

Convolutional Character Networks

Linjie Xing^{1,2}, Zhi Tian³, Weilin Huang^{*1,2}, and Matthew R. Scott^{1,2}

¹Malong Technologies, Shenzhen, China

²Shenzhen Malong Artificial Intelligence Research Center, Shenzhen, China

³University of Adelaide, Australia

Abstract

Recent progress has been made on developing a unified framework for joint text detection and recognition in natural images, but existing joint models were mostly built on two-stage framework by involving ROI pooling, which can degrade the performance on recognition task. In this work, we propose convolutional character networks, referred as CharNet, which is an one-stage model that can process two tasks simultaneously in one pass. CharNet directly outputs bounding boxes of words and characters, with corresponding character labels. We utilize character as basic element, allowing us to overcome the main difficulty of existing approaches that attempted to optimize text detection jointly with a RNN-based recognition branch. In addition, we develop an iterative character detection approach able to transform the ability of character detection learned from synthetic data to real-world images. These technical improvements result in a simple, compact, yet powerful one-stage model that works reliably on multi-orientation and curved text. We evaluate CharNet on three standard benchmarks, where it consistently outperforms the state-of-the-art approaches [25, 24] by a large margin, e.g., with improvements of 65.33%→71.08% (with generic lexicon) on ICDAR 2015, and 54.0%→69.23% on Total-Text, on end-to-end text recognition. Code is available at: <https://github.com/MalongTech/research-charnet>.

1. Introduction

Text reading in natural images has long been considered as two separate tasks: text detection and recognition, which are implemented sequentially. The two tasks have been advanced individually by the success of deep neural networks. Text detection aims to predict a bounding box for each text instance (e.g., typically a word) in natural images, and cur-



Figure 1: The proposed CharNet can directly output bounding boxes of words and characters, with corresponding character labels in one pass.

rent leading approaches are mainly extended from object detection or segmentation frameworks, such as [25, 41, 24]. Built on text detection, the goal of text recognition is to recognize a sequence of character labels from a cropped image patch including a text instance. Generally, it can be cast into a sequence labeling problem, where various recurrent models with CNN-extracted features have been developed, with state-of-the-art performance achieved [33, 4, 31, 10].

However, the two-step pipeline often suffers from a number of limitations. First, learning the two tasks independently would result in a sub-optimization problem, making it difficult to fully explore the potential of text nature. For example, text detection and recognition can work collaboratively by providing strong context and complementary information to each other, which is critical to improving the performance, as substantiated by recent work [12, 24]. Second, it often requires to implement multiple sequential steps, resulting in a relatively complicated system, where the performance of text recognition is heavily relied on text detection results.

陷入局部最优

* Corresponding author: whuang@malong.com.

Recent effort has been devoted to developing a unified framework that implements text detection and recognition simultaneously [12, 24, 25]. For example, in [12] and [24], text detection models were extended to joint detection and recognition, by adding a new RNN-based branch for recognition, leading to the state-of-the-art performance on end-to-end (E2E) text recognition. These approaches can achieve joint detection and recognition using a single model, but they are in the family of two-stage framework and thus have the following limitations. Firstly, the recognition branch often explores a RNN-based sequential model, which is difficult to optimize jointly with the detection task, by requiring a significantly larger amount of training samples. Thus the performance is heavily depended on a well-designed but complicated training scheme (e.g., [12] and [20]). This is the central issue that impedes the development of a united framework. Secondly, current two-stage framework commonly involves RoI cropping and pooling, making it difficult to crop an accurate text region for feature pooling, where a large amount of background information may be included. This inevitably leads to significant performance degradation on recognition task, particularly for multi-orientation or curved text.

To overcome the limitations of RoI cropping and pooling for two-stage framework, He *et al.* [12] proposed a text-alignment layer to precisely compute the convolutional features for a text instance of arbitrary orientation, which boosted the performance. In [24], multiple affinity transformations were applied to the convolutional features for enhancing text information in the RoI regions. However, these methods failed to work on curved text. In addition, many high-performance models consider words (for English) as detection units, but word-level detection often requires to cast text recognition into a sequence labelling problem, where a RNN model with additional modules, such as CTC [6, 11, 32] or attention mechanism [33, 4, 1, 12], was applied. Unlike English, words are not clearly distinguishable in some languages such as Chinese, where text instances can be defined and separated more clearly by characters. Therefore, characters are more clearly-defined elements that generalize better over various languages. Importantly, character recognition is straightforward, and can be implemented with a simple CNN model, rather than using a RNN-based sequential model.

Contributions. In this work, we present Convolutional Character Networks (referred as CharNet) for joint text detection and recognition, by leveraging character as basic unit. Moreover, for the first time, we provide an one-stage CNN model for the joint tasks, with significant performance improvements over the state-of-the-art results achieved by a more complex two-stage framework, such as [12], [25] and [24]. The proposed CharNet implements direct character detection and recognition, jointly with text instance (e.g.,

word) detection. This allows it to avoid the RNN-based word recognition, resulting in a simple, compact, yet powerful model that directly outputs bounding boxes of words and characters, as well as the corresponding character labels, as shown in Fig.1. Our main contributions are summarized as follows.

Firstly, we propose an one-stage CharNet for joint text detection and recognition, where a new branch for direct character detection and recognition is introduced, and can be integrated seamlessly into existing text detection framework. We explore character as basic unit, which allows us to overcome the main limitations of current two-stage framework using RoI pooling with RNN-based recognition.

Secondly, we develop an iterative character detection method which allows CharNet to transform the character detection capability learned from synthetic data to real-world images. This makes it possible for training CharNet on real-world images, without providing additional character-level bounding boxes.

Thirdly, CharNet consistently outperforms recent two-stage approaches such as [12, 25, 24, 35] by a large margin, with improvements of 65.33%→71.08% (generic lexicon) on ICDAR 2015, and 54.0%→69.23% (E2E) on Total-Text. Particularly, it can achieve comparable results, e.g., 67.24% on ICDAR 2015, even by completely removing a lexicon.

2. Related Work

Traditional approaches often regard text detection and recognition as two separate tasks that process sequentially [15, 37, 36, 41, 10, 32]. Recent progress has been made on developing a unified framework for joint text detection and recognition [12, 24, 25]. We briefly review the related studies on text detection, recognition and join of two tasks.

Text detection. Recent approaches for text detection were mainly built on general object detectors with various text-specific modifications. For instance, by building on Region Proposal Networks [29], Tian *et al.* [36] proposed a Connectionist Text Proposal Network (CTPN) to explore the sequence nature of text, and detect a text instance in a sequence of fine-scale text proposals. Similarly, Shi *et al.* [30] developed a method with linking segment which also localizes a text instance in a sequence, with the capability for detecting multi-oriented text. In [41], EAST was introduced by exploring IOU loss [39] to detect multi-oriented text instances (e.g., words), with impressive results achieved. Recently, a single-shot text detector (SSTD) [9] was proposed by extending SSD object detector [22] to text detection. SSTD encodes text regional attention into convolutional features to enhance text information.

Text recognition. Inspired from speech recognition, recent work on text recognition commonly cast it into a sequence-to-sequence recognition problem, where recurrent neural networks (RNNs) were employed. For exam-

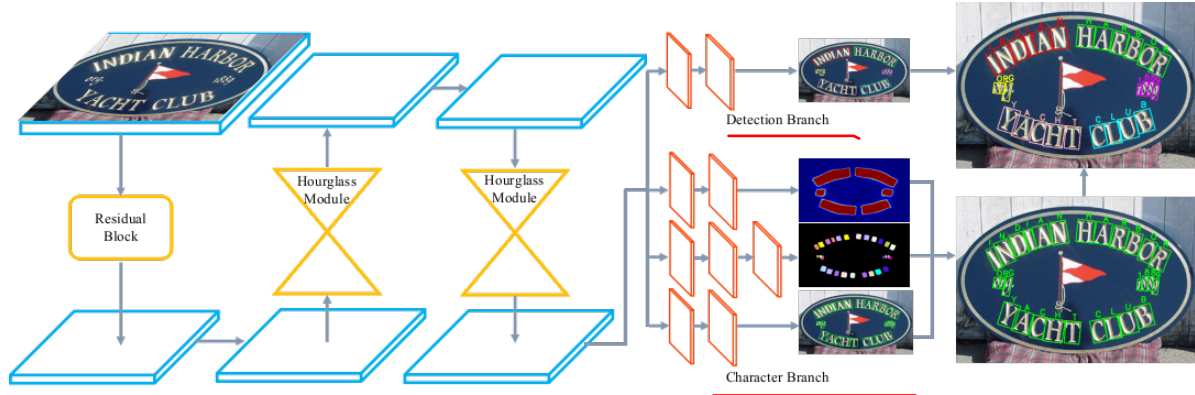


Figure 2: Overview of the proposed CharNet, which contains two branches working in parallel: a character branch for direct character detection and recognition, and a detection branch for text instance detection.

ple, He *et al.* [10] exploited convolution neural networks (CNNs) to encode a raw input image into a sequence of deep features, and then a RNN is applied to the sequential features for decoding and yielding confidence maps, where connectionist temporal classification CTC [6] is applied to generate final results. Shi *et al.* [31] improved such CNN+RNN+CTC framework by making it end-to-end trainable, with significant performance gain obtained. Recently, the framework was further improved by introducing various attention mechanisms, which are able to encode more character information explicitly or implicitly [33, 4, 1, 12].

End-to-end (E2E) text recognition. Recent work attempted to integrate text detection and recognition into a unified framework for E2E text recognition. Li *et al.* [20] drew inspiration from Faster R-CNN [29] and employed RoI pooling to obtain text features from a detection framework for further recognition. In [12], He *et al.* proposed an E2E framework by introducing a new text-alignment layer with character attention mechanism, leading to significant performance improvements by jointly training two tasks. Similar framework has been developed by Liu *et al.* in [24]. Both works have achieved strong performance on E2E text recognition, but they were built on two-stage models implementing ROI cropping and pooling operations, which may reduce the performance, particularly on the recognition task for multi-orientation or curved text.

Our work is related to character-based approaches for text detection or recognition. Hu *et al.* proposed a WordSup able to detect text instances at the character level [14], while Liu *et al.* [23] developed a character-aware neural network for distorted scene text recognition. However, they did not provide a full solution for E2E text recognition. The most closely related work is that of Mask TextSpotter [25] which is a two-stage character-based framework for E2E recognition, built on recent Mask R-CNN [9]. However, our CharNet has a number of clear distinctions: (1) CharNet is

the first one-stage model for E2E text recognition, which is different from the two-stage Mask TextSpotter, where RoI cropping and pooling operations are required; (2) CharNet has a character branch that directly outputs accurate char-level bounding boxes. This enables it to automatically identify characters, allowing it to work in a weakly-supervised manner by using the proposed iterative character detection; (3) This results in a distinct capability for training CharNet without additional char-level bounding boxes in real-world images, while Mask TextSpotter requires full char-level annotations which are often highly expensive; (4) CharNet achieved consistent and significant performance improvements over Mask TextSpotter, as shown in Table 4 and 5.

3. Convolutional Character Networks

In this section, we describe the proposed CharNet in details. Then an iterative character detection method is introduced for automatically identifying characters with bounding boxes from real-world images, by leveraging synthetic data. In this work, we use “text instance” as a higher level concept for text, which can be a word or a text-line, with multi-orientation or curved shape.

3.1. Overview

As discussed, existing approaches for E2E text recognition are commonly limited by using RoI cropping and pooling, with a RNN-based sequential model for word recognition. The proposed CharNet is an one-stage convolutional architecture consisting of two branches: (1) a character branch designed for direct character detection and recognition, and (2) a text detection branch predicting a bounding box for each text instance in an image. The two branches are implemented in parallel, which form an one-stage model for joint text detection and recognition, as shown in Fig. 2. The character branch can be integrated seamlessly into an one-stage text detection framework, resulting in an end-to-end

trainable model. Training the model requires both instance-level and char-level bounding boxes with character labels as supervised information. In inference, CharNet can directly output both instance-level and char-level bounding boxes with corresponding character labels in one pass.

Many existing text databases often do not include char-level annotations which are highly expensive to obtain. We develop an iterative learning approach for automatic character detection, which allows us to learn a character detector from synthetic data where full char-level annotations can be generated unlimitedly. Then the learned character detection capability can be transformed and adapted gradually to real-world images. This enables the model with ability to automatically identify characters in real-world images, providing a weakly-supervision learning manner for CharNet.

Backbone networks. We employ ResNet-50 [8] and Hourglass [19] networks as backbone for our CharNet framework. For ResNet-50, we follow [41], and make use of the convolutional feature maps with $4\times$ down-sampling ratio as the final convolutional maps to implement text detection and recognition. This results in high-resolution feature maps that enable CharNet to identify extremely small-scale text instances. For Hourglass networks, we stack two hourglass modules, as shown in Fig. 2, and the final feature maps are up-sampled to $\frac{1}{4}$ resolution of the input image. In this work, we use two variants of Hourglass networks, Hourglass-88 and Hourglass-57. Hourglass-88 is modified from Hourglass-104 in [19] by removing two down-sampling stages and reducing the number of layers in the last stage of each hourglass module by half. Hourglass-57 is constructed by further removing half number of layers in each stage of hourglass modules. Notice that, for both variants, we do not employ the intermediate supervision as did in CornerNet [19].

3.2. Character Branch

Existing RNN-based recognition methods were commonly built on word-level optimization with a sequential model, which has a significantly larger search space than direct character classification. This inevitably makes the models more complicated and difficult to train by requiring a significantly longer training time with a large amount of training samples. Recent work, such as [33, 4, 12], had shown that the performance of RNN-based methods can be improved considerably by introducing char-level attention mechanism which is able to encode strong character information implicitly or explicitly. This enables the models to have the ability to identify characters more accurately, and essentially adds additional constraints to the models which in turn reduce the search space, leading to performance boost. *This suggests that precise identification of characters is of great importance to RNN-based text recognition, which inspired the current work to simplify it into direct character*

recognition with an automatic character localization mechanism, resulting in a simple yet powerful one-stage fully convolutional model for E2E text recognition.

To this end, we introduce a new character branch which has the functions of direct character detection and recognition. The character branch uses character as basic unit for detection and recognition, and outputs char-level bounding boxes as well as the corresponding character labels. Specifically, the character branch is a stack of convolutional layers, which move densely over the final feature maps of the backbone. It has the input features maps with $\frac{1}{4}$ spatial resolution of the input image. This branch contains three sub-branches, for text instance segmentation, character detection and character recognition, respectively. The text instance segmentation sub-branch and character detection sub-branch have three convolutional layers with filter sizes of 3×3 , 3×3 and 1×1 , respectively. The character recognition sub-branch has four convolutional layers with one more 3×3 convolutional layer.

Text instance segmentation sub-branch exploits a binary mask as supervision, and outputs 2-channel feature maps indicating text or non-text probability at each spatial location. Character detection sub-branch outputs 5-channel feature maps, estimating a character bounding box at each spatial location. By following EAST [41], each character bounding box is parameterized by five parameters, indicating the distances of current location to the top, bottom, left and right sides of the bounding box, as well as the orientation of bounding-box. In character recognition sub-branch, character labels are predicted densely over the input feature maps, generating 68-channel probability maps. Each channel is a probability map for a specific character class among 68 character classes, including 26 English characters, 10 digital numbers and 32 special symbols. All of the output feature maps from three sub-branches have the same spatial resolution, which is exactly the same as that of the input feature maps ($\frac{1}{4}$ of the input image). Finally, the char-level bounding boxes are generated by keeping the bounding boxes having a confident value over 0.95. Each generated bounding box has a corresponding character label, which is computed at the corresponding spatial location from the 68-channel classification maps - by using the maximum of the computed softmax scores.

Training character branch requires char-level bounding boxes with the corresponding character labels. Compared to word-level annotations, acquiring char-level labels with bounding boxes is much more expensive and would significantly increase labor cost. To avoid such additional cost, we develop an iterative character detection mechanism which is described in Section 3.4.

3.3. Text Detection Branch

Text detection branch is designed to identify text instances at a higher level concept, such as words or text-lines. It provides strong context information which is used to group the detected characters into text instances. Because directly grouping characters by using characters information (e.g., character locations or geometric features) is heuristic and complicated when multiple text instances are located closely within a region, particularly for text instances with multiple orientations or in a curved shape. Our text detection branch can be defined in different forms subjected to the type of text instances, and existing instance-level text detectors can be adapted with minimum modification. We take text detectors for multi-orientation words or curved text-lines as examples.

Multi-Orientation Text. We simply modify EAST detector [41] as our text detection branch, which contains two sub-branches for text instance segmentation and instance-level bounding box regression using IoU loss. The predicted bounding boxes are parameterized by five parameters including 4 scalars for a bounding box with an orientation angle. We compute dense prediction at each spatial location of the feature maps by using two 3×3 convolutional layers, followed by another 1×1 convolutional layer. Finally, the text detection branch outputs 2-channel feature maps indicating text or non-text probability, and 5-channel detection maps for bounding boxes with orientation angles. We keep the bounding boxes having a confident value over 0.95.

Curved Text. For curved text, we modify Textfield in [38] by using a direction field, which encodes the direction information that points away from text boundary. The direction field is used to separate adjacent text instances, and can be predicted by a new branch in parallel with text detection branch and character branch. This branch is composed of two 3×3 convolutional layers, followed by another 1×1 convolutional layer.

Generation of Final Results. The predicted instance-level bounding boxes are applied to group the generated characters into text instances. We make use of a simple rule, by assigning a character to a text instance if the character bounding box have an overlap (e.g., with > 0 IoU) with an instance-level bounding box. The final outputs of our CharNet are bounding boxes of both text instances and characters, with the corresponding character labels.

3.4. Iterative Character Detection

Training our model requires both char-level and word-level bounding boxes as well as the corresponding character labels. However, char-level bounding boxes are expensive to obtain and are not available in many existing benchmark datasets such as ICDAR 2015 [17] and Total-Text [5]. We develop an iterative character detection method that enables our model to have capability for identifying charac-

Method	w/ Real.	Detection	E2E
CharNet R-50		65.38	33.69
CharNet R-50	✓	89.70	62.18
CharNet H-57		65.19	39.43
CharNet H-57	✓	89.66	66.92
CharNet H-88		65.11	39.94
CharNet H-88	✓	90.97	69.14

Table 1: Performance of CharNet with various backbone networks on ICDAR 2015. “Real.” denotes “CharNet trained on real-world images with the proposed iterative character detection”. Detection is compared by using F-measure.

ters by leveraging synthetic data, such as Synth800k [7], where multi-level supervised information can be generated unlimitedly. This allows us to train CharNet in a weakly-supervised manner by just using instance-level annotations from real-world images.

A straightforward approach is to train our model directly with synthetic images, and then run inference on real-world images. However, it has a large domain gap between the synthetic images and real ones, and therefore the model trained from synthetic images is difficult to work directly on the real-world ones, as shown in Table 1, where low performance is obtained on both text detection and E2E recognition. We observed that a text detector has relatively stronger generalization capability than a text recognizer. As shown in [36], a text detector trained solely on English and Chinese data can work reasonably on other languages, which inspired us to explore the generalization ability of a character detector to bridge the gap between the two domains.

Our intuition is to gradually improve the generalization capability of model which is initially trained from synthetic images where full char-level annotations are provided, and the key is to transform the capability of *character detection* learned from the synthetic data to real-world images. We develop an iterative process by gradually identifying the “correct” char-level bounding boxes from real-world images by the model itself. We make use of a simple rule that identifies a group of char-level bounding boxes as “correct” if *the number of character bounding boxes in a text instance is exactly equal to the number of character labels in the provided instance-level transcript*. Note that instance-level transcripts (e.g., words) are often provided in existing datasets for E2E text recognition. The proposed iterative character detection are described as follows.

- (i) We first train an initial model on synthetic data, where both char-level and instance-level annotations are available to our CharNet. Then we apply the trained model to the training images from a real-world dataset, where char-level bounding boxes are predicted by the learned model.
- (ii) We explore the aforementioned rule to collect the



Figure 3: Character bounding boxes generated at 4 interactive steps from left to right. Red boxes indicate the identified “correct” ones by our rule, while blue boxes mean invalid ones, which are not collected for training in next step.

“correct” char-level bounding boxes detected in real-world images, which are used to further train the model with the corresponding transcripts provided. Note that we do not use the predicted character labels, which are not fully correct and would reduce the performance in our experiments.

- (iii) This process is implemented iteratively to enhance the model capability gradually for character detection, which in turn continuously improves the quality of the identified characters, with an increasing number of the “correct” char-level bounding boxes generated, as shown in Fig. 3 and Table 2.

4. Experiments, Results and Comparisons

Our CharNet is evaluated on three standard benchmarks: ICDAR 2015 [17], Total-Text [5], and ICDAR MLT 2017 [27]. **ICDAR 2015** includes 1,500 images collected by using Google Glasses. The training set has 1,000 images, and the remaining 500 images are used for evaluation. This dataset is challenging due to the presence of multi-orientated and very small-scale text instances. **Total-Text** consists of 1,555 images with a variety of text types including horizontal, multi-orientated, and curved text instances. The training split and testing split have 1,255 images and 300 images, respectively. **ICDAR MLT 2017** is a large-scale multi-lingual text dataset, which contains 7,200 training images, 1,800 validation images, and 9,000 testing images. 9 languages are included in total.

4.1. Implementation Details

Similar to recent work in [12, 24], our CharNet is trained on both synthetic data and real-world data. The proposed iterative character detection is implemented by using 4 iterative steps. At the first step, CharNet is trained on synthetic data, Synth800k [7], for 5 epochs, where both char-level and word-level annotations are available. We use a mini-batch

Step	# Words	Ratio (%)	E2E	# Epochs
0	6033	64.95	39.3	5
1	8262	88.94	62.9	100
2	8494	91.44	65.0	400
3	8606	92.65	66.1	800

Table 2: 4-step iterative character detection with CharNet. “# Words” is the number of words identified as “correct” at each step iterative learning. “Ratio” denotes the ratio of the “correct” words to all words in the training images from Total-Text. “# Epochs” indicates the number of training epochs for each iterative step. At the Step 0, CharNet is trained on synthetic data for 5 epochs, while Step 1-3 are implemented on real-world images. “E2E” means “End-to-End Recognition with F-measure”.

of 32 images, with 4 images per GPU. On the synthetic data, we set a base learning rate of 0.0002, which is reduced according to $lr_{base} \times (1 - \frac{iter}{max.iter})^{power}$ with $power = 0.9$, by following [3]. The remained three iterative steps are implemented on real-world data, by training CharNet for 100, 400 and 800 epochs respectively, on the training set of a benchmark provided, e.g., ICDAR 2015 [17] or Total-Text [5]. On the real-world data, we set a base learning rate of 0.002, and use the char-level bounding boxes generated by the model trained from the previous step. We make use of similar data augmentation as [24] and OHM [34].

4.2. On Iterative Character Detection

Interactive character detection is an important function for CharNet that allows us to train the model on real-world images by only using text instance-level annotations. Thus accurate identification of characters is critical to the performance of CharNet. We evaluate the iterative character detection with CharNet by using various backbone networks on ICDAR 2015. Results are reported in Table 1. As can be found, CharNet has low performance on both text detection and E2E recognition when we directly apply the model trained from synthetic data to testing images from ICDAR 2015, due to a large domain gap between the two data sets. The performance can be improved considerably by training CharNet on real-world data with iterative character detection, which demonstrates its efficiency.

We further investigate the capability of our model for identifying the “correct” characters in real-world images. Experiments were conducted on Total-Text. In this experiment, the “correct” characters are grouped into words, and we calculate the number of correctly-detected words at each iterative step. As shown in Table 2, at the step 0, when CharNet is only trained on synthetic data, only 64.95% words are identified as “correct” from real-world training images. Interestingly, this number increases immediately from 64.95% to 88.94% at the step 1, when the proposed iterative character detection is applied. This also leads to a significant performance improvement, from 39.3% to 62.9% on E2E text recognition. The iterative training con-



Figure 4: CharNet improves both recall and precision on text detection by jointly learning with character recognition.

tinues until the number of the identified words dose not increase further. Finally, our method is able to collect 92.65% correct words from real-world images by implementing 4 iterative steps in total. We argue that this number of char-level annotations learned automatically by model is enough to train our CharNet, as evidenced by the state-of-the-art performance obtained, which is shown next.

4.3. Results on Text Detection

We evaluate the performance of CharNet on text detection task. To make a fair comparison, we use the same backbone ResNet-50 as FOTS [24]. As shown in Table 3, our CharNet achieves comparable performance with FOTS when both methods are trained without recognition branch. By jointly optimizing the model with text recognition, CharNet improves the detection performance by 4.13%, from a F-measure of 85.57% to 89.70%, which is more significant than 2.68% performance gain achieved by FOTS. It suggests that our one-stage model allows text detection and recognition to work more effectively and collaboratively. This enables CharNet with higher capability for identifying extremely challenging text instances with stronger robustness which also reduces false detections, as shown in Fig. 4. In addition, CharNet also has a performance improvement of 87.00% \rightarrow 89.70% on F-measure over that of [12] which uses a PVAnet [18] as backbone with multi-scale implementation.

Moreover, our one-stage CharNet achieves new stage-of-the-art performance on text detection on all three benchmarks, which improves recent strong baseline (e.g., He *et al.* [12], FOTS [24] and TextFiled [38]) by a large margin. For example, on single-scale case, the improvements on F-measure are: 87.99% \rightarrow 90.97% on ICDAR 2015 (in Table 4), 80.3% \rightarrow 85.6% on the Total-Text for curved text (in Table 5), and 67.25% \rightarrow 75.77% on ICDAR 2017 MLT (in Table 6). Notice that CharNet is designed by using characters as basic unit. This natural property allows it to be easily adapted to curved text, where FOTS is difficult to work reliably. TextFiled was designed specifically for curved text but only has a F-measure of 82.4% on ICDAR 2015. Several examples for detecting challenging text instances are presented in Fig. 5.

Method	Rec.	R	P	F	Gain
He <i>et al.</i> [12]		83.00	84.00	83.00	-
He <i>et al.</i> [12]	✓	86.00	87.00	87.00	+4.00
FOTS [24]		82.04	88.84	85.31	-
FOTS [24]	✓	85.17	91.00	87.99	+2.68
CharNet		81.37	90.23	85.57	-
CharNet	✓	88.30	91.15	89.70	+4.13

Table 3: Detection performance on ICDAR 2015. ResNet-50 was used by both FOTS and CharNet as backbone, while PVAnet [18] was applied in [12]. “Rec.” denotes “Recognition”. “Gain” is the performance gain obtained by joint optimization with text recognition. “R”, “P”, “F” indicate “Recall”, “Precision”, “F-measure”.

4.4. Results on End-to-End Text Recognition

For E2E text recognition task, we compare our CharNet with recent state-of-the-art methods on ICDAR 2015 [17] and Total-Text [5].

ICDAR 2015. As shown in Table 4, by using a same backbone ResNet-50, our CharNet has comparable results with Mask TextSpotter [25]. However, Mask TextSpotter has significant performance improvements by using additional char-level manual annotations on real-world images, with a weighted edit distance applied to a lexicon, e.g., 76.1% \rightarrow 79.3% (S), 67.1% \rightarrow 73.0% (W) and 56.7% \rightarrow 62.4% (G) on E2E recognition. Furthermore, CharNet also outperforms FOTS by 1.38% in terms of generic lexicon. Unlike FOTS, which makes use of a heavy recognition branch with 6.31M parameters, our one-stage model only employs a light-weight CNN-based character branch with 1.19M parameters. Importantly, our model can work reliably without a lexicon, with performance of 60.72%, which is comparable to 60.72% of FOTS with a generic lexicon. These lexicon-free results demonstrate the strong capability of our CharNet, making it better applicable to real-world applications where a lexicon is not always available.

We further employ Hourglass-57 [19] as backbone, which has the similar number of model parameters compared to FOTS (34.96M v.s. 34.98M). As shown in Table 4, our CharNet outperforms FOTS by 6.12% with generic lexicon. With a more powerful Hourglass-88, we set a new state-of-the-art single-scale performance on the benchmark, and improve both Mask TextSpotter and FOTS considerably in all terms. Finally, with multi-scale inference, CharNet surpasses the previous best results [24] by a large margin, e.g., from 65.33% to 71.08% with generic lexicon.

Total-Text. We conduct experiments on Total-text to show that the capability of our CharNet on curved text. We employ the protocol described in [5] to evaluate the performance of text detection, and follow the evaluation protocol presented in [25] for E2E recognition. No lexicon is used

Method	Params	Detection			Method	End-to-End Recognition			
		R	P	F		S	W	G	N
Single Scale									
WordSup [14]	-	77.03	79.33	78.16	Neumann <i>et al.</i> [28]	35.00	20.00	16.00	-
EAST [41]	-	78.33	83.27	80.72	Deep text spotter [2]	54.00	51.00	47.00	-
R2CNN [16]	-	79.68	85.62	82.54	TextProp.+DictNet [13, 40]	53.30	49.61	47.18	-
Mask TextSpotter [25] *	-	81.00	91.60	86.00	Mask TextSpotter [25] *	79.30	73.00	62.40	-
FOTS R-50 [24]	34.98 M	85.17	91.00	87.99	FOTS R-50 [24]	81.09	75.90	60.80	-
CharNet R-50	26.48 M	88.30	91.15	89.70	CharNet R-50	80.14	74.45	62.18	60.72
CharNet H-57	34.96 M	88.88	90.45	89.66	CharNet H-57	81.43	77.62	66.92	62.79
CharNet H-88	89.21 M	89.99	91.98	90.97	CharNet H-88	83.10	79.15	69.14	65.73
Multi-Scale									
He <i>et al.</i> MS [12]	-	86.00	87.00	87.00	He <i>et al.</i> MS [12]	82.00	77.00	63.00	-
FOTS R-50 MS [24]	34.98 M	87.92	91.85	89.84	FOTS R-50 MS [24]	83.55	79.11	65.33	-
CharNet R-50 MS	26.48 M	90.90	89.44	90.16	CharNet R-50 MS	82.46	78.86	67.64	62.71
CharNet H-57 MS	34.96 M	91.43	88.74	90.06	CharNet H-57 MS	84.07	80.10	69.21	65.26
CharNet H-88 MS	89.21 M	90.47	92.65	91.55	CharNet H-88 MS	85.05	81.25	71.08	67.24

Table 4: Results on ICDAR 2015. “R-*” and “H-*” denote “ResNet-*” and “Hourglass-*”. “MS” means multi-scale inference. “R”, “P”, “F” are “Recall”, “Precision”, “F-measure”. “S”, “W”, “G” and “N” mean F-measure using “Strong”, “Weak”, “Generic” and “None” lexicon.

Method	Detection			E2E
	R	P	F	
Textboxes [21]	45.5	62.1	52.5	36.3
Mask TextSpotter [25]	55.0	69.0	61.3	52.9
TextNet [35]	59.5	68.2	63.5	54.0
TextFiled [38]	79.9	81.2	80.6	-
CharNet H-57	81.0	88.6	84.6	63.6
CharNet H-88	81.7	89.9	85.6	66.6
CharNet H-57 MS	85.0	87.3	86.1	66.2
CharNet H-88 MS	85.0	88.0	86.5	69.2

Table 5: Results on Total-Text. “H-*” denotes “Hourglass-*”. “MS” indicates multi-scale inference. “R”, “P”, “F” are “Recall”, “Precision”, “F-measure”. “E2E” is “End-to-End Recognition using F-measure”.

Method	R	P	F
SARLFDU_RPN [26]	55.50	71.17	62.37
SCUT_DLVClab	54.54	80.28	64.96
FOTS [24]	57.51	80.95	67.25
FOTS MS [24]	62.30	81.86	70.75
CharNet R-50	70.10	77.07	73.42
CharNet H-88	70.97	81.27	75.77

Table 6: Text detection on ICDAR 2017 MLT. “R-*” and “H-*” denote “ResNet-*” and “Hourglass-*”. “R”, “P” and “F” represent “Recall”, “Precision” and “F-measure”. “MS” indicates multi-scale inference.

in E2E recognition. As shown in Table 5, CharNet outperforms current state-of-the-art methods by 5.9% F-measure on text detection, and 15.2% on E2E recognition. Compared to character-based method, Mask TextSpotter [25], our CharNet can obtain even larger performance improvements on curved text.



Figure 5: Full results by CharNet.

5. Conclusions

We have presented an one-stage CharNet for E2E text recognition. We introduce a new branch for direct character recognition, which can be integrated seamlessly into text detection framework. This results in the first one-stage fully convolutional model that implements two tasks jointly, setting it apart from existing RNN-integrated two-stage framework. We demonstrate that with CharNet, the two tasks can be trained more effectively and collaboratively, leading to significant performance improvements. Furthermore, we develop an iterative character detection able to transfer the character detection capability learned from synthetic data to real-world images. In addition, CharNet is compact with less parameters, and can work reliably on curved text. Extensive experiments were conducted on ICDAR 2015, MTL 2017 and Total-text, where CharNet consistently outperforms existing approaches by a large margin.

References

- [1] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou. Edit probability for scene text recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1508–1516, 2018. 2, 3
- [2] M. Busta, L. Neumann, and J. 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. 8
- [3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. 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. 6
- [4] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5076–5084, 2017. 1, 2, 3, 4
- [5] C. K. Ch'ng and C. S. 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. 5, 6, 7
- [6] A. Graves, S. Fernández, F. Gomez, and J. 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. 2, 3
- [7] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016. 5, 6
- [8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4
- [9] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li. Single shot text detector with regional attention. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 6, 2017. 2, 3
- [10] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang. Reading scene text in deep convolutional sequences. In *AAAI*, volume 16, pages 3501–3508, 2016. 1, 2, 3
- [11] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang. Reading scene text in deep convolutional sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 2
- [12] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. 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. 1, 2, 3, 4, 6, 7, 8
- [13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 8
- [14] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding. Wordsup: Exploiting word annotations for character based text detection. In *Proc. ICCV*, 2017. 3, 8
- [15] W. Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 2
- [16] Y. Jiang, X. Zhu, X. Wang, S. Yang, W. Li, H. Wang, P. Fu, and Z. Luo. R2cnn: rotational region cnn for orientation robust scene text detection. *arXiv preprint arXiv:1706.09579*, 2017. 8
- [17] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1156–1160. IEEE, 2015. 5, 6, 7
- [18] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park. Pvanet: Lightweight deep neural networks for real-time object detection. *arXiv preprint arXiv:1611.08588*, 2017. 7
- [19] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 734–750, 2018. 4, 7
- [20] H. Li, P. Wang, and C. Shen. Towards end-to-end text spotting with convolutional recurrent neural networks. In *Proc. ICCV*, pages 5238–5246, 2017. 2, 3
- [21] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu. Textboxes: A fast text detector with a single deep neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 8
- [22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2
- [23] W. Liu, C. Chen, and K.-Y. K. Wong. Char-net: A character-aware neural network for distorted scene text recognition. In *AAAI*, 2018. 3

- [24] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. 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. 1, 2, 3, 6, 7, 8
- [25] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. 2018. 1, 2, 3, 7, 8
- [26] J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*, 2018. 8
- [27] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, et al. Icdar2017 robust reading challenge on multilingual scene text detection and script identification-rrc-mlt. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, volume 1, pages 1454–1459. IEEE, 2017. 6
- [28] L. Neumann and J. Matas. Real-time lexicon-free scene text localization and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1872–1885, 2016. 8
- [29] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 2, 3
- [30] B. Shi, X. Bai, and S. Belongie. Detecting oriented text in natural images by linking segments. *arXiv preprint arXiv:1703.06520*, 2017. 2
- [31] B. Shi, X. Bai, and C. 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. 1, 3
- [32] B. Shi, X. Bai, and C. 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. 2
- [33] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 2, 3, 4
- [34] A. Shrivastava, A. Gupta, and R. Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 761–769, 2016. 6
- [35] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han, and E. Ding. Textnet: Irregular text reading from images with an end-to-end trainable network. *arXiv preprint arXiv:1812.09900*, 2018. 2, 8
- [36] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao. Detecting text in natural image with connectionist text proposal network. In *European conference on computer vision (ECCV)*, pages 56–72. Springer, 2016. 2, 5
- [37] Y. Q. Weilin Huang and X. Tang. Robust scene text detection with convolution neural network induced msr tree. In *European conference on computer vision (ECCV)*, 2014. 2
- [38] Y. Xu, Y. Wang, W. Zhou, Y. Wang, Z. Yang, and X. Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 2019. 5, 7, 8
- [39] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unit-box: An advanced object detection network. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 516–520. ACM, 2016. 2
- [40] Z. Zhang, W. Shen, C. Yao, and X. Bai. Symmetry-based text line detection in natural scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2558–2567, 2015. 8
- [41] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang. East: an efficient and accurate scene text detector. In *Proc. CVPR*, pages 2642–2651, 2017. 1, 2, 4, 5, 8