Multi-Granularity Prediction for Scene Text Recognition

Peng Wang [⋆], Cheng Da[⋆], and Cong Yao[†]

Alibaba DAMO Academy, Beijing, China
{wdp0072012,dc.dacheng08,yaocong2010}@gmail.com

Abstract. Scene text recognition (STR) has been an active research topic in computer vision for years. To tackle this challenging problem, numerous innovative methods have been successively proposed and incorporating linguistic knowledge into STR models has recently become a prominent trend. In this work, we first draw inspiration from the recent progress in Vision Transformer (ViT) to construct a conceptually simple yet powerful vision STR model, which is built upon ViT and outperforms previous state-of-the-art models for scene text recognition, including both pure vision models and language-augmented methods. To integrate linguistic knowledge, we further propose a Multi-Granularity Prediction strategy to inject information from the language modality into the model in an *implicit* way, *i.e.*, subword representations (BPE and WordPiece) widely-used in NLP are introduced into the output space, in addition to the conventional character level representation, while no independent language model (LM) is adopted. The resultant algorithm (termed MGP-STR) is able to push the performance envelop of STR to an even higher level. Specifically, it achieves an average recognition accuracy of 93.35% on standard benchmarks. Code is available at https://github.com/AlibabaResearch/AdvancedLiterateMachinery/tree/main/OCR/MGP-STR.

Keywords: Scene Text Recognition, ViT, Multi-Granularity Prediction

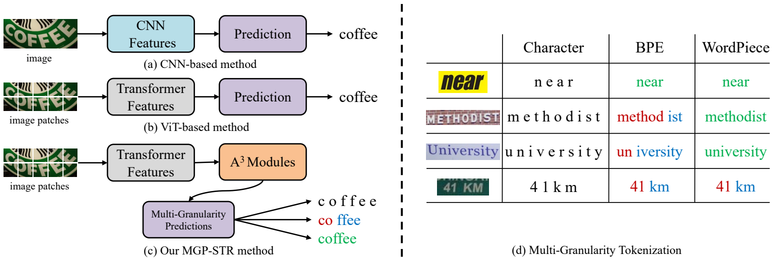
1 Introduction

Reading text from natural scenes is one of the most indispensable abilities when building an automated machine with high-level intelligence. This explains the reason why researchers from the computer vision community sedulously have explored and investigated this complex and challenging task for decades. Scene text recognition (STR) involves decoding textual content from natural images (usually cropped sub images), which is a key component in text reading pipelines.

Previously, a number of methods [39,5,41,30] have been proposed to address the problem of scene text recognition. Recently, there emerges a new trend that linguistic knowledge is introduced into the text recognition process. SRN [53] devised a global semantic reasoning module (GSRM) to model global semantic context. ABINet [9] proposed bidirectional cloze network (BCN) as the language

[⋆] Equal contribution. [†] Corresponding author.

model to learn bidirectional feature representation. Both SRN and ABINet adopt an independent and separate language model to capture rich language prior.



(a) CNN-based method

(b) ViT-based method

(c) Our MGP-STR method

	Character	BPE	WordPiece
near	n e a r	near	near
METHODIST	m e t h o d i s t	method ist	methodist
University	u n i v e r s i t y	un iversity	university
41 KM	4 1 k m	41 km	41 km

(d) Multi-Granularity Tokenization

Fig. 1. Pipelines of classic CNN-based, ViT-based and the proposed MGP-STR scene text recognition methods are illustrated in (a), (b) and (c), respectively. (d) Examples of Character, BPE and WordPiece subword tokenization. (Best viewed in color.)

In this paper, we propose to integrate linguistic knowledge in an ***implicit*** way for scene text recognition. Specifically, we first construct a pure vision STR model based on ViT [8] and a tailored Adaptive Addressing and Aggregation (A^3) module inspired by TokenLearner [36]. This model serves as a strong baseline, which already achieves better performance than previous methods for scene text recognition, according to the experimental comparisons. To further make use of linguistic knowledge to enhance the vision STR model, we explore a Multi-Granularity Prediction (MGP) strategy to inject information from the language modality. The output space of the model is expanded that subword representations (BPE and WordPiece) are introduced, *i.e.*, the augmented model would produce two extra subword-level predictions, besides the original character-level prediction. Notably, there is no independent and separate language model. In the training phase, the resultant model (named MGP-STR) is optimized with a standard multi-task learning paradigm (three losses for three types of predictions) and the linguistic knowledge is naturally integrated into the ViT-based STR model. In the inference phase, the three types of predictions are fused to give the final prediction result. Experiments on standard benchmarks verify that the proposed MGP-STR algorithm can obtain state-of-the-art performance. Another advantage of MGP-STR is that it does not involve iterative refinement, which could be time-consuming in the inference phase. The pipeline of the proposed MGP-STR algorithm as well as that of previous CNN-based and ViT-based methods are shown in Fig. 1. In a nutshell, the major difference between MGP-STR and other methods is that it generates three types of predictions, representing textual information at different granularities: from individual characters to short character combinations, and even whole words.

The contributions of this work are summarized as follows: (1) We construct a pure vision STR model, which combines ViT with a specially designed A^3 module. It already outperforms existing methods. (2) We explore an implicit way for incorporating linguistic knowledge by introducing subword representations to facilitate multi-granularity prediction, and prove that an independent language model (as used in SRN and ABINet) is not indispensable for STR models. (3) The proposed MGP-STR algorithm achieves state-of-the-art performance.

2 Related Work

Scene Text Recognition (STR) is a long-term subject of attention and research [58,28,4]. With the popularity of deep learning methods [42,13,21], its effectiveness in the field of STR has been extensively verified. Depending on whether linguistic information is applied, we roughly divide STR methods into two categories, *i.e.*, language-free and language-augmented methods.

2.1 Language-Free STR Methods

The mainstream way for image feature extraction in STR methods is CNN [42,13]. For example, previous STR methods [39,40,21] utilize VGG. Current STR methods [3,26,2,48] employ ResNet [13] for better performance. Based on the powerful CNN features, various methods [57,33,25] are proposed to tackle the STR problem. CTC-based methods [39,46,26,15,14] use the Connectionist Temporal Classication (CTC) [10] to accomplish sequence recognition. Segmentation-based methods [24,47,23,45] cast STR as a semantic segmentation problems.

Inspired by the great success of Transformer [44] in natural language processing (NLP) tasks, the application of Transformer in STR has also attracted more attention. Vision Transformer (ViT) [8] that directly processes image patches without convolutions opens the beginning of using Transformer blocks instead of CNNs to solve computer vision problems [27,52], leading to prominent results. ViTSTR [1] attempts to simply leverage the feature representations of the last layer of ViT for parallel character decoding. In general, language-free methods often fail to recognize low-quality images due to the lack of language information.

2.2 Language-Augmented STR Methods

Obviously, language information is favourable to the recognition of low-quality images. RNN-based methods [39,21,48] can effectively capture the dependency between sequential characters, which can be regarded as an implicit language model. However, they cannot execute decoding in parallel during training and inference. Recently, Transformer blocks are introduced into CNN-based framework to facilitate language content learning. SRN [53] proposes a Global Semantic Reasoning Module (GSRM) to capture the global semantic context through multiple parallel transmissions. ABINet [9] presents a Bidirectional Cloze Network (BCN) to explicitly model the language information, which is further used for iterative correction. VisionLAN [51] proposes a visual reasoning module that simultaneously captures visual and language information by masking input images at the feature level. The mentioned above approaches utilize a specific module to integrate language information. Meanwhile, most works [16,9] capture semantic information based on character-level or word-level. In this paper,

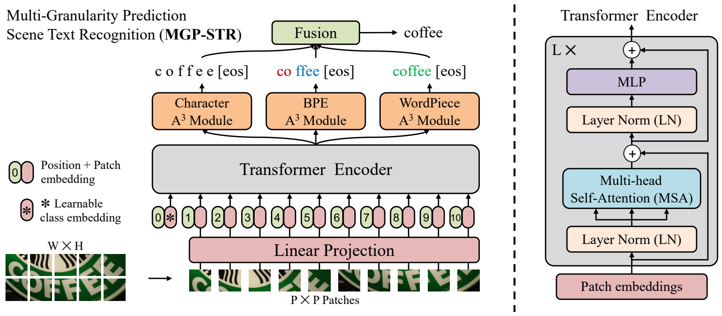


Fig. 2. The architecture of the proposed MGP-STR algorithm.

we manage to utilize multi-granularity (character, subword and even word) semantic information based on BPE and WordPiece tokenizations.

3 Methodology

The overview of the proposed MGP-STR method is depicted in Fig. 2, which is mainly built upon the original Vision Transformer (ViT) model [8]. We propose a tailored Adaptive Addressing and Aggregation (A^3) module to select a meaningful combination of tokens from ViT and integrate them into one output token corresponding to a specific character, denoted as Character A^3 module. Moreover, subword classification heads based on BPE A^3 module and WordPiece A^3 module are devised for subword predictions, so that the language information can be implicitly modelled. Finally, these multi-granularity predictions are merged via a simple and effective fusion strategy.

3.1 Vision Transformer Backbone

The fundamental architecture of MGP-STR is Vision Transformer [8,43], where the original image patches are directly utilized for image feature extraction by linear projection. As shown in Fig. 2, an input RGB image $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$ is split into non-overlapping patches. Concretely, the image is reshaped into a sequence of flattened 2D patches $\mathbf{x}_p \in \mathbb{R}^{N \times (P^2 C)}$, where $(P \times P)$ is the resolution of each image patch and $(P^2 C)$ is the number of feature channels of $\mathbf{x}_p$. In this way, a 2D image is represented as a sequence with $N = HW/P^2$ tokens, which serve as the effective input sequence of Transformer blocks. Then, these tokens of $\mathbf{x}_p$ are linear transcribed into D dimension patch embeddings. Similar to the original ViT [8] backbone, a learnable [*class*] token embedding with D dimension is introduced into patch embeddings. And position embeddings are also added to each patch embedding to retain the positional information, where the standard learnable $1D$ position embedding is employed. Thus, the generation of patch embedding vector is formulated as follows:

$$\mathbf{z}_0 = [\mathbf{x}_{class}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \ldots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{pos}, \tag{1}$$

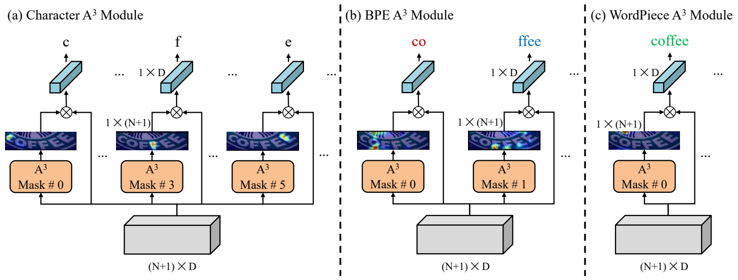


Fig. 3. The detailed architectures of the three A^3 modules.

where $\mathbf{x}_{class} \in \mathbb{R}^{1\times D}$ is the $[class]$ embedding, $\mathbf{E} \in \mathbb{R}^{(P^2C)\times D}$ is a linear projection matrix and $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$ is the position embedding.

The resultant feature sequence $\mathbf{z}_0 \in \mathbb{R}^{(N+1)\times D}$ serves as the input of Transformer encoder blocks, which are mainly composed of Multi-head Self-Attention (MSA), Layer Normalization (LN), Multilayer Perceptron (MLP) and residual connection as in Fig.2. The Transformer encoder block is formulated as:

$$\begin{aligned} \mathbf{z}'_l &= \mathrm{MSA}(LN(\mathbf{z}_{l-1})) + \mathbf{z}_{l-1} \\ \mathbf{z}_l &= \mathrm{MLP}(LN(\mathbf{z}'_l)) + \mathbf{z}'_l. \end{aligned} \tag{2}$$

Here, L is the depth of Transformer block and $l = 1 \ldots L$. The MLP consists of two linear layers with GELU activation. Finally, the output embedding $\mathbf{z}_L \in \mathbb{R}^{(N+1)\times D}$ of Transformer is utilized for subsequent text recognition.

3.2 Adaptive Addressing and Aggregation (A^3) Modules

Traditional Vision Transformers [8,43] usually prepend a learnable $\mathbf{x}_{class}$ token to the sequence of patch embeddings, which directly collects and aggregates the meaningful information and serves as the image representation for the classification of the whole image. While the task of scene text recognition aims to produce a sequence of character predictions, where each character is only related to a small patch of the image. Thus, the global image representation $\mathbf{z}_L^0 \in \mathbb{R}^D$ is inadequate for text recognizing task. ViTSTR [1] directly employs the first T tokens of $\mathbf{z}_L$ for text recognition, where T is the maximum text length. Unfortunately, the rest tokens of $\mathbf{z}_L$ are not fully utilized.

In order to take full advantage of the rich information of the sequence $\mathbf{z}_L$ for text sequence prediction, we propose a tailored Adaptive Addressing and Aggregation (A^3) module to select a meaningful combination of tokens $\mathbf{z}_L$ and integrate them into one token corresponding to a specific character. Specifically, we manage to learn T tokens $\mathbf{Y} = [\mathbf{y}_i]_{i=1}^T$ from the sequence $\mathbf{z}_L$ for the subsequent text recognizing task. An aggregation function is, thus, formulated as $\mathbf{y}_i = A_i(\mathbf{z}_L)$, which converts the input $\mathbf{z}_L$ to a token vector $\mathbf{y}_i : \mathbb{R}^{(N+1)\times D} \mapsto \mathbb{R}^{1\times D}$. And such T functions are constructed for the sequential output of text recognition.

Typically, the aggregation function $A_i(\mathbf{z}_L)$ is implemented via a spatial attention mechanism [36] to adaptively select the tokens from $\mathbf{z}_L$ corresponding to i_{th} character. Here, we employ function $\alpha_i(\mathbf{z}_L)$ and softmax function to generate precise spatial attention mask $\mathbf{m}_i \in \mathbb{R}^{(N+1)\times 1}$ from $\mathbf{z}_L \in \mathbb{R}^{(N+1)\times D}$. Thus, each output token $\mathbf{y}_i$ of A^3 module is produced by

$$\mathbf{y}_i = A_i(\mathbf{z}_L) = \mathbf{m}_i^T \tilde{\mathbf{z}}_L = \text{softmax}(\alpha_i(\mathbf{z}_L))^T (\mathbf{z}_L \mathbf{U})^T. \tag{3}$$

Here, $\alpha_i(\cdot)$ is implemented by group convolution with one 1×1 kernel. And $\mathbf{U} \in \mathbb{R}^{D\times D}$ is a linear mapping matrix for learning feature $\tilde{\mathbf{z}}_L$. Therefore, the resulting tokens of different aggregation functions are gathered together to form the final output tensor as follows:

$$\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2; \ldots; \mathbf{y}_T] = [A_1(\mathbf{z}_L); A_2(\mathbf{z}_L); \ldots; A_T(\mathbf{z}_L)]. \tag{4}$$

Owing to the effective and efficient A^3 module, the ultimate representation of the text sequence is denoted as $\mathbf{Y} \in \mathbb{R}^{T\times D}$ in Eq. (4). Then, a character classification head is built by $\mathbf{G} = \mathbf{Y}\mathbf{W}^T \in \mathbb{R}^{T\times K}$ for text sequence recognition, where $\mathbf{W} \in \mathbb{R}^{K\times D}$ is a linear mapping matrix, K is the number of categories and $\mathbf{G}$ is the classification logist. We regard this module as Character A^3 for character-level prediction, of which the detailed structure is illustrated in Fig. 3(a).

3.3 Multi-Granularity Predictions

Character tokenization that simply splits text into characters is commonly-used in scene text recognition methods. However, this naive and standard way ignores the language information of text. In order to effectively resort to linguistic information for scene text recognition, we incorporate subword [20] tokenization mechanism in NLP [7] into text recognition method. Subword tokenization algorithms aim to decompose rare words into meaningful subwords and remain frequently used words, so that the grammatical information of word has already been captured in the subwords. Meanwhile, since A^3 module is independent of Transformer encoder backbone, we can directly add extra parallel subword A^3 modules for subword predictions. In such a way, the language information can be implicitly injected into model learning for better performance. Notably, previous methods, *i.e.*, SRN [53] and ABINet [9], design an explicit transformer module for language modelling, while we cast linguistic information encoding problem as a character and subword prediction task without an explicit language model.

Specifically, we employ two subword tokenization algorithms Byte-Pair Encoding (BPE) [38] and WordPiece [37] [1] to produce various combinations as shown in Fig.1(b)(c). Thus, BPE A^3 module and WordPiece A^3 module are proposed for subword attention. And two subword-level classification heads are used for subword predictions. Since subwords could be whole words (such as “coffee” in

[1] Considering the potential out-of-vocabulary (OOV) issue in the inference phase, we did not directly predict whole words.

WordPiece), subword-level and even word-level predictions can be generated by the BPE and WordPiece classification heads. Along with the original character-level prediction, we denote these various outputs as multi-granularity predictions for text recognition. In this way, character-level prediction guarantees the fundamental recognition accuracy, and subword-level or word-level predictions can serve as complementary results for noised images via linguistic information.

Technically, the architecture of BPE or WordPiece A^3 module is the same as Character one. They are independent of each other with different parameters. And the numbers of categories are different for different classification heads, which depend on the vocabulary size of each tokenization method. The cross entropy loss is employed for classification. Additionally, the mask $\mathbf{m}_i$ precisely indicates the attention location of the i_{th} character in Character A^3 module, while it roughly shows the i_{th} subword region of the image in subword A^3 modules, due to the higher complexity and uncertainty of learning subword splitting.

3.4 Fusion Strategy for Multi-Granularity Results

Multi-granularity predictions (Character, BPE and WordPiece) are generated by different A^3 modules and classification heads. Thus, a fusion strategy is required to merge these results. At the beginning, we attempt to fuse multi-granularity information by aggregating text features $\mathbf{Y}$ of the output of different A^3 modules at feature level. However, since these features are from different granularities, the i_{th} token $\mathbf{y}_i^{char}$ of character level is not aligned with the i_{th} token $\mathbf{y}_i^{bpe}$ (or $\mathbf{y}_i^{wp}$) of BPE level (or WordPiece level), so that these features cannot be added for fusion. Meanwhile, even if we concatenate features by $[\mathbf{Y}_i^{char}, \mathbf{Y}_i^{bpe}, \mathbf{Y}_i^{wp}]$, only one character-level head can be used for final prediction. The subword information will be greatly impaired in this way, resulting in less improvement.

Therefore, decision-level fusion strategy is employed in our method. However, perfectly fusing these predictions is a challenging problem [11]. We, thus, propose a compromised but efficient fusion strategy based on the prediction confidences. Specifically, the recognition confidence of each character or subword can be obtained by the corresponding classification head. Then, we present two fusion functions $f(\cdot)$ to produce the final recognition score based on atomic confidences:

$$f_{Mean}([c_1, c_2, \ldots, c_{eos}]) = \frac{1}{n}\sum_{i=1}^{eos} c_i, \tag{5}$$

$$f_{Cumprod}([c_1, c_2, \ldots, c_{eos}]) = \prod_{i=1}^{eos} c_i. \tag{6}$$

We only consider the confidence of valid character or subword and ending symbol *eos*, and ignore padding symbol *pad*. "Mean" recognition score is generated by the mean value function as in Eq. (5). And "Cumprod" represents the score produced by cumulative product function. Then, three recognition scores of three classification heads for one image can be obtained by $f(\cdot)$. We simply pick the one with the highest recognition score as the the final predicted result.

4 Experiment

4.1 Datasets

For fair comparison, we use MJSynth [16,17] and SynthText [12] as training data. MJSynth contains $9M$ realistic text images and SynthText includes $7M$ synthetic text images. The test dataset consists of "regular" and "irregular" datasets. The "regular" dataset is mainly composed of horizontally aligned text images. IIIT 5K-Words (IIIT) [31] consists of 3,000 images collected on the website. Street View Text (SVT) [49] contains 647 test images. ICDAR 2013 (IC13) [19] contains 1,095 images cropped from mall pictures, but we eventually evaluate on 857 images, discarding images that contain non-alphanumeric characters or less than three characters. The text instances in the "irregular" dataset are mostly curved or distorted. ICDAR 2015 (IC15) [18] includes 2,077 images collected from Google Eyes, but we use 1,811 images without some extremely distorted images. Street View Text-Perspective (SVTP) [32] contains 639 images collected from Google Street View. CUTE80 (CUTE) [35] consists of 288 curved images.

4.2 Implementation Details

Model Configuration. MGP-STR is built upon DeiT-Base model [43], which is composed of 12 stacked transformer blocks. For each layer, the number of head is 12 and the embedding dimension D is 768. More importantly, square 224×224 images [8,43,1] are not adopted in our method. The height H and width W of the input image are set to 32 and 128. The patch size P is set to 4 and thus $N = 8 \times 32 = 256$ plus one $[class]$ tokens $\mathbf{z}_L \in \mathbb{R}^{257 \times 768}$ can be produced. The maximum length T of the output sequence $\mathbf{Y}$ of A^3 module is set to 27. The vocabulary size K of Character classification head is set to 38, including $0-9$, $a-z$, *pad* for padding symbol and *eos* for ending symbol. The vocabulary sizes of BPE and WordPiece heads are set to $50,257$ and $30,522$.

Model Training. The pretrained weights of DeiT-base [43] are loaded the initial weights, except the patch embedding model, due to inconsistent patch sizes. Common data augmentation methods [6] for text image are applied, such as perspective distortion, affine distortion, blur, noise and rotation. We use 2 NVIDIA Tesla V100 GPUs to train our model with a batch size of 100. Adadelta [55] optimizer is employed with an initial learning rate of 1. The learning rate decay strategy is Cosine Annealing LR [29] and the training lasts 10 epochs.

4.3 Discussions on Vision Transformer and A^3 Modules

We analyse the influence of the patch size of Vision Transformer and the effectiveness of A^3 module in the proposed MGP-STR method (shown in Table 1). MGP-STR$_{P=16}$ represents the model that simply uses the first T tokens of $\mathbf{z}_L$ for text recognition as in ViTSTR [1], where the input image is reshaped to

Table 1. The ablation study of the proposed vision model and the accuracy comparisons with some SOTA methods based on only vision information.

Methods	Vision	Image size (Patch)	IC13	SVT	IIIT	IC15	SVTP	CUTE	AVG
MASTER [30]	CNN	-	95.3	90.60	95.0	79.4	84.5	87.5	89.5
SRN$_V$ [53]		-	93.2	88.1	92.3	77.5	79.4	84.7	86.9
ABINet$_V$ [9]		-	94.9	90.4	94.6	81.7	84.2	86.5	89.8
MGP-STR$_{P=16}$	ViT	$224 \times 224 (16 \times 16)$	95.68	91.96	95.13	83.88	85.74	90.28	91.07
MGP-STR$_{P=4}$		$32 \times 128 (4 \times 4)$	96.62	92.27	95.40	84.76	86.98	88.54	91.58
MGP-STR$_{Vision}$		$32 \times 128 (4 \times 4)$	96.50	93.20	96.37	86.25	89.46	90.63	92.73

Table 2. The accuracies of MGP-STR$_{Fuse}$ with different fusion strategies.

Method	Mode	IC13	SVT	IIIT	IC15	SVTP	CUTE	AVG
MGP-STR$_{Fuse}$	Char	96.49	93.66	96.1	86.14	88.83	89.58	92.53
	Mean	97.31	94.28	96.60	86.97	90.23	90.97	93.28
	Cumprod	97.32	94.74	96.40	87.24	91.01	90.28	93.35

224 × 224 and the patch size is set to 16 × 16. In order to retain the significant information of the original text image, 32 × 128 images with 4 × 4 patches are employed in MGP-STR$_{P=4}$. MGP-STR$_{P=4}$ outperfrrms MGP-STR$_{P=16}$, which indicates that the standard image size of ViT [8,43] is incompatible with text recognition. Thus, 32 × 128 images with 4 × 4 patches are used in MGP-STR.

When the Character A^3 module is introduced into MGP-STR, denoted as MGP-STR$_{Vision}$, the recognition performance will be further improved. MGP-STR$_{P=16}$ and MGP-STR$_{P=4}$ cannot fully learn and utilize the all tokens, while the Character A^3 module can adaptively aggregate features of the last layer, resulting in more sufficient learning and higher accuracy. Meanwhile, compared with SOTA text recognition methods with CNN feature extractors, the proposed MGP-STR$_{Vision}$ method achieves substantially performance improvement.

4.4 Discussions on Multi-Granularity Predictions

Effect of Fusion Strategy. Since the subwords generated by subword tokenization methods contain grammatical information, we directly employ subwords as the targets of our method to capture the language information implicitly. As described in Sec. 3.2, two different subword tokenizations (BPE and WordPiece) are employed for complementary multi-granularity predictions. Besides the character prediction, we propose two fusion strategies to further merge these three results, denoted as "Mean" and "Cumprod" as mentioned in Sec. 3.4. We denote this method that merges three results as MGP-STR$_{Fuse}$, and the accuracy results of MGP-STR$_{Fuse}$ with different fusion strategies are listed in Table 2. Additionally, the first line "Char" in Table 2 records the result of character classification head in MGP-STR$_{Fuse}$. It is clear to see that both "Mean" and "Cumprod" fusion strategies can significantly improve the recognition accuracy over single character-level result. Due the better performance of "Cumprod" strategy, we employ it as the fusion strategy in the following experiments.

Table 3. The accuracy results of the four variants of MGP-STR model. “Char”, “BPE” and “WP” at “Output” represent predictions of Character, BPE and WordPiece classification head in each model, respectively. “Fuse” represents the fused results.

Methods	Char	BPE	WP	Output	IC13	SVT	IIIT	IC15	SVTP	CUTE	AVG
MGP-STR$_{Vision}$	✓	×	×	Char	96.50	93.20	96.37	86.25	89.46	90.63	92.73
MGP-STR$_{C+B}$	✓	✓	×	Char	97.43	93.82	96.53	85.92	89.15	90.28	92.84
				BPE	97.78	94.13	90.00	81.12	88.37	82.64	88.63
				Fuse	97.67	94.47	96.73	86.97	88.99	89.93	93.24
MGP-STR$_{C+W}$	✓	×	✓	Char	96.97	93.97	96.30	86.20	90.39	89.93	92.87
				WP	95.92	93.35	87.70	78.74	89.30	80.21	86.78
				Fuse	97.32	93.82	96.60	86.91	90.54	90.97	93.25
MGP-STR$_{Fuse}$	✓	✓	✓	Char	96.49	93.66	96.10	86.14	88.83	89.58	92.53
				BPE	95.56	93.66	88.73	79.84	89.76	83.33	87.63
				WP	95.79	94.59	86.37	77.36	89.61	79.86	85.99
				Fuse	97.32	94.74	96.40	87.24	91.01	90.28	93.35

Effect of Subword Representations. We evaluate four variants of the MGP-STR model, and the performances of these four methods are elaborately reported in Table 3, including the fused results and the results of each single classification. Specifically, MGP-STR$_{Vision}$ with only Character A^3 module has already obtained promising results. MGP-STR$_{C+B}$ and MGP-STR$_{C+W}$ incorporate Character A^3 module with BPE A^3 module and WordPiece A^3 module, respectively. No matter which subword tokenization is used alone, the accuracy of “Fuse” can exceed that of “Char” in both MGP-STR$_{C+B}$ and MGP-STR$_{C+W}$ methods, respectively. Notably, the performance of the classification of “BPE” or “WP” could be better than that of “Char” on SVP and SVTP datasets in the same model. These results show that subword predictions can boost text recognition performance by implicitly introducing language information. Thus, MGP-STR$_{Fuse}$ with three A^3 modules can produce complementary multi-granularity predictions (character, subword and even word). By fusing these multi-granularity results, MGP-STR$_{Fuse}$ obtains the best performance.

Comparison with BCN. Bidirectional cloze network (BCN) is designed in ABINet [9] for explicit language modelling, and it leads to favorable improvement over pure vision model. We equip MGP-STR$_{Vision}$ with BCN as a competitor of MGP-STR$_{Fuse}$ to verify the advantage of multi-granularity predictions. Concretely, we first reduce the dimension 768 of representation feature $\mathbf{Y}$ to 512 for feature fusion of the output of BCN. Following the training setting in [9], the model results are reported in Table 4. The accuracy of “V+L” is further imporved over the pure vision prediction “V” in MGP-STR$_{Vision}$+BCN, and better than the original ABINet [9]. However, the performance of MGP-STR$_{Vision}$+BCN is a little worse than that of MGP-STR$_{Fuse}$. In addition, we provide the upper bound on the performance of MGP-STR$_{Fuse}$, denoted as MGP-STR$^{*}_{Fuse}$ in Table 4. If one of the three predictions (“Char”, “BPE” and “WP”) is right, the final prediction is considered correct. The highest score

Table 4. The accuracy results of MGP-STR$_{Vision}$ equipped with BCN and multi-granularity prediction. "V" represents the results of the pure vision output. "V+L" represents the results based on the both vision and language parts.

Methods	Mode	IC13	SVT	IIIT	IC15	SVTP	CUTE	AVG
MGP-STR$_{Vision}$+BCN	V	96.97	93.82	95.90	85.53	89.15	89.58	92.40
	V+L	97.32	95.36	95.97	86.69	91.78	89.93	93.14
MGP-STR$_{Fuse}$	V+L	97.32	94.74	96.40	87.24	91.01	90.28	93.35
MGP-STR$^{*}_{Fuse}$	V+L	97.66	96.29	96.97	89.06	92.09	92.01	94.38

Table 5. The accuracy results of MGP-STR$_{Fuse}$ with different ViT backbones.

Backbone	Output	IC13	SVT	IIIT	IC15	SVTP	CUTE	AVG
DeiT-Tiny	Char	93.47	90.57	93.93	82.94	81.71	84.38	89.36
	BPE	87.40	84.39	83.17	73.72	77.83	71.53	80.48
	WP	53.79	45.44	60.07	52.57	42.79	42.71	53.92
	Fuse	94.05	91.19	94.30	83.38	83.57	84.38	89.91
DeiT-Small	Char	95.92	91.04	94.97	84.59	85.89	86.81	91.01
	BPE	96.27	93.35	89.37	79.74	86.67	82.29	87.61
	WP	75.50	70.48	74.70	66.81	68.06	62.15	71.36
	Fuse	96.38	93.51	95.30	86.09	87.29	87.85	91.96
DeiT-Base	Char	96.49	93.66	96.10	86.14	88.83	89.58	92.53
	BPE	95.56	93.66	88.73	79.84	89.76	83.33	87.63
	WP	95.79	94.59	86.37	77.36	89.61	79.86	85.99
	Fuse	97.32	94.74	96.40	87.24	91.01	90.28	93.35

of MGP-STR$^{*}_{Fuse}$ demonstrates the good potential of multi-granularity predictions. Moreover, MGP-STR$_{Fuse}$ only requires two new subword prediction heads, rather than the design of a specific and explicit language model in [9,53].

4.5 Results with Different ViT Backbones

All of the proposed MGP-STR models mentioned earlier are based on DeiT-Base [43]. We also introduce two smaller models, namely DeiT-Small and DeiT-Tiny as presented in [43] to further evaluate the effectiveness of MGP-STR$_{Fuse}$ method. Specifically, the embedding dimensions of DeiT-Small and DeiT-Tiny are reduced to 384 and 192, respectively. Table 5 records the results of each prediction head of the MGP-STR$_{Fuse}$ method with different ViT backbones. Clearly, fusing multi-granularity predictions can still improve the performance of pure character-level prediction in every backbone. And bigger models achieve better performance in the same head. More importantly, the results of "Char" in DeiT-Small and even DeiT-Tiny have already surpassed the SOTA pure CNN-based vision models, referring to Table 1. Therefore, MGP-STR$_{Vision}$ with small or tiny ViT backbone is also a competitive vision model and multi-granularity prediction can also work well in different ViT backbones.

Table 6. The comparisons with SOTA methods on several public benchmarks.

Methods	Regular Text			Irregular Text			AVG
	IC13	SVT	IIIT	IC15	SVTP	CUTE	
TBRA [2]	93.6	87.5	87.9	77.6	79.2	74.0	84.6
ViTSTR [1]	93.2	87.7	88.4	78.5	81.8	81.3	85.6
ESIR [56]	91.3	90.2	93.3	76.9	79.6	83.3	87.1
DAN [50]	93.9	89.2	94.3	74.5	80.0	84.4	87.2
SE-ASTER [34]	92.8	89.6	93.8	80.0	81.4	83.6	88.3
RobustScanner [54]	94.8	88.1	95.3	77.1	79.5	90.3	88.4
TextScanner [45]	92.9	90.1	93.9	79.4	84.3	83.3	88.5
SATRN [22]	94.1	91.3	92.8	79.0	86.5	87.8	88.6
MASTER [30]	95.3	90.6	95.0	79.4	84.5	87.5	89.5
SRN [53]	95.5	91.5	94.8	82.7	85.1	87.8	90.4
VisionLAN [51]	95.7	91.7	95.8	83.7	86.0	88.5	91.2
ABINet [9]	**97.4**	93.5	96.2	86.0	89.3	89.2	92.6
MGP-STR$_{Vision}$	96.50	93.20	96.37	86.25	89.46	**90.63**	92.73
MGP-STR$_{Fuse}$	97.32	**94.74**	**96.40**	**87.24**	**91.01**	90.28	**93.35**

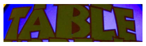
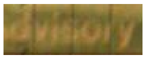
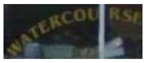
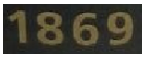
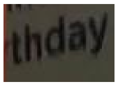
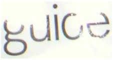
4.6 Comparisons with State-of-the-Arts

We compare the proposed MGP-STR$_{Vision}$ and MGP-STR$_{Fuse}$ methods with previous state-of-the-art scene text recognition methods, and the results on 6 standard benchmarks are summarized in Table 6. All of compared methods and ours are trained on synthetic datasets MJ and ST for fair evaluation. And the results are obtained without any lexicon based post-processing. Generally, language-aware methods (*i.e.*, SRN [53], VisionLAN [51], ABINet [9] and MGP-STR$_{Fuse}$) perform better than other language-free methods, showing the significance of linguistic information. Notably, MGP-STR$_{Vision}$ without any language information has already outperformed the state-of-the-art method ABINet with explicit language model. Owing to the multi-granularity prediction, MGP-STR$_{Fuse}$ obtains more impressive results further, which outperforms ABINet with 0.7% improvement on average accuracy.

4.7 Details of Multi-Granularity Predictions

We show the detailed prediction process of the proposed MGP-STR$_{Fuse}$ method on 6 test images from standard datasets. In the first three images, the results of character-level prediction are incorrect, due to irregular font, motion blur and curved shape, respectively. The scores of character prediction are very low, since the images are difficult to recognize and one character is wrong in each image. However, "BPE" and "WP" heads can recognize "table" image with high scores. And "BPE" can make correct predictions with two subwords on "dvisory" and "watercourse" images, while "WP" is wrong in "watercourse" image. After fusion, the mistakes can be corrected. From the rest three images, interesting phenomena can be observed. The predictions of "Char" and "BPE" conform to the images. The predictions of "WP", however, attempt to produce

Table 7. The details of multi-granularity prediction of MGP-STR$_{Fuse}$, including the scores of each prediction head, the intermediate multi-granularity (Gra.) results and the final prediction (Pred.). Best viewed in color.

Images	GT	Output	Char	BPE	WP	Fuse
	table	Score	0.1643	0.9813	0.9521	0.9813
		Gra.	tabbe	table	table	-
		Pred.	tabbe	table	table	table
	dvisory	Score	0.0316	0.8218	0.2574	0.8218
		Gra.	divsoory	d visory	dvisory	-
		Pred.	divsoory	dvisory	dvisory	dvisory
	watercourse	Score	0.1565	0.8295	0.632	0.8295
		Gra.	watercourss	water course	waterco	-
		Pred.	watercourss	watercourse	waterco	watercourse
	1869	Score	0.9999	0.9207	0.0354	0.9999
		Gra.	1869	18 69	18	-
		Pred.	1869	1869	18	1869
	thday	Score	0.9998	0.5983	0.7638	0.9998
		Gra.	thday	th day	today	-
		Pred.	thday	thday	today	thday
	guide	Score	0.9675	0.6959	0.1131	0.9675
		Gra.	guice	gu ice	guide	-
		Pred.	guice	guice	guide	guice

strings with more linguistic content, like “today” and “guide”. Generally, “Char” aims to produce characters one by one, while “BPE” usually generates n-gram segments related to image and “WP” tends to directly predict words that are linguistically meaningful. These prove the predictions of different granularities convey text information in different aspects and are indeed complementary.

4.8 Visualization of Spatial Attention Maps of A^3 Modules

Exemplar attention maps $\mathbf{m}_i$ of Character, BPE and WordPiece A^3 modules are shown in Fig. 4. Character A^3 module shows extremely precise addressing ability on a variety of text images. Specifically, for the “7” image with one character, the attention mask seems like the “7” shape. For the “day” and “bar” images with three characters, the attention masks of middle character “a” are completely different, verifying the adaptiveness of A^3 module. As depicted in Fig.1(d) and in Table 7, BPE tends to generate short segments, thus the attention masks of BPE are spilt into 2 or 3 areas as shown in “leaves” and “academy” images. This is probably because that performing subword splitting and character addressing simultaneously is difficult. Moreover, WordPiece often produces a whole word, and the attention maps should be the whole feature map. Since the attention maps produced by the softmax function are usually sparse, the attention maps of WordPiece are not as appealing as those of Character A^3 module. These results are consistent to those of Table 3, where the accuracies of “BPE” and “WP” are relatively lower than “Char”, due to the difficulty of precise subword prediction.

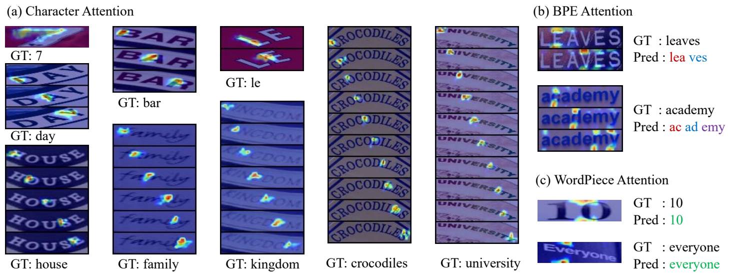


Fig. 4. The illustration of spatial attention masks on Character A^3 module, BPE A^3 module and WordPiece A^3 module, respectively.

4.9 Comparisons of Inference Time and Model Size

Table 8. Comparisons on inference time and model size.

Methods	Time (ms/image)	Parameters (1×10^6)
ABINet-S-iter1/iter2/iter3	13.7/18.6/24.3	32.8
ABINet-L-iter1/iter2/iter3	16.1/21.4/26.8	36.7
MGP-STR$_{Vision}$-tiny/small/base	10.6/10.8/10.9	5.4/21.4/85.5
MGP-STR$_{Fuse}$-tiny/small/base	12.0/12.2/12.3	21.0/52.6/148.0

The model sizes and latencies of the proposed MGP-STR with different settings as well as those of ABINet are depicted in Table. 8 [2]. Since MGP-STR is equipped with a regular Vision Transformer (ViT) and involves no iterative refinement, the inference speed of MGP-STR is very fast: 12.3ms with ViT-Base backbone. Compared with ABINet, MGP-STR runs much faster (12.3ms vs. 26.8ms), while obtaining higher performance. The model size of MGP-STR is relatively large. However, a large portion of the model parameter is from the BPE and WordPiece branches. For the scenarios that are sensitive to model size or with limited memory space, MGP-STR$_{Vision}$ is an excellent choice.

5 Conclusion

We presented a ViT-based pure vision model for STR, which shows its superiority in recognition accuracy. To further promote recognition accuracy of this baseline model, we proposed a Multi-Granularity Prediction strategy to take advantage of linguistic knowledge. The resultant model achieves state-of-the-art performance on widely-used dadatsets. In the future, we will extend the idea of multi-granularity prediction to broader domains.

[2] All the evaluations are conducted on a NVIDIA V100 GPU.

References

1. Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: ICDAR. vol. 12821, pp. 319–334 (2021)
2. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: ICCV. pp. 4714–4722 (2019)
3. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: Large scale system for text detection and recognition in images. In: Guo, Y., Farooq, F. (eds.) SIGKDD. pp. 71–79 (2018)
4. Chen, X., Jin, L., Zhu, Y., Luo, C., Wang, T.: Text recognition in the wild: A survey. ACM Computing Surveys (CSUR) **54**(2), 1–35 (2021)
5. Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., Zhou, S.: Focusing attention: Towards accurate text recognition in natural images. In: CVPR. pp. 5086–5094 (2017)
6. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: CVPR Workshops. pp. 3008–3017 (2020)
7. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT. pp. 4171–4186 (2019)
8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
9. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: CVPR. pp. 7098–7107 (2021)
10. Graves, A., Fernández, S., Gomez, F.J., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: ICML. vol. 148, pp. 369–376 (2006)
11. Gu, J., Meng, G., Da, C., Xiang, S., Pan, C.: No-reference image quality assessment with reinforcement recursive list-wise ranking. In: AAAI. pp. 8336–8343 (2019)
12. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: CVPR. pp. 2315–2324 (2016)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
14. He, P., Huang, W., Qiao, Y., Loy, C.C., Tang, X.: Reading scene text in deep convolutional sequences. In: AAAI. pp. 3501–3508 (2016)
15. Hu, W., Cai, X., Hou, J., Yi, S., Lin, Z.: GTC: guided training of CTC towards efficient and accurate scene text recognition. In: AAAI. pp. 11005–11012 (2020)
16. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. NIPS Deep Learning Workshop (2014)
17. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. Int. J. Comput. Vis. **116**(1), 1–20 (2016)
18. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S.K., Bagdanov, A.D., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., Shafait, F., Uchida, S., Valveny, E.: ICDAR 2015 competition on robust reading. In: ICDAR. pp. 1156–1160 (2015)

19. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazán, J., de las Heras, L.: ICDAR 2013 robust reading competition. In: ICDAR. pp. 1484–1493 (2013)
20. Labeau, M., Allauzen, A.: Character and subword-based word representation for neural language modeling prediction. In: SWCN@EMNLP. pp. 1–13 (2017)
21. Lee, C., Osindero, S.: Recursive recurrent nets with attention modeling for OCR in the wild. In: CVPR. pp. 2231–2239 (2016)
22. Lee, J., Park, S., Baek, J., Oh, S.J., Kim, S., Lee, H.: On recognizing texts of arbitrary shapes with 2d self-attention. In: CVPR Workshops. pp. 2326–2335 (2020)
23. Liao, M., Lyu, P., He, M., Yao, C., Wu, W., Bai, X.: Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. IEEE Trans. Pattern Anal. Mach. Intell. **43**(2), 532–548 (2021)
24. Liao, M., Zhang, J., Wan, Z., Xie, F., Liang, J., Lyu, P., Yao, C., Bai, X.: Scene text recognition from two-dimensional perspective. In: AAAI. pp. 8714–8721 (2019)
25. Liu, H., Wang, B., Bao, Z., Xue, M., Kang, S., Jiang, D., Liu, Y., Ren, B.: Perceiving stroke-semantic context: Hierarchical contrastive learning for robust scene text recognition. In: AAAI. pp. 1702–1710 (2022)
26. Liu, W., Chen, C., Wong, K.K., Su, Z., Han, J.: Star-net: A spatial attention residue network for scene text recognition. In: BMVC (2016)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. CoRR **abs/2103.14030** (2021)
28. Long, S., He, X., Yao, C.: Scene text detection and recognition: The deep learning era. IJCV **129**(1), 161–184 (2021)
29. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: ICLR (2017)
30. 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 Recognition **117**, 107980 (2021)
31. Mishra, A., Alahari, K., Jawahar, C.V.: Scene text recognition using higher order language priors. In: BMVC. pp. 1–11 (2012)
32. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: ICCV. pp. 569–576 (2013)
33. Qiao, Z., Zhou, Y., Wei, J., Wang, W., Zhang, Y., Jiang, N., Wang, H., Wang, W.: Pimnet: A parallel, iterative and mimicking network for scene text recognition. In: ACM MM. pp. 2046–2055 (2021)
34. Qiao, Z., Zhou, Y., Yang, D., Zhou, Y., Wang, W.: SEED: semantics enhanced encoder-decoder framework for scene text recognition. In: CVPR. pp. 13525–13534 (2020)
35. Risnumawan, A., Shivakumara, P., Chan, C.S., Tan, C.L.: A robust arbitrary text detection system for natural scene images. Expert Syst. Appl. **41**(18), 8027–8048 (2014)
36. Ryoo, M.S., Piergiovanni, A.J., Arnab, A., Dehghani, M., Angelova, A.: Tokenlearner: What can 8 learned tokens do for images and videos? CoRR **abs/2106.11297** (2021)
37. Schuster, M., Nakajima, K.: Japanese and korean voice search. In: ICASSP. pp. 5149–5152 (2012)
38. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: ACL. The Association for Computer Linguistics (2016)

39. 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 TPAMI **39**(11), 2298–2304 (2017)
40. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: CVPR. pp. 4168–4176 (2016)
41. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: ASTER: an attentional scene text recognizer with flexible rectification. IEEE TPAMI **41**(9), 2035–2048 (2019)
42. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
43. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. vol. 139, pp. 10347–10357 (2021)
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)
45. Wan, Z., He, M., Chen, H., Bai, X., Yao, C.: Textscanner: Reading characters in order for robust scene text recognition. In: AAAI. pp. 12120–12127 (2020)
46. Wan, Z., Xie, F., Liu, Y., Bai, X., Yao, C.: 2d-ctc for scene text recognition. arXiv preprint arXiv:1907.09705 (2019)
47. Wan, Z., Zhang, J., Zhang, L., Luo, J., Yao, C.: On vocabulary reliance in scene text recognition. In: CVPR. pp. 11422–11431 (2020)
48. Wang, J., Hu, X.: Gated recurrent convolution neural network for OCR. In: NeurIPS. pp. 335–344 (2017)
49. Wang, K., Babenko, B., Belongie, S.J.: End-to-end scene text recognition. In: ICCV. pp. 1457–1464 (2011)
50. Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M.: Decoupled attention network for text recognition. In: AAAI. pp. 12216–12224 (2020)
51. Wang, Y., Xie, H., Fang, S., Wang, J., Zhu, S., Zhang, Y.: From two to one: A new scene text recognizer with visual language modeling network. In: ICCV. pp. 1–10 (2021)
52. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: Segformer: Simple and efficient design for semantic segmentation with transformers. CoRR **abs/2105.15203** (2021)
53. Yu, D., Li, X., Zhang, C., Liu, T., Han, J., Liu, J., Ding, E.: Towards accurate scene text recognition with semantic reasoning networks. In: CVPR. pp. 12110–12119 (2020)
54. Yue, X., Kuang, Z., Lin, C., Sun, H., Zhang, W.: Robustscanner: Dynamically enhancing positional clues for robust text recognition. In: ECCV. vol. 12364, pp. 135–151 (2020)
55. Zeiler, M.D.: ADADELTA: an adaptive learning rate method. CoRR **abs/1212.5701** (2012)
56. Zhan, F., Lu, S.: ESIR: end-to-end scene text recognition via iterative image rectification. In: CVPR. pp. 2059–2068 (2019)
57. Zhang, X., Zhu, B., Yao, X., Sun, Q., Li, R., Yu, B.: Context-based contrastive learning for scene text recognition. In: AAAI. pp. 888–896 (2022)
58. Zhu, Y., Yao, C., Bai, X.: Scene text detection and recognition: Recent advances and future trends. Frontiers of Computer Science **10**(1), 19–36 (2016)