TextScanner: Reading Characters in Order for Robust Scene Text Recognition

Zhaoyi Wan[1*], **Minghang He**[2*], **Haoran Chen**[3], **Xiang Bai**[2†], **Cong Yao**[1†]

[1]Megvii, [2]Huazhong University of Science and Technology, [3]Beijing Institute of Technology

i@wanzy.me, {minghanghe09,xbai}@hust.edu.cn, xidianchr@163.com, yaocong2010@gmail.com

Abstract

Driven by deep learning and a large volume of data, scene text recognition has evolved rapidly in recent years. Formerly, RNN-attention-based methods have dominated this field, but suffer from the problem of *attention drift* in certain situations. Lately, semantic segmentation based algorithms have proven effective at recognizing text of different forms (horizontal, oriented and curved). However, these methods may produce spurious characters or miss genuine characters, as they rely heavily on a thresholding procedure operated on segmentation maps. To tackle these challenges, we propose in this paper an alternative approach, called TextScanner, for scene text recognition. TextScanner bears three characteristics: (1) Basically, it belongs to the semantic segmentation family, as it generates pixel-wise, multi-channel segmentation maps for character class, position and order; (2) Meanwhile, akin to RNN-attention-based methods, it also adopts RNN for context modeling; (3) Moreover, it performs paralleled prediction for character position and class, and ensures that characters are transcripted in the correct order. The experiments on standard benchmark datasets demonstrate that TextScanner outperforms the state-of-the-art methods. Moreover, TextScanner shows its superiority in recognizing more difficult text such as Chinese transcripts and aligning with target characters.

1 Introduction

In the past decades, scene text detection and recognition have drawn considerable attention from the computer vision community, due to its wide applications, e.g, automatic driving (Graves et al. 2006), visual auxiliaries (Ghosh, Valveny, and Bagdanov 2017), and human-computer interaction (Wang et al. 2012). As scene text provides pivotal and specific information, accurate recognition of text plays crucial roles in various real-world scenarios (Phan et al. 2013).

Among the state-of-the-art methods for scene text recognition, there are two prevalent paradigms: RNN-attention-based methods and semantic segmentation based algorithms.

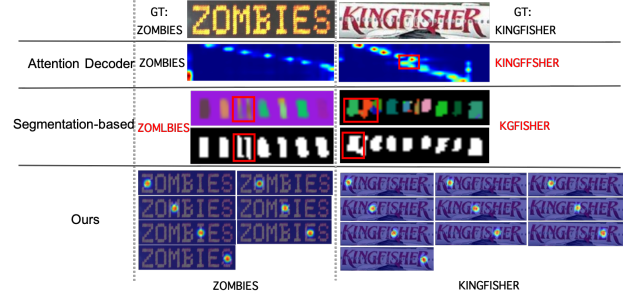*Authors contribute equally

†Corresponding author

Figure 1: Our Motivation. RNN-attention-based methods may encounter the problem of *attention drift* (Cheng et al. 2017) (see the red rectangle), thus leading to incorrect prediction of character class. In semantic segmentation based algorithms, the search of connected components depends on a thresholding operation, which is prone to over-segmentation or under-segmentation, thus generating spurious characters or missing genuine characters (see the red rectangles). In contrast, TextScanner scans characters one by one and ensures that characters are read in right order and separated properly.

The former (Shi et al. 2016; Cheng et al. 2018), drawing inspiration from neural machine translation (Bahdanau, Cho, and Bengio 2014), encodes images into features and employs an attention mechanism to align and decode characters. The latter (Lyu et al. 2018; Liao et al. 2019), approaching text recognition from a 2D perspective, first adopts a fully convolutional network (FCN) to perform semantic segmentation, then seeks connected components in the segmentation maps, and finally infers the class of each connected component (each is taken as a character).

Motivation Essentially, to correctly recognize the content in a cropped text image, the number of characters as well as the order and class of each character should be accurately predicted. RNN-attention based methods usually work well in most cases. However, when there is noise in the background or irregular text shape , the attention mechanism may fail that the center of the estimated attention map targets to
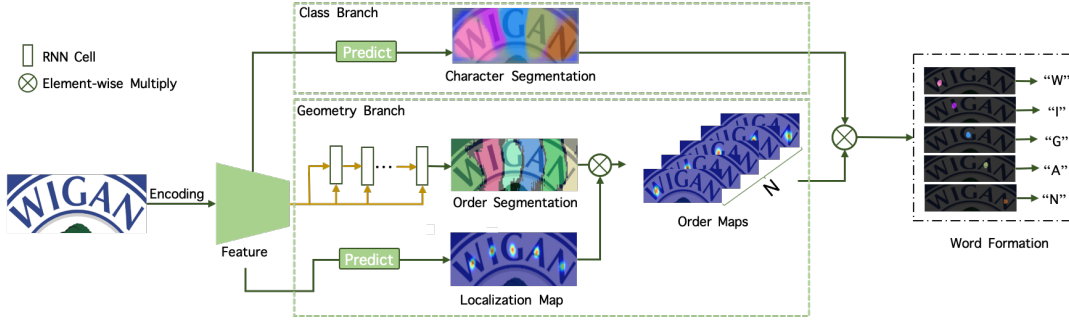
Figure 2: Schematic illustration of the proposed text recognition framework. Different colors in character segmentation map represent the values in different channels. The values in the localization map and order maps are visualized as heat maps. The predictions of the two branches are fused to extract characters (position, order, and class) and form the final output.

a wrong position, causing mistakes in character order and class (see Fig. 1). More seriously, due to the recurrent memory mechanism in the RNN module, such errors will accumulate and propagate, making the situation even worse.

Semantic segmentation based algorithms explore a different way and exhibit stronger adaptability to text of different shapes (horizontal, oriented and curved). However, it is difficult to successfully separate each character from the segmentation maps, since improper binarization will result in such embarrassments: one character might be split into multiple parts or multiple characters may stick together (see Fig. 1). In these cases, the predictions of number and class of characters would be wrong. In summary, existing approaches, either RNN-attention based or semantic segmentation based, are not able to commendably resolve the difficulties in scene text recognition.

The root cause for the attention drift problem in the RNN-attention based methods might be that the alignment operations (realized with attention maps) rely on both visual features and previous decoding results. Mutual interference might occur between these two types of information. Therefore, it is necessary to perform character alignment and classification in independent branches. Regarding semantic segmentation based algorithms, the assumption that characters can be sought via simple binarization does not hold in challenging scenarios. To address this issue, a natural and feasible solution is to represent the position and order of characters with different channels.

Our Work   In this paper, we propose a novel text recognition framework, called TextScanner. Like a scanner in the real world, TextScanner can read characters in correct order.

Generally, TextScanner is built upon semantic segmentation (Liao et al. 2019). It consists of two branches: one for character classification (class branch) and the other for character position and order prediction (geometry branch) (see Fig. 2). The class branch produces multi-channel segmentation maps, in which the values at each location represent the probabilities of character classes (including the background class). The geometry branch also produces multi-channel segmentation maps, but the meanings of the values at each location are different from those in the class branch. The

characters are sought by multiplying two groups of segmentation maps in an element-wise manner and acquiring the class with maximal probability for each channel. This procedure is termed as word formation in this work.

Since characters are well aligned and the order is ensured, TextScanner can avoid the attention drift phenomenon observed in RNN-attention based methods. Meanwhile, in the geometry branch different characters, even contiguous with each other or with the same class label, are strictly assigned into different channels, so they can be easily extracted.

As FAN (Cheng et al. 2017) and CA-FCN (Liao et al. 2019), TextScanner also requires character level annotations for training, since the geometry branch takes character centers as supervision signals. However, there are actually plenty of real image examples without character level annotations, which could be very beneficial for training text recognizers. To make use of such real data, we devise a mutual supervision mechanism. For image examples without character level annotations, the predictions of the class branch and the geometry branch can supervise each other with only sequence level annotations. In consequence, TextScanner is able to fully utilize all kinds of available training data, including both synthetic and real text images.

We conduct experiments on public benchmarks for scene text recognition to validate the proposed TextScanner. It achieves higher or highly competitive accuracy on regular text datasets and obtains significantly enhanced performance on irregular text datasets. The recognition accuracy increases 3.3% on ICDAR 2015 and 4% on CUTE80, compared with the previous art. We also evaluate TextScanner on a Chinese recognition task. The quantitative results further prove the superiority of the proposed algorithm. The contributions in this work are summarized as follows:

- We propose a novel text recognition framework, which predicts the class and geometry information (position and order) of characters with two separate branches.

- We devise a mutual-supervision mechanism, which endows the framework with the ability to make use of both synthetic and real data for training.

- The experiments demonstrate that the proposed TextScanner achieves state-of-the-art or highly competitive perfor-

mance on public benchmarks.

- Furthermore, TextScanner exhibits stronger adaptability to longer and more complex text (such as Chinese scripts).

2 Related Work

Text recognition has been a long-standing research topic in computer vision. Research efforts on text recognition can date back to the early age of AI (Herbert 1982; LeCun et al. 1998). With the rise of deep learning, scene text recognition has entered a new era. For recent progress in this field, please refer to the survey paper (Long, He, and Ya 2018; Zhu, Yao, and Bai 2016). In this section, we will concentrate on the most relevant works.

Inspired by speech recognition and natural language processing, CTC-based (Shi, Bai, and Yao 2017; Graves et al. 2006) soft-alignment methods and attention-based (Ghosh, Valveny, and Bagdanov 2017; Shi et al. 2018) methods are proposed to handle text recognition as a sequence recognition task. Among them, attention-based methods are prevalent recently, and achieve state-of-the-art performance on public benchmarks. On the other hand, there are still challenging problems in the field of scene text recognition. Text images in natural scenes suffer from the complex background, arbitrary text shape and severe image distortion. Most of current text recognition algorithms are not robust enough to solve hard cases such as text instances which are oriented, curved or extremely blurred.

As stated in Sec. 1, the problem of attention drift is exacerbated by recursive modeling. This problem is also observed in speech recognition (Kim, Hori, and Watanabe 2017), which is where the idea of attention decoder originated from. There are existing works aimed at alleviating this problem. (Cheng et al. 2017) proposed to correct attention positions using characters' class and localization label. New loss function motivated by the formulation of edit distance is presented(Bai et al. 2018) to improve the hard alignment of attention decoder. However, these methods do not change the nature of error accumulating which lies in the coupled modeling of attention generation and character classification. Different from these methods, the proposed algorithm uses two separated branches to classify characters and predict the positions and order of characters. The potential mutual interference between alignment and decoding is eliminated and the problem of attention drift can be avoided.

Recently, segmentation-based methods are also introduced to the field of text recognition (Lyu et al. 2018; Liao et al. 2019). Segmentation-based methods are usually more flexible than attention decoders in the recognition of irregular text such as oriented or curved text instances. However, the post-processing of these methods may fail to separate closely arranged characters as shown in Fig. 1. As the characters are recognized by finding and voting inside the connected components in the segmentation map, this restriction limits their recognition accuracy. Besides, the application of segmentation-based methods remains limited because these methods can not use real image examples with only sequence-level annotation. With our proposed method, the characters are naturally separated and ordered
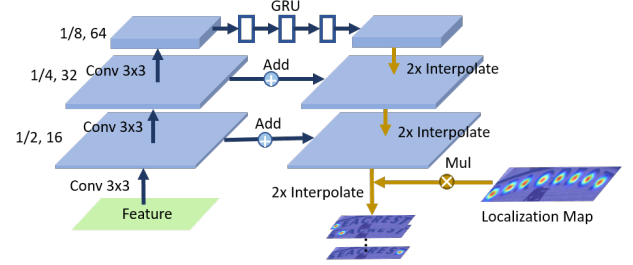

Figure 3: Illustration of the geometry branch. The feature maps are up-sampled and down-sampled by a pyramid architecture with skip connections. Features at the top layer is processed by an RNN module for context modeling.

by dispatching character localization to different channels. The mutual-supervision mechanism further enables the two branches to utilize sequence level annotations to supervise and enhance each other.

3 Methodology

Overview

The overall structure of the proposed method is illustrated in Fig. 2. The decoder of the network is composed of two branches: class branch and geometry branch.

Class Branch The class branch of TextScanner produces character segmentation $G \in R^{h \times w \times c}$, which is of resolution $h \times w$ and c denotes the number of classes (all character classes plus background). G is directly generated from visual features extracted by a CNN backbone. The prediction module is composed of two stacked convolutional layers with kernel size 3×3 and 1×1. A Softmax normalization is applied over the class dimension to generate the character segmentation maps.

Geometry Branch Firstly, a character localization map $Q \in R^{h \times w}$ is produced from the same visual features as the class branch, with a Sigmoid activation function. Concurrently, a top-down pyramid structure, in which features in the upsampling path is added by the features with the same resolution from the downsampling path is taken to generate order segmentation maps. Especially, the feature maps in the top layer of the downsampling path are encoded by an RNN (GRU(Chung et al. 2015), to be exact) module for context modeling. Following the upsampling path, two convolutional layers are employed to generate the order segmentation maps $S \in R^{h \times w \times N}$, where N is the pre-defined max sequence length. The order segmentation maps are also normalized by a Softmax operation. Then an order map $H_k \in R^{h \times w}$, which indicates the position of the k_{th} character in the sequence, can be computed from the $k_{th}(k \in [1, N])$ channel of order segmentation and the character localization map Q by an element-wise multiplication: $H_k = Q * S_k$. The detail of the geometry branch is depicted in Fig. 3.

Word Formation With the produced character clarification maps G and order maps H, we now present the de-
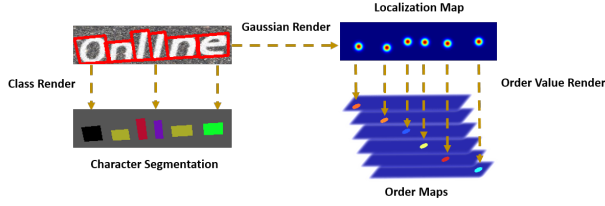
Figure 4: Ground truth generation for pre-training. Pixels outside shrunk boxes P' are represented as gray in character segmentation label, which are ignored in loss computation.

coding procedure which formats characters in order. As the order of character locations is encoded into the order maps H, the classification scores can be computed from the order maps and character clarification maps as:

$$p_k = \int_{(x,y)\in\Omega} G(x,y) * H_k(x,y) \qquad (1)$$

where $p_k \in R^c$ is the vector of scores representing the class probabilities of the k_{th} character. Ω is all valid spatial locations in the $h \times w$ space. Similar to attention decoders, once the maximal probability of a character is below a predefined threshold T_{score} or k reached the maximal value N, the decoding process is terminated.

This decoding procedure is totally differentiable. Therefore, it can be trained within the network using sequence level as well as character level annotations. The optimization process utilizing sequence level annotations is introduced in detail in the following section.

Pre-Training with Character-Level Annotations

When pre-training on synthetic data, TextScanner can be optimized with character-level annotations.

Label Generation Due to the rectangles are inaccurate in curved or dense text, we keep the definition of character regions polygons $P = \{(x_i, y_i)\}_{i=1}^{N_p}$, where N_p is the number of points in polygon.

To refrain the overlap caused by edges of adjacent characters, the polygon character bounding box P is shrunk to P' with the Vatti clipping algorithm(Vati 1992). Inside P' area, the class of the corresponding character is rendered as ground truth of the character segmentation. Pixels outside P' hardly contribute to the decoding of sequences and are ignored in the optimization of text segmentation.

To generate the ground truth of order maps with character-level annotations, the center of Gaussian maps is firstly detected by computing the central points of characters bounding boxes. As Fig. 4 shown,

2D Gaussian maps $\hat{Y}_k \in R^{h \times w}$ with σ and expectation at central points are generated for each character. Then the order of characters is rendered for pixels inside $\hat{Y}_k$ area:

$$\hat{Z}_k(i,j) = \begin{cases} k, & \text{if } \frac{\hat{Y}_k(i,j)}{max\hat{Y}_k(h,w)} > \zeta_{order}, (i,j) \in (h,w) \\ 0, & \text{otherwise} \end{cases}$$

$$(2)$$

where $\hat{Z}_k \in R^{h \times w}$ is the generated order map ground truth for each character, ζ_{order} is the order threshold, which set to 0.5 in our experiment. Finally $\hat{Z}_k$ is normalized to $[0, 1]$, to produce the ground truth Z_k of H_k.

Taking the same operation as $\hat{Z}_k$ to Z_k, 0-1 normalized Gaussian heatmap Y_k can be acquired from $\hat{Y}_k$. According to all Y_k, the ground truth of localization map Q can be generated by straightforwardly combining heat maps:

$$Y^* = \max_{k=1}^{N} Y_k \qquad (3)$$

Loss Function The overall loss function is a weighted sum of losses for the three mentioned tasks as

$$L = \lambda_l * L_l + \lambda_o * L_o + \lambda_m * L_m + L_s \qquad (4)$$

where L_c, L_o, L_s, L_m are the losses for localization map, order segmentation, text segmentation and mutual supervision loss respectively. The detail of mutual supervision loss is illustrated in the next section. In our experiments λ_l and λ_o are set to 10 for scaling the numerical values. λ_m is set to 0 during pretraining otherwise to 1.

The localization map loss is computed as an average smooth 11 loss. The losses for order segmentation(L_o) and character segmentation(L_s) are computed as cross entropy between the predicted scores and corresponding ground truth. The background class in both segmentation task is ignored in cross entropy computation.

Mutual-Supervision Mechanism

For previous semantic segmentation based methods, it's critical to obtain accurate locations of all the characters during model training, since the character classification is at pixel-level. This is problematic when character level annotations are not available. To reduce the reliance on character-level annotations, we devise a mutual-supervision mechanism based on the dual-branch structure of TextScanner. As shown in Fig. 2, text sequences can be generated by combining character segmentation maps G and order maps H. Given a sequence label and one of the two outputs, supervision signals can be generated for the other one.

G and H are further transformed by taking the index of max value across c and o channels into $\hat{G} \in R^{h \times w}$ and $\hat{H} \in R^{h \times w}$. Given the text sequence label T, the mutual-supervision process is carried out from the first character in T to the last. For the kth character in T, its order is k and the class is $T(k)$. Ψ_h^k and Ψ_g^k are coordinates of the pixels that corresponding to the kth character in H and G:

$$\Psi_h^k = \{(i,j)|\hat{H}(i,j) = k, Q(i,j) > \epsilon\}$$
$$\Psi_g^k = \{(i,j)|\hat{G}(i,j) = T(k), Q(i,j) > \epsilon\} \qquad (5)$$

Apart from the constraint of class or order, we add the constraint of $Q(i,j) > \epsilon$ to keep the attended regions in the center of characters. ϵ is set to 0.2 in our experiment.

For the mutual supervision purpose, we use Ψ_g^k, which is derived from G,in the supervision of H and Ψ_h^k, which is
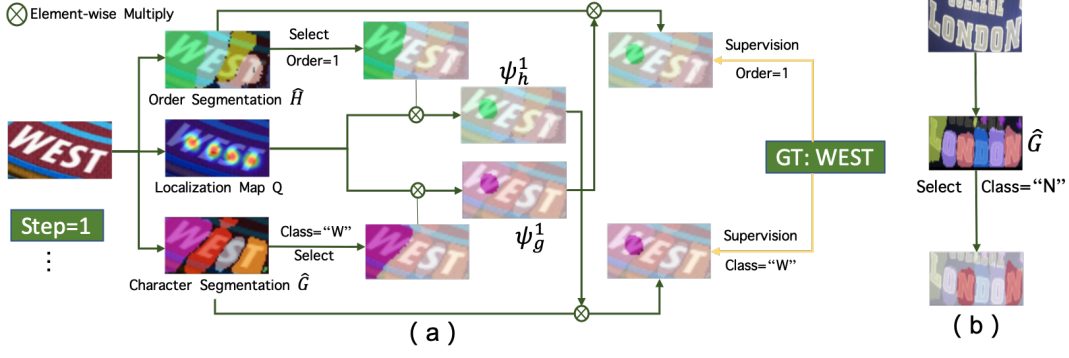
Figure 5: (a) Visualization of step 1 of mutual-supervision mechanism. The selected regions in $\hat{G}$ and $\hat{H}$ are refined using Q to get Ψ_g^1 and Ψ_h^1, which are then mapped into $\hat{H}$ and $\hat{G}$ separately. (b) Two regions in $\hat{G}$ are selected for 'N' in "LONDON".

derived from H, in the supervision of G:

$$L_g^k = \frac{1}{|\Psi_h^k|} \sum_{(i,j) \in \Psi_h^k} L_{CE}(G(i,j), onehot(T(k)))$$

$$L_h^k = \frac{1}{|\Psi_g^k|} \sum_{(i,j) \in \Psi_g^k} L_{CE}(H(i,j), onehot(k)) \tag{6}$$

L_{CE} is the cross entropy loss and $onehot(\cdot)$ is the one hot encoding function. The step 1 of cross supervision process is shown in Fig 5(a). The process is carried on till the last character of T. Note that multiple regions of $\hat{G}$ are chosen in one selection for characters occur more than once in T and can't be used in supervision of H, as shown in Fig 5(b). So we remove these samples from the cross supervision process. The confidences for G and H are denoted as Φ_g and Φ_h:

$$n_g^k = \begin{cases} 1, & \text{if } \Psi_g^k \neq \varnothing \\ 0, & \text{otherwise} \end{cases}, \quad \Phi_g = \frac{\sum_{k=1}^{|T|} n_g^k}{|T|}$$

$$n_h^k = \begin{cases} 1, & \text{if } \Psi_h^k \neq \varnothing \\ 0, & \text{otherwise} \end{cases}, \quad \Phi_h = \frac{\sum_{k=1}^{|T|} n_h^k}{|T|} \tag{7}$$

For the kth character in T, if the number of pixels in its corresponding region in G or H is larger than 0, the character is considered exists in G or H. $|T|$ is the length of T. The more the characters in T exists, the higher Φ is. $T = 1$ when all the characters exists in the prediction map. Φ_g and Φ_h are included in the loss computation of the text sequence:

$$L_m = L_h + \lambda * L_g$$

$$L_g = \frac{(\Phi_h)^\gamma}{|T|} \sum_{k=1}^{|T|} L_g^k, \quad L_h = \frac{(\Phi_g)^\gamma}{|T|} \sum_{k=1}^{|T|} L_h^k \tag{8}$$

In our experiment, λ is set to 0.2, γ is set to 2 to further reduce the impact of inaccurate predictions. After being pre-trained on synthetic datasets with character level annotations, our model can be further fine-tuned on real-world datasets or synthetic datasets with only sequence level annotation by adopting this cross supervision mechanism, which is impossible in previous segmentation-based methods.

Other Details

Our model is built on top of the backbone from CA-FCN, in which the character attentions are removed and VGG blocks are replaced with a ResNet-50(He et al. 2016) base model. The score threshold ζ_{score} is set as 0.3 empirically, and the max size N is set as 32 in our implementations.

4 Experiments

We conduct experiments on standard benchmarks to evaluate TextScanner and compare it with other competitors.

Datasets

ICDAR 2013(IC13) (Karatzas et al. 2013) recognition task provides 288 scene images with annotations, from which 1015 word images are cropped. Besides, the dataset provides character-level bounding boxes.

ICDAR 2015(IC15) (Karatzas et al. 2015) consists of 1000 images with word-level quadrangles annotation for training and 500 for testing.

IIIT 5K-Words(IIIT) (Mishra, Alahari, and Jawahar 2012) dataset contains 5K word images for scene text recognition.

Street View Text(SVT) (Wang, Babenko, and Belongie 2011) dataset has 350 images and only word-level annotations are provided.

SVT-Perspective(SVTP) (Phan et al. 2013) dataset contains 639 cropped images for testing. Many images in the dataset are heavily distorted.

CUTE80(CT) (Risnumawan et al. 2014) dataset is taken in natural scene. It consists of 80 high-resolution images with no lexicon.

ICDAR 2017 MLT(MLT-2017) (Nayef et al. 2017) is comprised of 9000 training images and 9000 test images. We acquire cropped word instances for recognition by using the quadrilateral word-level annotation.

SynthText (Gupta, Vedaldi, and Zisserman 2016) consists of 80k images for training. We cropped about 7 million instances with character and word-level bounding-boxes annotations from the training set.

Synth90k (Jaderberg et al. 2014b) contains 8 millions word images from 90k English words with word-level annotation.

Methods	Training Data	IIIT			SVT		IC13	IC15	SVTP	CT
		50	1k	0	50	0	0	0	0	0
Almazán et al.(?)	-	91.2	82.1	-	89.2	-	-	-	-	-
Strokelets(?)	-	80.2	69.3	-	75.9	-	-	-	-	-
Jaderberg et al.(Jaderberg et al. 2014a)	-	-	-	86.1	-	-	-	-	-	-
Su et al.(?)	-	-	-	-	83.0	-	-	-	-	-
Gordo(?)	-	93.3	86.6	-	91.8	-	-	-	-	-
Jaderberg et al.(Jaderberg et al. 2016)	90k	97.1	92.7	-	95.4	80.7	90.8	-	-	-
Jaderberg et al.(Jaderberg et al. 2014a)	90k	95.5	89.6	-	93.2	71.7	81.8	-	-	-
CRNN(Shi, Bai, and Yao 2017)	90k	97.6	94.4	78.2	96.4	80.8	86.7	-	-	-
RARE(Shi et al. 2016)	90k	96.2	93.8	81.9	95.5	81.9	88.6	-	71.8	59.2
R^2AM(Lee and Osindero 2016)	90k	96.8	94.4	78.4	96.3	80.7	90.0	-	-	-
Yang et al.(Yang et al. 2017)	90k	97.8	96.1	-	95.2	-	-	-	75.8	69.3
CA-FCN(Liao et al. 2019)	ST	99.8	98.8	91.9	**98.8**	86.4	91.5	-	-	79.9
TextScanner-pre	ST	99.8	99.0	92.1	97.7	89.4	**92.7**	79.6	82.6	83.5
TextScanner-mutual	ST	**99.8**	**99.2**	**92.6**	98.4	**89.5**	92.5	**80.1**	**83.7**	**85.2**
FAN^*(Cheng et al. 2017)	ST + 90k	99.3	97.5	87.4	97.1	85.9	93.3	70.6	-	-
AON^*(Cheng et al. 2018)	ST + 90k	99.6	98.1	87.0	96.0	82.8	-	68.2	73.0	76.8
EP^*(Bai et al. 2018)	ST + 90k	99.5	97.9	88.3	96.6	87.5	**94.4**	73.9	-	-
$ASTER^*$(Shi et al. 2018)	ST + 90k	99.6	98.8	93.4	97.4	89.5	91.8	76.1	78.5	79.5
TextScanner+90k*	ST + 90k	**99.7**	**99.1**	**93.9**	**98.5**	**90.1**	92.9	**79.4**	**84.3**	83.3
$SAR^\dagger$(Li et al. 2019)	ST + 90k + real	99.4	98.2	95.0	98.5	91.2	94.0	78.8	**86.4**	89.6
TextScanner+real†	ST + 90k + real	**99.8**	**99.5**	**95.7**	**99.4**	**92.7**	**94.9**	**83.5**	84.8	**91.6**

Table 1: Performance comparison of our methods and others. "ST", "90k", and "real" are the training data of SynthText, 90k, and real data, respectively. The methods marked with star mix SynthText and 90k dataset for training and methods marked with "†" use the training set of real data. "0", "50" and "1k" indicate the size of the lexicons, "0" means no lexicon.

In addition, approximately 16k real images are collected from the training sets released by the mentioned datasets for fine-tuning.

Training Strategy

The training procedure of TextScanner includes two stages: we firstly use the synthetic dataset SynthText with character-level annotations to pre-train the model for 5 epochs, then the real image examples with sequence-level annotations are mixed into the training set for fine-tuning the network for 1 more epoch. Our methods in Tab.1 use different data for training, "TextScanner-pre" and "TextScanner-mutual" are the models trained on SynthText with and without mutual-supervision, respectively; "TextScanner+90k" and "TextScanner+real" are fine-tuned with the 90k dataset and the training set of real data.

We use Adam optimizer for training of all experiments. The learning rate is initialized as 10^{-3} and the decays to 10^{-4} and 10^{-5}. During training and inference, the input images are resized to 64×256.

Recognition Performance Evaluation

The recognition accuracies of different methods on standard benchmarks, including regular (IIIT, SVT, IC13) and irregular (IC15, SVTP, CT) text, are shown in Tab. 1.

The natural modeling of TextScanner makes it more robust to hard cases where the text is curved or oriented. The three variants of TextScanner in Tab. 1 consistently outperform previous methods in comparison with the same training
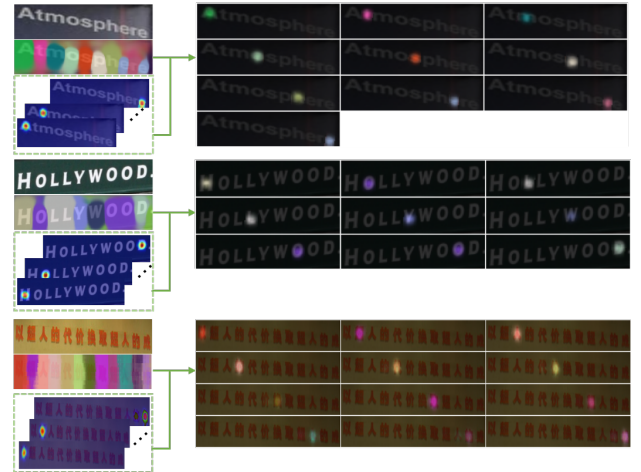

Figure 6: Intermediate results of TextScanner. Obviously TextScanner can track the arrangement of characters well, for long or oriented text. The geometry branch can separate adjacent characters even their segmentation masks connect to each other (note the two adjacent 'O' the middle row).

data. Especially on curved text, "TextScanner+90k", which is trained with synthetic data, achieves an improvement of 3.3% on IC15, 4.1% on SVTP, and 4.0% on CT. The advantages of TextScanner stem from aspects: (1) It is segmentation based, which makes the prediction more relevant

Methods	Acc(%)	NED
CRNN(Shi, Bai, and Yao 2017)	59.2	0.68
ASTER(Shi et al. 2018)	57.4	0.69
TextScanner	64.1	0.75

Table 2: Results comparison on MLT-17. "NED" is short for normalized edit distance.

to visual features and free from error accumulation brought by the recursive modeling. (2) It scans characters one by one and ensures they are read in the right order and separated properly. Some recognition examples are visualized in Fig. 6.

"TextScanner+real" brings an even more significant boost in performance and demonstrates the capability of TextScanner to utilize real data for training, which also verifies the effectiveness of the proposed mutual-supervision mechanism. Moreover, although the mutual-supervision mechanism is designed for making use of real-world data with sequence-level annotations, fine-tuning with synthetic SynthText and 90k datasets consistently bring performance improvement.

Chinese Recognition Evaluation

Preceding experiments are based on datasets with English text, whose alphabet is relatively small. To further validate the capability of TextScanner, we conduct experiments on a Chinese recognition task, which is more challenging due to its larger alphabet and more complex visual variations of characters. We compare the performance of TextScanner in Chinese recognition with two representative methods, CRNN(Shi, Bai, and Yao 2017) and ASTER(Shi et al. 2018). We use their open-source implementations for comparison. The models are trained with the same training data, which is generated by the synthetic engine released with SynthText and evaluated on cropped text images from the validation set of MLT-17.

The quantitative results are shown in Tab.2. In addition to accuracy, we also evaluate the Normalized Edit Distance of the methods following the ICDAR-2017 competition RCTW-17(Shi et al. 2017):

$$Norm = 1 - \frac{1}{N_t} \sum_i^{N_t} \frac{ED(s_i, \hat{s}_i)}{\max(|s_i|, |\hat{s}_i|)} \qquad (9)$$

where N_t is the number of text instances.

As shown in Tab. 2, TextScanner outperforms attention decoder with a large margin, 6.7% in recognition accuracy and 0.06 in normalized edit distance. The results demonstrate that TextScanner can handle such challenging recognition tasks better.

The main reason for this improvement is that TextScanner separate geometry branch from the class branch. For the Chinese recognition task, models are more prone to classification errors due to the much more complex structures and appearances of Chinese characters.

In contrast to attention decoders, the dual-branch architecture of TextScanner is more robust to the problem of error accumulating towards difficulty in classification. The class
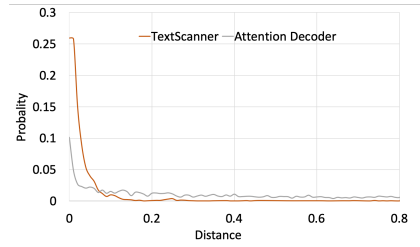

Figure 7: Probability density of deviation of character localization. "Distance" denotes the distance from predicted character center or attention position to the ground truth character center, which is normalized by the image width.

branch and the geometry branch are optimized individually in pre-training, therefore the extraction of character orders is not affected by the probable errors in classification. In the fine-tuning stage, the accurate order extraction can improve the class branch in return through the mutual-supervision mechanism.

Character Localization Accuracy of TextScanner

For both attention decoder and TextScanner, accurate prediction of attention position or character localization is fundamental for recognition. As they both produce the center of characters, we compare their performance in character localization on the IC13 dataset. As IC13 provides annotations of character positions in the image, the two methods are evaluated by measuring the normalized distance D between the produced character center and ground truth center position in the width axis.

The probability density of center distance in IC13 is illustrated in Fig.7. The probability of TextScanner to have accurate localization($D < 0.1$) is obviously greater than attention decoder. This proves that TextScanner gives more accurate character localization results.

5 Ablation Study

Geo	Ord	Accuracy (%)					
		IIIT	SVT	IC13	IC15	SVTP	CT
×	×	90.6	84.7	90.5	72.6	73.5	81.9
✓	×	91.2	85.6	90.9	75.8	75.6	82.7
✓	✓	92.6	89.5	92.5	80.1	83.7	85.2

Table 3: Recognition performance with different settings. "Geo" denotes the geometry branch, "Ord" denotes word formation using the order maps.

The superiority of TextScanner has been validated via the experiments in Chinese recognition task, here we specifically explore the effect of the geometry branch and the word formation process based on order maps. In Tab.3, the geometry branch and word formation are ablated respectively. Note that the word formation with order maps relies on the output of the geometry branch, therefore we use the post-processing procedure of CA-FCN as a replacement.

The experimental results clearly show the improvements brought by the geometry branch and its decoding process (the second and the third row in Tab.3). As the order maps ensure the characters are scanned in correct order, the recognition performance is significantly elevated, especially on irregular datasets: 7.4% on IC15 and 10.2% on SVTP. Besides, even with the regular post-processing, the geometry branch still achieves better performance, proving it can facilitate the optimization of the class branch.

6   Conclusion

In this paper, we have presented TextScanner, an effective segmentation-based dual-branch framework for scene text recognition. TextScanner can overcome the problems and defects of previous methods, and work well under various challenging scenarios. A novel mutual-supervision mechanism, which makes it possible to take full advantage of both real and synthetic data, is also proposed. Besides, TextScanner shows stronger adaptability in handling difficult text.

Acknowledgements

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bai, F.; Cheng, Z.; Niu, Y.; Pu, S.; and Zhou, S. 2018. Edit probability for scene text recognition. In *CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 1508–1516.

Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; and Zhou, S. 2017. Focusing attention: Towards accurate text recognition in natural images. In *ICCV 2017*, 5086–5094.

Cheng, Z.; Xu, Y.; Bai, F.; Niu, Y.; Pu, S.; and Zhou, S. 2018. AON: towards arbitrarily-oriented text recognition. In *CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 5571–5579.

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, 2067–2075.

Ghosh, S. K.; Valveny, E.; and Bagdanov, A. D. 2017. Visual attention models for scene text recognition. In *2017 14th IAPR ICDAR*, volume 01, 943–948.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine learning*, 369–376. Pittsburgh, Pennsylvania, USA: IMLS.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *CVPR*, 2315–2324.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *PAMI*, 770–778.

Herbert, H. 1982. The history of ocr, optical character recognition. *Manchester Center, VT: Recognition Technologies Users Association*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014a. Deep structured output learning for unconstrained text recognition. *arXiv preprint arXiv:1412.5903*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014b. Synthetic data and artificial neural networks for natural scene text recognition. *NIPS Deep Learning Workshop*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* 116(1):1–20.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i. Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazn, J. A.; and de las Heras, L. P. 2013. Icdar 2013 robust reading competition. In *2013 12th ICDAR*, 1484–1493.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. Icdar 2015 competition on robust reading. In *2015 13th ICDAR*, 1156–1160. IEEE.

Kim, S.; Hori, T.; and Watanabe, S. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 4835–4839. IEEE.

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.; et al. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

Lee, C.-Y., and Osindero, S. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, 2231–2239.

Li, H.; Wang, P.; Shen, C.; and Zhang, G. 2019. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, volume 33, 8610–8617.

Liao, M.; Zhang, J.; Wan, Z.; Xie, F.; Liang, J.; Lyu, P.; Yao, C.; and Bai, X. 2019. Scene text recognition from two-dimensional perspective. In *AAAI*.

Long, S.; He, X.; and Ya, C. 2018. Scene text detection and recognition: The deep learning era. *arXiv preprint arXiv:1811.04256*.

Lyu, P.; Liao, M.; Yao, C.; Wu, W.; and Bai, X. 2018. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *ECCV*, 67–83.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. In *BMVC-British Machine Vision Conference*. BMVA.

Nayef, N.; Yin, F.; Bizid, I.; Choi, H.; Feng, Y.; Karatzas, D.; Luo, Z.; Pal, U.; Rigaud, C.; Chazalon, J.; Khlif, W.; Luqman, M. M.; Burie, J.; Liu, C.; and Ogier, J. 2017. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt. In *2017 14th IAPR ICDAR*, volume 01, 1454–1459.

Phan, T. Q.; Shivakumara, P.; Tian, S.; and Tan, C. L. 2013. Recognizing text with perspective distortion in natural scenes. In *2013 IEEE International Conference on Computer Vision*, 569–576.

Risnumawan, A.; Shivakumara, P.; Chan, C. S.; and Tan, C. L. 2014. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications* 41(18):8027 – 8048.

Shi, B.; Bai, X.; and Yao, C. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *PAMI* 39(11):2298–2304.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *CVPR*, 4168–4176.

Shi, B.; Yao, C.; Liao, M.; Yang, M.; Xu, P.; Cui, L.; Belongie, S.; Lu, S.; and Bai, X. 2017. Icdar2017 competition on reading chinese text in the wild (rctw-17). In *2017 14th IAPR ICDAR*, volume 1, 1429–1434. IEEE.

Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2018.

Aster: An and attentional scene and text recognizer and with flexible and rectification. In *PAMI*, 1–1. IEEE.

Vati, B. R. 1992. A generic solution to polygon clipping. *Communications of the ACM* 35(7):56–64.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *ICCV*, ICCV '11, 1457–1464. Washington, DC, USA: IEEE Computer Society.

Wang, T.; Wu, D. J.; Coates, A.; and Ng, A. Y. 2012. End-to-end text recognition with convolutional neural networks. In *ICPR2012*, 3304–3308.

Yang, X.; He, D.; Zhou, Z.; Kifer, D.; and Giles, C. L. 2017. Learning to read irregular text with attention mechanisms. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 3280–3286.

Zhu, Y.; Yao, C.; and Bai, X. 2016. Scene text detection and recognition: Recent advances and future trends. *Frontiers of Computer Science* 10(1):19–36.