A Length-Sensitive Language-Bound Recognition Network for Multilingual Text Recognition

Ming Gao[1], Shilian Wu[2], and Zengfu Wang[2(✉)]

[1] Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China
vivigreeeen@mail.ustc.edu.cn

[2] University of Science and Technology of China, Hefei, China
wushilia@mail.ustc.edu.cn, zfwang@ustc.edu.cn

Abstract. Due to the widespread use of English, considerable attention has been paid to scene text recognition with English as the target language, rather than multilingual scene text recognition. However, it is increasingly necessary to recognize multilingual texts with the continuous advancement of global integration. In this paper, a Length-sensitive Language-bound Recognition Network (LLRN) is proposed for multilingual text recognition. LLRN follows the traditional encoder-decoder structure. We improve the encoder and decoder respectively to better adapt to multilingual text recognition. On the one hand, we propose a Length-sensitive Encoder (LE) to encode features of different scales for long-text images and short-text images respectively. On the other hand, we present a Language-bound Decoder (LD). LD leverages language prior information to constrain the original output of the decoder to further modify the recognition results. Moreover, to solve the problem of multilingual data imbalance, we propose a Language-balanced Data Augmentation (LDA) approach. Experiments show that our method outperforms English-oriented mainstream models and achieves state-of-the-art results on MLT-2019 multilingual recognition benchmark.

Keywords: Multilingual text recognition · Transformer

1 Introduction

Nowadays, in the research of scene text recognition, most existing methods mainly focus on Latin-alphabet languages, even only case-insensitive English characters. However, text recognition in other languages has become increasingly valuable with the trend of globalization and international cultural exchange. At present, the most common method for multilingual recognition is to apply the method that works in English directly to all kinds of languages. In this way, all characters are treated as different categories without distinguishing languages, and the data of all languages are mixed and trained together to get a universal network that can recognize all characters. But there are some disadvantages:

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
D.-T. Dang-Nguyen et al. (Eds.): MMM 2023, LNCS 13834, pp. 139–150, 2023.
https://doi.org/10.1007/978-3-031-27818-1_12

- Too many categories lead to poor recognition accuracy.
- Different languages have different characteristics, so they may adapt to different recognizers. It is difficult to achieve the optimal solution in every language using the same network.
- There are visually similar characters with different labels in different languages. Like "고" in Korean and "卫" in Chinese, "—" in Symbols and "—" in Chinese, "ん" in Japanese and "h" in Latin. Multilingual mixed training makes these similar characters difficult to identify.
- Latin data tend to be much more than other languages, so the prediction results will be more in favor of Latin.

In this paper, a Length-sensitive Language-bound Recognition Network (LLRN) is designed for multilingual recognition. At the same time, Language-balanced Data Augmentation(LDA) is applied to balance the multilingual data.

In summary, the main contributions of this paper are as follows:

- We use LDA to solve the problem of data imbalance in different languages and significantly improve recognition accuracy.
- We propose a Length-sensitive Encoder (LE) adapted to different lengths of text images, and a Language-bound Decoder (LD) adapted to different languages. These are the two main components that LLRN has innovated for multilingual recognition.
- The proposed LLRN achieves state-of-the-art (SOTA) performance on mainstream benchmarks on the MLT-2019 dataset.

2 Related Work

2.1 Scene Text Recognition

Basically, the mainstream approach after 2015 is based on two ideas. One is based on Connectionist Temporal Classification (CTC) [3], especially the combination of CTC and neural networks. The typical representative method is CRNN [15]. CRNN uses Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) for feature extraction. Then, the feature map is decoded to an output sequence, and the problem of constructing the loss function of the indefinite time sequence is solved by calculating the conditional probability. Another approach is based on the attention mechanism [17], which is usually combined with the sequence-to-sequence encoder-decoder framework to help feature alignment through the attention module. The typical representative method is ASTER [16]. ASTER uses CNN to extract feature maps from the input images, and then the feature maps are encoded by a Bidirectional Long Short-Term Memory (BiLSTM) network [6]. According to the weight given by the attention model, the features of different positions are weighted as the input of the decoding model, and then the decoding is carried out by the RNN based on the attention mechanism.

2.2 Multilingual Scene Text Recognition

Most of the above studies are for English. There are two ways to extend them to multilingual scene text recognition. The first way is to identify the language script of the scene text images and then send it to the recognition network of the corresponding script. For example, in [7], text images in different languages are sent to the corresponding recognition network to get the result. The second way is undifferentiated regarding all the characters of all languages as different categories and getting a general network that can recognize all languages through mixed training. For example, E2E-MLT [1] forgoes script identification and performs text recognition directly.

3 Methodology

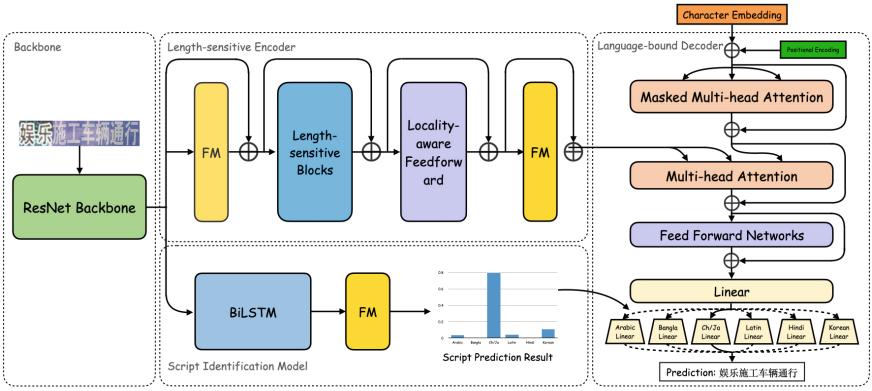

Fig. 1. A schematic overview of LLRN. LLRN is composed of four parts: a ResNet Backbone, a Length-sensitive Encoder (LE), a Script Identification Module (SIM) and a Language-bound Decoder (LD).

In this paper, we propose a multilingual text recognition pipeline called Length-sensitive Language-bound Recognition Network (LLRN). The network is composed of four parts: a backbone based on ResNet [5], a Length-sensitive Encoder (LE), a Script Identification Module (SIM) and a Language-bound Decoder (LD). Moreover, in order to address the multilingual data imbalance, we also introduce Language-balanced Data Augmentation (LDA) to balance the amount of data in different languages.

Figure 1 shows the pipeline of our method. First, we utilize LDA to equalize the amount of data in each language. Then, the language-balanced data are fed into the LLRN for training. During the training stage, first of all, like most text recognition methods, the backbone uniformly extracts features from the

input images to obtain feature maps. Secondly, the feature maps are sent to the corresponding LE according to the aspect ratio of the original input images. At the same time, the feature maps are also fed into SIM to get the language classification. After that, the Transformer based LD, through language-bound linear layers guided by language classification, further calculates the final result.

In the following, we will introduce the implementation details of each module.

3.1 Language-Balanced Data Augmentation (LDA)

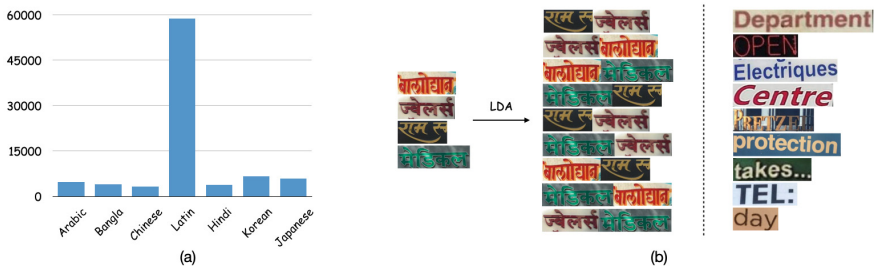

Fig. 2. (a) Distribution of data in different languages in MLT-2019. (b) Left: LDA for languages with rare data. Right: Keep Latin for excessive data unchanged.

As we can see from Fig. 2(a), Latin data are excessive while other language data are relatively rare. This often leads to the network overfitting on Latin data and inadequate training on data in other languages. Therefore, we draw inspiration from [19] and apply the context-based data augmentation (ConAug) proposed therein to enlarge the dataset so that all languages have the same amount of data as Latin. In our work, we refer to it as Language-balanced Data Augmentation (LDA) because of its ability to balance multilingual data. Figure 2(b) shows that we use LDA for languages with rare data, while keeping the Latin for excessive data unchanged. Not only that, this simple but effective data augmentation approach can change the background of the on-image text and force the network to learn a more diversified context so as to improve the generalization ability of the model.

To do this, first, all images are normalized to the same height while keeping the aspect ratio constant. Then, two different images of the same language are randomly selected and concatenated to a new view. The labels of the new view are the concatenation labels of the two original images. With LDA, we have achieved the same amount of data for all languages.

LDA is not only effortless to realize but also provides a considerable improvement, as our subsequent experiments (Sect. 4.5) will demonstrate.

3.2 Script Identification Module (SIM)

The Script Identification Module(SIM) connects to the backbone. As shown in Fig. 1, it contains a BiLSTM layer and a Feed-forward Module(FM). FM is composed of three linear layers sandwiched between ReLU activation and dropout.

3.3 Length-Sensitive Encoder (LE)

Fig. 3. (a) Architecture of Length-sensitive Encoder (LE). (b) Left: Examples of short-text images in each language. Right: Examples of long-text images in each language.

Figure 3(a) shows the specific architecture of the Length-sensitive Encoder(LE). We generally follow a Transformer-based encoder similar to SATRN [9]. The encoder of SATRN is composed of N self-attention blocks connected with a locality-aware feedforward network, while our network changes the self-attention blocks to length-sensitive blocks, which adapt to texts with different lengths. In addition, we learn from the architecture of Conformer, a convolution-augmented Transformer for speech recognition proposed by [4], to add two macaron-like Feedforward Modules (FM) with half-step residual connections. FM is exactly the same as it in [4].

Length-Sensitive Blocks. In the multilingual text recognition task, the length of different text images varies greatly. Some text images contain only one character, while others have more than a dozen characters. According to SATRN [9], self-attention layer itself is good at modeling long-term dependencies, but is not equipped to give sufficient focus on local structures. Therefore, we have reason to believe that the self-attention layer works well for long-text images. But for

short-text images, especially those with only one character, we should pay enough attention to local structures rather than the context relationship, which is suitable to be realized by a convolutional network. Thus, for long-text images, we keep the multi-head attention structure unchanged, while for short-text images, we design two continuous 3×3 convolutional layers for feature extraction, so that the network focuses more on the character itself in short-text images rather than the dependencies between contexts.

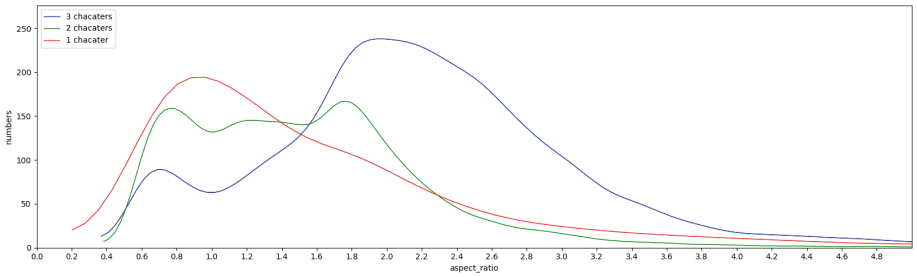

Fig. 4. The distribution of aspect ratio with 1 character, 2 characters and 3 characters on MLT-2019 dataset.

We perform statistics on MLT-2019 dataset of real scenes. Figure 4 shows the aspect ratio distribution of text images with one character, two characters and three characters. Through the observation of Fig. 4, we believe that most of the aspect ratio less than 1.8 contains only one or two characters, which is considered as short-text images, and vice versa. Therefore, we choose 1.8 as the dividing line between short-text and long-text images.

As shown in Fig. 3(b), the left side is short-text images with aspect ratios less than 1.8, and the right side is long-text images with aspect ratios greater than 1.8. As we can see, short-text images usually contain only one or two characters, so there is no need for excessive long-term dependencies.

3.4 Language-Bound Decoder (LD)

The traditional decoder designed for English can only capture the specific sequence feature. Since different languages have their own characteristics, we design the Language-bound Decoder (LD) to adapt to their own characteristics. Moreover, LD obtains the prior information of language by accepting the prediction of SIM. This further constrains the output result, increasing the probability of predicting characters in the target language and limiting the output of characters in non-target languages.

The decoder retrieves enriched two-dimensional features from the encoder to generate a sequence of characters. As shown in Fig. 1, the multi-head attention and point-wise feedforward layers are identical to the decoder of the Transformer

[17]. After that, we add language-bound linear layers to accommodate multilingual text recognition. It is essentially N_{lang} fully connected layers, where N_{lang} is the number of languages. In our work, language-bound linear layers contain Arabic, Bangla, Chinese-Japanese (Ch/Ja), Latin, Hindi and Korean linear layer. Decoded features are sent into the language-bound linear layer of the corresponding language according to the prediction of SIM. Since the language-bound linear layers take advantage of the prior information of languages, the output will be further corrected. We prove the effectiveness of LD in Sect. 4.5.

It is worth mentioning that we have tried to design separate decoders for different languages before. However, separate decoders lead to a shortage of training data in each language, so the network performance will decline sharply.

3.5 Loss

Equation(1) shows the two components of the loss function: script identification loss L_{lang} and multilingual text recognition loss L_{rec}.

$$L = \alpha L_{lang} + \sum_{l=1}^{N_{lang}} L_{rec(l)}, \tag{1}$$

where L_{rec} represents the text recognition loss of a single language. N_{lang} is the number of languages. α is a balanced factor. In our experiment, we set it equal to the language number.

As shown in Eq. (2), cross-entropy is used to compute the script identification loss.

$$L_{lang} = -\sum_{l=1}^{N_{lang}} I(l = l_{gt}) \log p(l), \tag{2}$$

where $I(l = l_{gt})$ is the binary indicator (0 or 1) if the language matches the ground truth, and $p(l)$ is the probability inferred by SIM that the word belongs to language l.

4 Experiments

4.1 Datasets

Our experiments are conducted on the following multilingual datasets.

MLT-2019. [13] releases the MLT-2019(MLT19) dataset of real images, which contains a total of 20 K real text-embedded natural scene images in 10 languages, including street signs, street billboards, shop names, passing vehicles, and so on. The ten languages are: Arabic, Bangla, Chinese, Devanagari, English, French, German, Italian, Japanese and Korean. Those languages belong to one of the following seven scripts: Arabic, Bangla, Latin, Chinese, Japanese, Korean, Hindi, Symbols and Mixed. The images are taken with different mobile phone cameras

or obtained for free from the Internet. The text in the scene images of the dataset is annotated at word level. Cropped scene text images of these nine scripts are used as datasets in our experiments.

SynthTextMLT. [13] also provides a synthetic dataset in seven scripts (without Symbols and Mixed) called SynthTextMLT, which we use to supplement our training dataset. The SynthTextMLT contains text rendered over natural scene images selected from the set of 8,000 background images. The dataset has 277 K images with thousands of images for each language.

UnrealText. [11] proposes a method to generate synthetic scene text images. With the help of this approach, the authors also generate a multilingual version with 600K images containing 10 languages as included in MLT19. Text contents are sampled from corpus extracted from the Wikimedia dump.

4.2 Data Preprocessing

Data Filtering. We discard images with widths shorter than 32 pixels as they are too blurry. Also, we eliminate the images with empty labels. After filtering, UnrealText, SynthTextMLT and MLT19 have 2.88 M, 886 K and 86 K respectively.

Script Reclassification. First, following the principle of [11], we randomly select 1500 images from each of the nine scripts in the MLT19 to ensure the test samples follow the original script classification. After that, we reclassify the remaining data according to the Unicode of characters in the text label.

Data Augmentation. We first normalize the height of all images to 32 pixels, keeping the aspect ratio. Then, we use LDA to expand the data of all scripts to the same amount as Latin. For the UnrealText and SynthTextMLT, we add some extra data from MLT19 for concatenation to obtain more real scene textures.

4.3 Comparisons with Other Methods

As shown in Table 1, we compare our results against existing methods that perform well in English, as well as some classical methods. To strictly perform a fair comparison, we reproduce these methods using the code provided by the MMOCR [8], which shares the same experiment configuration with LLRN. These methods are listed in the first column. Among them, SATRN is the small model mentioned in [9], and ABINet does not use language model.

We observe that the average recognition accuracy (Mean) of our method outperforms the other methods in all benchmarks, even without using LDA to augment the dataset. LLRN improves upon the second best method (SATRN) with 2.1% on average. After adding LDA, the accuracy is further improved, outperforming the second best method by a large margin of 5.47% on average.

Table 1. Multilingual scene text recognition results (word level accuracy) on the MLT19 dataset. The title of the last column "Mean" means the average recognition accuracy across the nine scripts. "Ours" represents the results of LLRN without LDA, while "Ours*" represents the results of LLRN with LDA.

	Arabic	Bangla	Chinese	Latin	Hindi	Korean	Japanese	Mixed	Symbols	Mean
CRNN [15]	0.93	19.80	45.40	69.00	23.87	63.13	42.20	27.39	21.90	34.85
NRTR [14]	60.80	52.73	69.60	80.80	52.27	72.33	56.60	41.74	18.25	56.12
SAR [10]	64.00	55.20	68.87	82.53	57.33	74.33	56.93	37.39	18.98	57.28
RS [18]	48.67	47.00	66.87	80.87	50.67	72.47	52.67	40.00	18.73	53.11
SATRN [9]	67.40	54.13	73.47	**84.93**	60.80	75.80	56.67	**45.65**	16.55	59.49
Master [12]	59.80	50.80	67.73	82.27	55.07	73.40	55.87	37.39	17.15	55.50
ABINet [2]	59.67	31.20	63.07	82.67	40.33	73.60	50.87	44.78	20.68	51.87
Ours	**69.93**	**59.07**	73.33	84.33	**66.53**	**75.93**	**57.93**	44.35	**22.87**	**61.59**
Ours*	**75.40**	**71.53**	**76.00**	84.47	**76.40**	**78.07**	**61.00**	43.04	18.73	**64.96**

However, LLRN is not outstanding in Latin recognition. We conjecture that, for language-insensitive recognition methods, the network is more inclined to learn features of Latin images, as Latin images are extremely abundant. In our method, due to the addition of language constraints, the overfitting of Latin is alleviated to a certain extent, resulting in reduced recognition accuracy.

Moreover, we also noticed that the accuracy of Mixed decreased, which is quite reasonable. Although we do not have a rigid restriction that output characters in a word must belong to the same language, each language-bound linear layer prefers to map output to characters in the corresponding language rather than in other languages.

In addition, we find that the CTC-based CRNN method is almost completely invalid in Arabic recognition. CTC-based methods rely heavily on character order, as it is decoded sequentially in time steps and the time steps are independent of each other. While words of most languages are written from left to right, Arabic is the opposite, written from right to left. Therefore, CRNN can hardly recognize Arabic texts when all languages are mixed for training. We conclude that the CTC-based method is not suitable for mixed training of languages with different writing orders.

4.4 Ablation Experiments

We perform ablation experiments on the three proposed improvements, including Language-bound Decoder (LD), Language-balanced Data Augmentation (LDA), and Length-sensitive Encoder (LE). We add these three components to the baseline in turn to evaluate their significance. Table 2 reports the recognition accuracy when each module is added.

Table 2. Ablation study result. "LD" means that we replace the decoder of baseline with the Language-bound Decoder. "LE" means that we replace the encoder of baseline with the Length-sensitive Encoder. "LDA" means that we use Language-balanced Data Augmentation.

LD	LDA	LE	Arabic	Bangla	Chinese	Latin	Hindi	Korean	Japanese	Mixed	Symbols	Mean
			64.27	54.87	73.20	82.67	59.47	74.87	57.53	**45.65**	20.56	59.23
✓			68.27	56.20	73.27	81.80	65.47	74.93	56.27	43.48	**26.89**	60.73
✓	✓		74.13	71.00	75.40	82.80	73.40	76.73	60.20	39.57	12.53	62.86
✓	✓	✓	**75.40**	**71.53**	**76.00**	**84.47**	**76.40**	**78.07**	**61.00**	43.04	18.73	**64.96**

Baseline. The baseline is similar to SATRN [9], but slightly different. We replace the backbone of SATRN with a ResNet as same as it in ABINet [2]. Then we set the channel dimensions in all layers to 512. The number of encoder layers is reduced to 1, and the number of decoder layers is reduced to 3. Moreover, there is no SIM in the baseline.

Impact of LD. In LD ticked row, we replace the decoder of baseline with the Language-bound Decoder (LD) and add SIM to the baseline. The experimental results show that the average accuracy increases by 1.5% after using the LD. However, on the one hand, we observe a slight decrease in accuracy for Latin and Japanese compared to baseline, by 0.87% and 1.26%, respectively. Our explanation is that part of the reason is the introduction of SIM, which will produce recognition errors caused by wrong script prediction. In addition, the network tends to predict Latin characters for uncertain characters in mixed training due to plentiful Latin data. This tendency is somewhat weakened by the addition of SIM. The accuracy of Mixed also decreases for the same reason as described earlier in Sect. 4.4. On the other hand, we also find that the accuracy of Arabic, Hindi and Symbols improve greatly, by 4%, 6%, and 6.33%, respectively. This shows our advantage in distinguishing languages and introducing LD.

Impact of LDA. In LDA ticked row, we augment the dataset with the Language-balanced Data Augmentation (LDA). We believe that different languages have different characteristics. Due to data imbalance, the features extracted by the network will be more inclined to Latin with a large amount of data, while other languages will face the problem of insufficient training. At the same time, SIM may habitually predict text to be Latin based on previous experience. By LDA, all languages achieve the same amount of text as Latin, which not only ensures data balance but also expands the dataset. The experimental results show that this simple data augmentation method results in an average improvement of 2.13%. However, as mentioned before, Mixed and Symbols do not have exclusive training data. As the amount of data in other scripts increases, it will be more difficult for the network to recognize unfamiliar text images. Therefore, the results of Mixed and Symbols decrease.

Impact of LE. In LE ticked row, the encoder is changed into the Length-sensitive Encoder (LE). Note that we only change the structure of the encoder, while the number of layers and channel dimensions remain the same. Since almost all images in Symbols are short-text images, the Symbols increases greatly by 6.2%, which shows the effectiveness of LE. Meanwhile, the average accuracy increases by 2.1% after using LE. This is also the final result of our method.

5 Conclusion

In this paper, we propose a new multilingual recognition network LLRN based on the attention mechanism, which can be used as the baseline of future multilingual recognition research. The LLRN is 1) length-sensitive that adapts to different lengths of text images; 2) language-bound that fits for characteristics of different languages. Based on the LLRN, we further propose a language-balanced data augmentation approach to expand the multilingual dataset and solve the problem of data imbalance between different languages. Experiments show that we achieve the best results on the MLT19 dataset. In the future, we will continue to do research on multilingual text recognition and try to apply the method that proved to be good in Latin recognition to more languages.

Acknowledgments. This work was supported by Strategic Priority Research Program of the Chinese Academy of Sciences (XDC08020400).

References

1. Bušta, M., Patel, Y., Matas, J.: E2E-MLT - an unconstrained end-to-end method for multi-language scene text. In: Carneiro, G., You, S. (eds.) ACCV 2018. LNCS, vol. 11367, pp. 127–143. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-21074-8_11
2. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7098–7107 (2021)
3. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
4. Gulati, A., et al.: Conformer: convolution-augmented transformer for speech recognition. arXiv preprint arXiv:2005.08100 (2020)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Huang, J., et al.: A multiplexed network for end-to-end, multilingual OCR. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4547–4557 (2021)

8. Kuang, Z., et al.: Mmocr: a comprehensive toolbox for text detection, recognition and understanding. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 3791–3794 (2021)
9. Lee, J., Park, S., Baek, J., Oh, S.J., Kim, S., Lee, H.: On recognizing texts of arbitrary shapes with 2d self-attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 546–547 (2020)
10. Li, H., Wang, P., Shen, C., Zhang, G.: Show, attend and read: A simple and strong baseline for irregular text recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 8610–8617 (2019)
11. Long, S., Yao, C.: Unrealtext: synthesizing realistic scene text images from the unreal world. arXiv preprint arXiv:2003.10608 (2020)
12. Lu, N., Yu, W., Qi, X., Chen, Y., Gong, P., Xiao, R., Bai, X.: Master: multi-aspect non-local network for scene text recognition. Pattern Recogn. **117**, 107980 (2021)
13. Nayef, N., et al.: ICDAR 2019 robust reading challenge on multi-lingual scene text detection and recognition–RRC-MLT-2019. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1582–1587. IEEE (2019)
14. Sheng, F., Chen, Z., Xu, B.: NRTR: a no-recurrence sequence-to-sequence model for scene text recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 781–786. IEEE (2019)
15. Shi, B., Bai, X., Yao, C.: An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **39**(11), 2298–2304 (2016)
16. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: Aster: An attentional scene text recognizer with flexible rectification. IEEE Trans. Pattern Anal. Mach. Intell. **41**(9), 2035–2048 (2018)
17. Vaswani, A., et al.: Attention is all you need. In: 30th Advances in Neural Information Processing Systems (2017)
18. Yue, X., Kuang, Z., Lin, C., Sun, H., Zhang, W.: RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 135–151. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_9
19. Zhang, X., Zhu, B., Yao, X., Sun, Q., Li, R., Yu, B.: Context-based contrastive learning for scene text recognition. In: AAAI (2022)