Spatial Aggregation for Scene Text Recognition

Yili Huang[1]
s15hyl@sjtu.edu.cn
Chengyu Gu[1]
gcy950912@sjtu.edu.cn
Shilin Wang[1,2]
wsl@sjtu.edu.cn
Zheng Huang[1]
huang-zheng@sjtu.edu.cn
Kai Chen[1]
kchen@sjtu.edu.cn

[1] School of Electronic Information and Electrical Engineering
Shanghai Jiao Tong University
Shanghai, China

[2] Diagnosis and Treatment Engineering Technology Research Center of Nervous System Diseases of Ningxia Hui Autonomous Region
Yinchuan, China

Abstract

Text recognition in natural images is an important research topic that has attracted widespread interest in recent years. Without character-level annotations, most existing state-of-the-art scene text recognition methods adopt CTC or attention-based decoders in the prediction stage to obtain the final word-level recognition results. However, these methods bring strong vocabulary reliance and fail to obtain satisfactory results when the predicting sample is out of the vocabulary in the training set. Moreover, predicting character-by-character in serial also limits efficiency. To solve these problems, in this paper, a new structure for the prediction stage is proposed to alleviate vocabulary reliance and accelerate prediction. In the new prediction stage, two classification layers are adopted on each feature vector to predict i) the character and ii) the order of the character in the word from the image region represented by the feature vector. Then, a spatial aggregation layer is designed to comprehensively integrate the character classification and the order estimation results to derive text recognition. In addition, a self-attention layer is adopted between the feature extraction stage and prediction stage to model the context. The experiment results on various benchmarks have demonstrated that compared with several state-of-the-art approaches, the proposed model achieves better performance in recognition accuracy and efficiency.

1 Introduction

Scene text recognition (STR), *i.e.*, text recognition from digital images in natural scenes, is an active and challenging research area. Its popularity stems from many real-life applications such as natural scene understanding and multimedia content analysis. However, variations in lighting, background, perspective, font, and layout in natural scenes have brought great difficulties to the traditional Optical Character Recognition (OCR) technology.

Recently, deep learning technology has shown its prominent advantages in many Computer Vision (CV) and Natural Language Processing (NLP) tasks and has been successfully applied in STR. According to [1], a typical deep-learning-based pipeline usually includes four stages: transformation, feature extraction, context modelling, and prediction. To the best of our knowledge, most STR methods without character-level annotations adopted Connectionist Temporal Classification (CTC) or attention-based methods which specializes in capturing the character-level semantic context information (*e.g*. the regularity of the arrangement of letters in all the words in the vocabulary) which help the model to rectify missing/erroneous characters. The CTC and attention mechanism are generally adopted in the context modelling and prediction stage. However, the strength in utilizing semantic context information can also lead to "vocabulary reliance" [25] . Although performing well on various public benchmarks, CTC and attention-based methods derive obviously lower performance on images with words outside the vocabulary. Another noticeable issue for the attention-based approach is that the serial decoding process in these methods limits the inference efficiency.

To overcome the problems mentioned above, a new vocabulary insensitive method for the prediction stage is proposed in this paper. The main contributions of the proposed method are three-folds: 1) A new method for the prediction stage in STR is proposed, which provides a solution and thus the vocabulary reliance problem can be alleviated; 2) The proposed method achieves fast inference speed and strong interpretability while achieving better accuracy; 3) The proposed method achieves the best STR performance, considering both accuracy and inference speed, compared with several state-of-the-art methods when expensive character-level annotations are not available.

2 Related Works

2.1 CTC and ACE

CTC was first proposed as a loss function for speech recognition. Convolutional Recurrent Neural Network (CRNN) [21] was one of the pioneering word-level STR methods based on deep learning, which adopted CTC in the prediction stage. Hu *et al*. [8] further improved the CTC-based method by adopting a graph convolutional network and an auxiliary attention-based branch. Xie *et al*. [28] attempted to implement the prediction stage without any language-based method for scene text recognition by adopting a new character-counting loss function, *i.e*., the Aggregation Cross-Entropy (ACE), and achieved comparable word-level recognition accuracy with CRNN. Addressing the complicated and time-consuming calculation process of CTC, ACE was proposed as a better loss function and can be easily adapted to the 2D prediction problem. The ACE-based method first decoded each feature vector in the feature map to the probability vector denoting the character it represents. Then, the ACE loss function was applied to supervise the summation of all the values in the predicted probability map for the character converges to the number of appearances in this character. Owing to the simplicity of the aggregation strategy, the prediction process can work in parallel and thus the prediction time can be reduced.

2.2 Attention-based Decoder

In recent years, the attention mechanism has achieved outstanding performance in sequence modelling and many state-of-the-art STR methods adopt the attention-based framework. Zhu

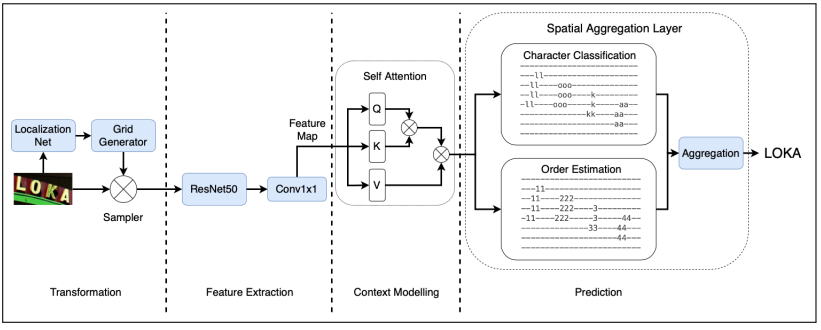

Figure 1: The overall network structure. '-' in the spatial aggregation layer means the background area.

et al. [33] and Li *et al*. [13] proposed new attention-based approaches on the 2D feature map to better recognize slanted texts. Borrowing the idea from text detection, a segmentation-based decoder is becoming a new trend for the prediction stage to addresses and recognizes characters. Usually combined with the attention mechanism, these methods [17] can achieve higher accuracies at the cost of expensive character-level annotations. Built upon the attention-based method, Shi *et al*. [22] proposed a text rectification module in the transformation stage to normalize various text layout, which is a new direction to improve recognition performance. Note that the attention-based methods usually decode characters with a RNN decoder or a Transformer decoder [23] , which will make the model focus more on the semantic context information rather than the image features to some extent and thus will lead to the vocabulary reliance problem.

3 Proposed Method

To overcome the weaknesses of CTC and attention-based decoders, we propose a new spatial aggregation layer in the prediction stage. Our motivation is to estimate both the character and its order in the sequence from the feature map extracted from the image. The overall structure of our network is shown in Fig.1. Following the mainstream pipeline concluded in [1] , our network also consists of four stages.

3.1 Stages Before Prediction

The input image is firstly transformed by a Spatial Transformer Network (STN) [10] and the image feature map can be extracted from the output of STN by the ResNet [7] backbone. Then, several stacked self-attention layers are adopted to further enrich the context information obtained by the feature vectors.

Transformation Stage. To handle complex distortion and layout in scene text images, the input images are usually transformed by an STN in the transformation stage. Our method adopts an STN with Thin-Plate-Spline [3] in this stage, which is similar to [22] .

Feature Extraction Stage. The feature extraction stage is usually implemented by a deep CNN which transforms an image into highly abstract feature maps whose height and width are 1/4 of the input image. An additional convolutional layer with kernel size of 1×1 is appended following the deep CNN to reduce the dimension of feature vectors. Each feature

vector in the resulting feature map has a corresponding distinguishable receptive field and contains the information of the corresponding local area of the input image.

Image Context Modelling Stage. To model image context on a 2D feature map, our method adopts several stacked self-attention layers [23] in this stage. According to [27] , self-attention can be viewed as a form of the non-local mean [4] . Thus, the output vectors of self-attention layers contain both the global and local information depicting the image.

3.2 Prediction Stage

For the prediction stage, we propose a new solution in this section. As shown in Fig.1, in the spatial aggregation layer, every feature vector is classified according to both the character it represents and its reading order in the sequence (*e.g.* 1 denotes the first reading character), and then is aggregated into a prediction. In this way, semantic context information can be directly modeled according to the related feature vectors in the image and thus the vocabulary reliance problem can be alleviated.

Order Estimation and Character Classification. Both the order estimation and the character classification are ordinary multi-class classification tasks which can be implemented by a fully connected layer with a softmax activation function, *i.e.*,

$$\begin{aligned} L_{x,y,i} &= SoftMax(\boldsymbol{W}_L \boldsymbol{f}_{x,y} + \boldsymbol{b}_L), \\ C_{x,y,j} &= SoftMax(\boldsymbol{W}_C \boldsymbol{f}_{x,y} + \boldsymbol{b}_C), \end{aligned} \tag{1}$$

where $\boldsymbol{f}_{x,y}$ is the (x,y)-th feature vector in the feature map, $L_{x,y,i}$ is the probability that the corresponding feature vector representing the i-th character in the output sequence, $C_{x,y,j}$ is the probability that the corresponding feature vector representing the j-th character in the character set (S_j in short), and $\boldsymbol{W}_C$, $\boldsymbol{b}_C$, $\boldsymbol{W}_L$, $\boldsymbol{b}_L$ are trainable parameters.

The Aggregation Algorithm. The aggregation algorithm takes $L_{x,y,i}$ and $C_{x,y,j}$ as inputs and calculates $P_{i,j}$ as the prediction output, which indicates the probability that there exists at least one feature vector which represents the character S_j and is ranked as the i-th character in the entire sequence. Note that the summation of $P_{i,j}$'s components, whether row-wise or column-wise, is not necessarily unity since they do not represent mutually exclusive events.

Let $F_{x,y,i,j}$ be the probability that the corresponding feature vector represents S_j which locates at the i-th location in the sequence. Assume that $L_{x,y,i}$ and $C_{x,y,j}$ are independent of each other, then $F_{x,y,i,j}$ can be calculated as,

$$F_{x,y,i,j} = L_{x,y,i} C_{x,y,j}. \tag{2}$$

For simplicity, we assume independence between different feature vectors, then $P_{i,j}$ can be calculated as,

$$\begin{aligned} 1 - P_{i,j} &= \prod_{x,y} (1 - F_{x,y,i,j}), \\ P_{i,j} &= 1 - \prod_{x,y} (1 - F_{x,y,i,j}). \end{aligned} \tag{3}$$

However, the floating precision easily overflows in the calculation of $P_{i,j}$ due to the multiplication of multiple probabilities and the independence assumption is difficult to satisfy. To overcome these problems, an approximate formula is adopted. Intuitively, the probability that the i-th output character is S_j mainly depends on whether a feature vector exists in the

Algorithm 1 The Inference Algorithm

Input: $P_{i,j}$, S
Output: The prediction of the string in the image.

Let s = ''
Let $minLoss = +\infty$
for i **from** to $max\_outputlength$ **do**
    $T = encode(s)$
    $loss = Loss(P, T)$
    if $loss < minLoss$ **then**
        $minloss = loss$
        $predictString = s$
    $c = \underset{j \in 1 \sim |S|}{argmax}(P_{i,j})$
    $s = s + S_c$
return $predictString$

feature map representing both S_j and the i-th output character. Inspired by the form of the sigmoid function, a feasible approximation is given in eq.4 which can be calculated without precision overflowing according to [2].

$$P_{i,j} \approx \frac{\sum_{x,y} \exp(logit(F_{x,y,i,j}))}{1 + \sum_{x,y} \exp(logit(F_{x,y,i,j}))}, \tag{4}$$

where $logit(\cdot)$ is the inverse *sigmoid* function.

3.3 Loss Function Design

Let $T_{i,j}$ be the one-hot code of the ground truth of a sample whose *(i, j)*-th component is unity if and only if the i-th character in the annotation is S_j, *i.e.*,

$$T_{i,j} = encode(gt) = \begin{cases} 1, & gt_i = S_j \\ 0, & gt_i \neq S_j \end{cases}, \tag{5}$$

where gt is the ground truth, gt_i is the i-th character in the ground truth, and $encode(\cdot)$ maps a string to its corresponding binary matrix that indicates its one-hot code.

Specifically, the *0*-th character in the alphabet represents the background noise and can be excluded from the calculation of loss function because it does not influence the inference result. The loss function can be calculated by the cross-entropy of $P_{i,j}$ and $T_{i,j}$ as eq.6:

$$\begin{aligned} Loss(P, T) &= \frac{1}{Len+1} \sum_{i=1}^{Len+1} \sum_{j=1}^{|S|} CE(sigmoid(P_{i,j}), T_{i,j}) \\ &= \frac{1}{Len+1} \sum_{i=1}^{Len+1} \sum_{j=1}^{|S|} \ln(1 + \exp(P_{i,j})) - T_{i,j} P_{i,j} \end{aligned}, \tag{6}$$

where $CE(\cdot)$ is the cross-entropy function, Len is the length of the annotation, and S is the character set. The loss where $i = Len + 1$ indicates the end of the prediction.

3.4 Inference Algorithm

Conducting the inference is tantamount to searching for a possible string that minimizes the loss function in eq.6. Without lexicon, the inference algorithm follows Algorithm 1 that decodes the characters under the maximum likelihood principle. The i-th output character is derived using *argmax* function and noted as S_c, $c \in 1 \sim |S|$. S_0 is regarded as the 'eos' character. By enumerating each position i as the end of the predicted string, which means the i-th character is assumed to be S_0, the set of the predicted string s with different lengths can be obtained. Then, each s in the set is encoded as T, and the loss can be calculated. Finally, the output string with the minimal loss will be selected as the predicted string.

4 Experiments

4.1 Datasets

Synth90k [9] . It is a synthetic text dataset generated by blending 90k common English words with natural scene images. The annotations are in word-level. All the images in this dataset are taken for training.

SynthText [6] . It is a synthetic text dataset that is generated in a similar way as Synth90k, but the words are rendered onto full images. The vocabulary is taken from the Newsgroup20 dataset. All cropped images by bounding boxes are taken for training.

SynthAdd [13] . This is a synthetic text dataset generated with the engine as same as [6] . Some special characters are randomly inserted into the words.

ICDAR2013 (IC13) [11] . This dataset contains 848 cropped text images for training and 1,015 cropped text images for testing. Most of its data are inherited from ICDAR2003.

ICDAR2015 (IC15) [12] . This dataset contains 4,468 cropped images for training 2,077 cropped text images for testing. Following the protocol of [5] , 1,811 images without non-alphanumeric characters are included in our test phrase.

Street View Text (SVT) [26] . This dataset is collected from Google Street View. It contains 257 cropped images for training and 647 images for testing.

SVT-Perspective (SVTP) [19] . This dataset contains and 639 cropped images for testing. It is proposed for evaluating the performance of recognizing perspective text and many images in this dataset are heavily distorted.

IIIT-5K [18] . This dataset contains 2000 images for training and 3,000 images for testing. All the images are collected from the webs. According to whether the word annotation belongs to the Newgroup20 dataset, IIIT5-K is further divided into two sub datasets: IIIT5K-I and IIIT5K-O. IIIT5K-I contains 2429 test samples whose word annotation belongs to Newsgroup20 and IIIT5K-O contains 571 test samples whose word annotation does not belong to Newgroup20.

CUTE80 (CUTE) [20] . This dataset contains 288 cropped images. Most of the texts are curved. No lexicon is associated.

COCOText (COCO) [24] . Samples in this dataset are cropped from COCO images. It contains 42618 cropped images for training, 9896 for validation, and 9837 for testing.

4.2 Implementation details

Network Configurations. The STN structure in [10] was adopted in our transformation stage and ResNet50 [7] was employed for feature extraction. Then, a convolutional layer

with the kernel size of 1×1 reduced the dimension of feature vectors from 2048 to 512. Two stacked self-attention layers were applied for context modelling. The hidden size and the number of heads were set to 512 and 8, respectively, for both self-attention layers.

Training. The input image was resized to 32×100 and data augmentation processes including random rotation and random color jittering were applied. All the non-alphanumeric characters were removed from the annotations and all the upper-case letters were turned into lower cases. The max length of the output sequence was set to 25. Besides, the batch size was set to 256 and the ADADELTA optimizer [31] was adopted. The learning rate was set to 1 and was decayed to 0.1 and 0.01 at step 0.6M and 0.8M, respectively. The training phase terminated at the step of 1M. The proposed model was trained on both synthetic datasets and the aforementioned real datasets which follows [8] and [13]. The sampling weights were adjusted so that the synthetic samples and real samples each account for half of the training data.

Evaluation. By default, the trained model was evaluated on all the benchmarks by case insensitive word accuracy.

Methods	IIIT5K	IIIT5K-I	IIIT5K-O
CRNN [21]	86.8	91.1	68.7
FAN [5]	**89.9**	**93.1**	75.3
CA-FCN [14]	89.3	91.6	76.3
ASTER [22]	89.2	92.9	74.6
Ours	**89.9**	92.9	**77.2**

Table 1: The comparison on IIIT5K-I and IIIT5K-O. The experiment results of other methods are implemented by [25].

4.3 Vocabulary Reliance Analysis

To evaluate the ability of vocabulary generalization, we have carried out experiments following those in [25] , where the model was only trained on SynthText and evaluated on IIIT5K-I and IIIT5K-O. The recognition accuracies in IIIT5K-I and IIIT5K-O reflect the ability of the model to predict seen and unseen words, respectively and the latter can be adopted as a metric to evaluate the vocabulary generalization ability. According to the results listed in Table 1, it is observed that our proposed method not only has comparable performance on IIIT5K-I but also outperforms other methods on IIIT5K-O. Generally speaking, segmentation-based (*e.g.*, CA-FCN [14]) methods require expensive character-level annotations for training. They are usually able to obtain more accurate visual features by locating characters. In comparison, our method does not rely on character-level annotations and alleviates vocabulary reliance by modelling the semantic context relationships according to the image features rather than text information.

Methods	IIIT5K	IC13	IC15	SVT	SVTP	CUTE	COCO
Baseline	90.3/0.0	89.9/0.0	76.7/0.0	87.3/0.0	72.1/0.0	77.6/0.0	54.6/0.0
Baseline+Tran.	92.8/2.5	93.0/3.1	82.3/5.6	90.1/2.8	80.3/8.2	83.2/5.6	61.9/6.3
Baseline+Cont.	94.8/4.5	93.4/3.5	84.1/7.4	91.3/4.0	82.3/10.2	88.1/10.5	65.0/10.4
Proposed	95.5/5.2	94.1/4.2	85.6/8.9	92.4/5.1	86.7/14.6	88.5/10.9	67.7/13.1

Table 2: The recongnition results of the module adopted in our method. 'A/B' means 'accuracy/improvement'.

Methods	Prediction Layer	IIIT5K	IC13	IC15	SVT	SVTP	CUTE	COCO
Shi *et al.* [21]	CTC	81.2	89.6	-	82.7	-	-	-
Hu *et al.* [8]	CTC+Attention	**95.5**	94.3	82.5	**92.9**	86.2	**92.3**	-
Liu *et al.* [16]#	Segmentation	83.6	90.8	60.0	84.4	73.5	-	-
Lyu *et al.* [17]#	Segmentation	95.3	95.3	78.2	91.8	83.6	88.5	-
Liao *et al.* [14]#	Segmentation	91.9	91.5	-	86.4	-	79.9	-
Cheng *et al.* [5]#	Attention	87.4	93.3	85.3	85.9	-	-	-
Zhan *et al.* [32]	Attention	93.3	91.3	76.9	90.2	79.6	79.5	-
Litman *et al.* [15]	Attention	93.7	93.9	82.2	92.7	**86.9**	87.5	-
Yang *et al.* [29]#	Attention	93.9	93.9	78.7	90.6	82.2	87.8	-
Li *et al.* [13]	Attention	95.0	94.0	78.8	91.2	86.4	89.6	66.8
Yu *et al.* [30]	Attention	94.8	**95.5**	82.7	91.5	85.1	87.8	-
Shi *et al.* [22]	Attention	93.4	91.8	76.1	89.5	78.5	76.8	-
Xie *et al.* [28]	ACE	82.3	89.7	68.9	82.6	70.1	82.6	-
Ours(Baseline)	Spatial Aggregation	90.3	89.9	76.7	87.3	72.1	77.6	54.6
Ours*		92.5	93.9	80.8	89.8	83.3	83.2	54.1
Ours		**95.5**	94.1	**85.6**	92.4	86.7	88.5	**67.7**

Table 3: The accuracy comparison on various benchmarks. '#' indicates that the methods are trained with both word-level and character-level annotations. '*' indicates that the model is trained only on Synth90k and SynthText following the mainstream protocol suggested by [1].

4.4 Ablation Study

To analyse the impact of each module adopted in our method, a series of ablation studies were performed, and the results were listed in Table 2. Our baseline only adopted ResNet50 for feature extraction and the spatial aggregation layer for prediction. Then, the transformation stage (Tran.) and the context modelling stage (Cont.) were applied to the model respectively to evaluate the benefit they brought. According to the results, it is observed that the recognition accuracies in all the datasets increase substantially with the employment of STN and/or the context modelling layer.

4.5 Comparisons with SOTA Methods

Table 3 lists the scene text recognition accuracies by the proposed approach compared with those obtained by other state-of-the-art approaches. From the table, the following observations can be made: i) Compared with the existing method without the character-level semantic context information, *i.e.*, ACE, even the baseline model (without the transformation stage and context modelling stage) can significantly improve the recognition performance. ii) Compared with the methods [21, 22, 32] converting the image map to one dimension and apply RNN encoders to model the context, our method keep the feature map in the 2D form, maintaining more layout information, and significantly improve the performance on benchmarks with many distorted samples (*i.e.*, IC15, SVTP, CUTE). iii) Compared with the recent state-of-the-art method in [13] which also decodes characters on a 2D feature map, our method not only maintains the same simplicity (*i.e.*, without additional supervision) but also gain improvement on most of the benchmarks. Moreover, our method also reaches comparable performance with [8] which applies a complex time-consuming hybrid mechanism during training.

Methods	Time(ms)	GPU Memory(MB)
ACE [28]	8.12	652
ASTER [22]	42.36	698
CRNN [21]	15.18	687
Ours(Baseline)	10.02	661
Ours	13.4	683

Table 4: The complexity comparison.

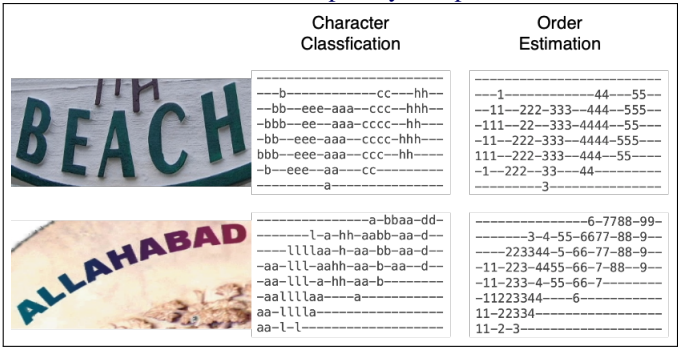

Figure 2: Visualization of the outputs of the character classification and order estimation in the spatial aggregation unit.

4.6 Complexity Analysis

To investigate the efficiency of the proposed method, a comparison with other mainstream methods in time and memory consumption was performed and the results were shown in Table 4. For a fair comparison, all the methods investigated adopt the same ResNet50 backbone. Note that all the models are implemented with the batch size of 1. The average inference time per image and the maximum GPU memory usage are listed in Table 4. Note that for the attention-based methods, the ASTER method is selected for its best recognition accuracy. It can be observed from Table 4 that ACE consumes the shortest time since it requires few calculations other than feature extraction. Although our method takes about 2ms more on the spatial aggregation layer per graph than ACE, it is still significantly faster than CRNN and other attention-based methods.

4.7 Interpretability Analysis

Similar to the attention-based methods and ACE, our proposed method also has strong interpretability from a 2D perspective. Fig.2 visualizes some character classification and order estimation results in the spatial aggregation layer. It is observed from the figure that the layout of text in an image is clearly expressed by the output of the spatial aggregation layer, which demonstrates the effectiveness of the character classification and order estimation modules in the spatial aggregation layer. The difference is that for one character, our method neither limits the total probability like ACE nor scales the importance weight spatial-wise with the SoftMax function like attention-based methods. Thus, more feature vectors in the feature map tend to be related to one reading character.

(a) GT: belgium
Pred: belvium

(b) GT: chimney
Pred: chimve

(c) GT: bloom
Pred: obloom

(d) GT: thunderball
Pred: thunderbal

Figure 3: Some failure cases of the proposed method. 'GT' is the ground-truth annotation, and 'Pred' is the predicted result.

4.8 Failure Cases Analysis

Some failure cases are shown in Fig. 3. The proposed method may fail to recognize the images which suffer from low quality (*e.g.*, particularly blurry, distorted images) or have complicated font style, as shown in strings 'belgium' and 'chimney'. Moreover, irrelevant patterns in the picture (*e.g.*, the circle in the image of 'bloom') or characters at the edge of the image (*e.g.*, the last 'l' in the string 'thunderball') may be misrecognized or missed.

5 Conclusion

In this paper, we propose a new method for the prediction stage in the scene text recognition task and make a novel attempt to discard language-based methods (*i.e.*, CTC and attention-based methods) in the prediction stage. The proposed method not only alleviates vocabulary reliance but also boosts the inference speed. The proposed method, *i.e.*, the spatial aggregation layer comprehensively considers the character recognition and the order estimation information to generate the text prediction results. Since it abandons traditional linguistic-based methods, it will not overly rely on the seen vocabulary and will be able to work in parallel. Extensive experiments on various datasets have demonstrated that our method outperforms several state-of-the-art approaches when taking both accuracy and efficiency into consideration.

6 Acknowledgements

The work described in this paper was fully supported by National Natural Science Foundation of China (61771310).

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4715–4723, 2019.

[2] Pierre Blanchard, Desmond J Higham, and Nicholas J Higham. Accurate computation of the log-sum-exp and softmax functions. *arXiv preprint arXiv:1909.03469*, 2019.

[3] F.L. Bookstein. Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[4] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 60–65, 2005.

[5] 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/CVF International Conference on Computer Vision*, pages 5076–5084, 2017.

[6] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11005–11012, 2020.

[9] 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.

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 2017–2025, 2015.

[11] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *Proceedings of the 12th International Conference on Document Analysis and Recognition*, pages 1484–1493, 2013.

[12] 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 *Proceedings of the 13th International Conference on Document Analysis and Recognition*, pages 1156–1160, 2015.

[13] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8610–8617, 2019.

[14] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8714–8721, 2019.

[15] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11962–11972, 2020.

[16] Wei Liu, Chaofeng Chen, and Kwan-Yee Wong. Char-net: A character-aware neural network for distorted scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7154–7161, 2018.

[17] 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*, pages 67–83, 2018.

[18] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *Proceedings of the British Machine Vision Conference*, pages 1–11, 2012.

[19] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *PProceedings of the IEEE/CVF International Conference on Computer Vision*, pages 569–576, 2013.

[20] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014.

[21] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304, 2016.

[22] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2035–2048, 2018.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

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

[25] Zhaoyi Wan, Jielei Zhang, Liang Zhang, Jiebo Luo, and Cong Yao. On vocabulary reliance in scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11425–11434, 2020.

[26] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1457–1464, 2011.

[27] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[28] Zecheng Xie, Yaoxiong Huang, Yuanzhi Zhu, Lianwen Jin, Yuliang Liu, and Lele Xie. Aggregation cross-entropy for sequence recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6538–6547, 2019.

[29] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9147–9156, 2019.

[30] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020.

[31] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[32] Fangneng Zhan and Shijian Lu. Esir: End-to-end scene text recognition via iterative image rectification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2059–2068, 2019.

[33] Yiwei Zhu, Shilin Wang, Zheng Huang, and Kai Chen. Text recognition in images based on transformer with hierarchical attention. In *Proceedings of the IEEE International Conference on Image Processing*, pages 1945–1949, 2019.