Towards Unconstrained End-to-End Text Spotting

Siyang Qin, Alessandro Bissacco, Michalis Raptis, Yasuhisa Fujii, Ying Xiao
Google AI
{qinb,bissacco,mraptis,yasuhisaf,yingxiao}@google.com

Abstract

We propose an end-to-end trainable network that can simultaneously detect and recognize text of arbitrary shape, making substantial progress on the open problem of reading scene text of irregular shape. We formulate arbitrary shape text detection as an instance segmentation problem; an attention model is then used to decode the textual content of each irregularly shaped text region without rectification. To extract useful irregularly shaped text instance features from image scale features, we propose a simple yet effective RoI masking step. Additionally, we show that predictions from an existing multi-step OCR engine can be leveraged as partially labeled training data, which leads to significant improvements in both the detection and recognition accuracy of our model. Our method surpasses the state-of-the-art for end-to-end recognition tasks on the ICDAR15 (straight) benchmark by 4.6%, and on the Total-Text (curved) benchmark by more than 16%.

1. Introduction

Automatically detecting and recognizing text in images can benefit a large number of practical applications, such as autonomous driving, surveillance, or visual search and can increase the environmental awareness of visually impaired people [44].

Traditional optical character recognition (OCR) pipeline methods usually partition the scene text reading task into two sub-problems, *scene text detection* and *cropped text line recognition*. Text detection methods try to spot text instances (words or lines) in the input image, while text recognition models take a cropped text patch and decode its textual content. Since most scene text detection methods are unable to directly predict the correct text reading direction, an additional direction identification step is necessary for successful OCR engines [56].

Despite their long history and great success, the use of multiple models within an OCR pipeline engine has several disadvantages: errors can accumulate in such a cascade which may lead to a large fraction of garbage predictions.

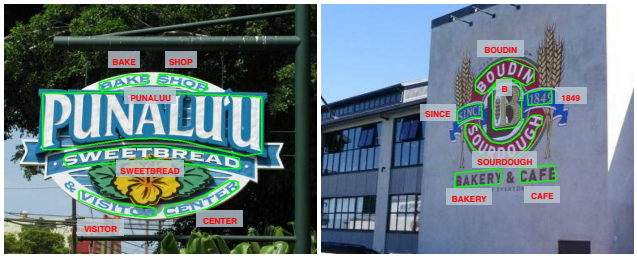

Figure 1. Our end-to-end model can predict the locations and transcriptions of text with arbitrary shape in a single forward pass.

Furthermore, each model in the pipeline depends on the outputs of the previous step, which makes it hard to jointly maximize the end-to-end performance, and fine-tune the engine with new data or adapt it to a new domain. Finally, maintaining such a cascaded pipeline with data and model dependencies requires substantial engineering effort.

End-to-end OCR models overcome those disadvantages and thus have recently started gaining traction in the research community [42, 54, 37, 24, 32]. The basic idea behind end-to-end OCR is to have the detector and recognizer share the same CNN feature extractor. During training, the detector and recognizer are jointly optimized; at inference time, the model can predict locations and transcriptions in a single forward pass. While producing superior accuracy in straight text reading benchmarks, these methods struggle to generalize and produce convincing results on more challenging datasets with curved text, which arise naturally and frequently in everyday environments (see Figure 1 for two such examples). Handling arbitrary shaped text is a crucial open problem in order for OCR to move beyond its traditional straight text applications.

In this paper, we propose a simple and flexible end-to-end OCR model based on a Mask R-CNN detector and a sequence-to-sequence (seq2seq) attention decoder [3]. We make no assumptions on the shape of the text: our model can detect and recognize text of arbitrary shape, not just the limited case of straight lines. The key idea underlying our model is to skip the feature rectification step between the detector and the recognizer, and directly feed cropped and

masked text instance features to the decoder. We show that our model is able to recognize text in different orientations and even along curved paths. Our model learns where to start decoding, and how to update the attention weights to follow the unrectified text path. Our detector is based on Mask R-CNN: for each text instance, it predicts an axis-aligned rectangular bounding box and the corresponding segmentation mask. Using these, our model works seamlessly on both straight and curved text paths.

Typically the recognizer requires far more data to train than the detector. Unlike the case of multi-step OCR models where cropped text lines (easier to collect and synthesize) are used to train the recognizer, previous end-to-end models demand fully labeled images as training data. This makes end-to-end training challenging due to the short of fully annotated images. Furthermore, by the time the recognizer has converged, the detector is often substantially overfitted. In this work, we solve both issues by adding additional large scale partially labeled data which is automatically labeled by an existing multi-step OCR engine [1] [4]. If an input training sample is partially annotated, only the recognizer branch is trained. We find that this significantly boosts the performance of our model.

Our method surpasses the previous state-of-the-art results by a large margin on both straight and curved OCR benchmarks. On the popular and challenging ICDAR15 (straight) dataset, our model out-performs the previous highest by 4.6% on end-to-end F-score. On the Total-Text (curved) dataset, we significantly increase the state-of-the-art by more than 16%.

In summary, the contributions of this paper are threefold:

- We propose a flexible and powerful end-to-end OCR model which is based on Mask R-CNN and attention decoder. Without bells and whistles, our model achieves state-of-the-art results on both straight and curved OCR benchmarks.

- We identify feature rectification as a key bottleneck in generalizing to irregular shaped text, and introduce a simple technique (RoI masking) that makes rectification unnecessary for the recognizer. This allows the attention decoder to directly operate on arbitrarily shaped text instances.

- To the best of our knowledge, this is the first work to show that end-to-end training can benefit from partially labeled data bootstrapped from an existing multi-step OCR engine.

[1] Publicly available via Google Cloud Vision API.

2. Related Work

In this section, we briefly review the existing text detection and recognition methods, and highlight the differences between our method and current end-to-end models. For a more detailed review, the reader is referred to [40].

Scene Text Detection: Over the years, the traditional sliding window based methods [28, 8, 62] and connected-component based methods [5, 26, 46, 45, 13, 47] have been replaced by deep learning inspired methods with a simplified pipeline. These newer methods have absorbed the advances from general object detection[36, 50, 49] and semantic segmentation [39, 7] algorithms, adding well-designed modifications specific to text detection. Modern scene text detection algorithms can directly predict oriented rectangular bounding boxes or tighter quadrilaterals via either single-shot [33, 61, 22, 38], or two-stage models [29, 34, 43, 48].

Recently, detecting curved text in images has become an emerging topic: a new dataset containing curved text was introduced in [11] which provides tight polygon bounding boxes and ground-truth transcriptions. In [12], Dai *et al*. formulate text detection as an instance segmentation problem, and in [41], the authors proposed representing a text instance as a sequence of ordered, overlapping disks, which is able to cover curved cases. Despite the success in detecting curved text, *reading* the curved text is an unsolved problem.

Scene Text Recognition: The goal of scene text recognition algorithms is to decode the textual content from *cropped text patches*. Modern scene text recognition methods can be grouped into two main categories, CTC (Connectionist Temporal Classification [19]) based methods [52, 23, 14] and attention based methods [57, 9, 10, 16, 31, 53]. Most scene text recognition methods from both categories assume the input text is rectified (straight line, read from left to right): the input is first resized to have a constant height, then fed to a fully convolutional network to extract features. To capture long range sequence context, some CTC-based architectures stack an RNN on top of a CNN, while others use stacked convolution layers, with a large receptive field. Finally, each feature column predicts a symbol and duplicated symbols are removed to produce the final prediction. In attention models, RNN is often used to predict one symbol per step based on the prediction at the previous step, the hidden state, and a weighted sum of the extracted image features (context). The process stops when the end-of-sequence symbol is predicted, or the maximum number of iterations is reached.

End-to-End OCR: The work from Li *et al*. [32] is the first successful end-to-end OCR model which only supports horizontal text. The multi-oriented end-to-end OCR architectures of Liu *et al*. [37], He *et al*. [24] and Sun *et al*. [54] share a common idea: they feed *rectified text region fea-*

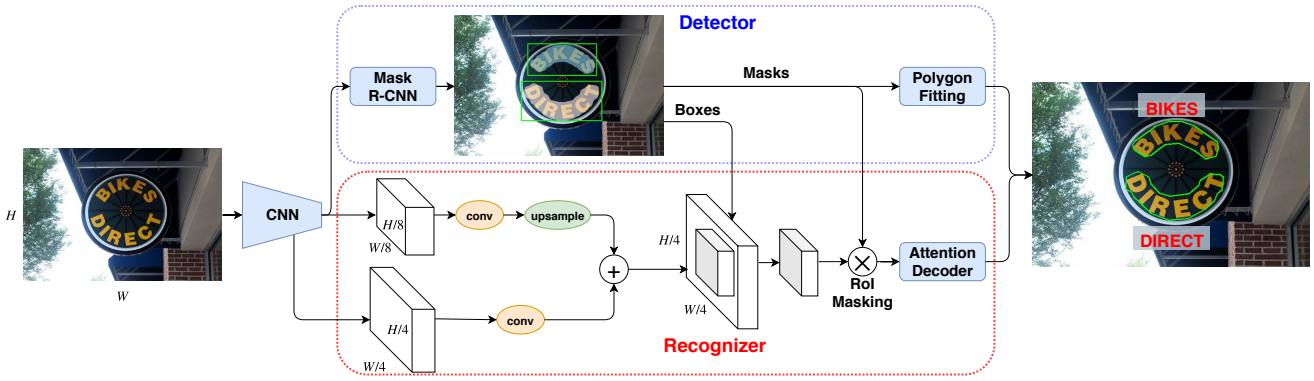

Figure 2. Overall architecture of our end-to-end OCR model.

tures to the recognizer to enable end-to-end training. In [37], the model outputs rotated rectangles in the detection stage, and used a CTC-based recognizer which can not generalize to curved cases. In [54], the detector outputs quadrilaterals and an attention-based model is used to decode the textual content. In contrast, our detector produces rectangular bounding boxes and the corresponding instance segmentation masks, which is a more general way to represent text in arbitrary shape. In addition, we remove the feature rectification step which is designed for straight text, and let the attention decoder directly operates on cropped and masked text instance features. This leads to better flexibility and performance in curved text.

Lyu *et al*. [42] proposed an end-to-end OCR engine which is based on Mask R-CNN. They adopted a simple recognition by detection scheme: in order to recognize the text, all the characters are detected individually. This method is not ideal because much of the sequential information is lost. Furthermore, detecting individual character can be difficult or even impossible in many cases. And even if all the characters are correctly detected, it is highly unclear how to link them into a correct sequence. In [42], the authors simply group characters from left to right, which precludes correct recognition of text in non-traditional reading directions. On the other hand, by leveraging sequential information, our method is able to correctly recognize text in more challenging situations and non-traditional reading directions.

3. Model Architecture and Training

Figure 2 shows the design of our end-to-end OCR model. The detector part of the model is based on Mask R-CNN which has been widely used in instance segmentation and other related tasks. For each text region (word or text line), Mask R-CNN can predict an axis-aligned rectangular bounding box and the corresponding instance segmentation mask. For the straight text case, the final detection results are obtained by fitting a min-area rotated rectangle to each segmentation mask, while a general polygon is fitted to each mask for the curved text case. By using Mask R-CNN as the detector, our model works seamlessly with straight and curved text paths.

A novel feature of our architecture is that *we do not rectify the input to the recognizer*. This renders traditional CTC-based decoders unsuitable. Instead, we use a seq2seq model (with attention) as recognizer. At each step, the decoder makes the prediction based on the output and state from the previous step, as well as a convex combination of the text instance features (context). In order to extract arbitrary shaped text instance features from image level features, we introduce *RoI masking* which multiplies the cropped features with text instance segmentation masks. This removes neighboring text and background, and ensures that the attention decoder will focus only on the current text instance.

3.1. Feature Extractor

We explore two popular backbone architecture, ResNet-50[21] and Inception-ResNet [55]; the latter model is far larger, and consequently yields better detection and recognition results. Scene text usually has large variance in scale; in order to capture both large and tiny text, the backbone should provide dense features while maintaining a large receptive field. To achieve this, we follow the suggestions from [25]: both backbones are modified to have an effective output stride of 8. In order to maintain a large receptive field, atrous convolutions are used to compensate for the reduced stride.

For ResNet-50, we modified the *conv4_1*[2] layer to have stride 1 and use atrous convolution for all subsequent layers. We extract features from the output of the third stage. The Inception-ResNet is modified in a similar way to have output stride 8, taking the output from the second repeated

[2]Our naming convention follows [25].

block (layer *PreAuxLogits*).

3.2. Detector

We follow the standard Mask R-CNN implementation. In the first stage, a region proposal network (RPN) is used to propose a number of candidate text regions of interest (RoIs). In the second stage, each RoI is processed by three prediction heads: a class prediction head to decide if it is text or not, a bounding box regression head to predict an axis-aligned rectangular box, and finally a mask prediction head to predict the corresponding instance segmentation mask.

The RPN anchors span four scales (64, 128, 256, 512) and three aspect ratios (0.5, 1.0, 2.0); using more scales and aspect ratios may increase the model's performance at the cost of longer inference time. Non-maximum suppression (NMS) is used to remove highly overlapping proposals with intersection-over-union (IoU) threshold set to 0.7. The top 300 proposals are kept. In the second stage, features from each RoI are cropped and resized to 28×28 followed by a 2×2 max pooling, which lead to 14×14 features for each RoI. At training time, RoIs are grouped into mini batches of size 64, and then fed to a class prediction head and bounding box refinement head. A second NMS is performed on top of refined boxes (IoU is set to 0.7). During inference time, the top 100 regions are sent to the mask prediction head. The final detection output is obtained after the final NMS step, which computes the IoU based on the mask instead of bounding boxes like the first two NMS steps.

3.3. Multi-Scale Feature Fusion and RoI Masking

In our experiments, we found that stride 8 features and multi-scale anchors are sufficient for the text detection task for both large and small text. However, for text recognition, a finer-grained task, denser features are needed. Inspired by the feature pyramid network [35], we gradually upsample lower resolution, but context rich features, and fuse them with higher resolution features from earlier CNN layers using element-wise addition. A 1×1 convolution (with 128 channels) is applied to all features to reduce dimensionality, and to ensure uniform shapes, before element-wise addition. This produces a dense feature map which encodes both local features and longer range contextual information, which can improve recognition performance especially for small text. In practice, we find that fusing features with stride 8 and 4 leads to the best results. More specifically, for ResNet-50, we use features after the first (stride 4), second (stride 8) and third stage (stride 8); the corresponding receptive field sizes are 35, 99 and 291 respectively. For Inception-ResNet, we use features after layer *Conv2d_4a_3x3* (stride 4), *Mixed_5b* (stride 8) and *PreAuxLogits* (stride 8), the corresponding receptive field sizes are 23, 63 and 2335 respectively.

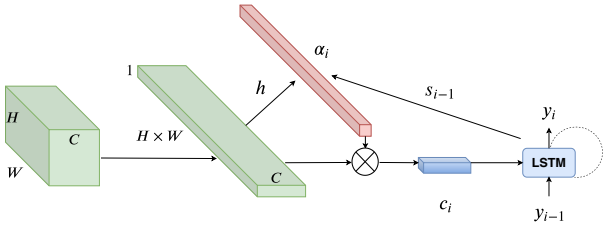

Figure 3. Our seq2seq based recognizer.

In multi-step OCR engines, each text instance is cropped out from the input image before being fed to the recognizer. In contrast, in end-to-end models, instead of cropping out the image patch, a more involved method is used to extract text instance features from the image level features output by the backbone CNN. For object detection models, axis-aligned bounding boxes are used to crop out features [17]. For text, the work in [37] and [54] proposed RoI rotate and perspective RoI transforms to compute rectified text instance features using rotated rectangles or quadrilaterals. This works well for straight text but fails in the curved text case. In this work, we propose a simple and more general way to extract text instance features that works for any shape, called *RoI masking*: first the predicted axis-aligned rectangular bounding boxes are used to crop out features, and then we multiply by the corresponding instance segmentation mask. Since we *do not* know the reading direction of the text at this point, features from each region are resized so that the shorter dimension is equal to 14 while maintaining the overall aspect ratio. RoI masking filters out the neighboring text and background, ensuring that the attention decoder will not accidentally focus on areas outside the current decoding region. Our ablation experiments in Section 4.3 show that RoI masking substantially improves the recognizer's performance.

3.4. Recognizer

The recognizer is a seq2seq model with Bahdanau-style attention proposed in [3], shown in Figure 3. At the first step, the model takes a *START* symbol and zero LSTM initial state; we then produce symbols until the End-of-Sequence (*EOS*) symbol is predicted. At each step, the final predicted distribution over possible symbols is given by:

$$p(y_i|y_1, ..., y_{i-1}, h) = \text{softmax}(W_o o_i + b_o) \qquad (1)$$

Where y_i is the predicted character, o_i is the LSTM output at time step i respectively, and h represents the flattened extracted text instance features. At each step, the LSTM takes the prediction of the previous step y_{i-1}, the previous hidden state s_{i-1} and a weighted sum of the image feature c_i (context) to compute the output o_i and new state vector s_i.

$$(o_i, s_i) = LSTM(y_{i-1}, s_{i-1}, c_i) \quad (2)$$

At each step, the decoder is able to pay attention to some specific image region and use the corresponding image features to help make the right prediction. The context vector c_i is a weighted sum of the flattened image feature h and learned weight vector α_i: $c_i = \sum_j \alpha_{ij} h_j$. The weight vector α_i is defined as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})} \quad (3)$$

$$e_{ij} = V^T \tanh(W_s s_{i-1} + W_h h_j). \quad (4)$$

The attention weight for each feature position is determined by image feature (h) and previous LSTM state (s_{i-1}) which encode the shift of the attention mask. This enables the recognizer to follow arbitrary shaped text lines.

By feeding the predicted symbol to the next step, the model can learn an implicit language model. At inference time, the predicted symbol is fed to the next step while the ground-truth one is used during training (i.e., teacher forcing).

3.5. Joint Training and Loss Function

We observe that the recognizer requires far more data and training iterations in comparison to the detector; this makes joint training difficult as the existing public datasets are not large enough to train a high performance attention decoder, especially when the input features are not rectified. Furthermore, if we train long enough to achieve convergence in the recognizer, there is a strong risk of overfitting the detector. In this work, we solve both issues by adding additional large scale partially labeled data which is automatically labeled by an existing multi-stage OCR engine from Google Cloud Vision API. If the input training sample is fully labeled, we update the weights of both detector and recognizer. If it has been automatically annotated by an OCR engine (and thus may have unlabeled text), only the recognizer branch is trained. Thus the total multitask loss is defined as:

$$L = \delta(L_{rpn} + \alpha L_{rcnn} + \beta L_{mask}) + \gamma L_{recog}. \quad (5)$$

Here, δ is 1 if the input is fully labeled, 0 otherwise. In our implementation, both α, β and γ are set to 1.0. Adding machine labeled, partially labeled data ensures that the recognizer "sees" enough text while preventing the detector from overfitting. For the machine labeled data, since the detection of all the text is not required, we could increase the confidence threshold to filter out noisy low confidence outputs.

The detection losses are the same as the original Mask R-CNN paper [20]. The recognizer loss L_{recog} is the cross entropy loss with label smoothing set to 0.9, as suggested by [57]. During training, the ground-truth boxes and masks are used for RoI cropping and RoI masking while the predicted ones are used at inference time. We also tried to use predicted bounding boxes and masks during training but found no improvement.

3.6. Implementation Details

The data used to train our model contains images from the training portion of popular public datasets, including SynthText, ICDAR15, COCO-Text, ICDAR-MLT and Total-Text. The number of images we used from each dataset are 200k, 1k, 17k, 7k and 1255 respectively. Besides public datasets, we also collected 30k images from the web and manually labeled each word, providing oriented rectangular bounding boxes and transcriptions. The number of fully labeled real images is too low to train a robust end-to-end OCR model. To solve this issue, as mentioned in Section 3.5, we run an existing OCR engine on one million images with text and use the predictions (oriented rectangles and transcriptions) as the partially labeled ground-truth. Our experiments (see Section 4.3) show this can significantly improve the end-to-end performance. To prevent the large volumes of synthetic and partially labeled data from dominating the training data, extensive data augmentations are applied to fully labeled real images. First, the shorter dimension of input image is resized from 480 to 800 pixels, then random rotation, random cropping and aspect ratio jittering are used.

We adopt a single-step training strategy. The backbone CNN is pre-trained on ImageNet; the detector and recognizer are jointly optimized using both fully and partially annotated images. Our ablation experiment (see Section 4.3) shows that this achieves better accuracy than a two-step training strategy, where we first use all the fully labeled data to train the detector and then jointly fine-tune the detector and recognizer using both fully and partially labeled data. We train our model with asynchronous SGD with momentum of 0.9. The initial learning rate depends on the backbone network, 10^{-3} for Inception-ResNet and 3×10^{-4} for ResNet-50. We reduce the learning rate by a factor of 3 every 2M iterations, with a total number of 8M iterations. During training, each GPU takes a single training sample per iteration, and 15 Tesla V100 GPUs are used. We implement the model using TensorFlow [1], the training process takes about three days to finish. In the recognizer, we use a single layer LSTM with 256 hidden units. Recurrent dropout [15] and layer normalization [2] are used to reduce overfitting. The total number of symbols are 79, which includes digits, upper and lower cases of English characters as well as several special characters.

Method	Detection			Method	End-to-End		
	P	R	F		S	W	G
SSTD [22]	80.23	73.86	76.91	Stradvision [30]	43.70	-	-
EAST [61]	83.27	78.33	80.72	TextProposals+DictNet [18, 27]	56.0	52.3	49.7
TextSnake [41]	84.9	80.4	82.6	HUST_MCLAB [51, 52]	67.86	-	-
RRD MS [34]	88	80	83.8	E2E-MLT [6]	-	-	55.1
Mask TextSpotter [42]	91.6	81.0	86.0	Mask TextSpotter [42]	79.3	73.0	62.4
TextNet [54]	89.42	85.41	87.37	TextNet [54]	78.66	74.90	60.45
He *et al*. [24]	87	86	87	He *et al*. [24]	82	77	63
FOTS [37]	91.0	85.17	87.99	FOTS [37]	81.09	75.90	60.80
FOTS MS [37]	**91.85**	87.92	**89.84**	FOTS MS [37]	83.55	79.11	65.33
Ours (ResNet-50)	89.36	85.75	87.52	Ours (ResNet-50)	83.38	79.94	67.98
Ours (Inception-ResNet)	91.67	**87.96**	89.78	Ours (Inception-ResNet)	**85.51**	**81.91**	**69.94**

Table 1. Comparison on ICDAR15. "MS" represents multi-scale testing. "P", "R" and "F" stand for precision, recall, and F-score respectively. In the end-to-end evaluation, F-score under three lexicon settings are shown. "S" (strong) means 100 words, including the ground-truth, are given for each image. For "W" (weak), a lexicon includes all the words appeared in the test set is provided. For "G", a generic lexicon with 90k words is given, which is not used by our model.

4. Experiments

We evaluate the performance of our model on the ICDAR15 benchmark [30] (straight text) and recently introduced Total-Text [11] (curved text) dataset.

4.1. Straight Text

We show the superior performance of our model on detecting and recognizing oriented straight text using the ICDAR15 benchmark introduced in Challenge 4 of the ICDAR 2015 Robust Reading Competition. The dataset consists of 1000 training images and 500 testing images. Images in this dataset are captured by wearable cameras, without intentional focus on text regions. There are large variations in text size, orientation, font, and lighting conditions. Motion blur is also common. In this dataset, text instances are labeled at the word level. Quadrilateral bounding boxes and transcriptions are provided. For detection evaluation, a prediction is counted as a true positive if the IoU with the closest ground-truth is larger than 0.5. For end-to-end evaluation, the predicted transcription needs to be identical to the corresponding ground-truth in order to be considered as a true positive. Some unreadable words are marked as "do not care". The Evaluation metrics of interest are precision (true positives count over detection count), recall (true positives count over ground-truth count), and F-score (harmonic mean of precision and recall).

The results are summarized in Table 1. At inference time, the shorter dimension of the image is resized to 900 pixels. Note we only use a single scale input. In the detection only task, our method (with Inception-ResNet backbone) surpasses the best single scale model (FOTS) by 1.8%. For end-to-end performance, our method outperforms the highest single scale model (He *et al*.) by about 7%. Compared to the multi-scale version of the FOTS model, the current state-of-the-art, our method matches the detection performance while still achieving 4.6% higher end-to-end F-score.

Method	Detection			E2E
	P	R	F	None
Baseline [11]	40.0	33.0	36.0	-
Textboxes [33]	62.1	45.5	52.5	36.3
TextSnake [41]	82.7	74.5	78.4	-
MSR [60]	85.2	73.0	78.6	-
TextField [59]	81.2	79.9	80.6	-
FTSN [12]	84.7	78.0	81.3	-
Mask TextSpotter [42]	69.0	55.0	61.3	52.9
TextNet [54]	68.21	59.45	63.53	54.0
Ours (ResNet-50)	83.3	83.4	83.3	67.8
Ours (Inc-Res public)	86.8	84.3	85.5	63.9
Ours (Inc-Res)	**87.8**	**85.0**	**86.4**	**70.7**

Table 2. Results on Total-Text. No lexicon is used in end-to-end evaluation. "Inc-Res" stands for Inception-Resnet. "Inc-Res public" represents our model with Inception-ResNet backbone, trained using only public datasets.

4.2. Curved Text

The biggest advantage of our method is the outstanding performance on irregular shaped text. We conducted an experiment on the recently introduced curved text dataset called Total-Text [11]. Total-Text contains 1255 images for training and another 300 for testing, with a large number of curved text. In each image, text is annotated at word level, each word is labeled by a bounding polygon. Ground truth transcriptions are provided. The evaluation protocol for detection is based on [58], the one for end-to-end recognition is based on ICDAR15's end-to-end evaluation protocol (the evaluation script is modified to support general polygons).

Figure 4. Qualitative results of our method on ICDAR15 (first two columns) and Total-Text (last two columns) datasets. In the bottom right image, prediction errors are shown in blue, some predictions are skipped for better visualization. All the skipped predictions are correctly predicted by our method.

During training, we pre-train the model on straight text and fine-tune it using *only images from the training portion of the Total-Text dataset*. At inference time, the shorter dimension of each image is resized to 600 pixels. We compare the results of our model with both backbones against previous work in Table 2. We also list results *trained using only publicly available datasets* for the Inception-ResNet backbone. Our method out-performs the previous state-of-the-art by a large margin in both detection and end-to-end evaluations. Specifically, for detection, our best model surpasses the previous highest by 5.1%. In the end-to-end recognition task, our best model significantly raises the bar by 16.7%. In the absence of our internal fully labeled data and partially machine annotated data, our method still achieves far better performance in both the detection and recognition tasks, at +4.2% and +9.9% respectively.

Several qualitative examples are shown in Figure 4 (third and fourth column). Our method produces high quality bounding polygons and transcriptions. Surprisingly, we find that our method can also produce reasonable predictions in partially occluded cases (top right image, "ANTIONE") by utilizing visible image features and the learned implicit language model from the LSTM. In the bottom right image, we show some failure cases, where the text is upside down and read from right to left. These cases are very rare in the training data, we believe more aggressive data augmentation may mitigate these issues.

We can visualize the attention weight vector at each step by reshaping it to 2D and projecting back to image coordinates. This provides a great tool to analyze and debug the model performance. In Figure 5, we find that the seq2seq model focuses on the right area when decoding each symbol and is able to follow the shape of the text. In the middle row (last image), we show the attention mask corresponding to the *EOS* symbol. The attention is spread across both the start and end positions.

4.3. Ablation Experiments

We conduct a series of ablation experiments to better understand our end-to-end OCR model. In this section, we report *average precision* (AP) score, which is often a better evaluation metric than F-score (which display sensitivity to a specific threshold). We use the ICDAR15 test set in this section. Table 3 summarizes the experimental results.

Baselines: We build a *detection-only baseline* (first row in Table 3) and an *end-to-end baseline* (third row in Table 3). In the detection-only baseline, we train a model with only the detection branch. In the end-to-end baseline, we train a model with both detection and recognition branches, but do not use partially labeled data or RoI masking, and adopt a single-step training strategy (described in Section 3.6). The end-to-end baseline exhibits stronger detection results than the detection-only baseline (with a ResNet-50 backbone, the improvement is 1.6%) despite being trained on exactly the same data. This suggests that training a recognizer improves the feature extractor for the detection task.

Backbones: From Table 3 we find that on the detection task, the more powerful Inception-ResNet backbone consistently out-performs ResNet-50. On the end-to-end task, our model with the ResNet-50 backbone actually achieves better performance when the training data is limited (without large scale partially labeled data). For our *full end-to-end model*, the Inception-ResNet backbone achieves marginal improvement on end-to-end AP score (59.5% *vs*. 59.0%).

Partially Labeled Data: The use of partially labeled data provides significant improvements in end-to-end performance across all configurations of our model (row 4 *vs*. row 6, or row 3 *vs*. row 5). Interestingly, *it also improves the de-*

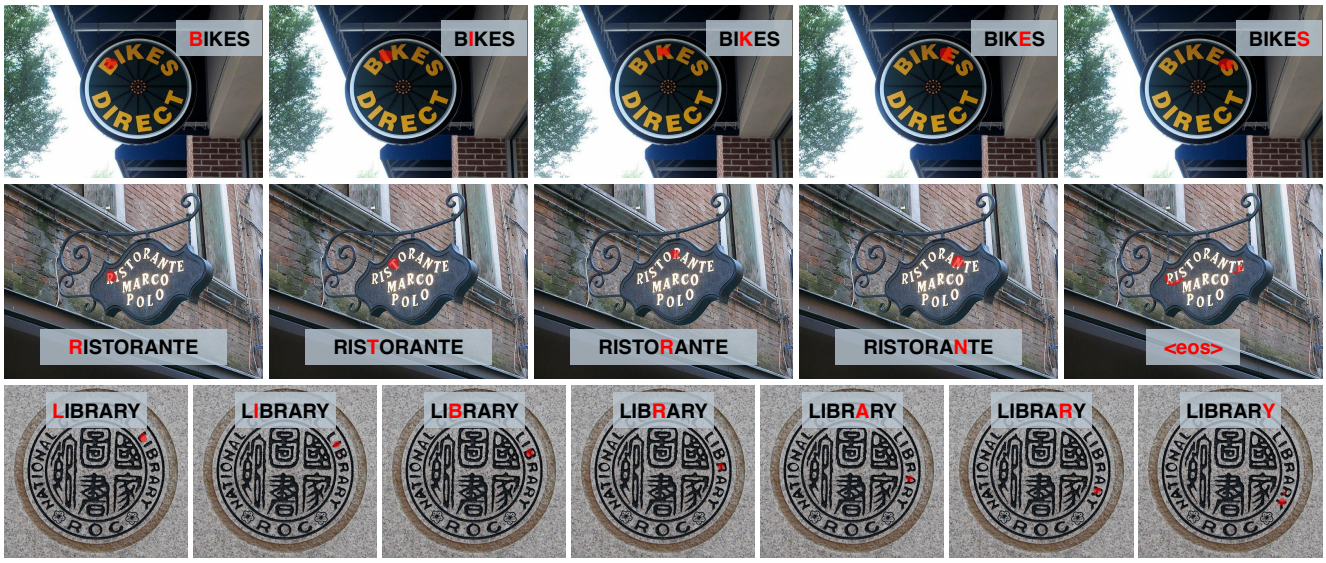

Figure 5. Visualization of the attention weights. Some steps are skipped for better visualization.

		PD	Mask	ResNet-50		Inc-Res	
				AP_{Det}	AP_{E2E}	AP_{Det}	AP_{E2E}
Two-step	Det-baseline			85.5	-	88.2	-
	E2E-full	✓	✓	86.9	55.3	89.1	57.4
Single-step	E2E-baseline			87.1	52.8	88.2	51.7
	+ Mask		✓	86.7	53.9	88.9	53.1
	+ PD	✓		**87.5**	55.7	89.9	58.7
	E2E-full	✓	✓	87.2	**59.0**	**90.8**	**59.5**

Table 3. Results on the ICDAR15 test set under different model configurations and training strategies. AP numbers are reported. "PD", "Mask" and "Inc-Res" stand for partially labeled data, RoI masking and Inception-ResNet respectively. "Det-baseline" refers to the first step (training the detector using fully labeled data) of the "two-step" training process.

tector without training the detection branch directly (Section 3.5). Once again, this suggests that we can improve the feature extractor by receiving training signal through recognition branch.

RoI Masking: In Table 3, we show the effectiveness of RoI masking (row 3 *vs*. row 4, or row 5 *vs*. row 6). Higher end-to-end AP scores are consistently achieved in the presence of RoI masking (*e.g*. +3.3% AP with the ResNet-50 backbone when using partially labeled data). This demonstrates that the recognizer benefits from RoI masking. The improvement is more significant for a lighter weight backbone with smaller receptive field. For detection performance, we observe mixed results: a marginal improvement for the Inception-ResNet backbone, and some degradation when using ResNet-50.

Training Strategy: Row 2 and row 6 of Table 3 compare the effect of single-step and two-step training strategies as described in Section 3.6. In single-step training, we jointly optimize the detector and recognizer together using both fully and partially labeled data. In two-step training, we first train the detection-only baseline, and then add a recognizer into joint training. We find that single-step training consistently out-performs two-step training in both detection and end-to-end evaluations. Single step training is far simpler, and makes it easier to apply automatic hyperparameter tuning and neural architecture search, which we will study in future work.

4.4. Speed

For images from the ICDAR15 dataset (with resolution 1280×720), the end-to-end inference time is 210 ms for the ResNet-50 backbone and 330 ms for the Inception-ResNet backbone (on a Tesla V100 GPU). If we only run the detection branch, the corresponding inference time are 180 ms and 270 ms respectively. Thus, for scene text images, the computational overhead of the recognition branch is quite small. Sharing the same CNN feature extractor makes end-to-end model more computationally efficient than two-step methods.

5. Conclusion

In this paper, we present an end-to-end trainable network that can simultaneously detect and recognize text in arbitrary shape. The use of Mask R-CNN, attention decoder and a simple yet effective RoI masking step leads to a flexible and high performance model. We also show that end-to-end training can benefit from partially machine annotated data. On the ICDAR15 and Total-Text benchmarks, our method significantly surpasses previous methods by a large margin while being reasonably efficient.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Alessandro Bissacco, Mark Cummins, Yuval Netzer, and Hartmut Neven. Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 785–792, 2013.

[5] Michal Busta, Lukas Neumann, and Jiri Matas. Fastext: Efficient unconstrained scene text detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1206–1214, 2015.

[6] Michal Bušta, Yash Patel, and Jiri Matas. E2e-mlt-an unconstrained end-to-end method for multi-language scene text. *arXiv preprint arXiv:1801.09919*, 2018.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[8] Xiangrong Chen and Alan 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. IEEE, 2004.

[9] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5076–5084, 2017.

[10] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5571–5579, 2018.

[11] 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.

[12] Yuchen Dai, Zheng Huang, Yuting Gao, Youxuan Xu, Kai Chen, Jie Guo, and Weidong Qiu. Fused text segmentation networks for multi-oriented scene text detection. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3604–3609. IEEE, 2018.

[13] Boris Epshtein, Eyal Ofek, and Yonatan Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970. IEEE, 2010.

[14] Yasuhisa Fujii, Karel Driesen, Jonathan Baccash, Ash Hurst, and Ashok C Popat. Sequence-to-label script identification for multilingual ocr. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 161–168. IEEE, 2017.

[15] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.

[16] Suman K Ghosh, Ernest Valveny, and Andrew D Bagdanov. Visual attention models for scene text recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 943–948. IEEE, 2017.

[17] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[18] Lluís Gómez and Dimosthenis Karatzas. Textproposals: a text-specific selective search algorithm for word spotting in the wild. *Pattern Recognition*, 70:60–74, 2017.

[19] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3047–3055, 2017.

[23] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[24] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5020–5029, 2018.

[25] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[26] Weilin Huang, Zhe Lin, Jianchao Yang, and Jue Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1241–1248, 2013.

[27] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

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

[29] Yingying Jiang, Xiangyu Zhu, Xiaobing Wang, Shuli Yang, Wei Li, Hua Wang, Pei Fu, and Zhenbo Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017.

[30] 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.

[31] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2231–2239, 2016.

[32] Hui Li, Peng Wang, and Chunhua Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5238–5246, 2017.

[33] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[34] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5909–5918, 2018.

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[36] 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.

[37] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018.

[38] Yuliang Liu and Lianwen Jin. Deep matching prior network: Toward tighter multi-oriented text detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1962–1969, 2017.

[39] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[40] Shangbang Long, Xin He, and Cong Ya. Scene text detection and recognition: The deep learning era. *arXiv preprint arXiv:1811.04256*, 2018.

[41] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 20–36, 2018.

[42] 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)*, pages 67–83, 2018.

[43] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 20(11):3111–3122, 2018.

[44] Leo Neat, Ren Peng, Siyang Qin, and Roberto Manduchi. Scene text access: A comparison of mobile ocr modalities for blind users. 2019.

[45] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3538–3545. IEEE, 2012.

[46] Lukas Neumann and Jiri Matas. Scene text localization and recognition with oriented stroke detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 97–104, 2013.

[47] Siyang Qin and Roberto Manduchi. A fast and robust text spotter. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[48] Siyang Qin and Roberto Manduchi. Cascaded segmentation-detection networks for word-level text spotting. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1275–1282. IEEE, 2017.

[49] 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.

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

[51] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2550–2558, 2017.

[52] 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, 2017.

[53] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4168–4176, 2016.

[54] Yipeng Sun, Chengquan Zhang, Zuming Huang, Jiaming Liu, Junyu Han, and Errui Ding. Textnet: Irregular text reading from images with an end-to-end trainable network. *arXiv preprint arXiv:1812.09900*, 2018.

[55] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[56] Jake Walker, Yasuhisa Fujii, and Ashok C Popat. A web-based ocr service for documents. In *Proceedings of the 13th IAPR International Workshop on Document Analysis Systems (DAS), Vienna, Austria*, 2018.

[57] Zbigniew Wojna, Alexander N Gorban, Dar-Shyang Lee, Kevin Murphy, Qian Yu, Yeqing Li, and Julian Ibarz. Attention-based extraction of structured information from street view imagery. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 844–850. IEEE, 2017.

[58] Christian Wolf and Jean-Michel Jolion. Object count/area graphs for the evaluation of object detection and segmentation algorithms. *International Journal of Document Analysis and Recognition (IJDAR)*, 8(4):280–296, 2006.

[59] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 2019.

[60] Chuhui Xue, Shijian Lu, and Wei Zhang. Msr: Multi-scale shape regression for scene text detection. *arXiv preprint arXiv:1901.02596*, 2019.

[61] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[62] Siyu Zhu and Richard Zanibbi. A text detection system for natural scenes with convolutional feature learning and cascaded classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 625–632, 2016.