is less significant and the end-to-end results remain the same. We use the full matching criteria for our fully-supervised training.

The query embeddings which go into the recognition head in TTS are one dimensional and have no spatial or sequential structure, in contrast to previous recognition architectures. In addition, the recognition head is trained without using the ground truth transcription during the forward pass like previous approaches, since the matching is performed only at the end of the forward pass. Taking into account these two significant changes, we aim to study the contribution of using an RNN compared to using linear layers. The results using the two different recognition heads are presented in Table 5. Using a linear head lowers the results compared to an RNN head, showing that the recurrent output formulation is beneficial for the recognition task while reducing the number of parameters in the recognition head.

5. Conclusions

We presented the first text spotting framework that can be trained in both fully- and weakly-supervised settings. By using a transformer encoder-decoder to learn a joint representation for the recognition and detection tasks, we can forgo much of the expensive annotations that are required in other approaches, and trade-off model accuracy vs. annotation time. The transformer's attention mechanism helps achieve accurate results on difficult cases, such as curved, rotated, dense, and even upside-down text. Our novel Text Hungarian Loss includes the recognition information in the detection optimization and permits training without the detection supervision altogether. Our method achieves state-of-the-art results on several benchmarks in the fully-supervised approach, and competitive results in the weakly-supervised setting. We hope that this work will open the door to new research directions in the field of text spotting, and to new views regarding which annotations are truly required for this task, examining trade-offs and combinations of weakly and fully supervised data.

References

[1] Youngmin Baek, Seung Shin, Jeonghun Baek, Sungrae Park, Junyeop Lee, Daehyun Nam, and Hwalsuk Lee. Character region attention for text spotting. *ArXiv*, abs/2007.09629, 2020. 2, 3, 6

[2] Christian Bartz, Haojin Yang, and Christoph Meinel. See: towards semi-supervised end-to-end scene text recognition. In *Thirty-second aaai conference on artificial intelligence*, 2018. 3

[3] Michal Busta, Lukas Neumann, and Jiri Matas. Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In *Proceedings of the IEEE international conference on computer vision*, pages 2204–2212, 2017. 1

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 3, 4, 5, 8

[5] Xiaoxue Chen, Lianwen Jin, Yuanzhi Zhu, Canjie Luo, and Tianwei Wang. Text recognition in the wild: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–35, 2021. 1, 2

[6] Xiangrong Chen and A.L. Yuille. Detecting and reading text in natural scenes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II, 2004. 1

[7] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017. 2, 5, 8

[8] Chee Kheng Chng, Yuliang Liu, Yipeng Sun, Chun Chet Ng, Canjie Luo, Zihan Ni, ChuanMing Fang, Shuaitao Zhang, Junyu Han, Errui Ding, et al. Icdar2019 robust reading challenge on arbitrary-shaped text-rrc-art. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1571–1576. IEEE, 2019. 8

[9] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021. 2

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 2

[11] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3, 6

[12] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 5

[13] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International journal of computer vision*, 116(1):1–20, 2016. 1

[14] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *European conference on computer vision*, pages 512–528. Springer, 2014. 1

[15] Klara Janouskova, Jiri Matas, Lluis Gomez, and Dimosthenis Karatzas. Text recognition-real world data and where to find them. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4489–4496. IEEE, 2021. 2, 3

[16] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 2, 5

[17] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazàn Almazàn, and Lluís Pere de las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493, 2013. 5

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 3, 4

[19] Huiping Li, David Doermann, and Omid Kia. Automatic text detection and tracking in digital video. *IEEE transactions on image processing*, 9(1):147–156, 2000. 1

[20] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5248–5256, 2017. 1, 2

[21] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):532–548, 2021. 6

[22] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020. 2, 3, 5, 6, 7

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 6

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[25] X. Liu, Ding Liang, Shihan Yan, D. Chen, Y. Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified net-

work. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5676–5685, 2018. 2

[26] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2020. 3, 6

[27] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *arXiv preprint arXiv:2105.03620*, 2021. 6

[28] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2, 3, 6, 7

[29] Qi Ming, Lingjuan Miao, Zhiqiang Zhou, Xue Yang, and Yunpeng Dong. Optimization for arbitrary-oriented object detection via representation invariance loss. *IEEE Geoscience and Remote Sensing Letters*, 2021. 2

[30] Liang Qiao, Ying Chen, Zhanzhan Cheng, Yunlu Xu, Yi Niu, Shiliang Pu, and Fei Wu. Mango: A mask attention guided one-stage scene text spotter. *arXiv preprint arXiv:2012.04350*, 2020. 2, 3, 6

[31] Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, and Ying Xiao. Towards unconstrained end-to-end text spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4704–4714, 2019. 3

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 3

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 2, 3

[34] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 2

[35] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14454–14463, 2021. 3, 4

[36] Zhiqing Sun, Shengcao Cao, Yiming Yang, and Kris Kitani. Rethinking transformer-based set prediction for object detection. *arXiv preprint arXiv:2011.10881*, 2020. 3

[37] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 3

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2

[39] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*, 2016. 5

[40] Victor Wu, R. Manmatha, and Edward M. Riseman. Finding text in images. In *Proceedings of the Second ACM International Conference on Digital Libraries*, DL '97, page 3–12, New York, NY, USA, 1997. Association for Computing Machinery. 1

[41] Linjie Xing, Zhi Tian, Weilin Huang, and Matthew R Scott. Convolutional character networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9126–9136, 2019. 2, 7

[42] Mengbiao Zhao, Wei Feng, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Weakly-supervised arbitrary-shaped text detection with expectation-maximization algorithm. *ArXiv*, abs/2012.00424, 2020. 3, 8

[43] Zhuoyao Zhong, Lianwen Jin, Shuye Zhang, and Ziyong Feng. Deeptext: A unified framework for text proposal generation and text detection in natural images. *ArXiv*, abs/1605.07314, 2016. 6

[44] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 3, 4, 8