PETR: Rethinking the Capability of Transformer-Based Language Model in Scene Text Recognition

Yuxin Wang, Hongtao Xie, Shancheng Fang, Mengting Xing, Jing Wang, Shenggao Zhu, and Yongdong Zhang, *Senior Member, IEEE*

***Abstract*— The exploration of linguistic information promotes the development of scene text recognition task. Benefiting from the significance in parallel reasoning and global relationship capture, transformer-based language model (TLM) has achieved dominant performance recently. As a decoupled structure from the recognition process, we argue that TLM's capability is limited by the input low-quality visual prediction. To be specific: 1) The visual prediction with low character-wise accuracy increases the correction burden of TLM. 2) The inconsistent word length between visual prediction and original image provides a wrong language modeling guidance in TLM. In this paper, we propose a Progressive scEne Text Recognizer (PETR) to improve the capability of transformer-based language model by handling above two problems. Firstly, a Destruction Learning Module (DLM) is proposed to consider the linguistic information in the visual context. DLM introduces the recognition of destructed images with disordered patches in the training stage. Through guiding the vision model to restore patch orders and make word-level prediction on the destructed images, visual prediction with high character-wise accuracy is obtained by exploring inner relationship between the local visual patches. Secondly, a new Language Rectification Module (LRM) is proposed to optimize the word length for language guidance rectification. Through progressively implementing LRM in different language modeling steps, a novel progressive rectification network is constructed to handle some extremely challenging cases (*e.g.* distortion, occlusion, etc.). By utilizing DLM and LRM, PETR enhances the capability of transformer-based language model from a more general aspect, that is, focusing on the reduction of correction burden and rectification of language modeling guidance. Compared with parallel transformer-based methods, PETR obtains 1.0% and 0.8% improvement on regular and irregular datasets respectively while introducing only 1.7M additional parameters. The extensive experiments on both English and Chinese benchmarks demonstrate that PETR achieves the state-of-the-art results.**

***Index Terms*— Scene text recognition, language model, deep neural networks.**

I. Introduction

SCENE text recognition (STR) [1], [2], [3], [4], [5] is a fundamental computer vision task, which aims to recognize text from the natural images. Because of the wide application of STR (*e.g.* automatic driving, visual auxiliaries etc.), it has attracted great interest from academia and industry [6], [7], [8], [9]. Due to the complexity of the scene text image, it is still a challenge to accurately recognize texts from noisy images (*e.g.* blur, occlusion etc.)

As text image contains two level contents: visual texture and linguistic information, linguistic information has been proved to improve the recognition performance on the low-quality images [5], [10], [11]. In the past years, STR methods attempted to learn robust linguistic rules from the input word image [12], [13], and recurrent neural network (RNN) [14], [15] is widely used to capture the linguistic information. This attention mechanism (RNN-based recognition) learns the alignment between the input image and output text sequences [16], which is easily disturbed by the attention drift problem. Thus, Cheng *et al.* [17] introduces a focusing attention network to automatically draw back the drifted attention. DAN [12] proposes a decoupled alignment operation to generate the attention map by using only visual information. However, due to the serial reasoning structure and local linguistic perception, the efficiency and the effectiveness of RNN-based language model needs to be explored. Benefiting from the global linguistic perception and parallel reasoning structure, Transformer [18] has received much attention from STR task [19], [20] recently. As a decoupled structure from recognition process, the transformer-based language model (TLM) takes the prediction of vision model as input, and rectifies the visual prediction based on the learned linguistic rules [21]. For example, SRN [19] constructs two unidirectional transformer encoders to capture the linguistic information from sequential and converse directions. To further

Manuscript received 1 August 2021; revised 10 April 2022 and 7 June 2022; accepted 18 July 2022. Date of publication 23 August 2022; date of current version 30 August 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB0804203; in part by the National Nature Science Foundation of China under Grant 62121002, Grant 62022076, and Grant U1936210; in part by the Youth Innovation Promotion Association Chinese Academy of Sciences under Grant Y2021122; and in part by the Fundamental Research Funds for the Central Universities under Grant WK3480000011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xu-Yao Zhang. *(Corresponding author: Hongtao Xie.)*

Yuxin Wang, Hongtao Xie, Shancheng Fang, and Yongdong Zhang are with the School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China (e-mail: wangyx58@mail.ustc.edu.cn; htxie@ustc.edu.cn; fangsc@ustc.edu.cn; zhyd73@ustc.edu.cn).

Mengting Xing is with Baidu Intelligent Cloud, Chengdu 610021, China (e-mail: xingmengting@baidu.com).

Jing Wang and Shenggao Zhu are with Huawei Cloud, Shenzhen 518129, China (e-mail: wangjing105@huawei.com; zhushenggao@huawei.com).

Digital Object Identifier 10.1109/TIP.2022.3197981

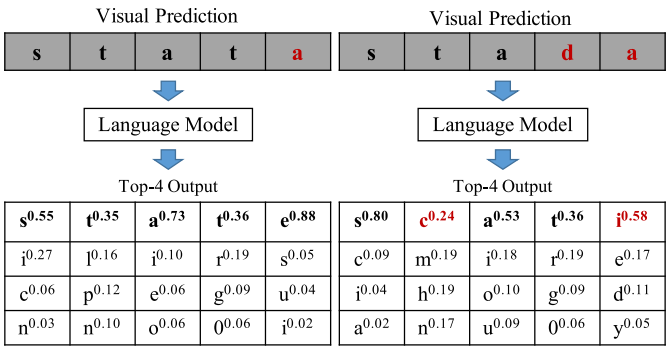

Fig. 1. The top-4 recognition results of language model. Left: the input word is "stata". Right: the input word is "stada". The ground truth is the word "state". The number in the upper right corner represents the probability of the output character. The rectified capability of transformer-based language model is limited when the vision model provides prediction with low character-wise accuracy. The red characters mean the wrong prediction.

improve the reasoning efficiency, ABINet [21] introduces a bi-directional language model to replace the two unidirectional linguistic learning. Though these methods achieve promising results, in this paper, we argue that low-quality visual predictions affect the capability of TLM. To be specific, we summarize the low-quality visual prediction into two categories: low character-wise accuracy and inaccurate word length. Specifically, the character-wise accuracy is defined as the rate of accurately predicted characters.

On the one hand, we propose the opinion that visual prediction with low character-wise accuracy limits the capability of transformer-based language model (TLM). As existing vision models focus on only the visual texture information [5], [8], [9], it is difficult for the vision model to generate predictions with high character-wise accuracy for challenging cases (*e.g.*blur, noise, etc.). However, the visual prediction with low character-wise accuracy will significantly increase the correction burden of TLM, further limiting the network capability. We visualize the top-4 probability of output from TLM in Fig. 1. Taking the word "stat-" as input, the language model corrects the word to "state" (left of Fig. 1). However, when the number of incorrectly predicted characters increases, it is hard for the language model to correct the input word (*e.g.* the input word is "sta–" in right of Fig. 1, the rectified word is "scati"). As VisionLAN [22] proves that the linguistic learning in the visual space is important for the performance boosting, how to concurrently consider linguistic information in the visual space to improve character-wise accuracy needs to be explored.

On the other hand, the inaccurate word length of visual prediction will cause the misalignment problem between vision model and TLM, which is firstly noticed by [19] without feasible solution. In this paper, we attribute the influence of such misalignment problem to the wrong language modeling guidance in TLM. We visualize the language modeling process of TLM in Fig. 2. To eliminate the interference from redundant time steps, padding mask determines the number of characters considered in the language modeling process. The sequence mask generated from the padding mask is used to determine the reasoning order. However, when facing

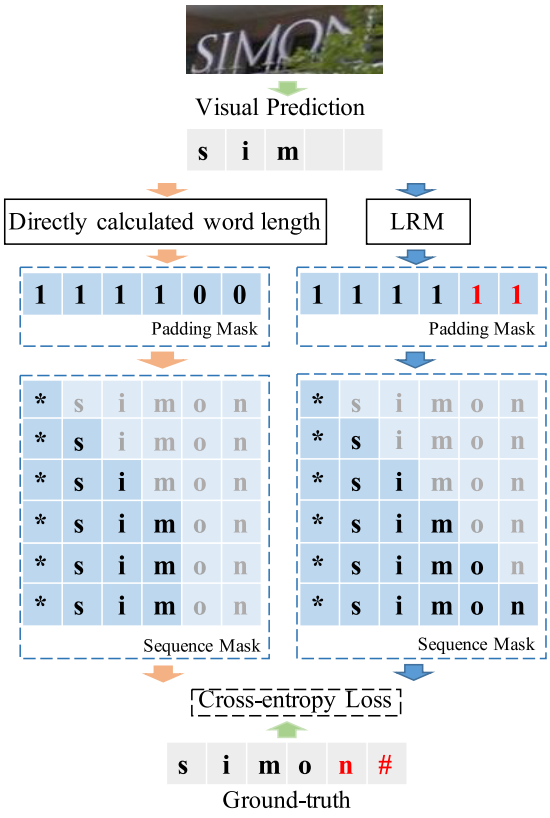

Fig. 2. The language modeling guidance in the transformer-based language model. The value of padding mask is $wordlength+1$ (end-of-sequence token). * and # are the start token and end-of-sequence token. Left: the padding mask and sequence mask are directly generated from the word length of visual prediction. Right: the padding mask and sequence mask are generated by our Language Rectification Module (LRM). The proposed LRM effectively provides the correct word length when input image is under difficult conditions (*e.g.* occlusion, noise, etc.). Finally, an accurate language modeling guidance (sequence mask) is provided in our method.

some difficult conditions (*e.g.* occlusion, noise, etc.), the word length is usually inconsistent between visual prediction and original image. Thus, inaccurate input masks are obtained by directly calculating from the visual prediction (left of Fig. 2). We argue that such inaccurate masks will limit the capability of TLM from following two aspects: 1) Limiting the effectiveness of language modeling. The inaccurate masks provide a wrong language modeling guidance, for example, resulting in the incomplete reasoning process (*e.g.* missing the reasoning process of character "n" and "#"in left of Fig. 2). Though supervised by word-level annotations, it is hard for TLM to learn robust linguistic rules with the inaccurate language modeling guidance. In addition, the stacked language modeling architecture [19], [20] further results in the error accumulation. 2) Being difficult to converge. The inaccurate language modeling guidance is harmful for the network convergency (discussed in Sec. IV-H.1). Thus, it is necessary to decouple the mask generation from the visual prediction and introduce a new generation approach.

Based on above analyses, in this paper, we rethink the capability of transformer-based language model, and introduce a **P**rogressive sc**E**ne **T**ext **R**ecognizer (PETR) to solve above two problems. The pipeline of our method is shown

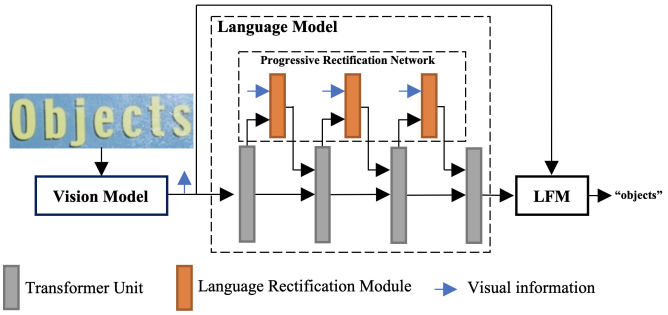

Fig. 3. The pipeline of proposed PETR. The input image is firstly processed by vision model. Then, the language model captures the relationship between characters. The language model contains two parts: progressive rectification network and transformer units. In the progressive rectification network, we iteratively use Language Rectification Module to optimize the word length to provide accurate language modeling guidance for transformer units. Finally, we use a Learnable Fusion Module (LFM) to sufficiently aggregate the visual and linguistic information for final prediction.

in Fig. 3. Specifically, we firstly introduce the thought of jigsaw puzzle [23] into the vision model. Besides classifying and reconstructing destructed images [23], we pioneer the introduction of text recognition on destructed images with disordered patches. Benefiting from the inner relationship capture between local visual patches, vision model can concurrently consider linguistic information in the visual context for high character-wise accuracy prediction. Secondly, instead of simply using the word length of visual prediction for mask generation, we propose a Language Rectification Module (LRM) to optimize the word length. To guarantee the accurate word length prediction, LRM considers not only the visual features from vision model but also the linguistic characteristics from last language modeling step. Through progressively implementing LRM in different language modeling steps, a novel progressive rectification network is constructed to handle extremely challenging cases (shown in Fig. 7). Finally, due to the feature heterogeneity, directly fusing visual and linguistic features for final prediction will limit the expression of linguistic information in multimodal characteristics. Thus, we introduce a Learnable Fusion Module (LFM) to adaptively aggregate the visual and linguistic features in a common semantic space, which is a bridge between the vision and language module, connecting DLM and LRM. To best of our knowledge, this is **the first work** to essentially study the capability of transformer-based language model, and prove that the great correction burden and wrong language guidance are the major limiting factors. Extensive experiments demonstrate that our method obtains state-of-the-art results on both English and Chinese benchmarks, especially for blurry and irregular text images.

The main contributions of this paper are fourfold:

- We analyze the relationship between the quality of visual predictions and the correction burden of TLM. Based on the analysis, we introduce a Destructed Learning Module (DLM) to help the vision model to concurrently consider the linguistic information in the visual context.
- A new Language Rectification Module (LRM) is proposed to eliminate the wrong language modeling guidance. By taking both visual and linguistic information into account, the progressive rectification network effectively provides an accurate language modeling guidance for robust linguistic rules learning.
- The proposed Learnable Fusion Module (LFM) sufficiently expresses linguistic information in the multimodal characteristics, which aggregates the visual and linguistic features in a common semantic space.
- This paper essentially explores the language modeling process in transformer-based language model (TLM), and enhances the capability of TLM from a general aspect. The exhaustive ablation studies are also beneficial to other related researches. Our PETR achieves a new state-of-the-art performance in both regular and irregular text recognition.

II. Related Work

A. Scene Text Recognition

Scene text recognition (STR) [24], [25], [26] has made significant strides in the past few years. As this paper focuses on the language modeling in STR task, we group STR works into two categories: language-free methods and language-based methods, according to whether the linguistic information is used.

1) Language-Free Methods: Language-free methods mainly use the visual texture for recognition without considering the linguistic information between characters [8], [27], [28]. The traditional STR methods firstly use connected components [29] or a sliding window [30], [31] to typically localize individual characters. Then, the character-wise classification combining various feature descriptors is utilized. Finally, a full word is generated by integrating the characters [32]. The CTC-based methods [33], [34] first use the convolutional neural networks (CNNs) to extract visual features, and then utilize the RNN for feature sequence modeling. Zhang *et al.* [8] calculate the similarity between pre-defined alphabet and visual features of input images, which use a visual matching approach for recognition. Benefiting from the improvement of object segmentation, some methods also regard the STR as a pixel-wise classification task [7], [35]. Particularly, Textscanner [7] further utilizes an order map to guarantee a more accurate transcription. Jaberberg *et al.* [27] use a classification network to directly classify 90k categories of word images. In general, language-free methods mainly focus on the texture features in the visual space, which usually fail on low-quality or occluded text images [13], [19].

2) Language-Based Methods:

a) Relationship between vision and language: Attention-based methods [9], [36], [37] are popular in scene text recognition task, which follow the encoder-decoder structure using RNN [11], CNN [5] or Transformer [20] to replace the complex N-grams for language modeling. The encoder extracts visual features and the decoder predicts characters by focusing on generated visual features following linguistic rules. For example, DAN [12] firstly uses convolutional alignment module (CAM) to generate the visual features of each character, then adopts the gated recurrent unit (GRU) to predict the word

sequence by focusing on the character visual cues at current time step and the embedding of predicted character from last step. Lee and Osindero [15] use recursive CNN as the feature extractor and implement LSTM to learn linguistic information in character-level. Following these methods, we adopt a similar encoder-decoder structure for scene text recognition, which uses a vision model as the encoder to extract visual information and adopts a language model as the decoder to capture the linguistic rules.

b) Language model with different structures: The linguistic information is widely used in Neuro-Linguistic Programming (NLP) methods [18], [38]. Benefiting from the NLP methods, linguistic information is proved to improve the performance of recent STR methods on low-quality images [13], [19]. Different from language-free methods, language-based methods adopt a language model to capture linguistic information between characters for performance boosting. As deep learning becomes the most promising machine learning tool, the structure of the language model is also changing and has different implementation forms. In this paper, we divide the language-based methods into two categories: Serially Reasoning Methods and Parallelly Reasoning Methods.

Serially Reasoning Methods (SRMs) model the linguistic information by sequentially using the character predicted from the last time step. Lee and Osindero [15] utilize RNNs for the sequential dynamics learning, which eliminate the dependence on manually defining N-grams. To handle the distorted text images, Aster [11] firstly introduces a rectification module for image preprocessing, then utilizes RNNs to learn the linguistic information. To handle the oriented text recognition, AON [39] firstly encodes the visual features in four directions. Then AON aggregates the extracted four sequences through a weighting mechanism. In order to solve the gradient vanishing problem caused by RNNs, Fang *et al.* [5] propose a completely CNN-based text recognizer. Specifically, they firstly implement a parallel branch to model linguistic rules, and then consider the visual and linguistic information as an ensemble to boost the recognition performance. To handle the attention drift problem, Cheng *et al.* [17] introduce a focusing attention network and use the location supervision to focus on the target regions. Wang *et al.* [40] handle the attention drift problem by further considering character center masks and encoded coordinates.

Parallelly Reasoning Methods (PRMs) simultaneously reason the linguistic information for each character, which propose an efficient language modeling process for scene text recognition. Benefiting from the wide usage of Transformer [18] in linguistic learning in NLP approaches, PRMs abandon auto-regression and use a parallel prediction to improve the efficiency. Lyu *et al.* [20] supervise the predictions in parallel execution, where the inputs at each time step are character-wise embedding from visual predictions. SRN [19] implements the transformer module [18] in the decode layer. Different from these methods, VisionLAN [22] endows the vision model with language capability, and uses a unified transformer structure for both visual and linguistic reasoning. In this paper, our method falls into parallel execution, but we try to enhance the capability of language model in PRMs. PETR essentially explores the language modeling process, and enhances the capability of transformer-based language model from a general aspect.

B. Progressive Structure in Recent Researches

As the performance of single-step regression is usually limited in some challenging conditions, the progressive structure is popular by its impressive property. The application of progressive structure ranges from detection [41], [42] to recognition [21], [37], which is demonstrated to be an effective approach for performance boosting [43], [44]. To handle the text instance with extreme ratios, LOMO [42] iteratively regresses the offsets to optimize the bounding box. To ensure a more robust image transformation, ESIR [37] uses several rectification module to progressively rectify the distorted text images. To be specific, the input of the rectification module is the output image from the last rectification step. ABINet [21] iteratively uses the language model to correct the visual recognition results. Different from these methods, the motivation of progressive structure in this paper is to enhance the capability of language model. To best of our knowledge, this is the first work to focus on the capability of transformer based language model in scene text recognition. The elaborately designed progressive rectification network is proved to be effective for performance boosting.

III. Methodology

As shown in Fig. 3, the proposed PETR mainly contains three parts: Vision Model, Language Model and Learnable Fusion Module. In this section, we first introduce the pipeline of our method, and then we illustrate each part in detail sequentially.

A. Pipeline

As shown in Fig. 3, given an input image, the vision model firstly gives a preliminary recognition result. Then, we send the visual recognition results to the language model for linguistic rules learning. The language model consists of two parts: progressive rectification network and transformer unit [18]. The progressive rectification network consists of three Language Rectification Modules (LRMs). LRM firstly predicts the word length of input text image by taking both visual and linguistic information into account. Then, LRM generates the related padding and sequence masks for following transformer unit (detailed in Sec. III-C). Transformer unit is used to achieve language modeling among input characters [19], [20]. Finally, the Learnable Fusion Module aggregates the visual and linguistic information for accurate prediction.

B. Vision Model

As shown in Fig. 4, the vision model contains three parts: backbone, Destruction Learning Module (DLM) and Visual Reasoning Module (VRM). Considering only visual textures

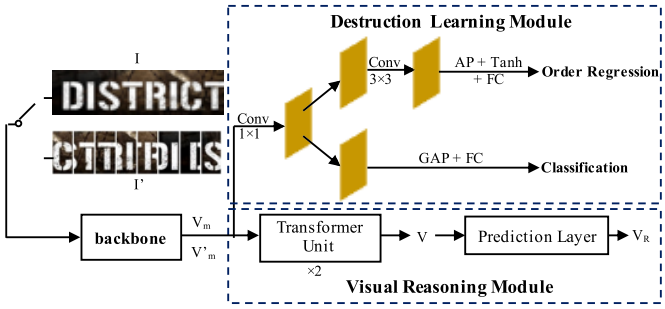

Fig. 4. The architecture of vision model. AP and GAP are short for average pooling and global average pooling. Tanh means tanh activation layer. FC is fully connected layer.

is not able to handle some difficult cases beyond the visual discrimination (*e.g.* blur, noise, etc.), how to capture the linguistic information in visual space for assistant prediction is still a challenge. Inspired by the significance of jigsaw puzzle [23] in capturing inner relationship between local visual patches, we introduce the recognition of destructed images in the training process. Specially, we firstly generate destructed image with disrupted patches (detailed in Sec. IV-B). Then, we combine both original and destructed images in a batch, and send them to vision model for destruction learning and text recognition (shown in Fig. 4). The introduction of destructed images in the training stage has following two advantages: 1) the recovery of the disrupted patches guides backbone to perceive the linguistic information during feature extraction. 2) Text recognition on destructed images guides VRM to concurrently consider linguistic information in the visual context.

1) Destruction Learning Module: The Destruction Learning Module (DLM) aims to classify the destructed images and restore the disrupted patches. As shown in Fig. 4, the visual feature map $V_m \in R^{H\times W}$ from the original image I or $V'_m \in R^{H\times W}$ from destructed image I' is first acquired through backbone network. H and W is the height and width of features. DLM contains two parallel branches, the first branch is guided to learn the destruction rules and the second branch is trained to classify destructed images.

Through implementing the average pooling with kernel size $H \times (W/M)$, tanh activation layer and fully connected layer sequentially, the first branch maps V_m and V'_m into a probability distribution vector $D \in R^{1\times M}$ or $D' \in R^{1\times M}$. D and D' correspond to the reconstruction sequence (*e.g.* "12345678" for I and "74865213" for I' in Fig. 4, $M = 8$). The second branch outputs vector $C \in R^{1\times 2}$ or $C' \in R^{1\times 2}$ through the global average pooling and a fully connected layer. Such vector is used to classify whether the input image is destructed. Benefiting from learning reconstruction rules and classifying discriminative characteristics, DLM helps backbone to perceive the linguistic information during feature extraction.

2) Visual Reasoning Module: The Visual Reasoning Module (VRM) aims to concurrently use the linguistic information in the visual context for high character-wise accuracy prediction. To achieve this purpose, we introduce the recognition of destructed images in the training process. Thus, VRM is trained to directly make word-level predictions on destructed images. In the testing stage, VRM is able to concurrently use the linguistic information in the visual context to assist the recognition process (shown in Fig. 10).

In order to capture the long-range dependencies in the 2d visual space, inspired by the significance of transformer [18] in computer vision tasks [19], [20], [45], we stack two transformer units [18] to model the long-range dependencies in VRM (shown in the bottom of Fig. 4). Then, the output feature V is sent to the prediction layer to generate visual predictions. The predicting process is formulated in Eq. (1), where $A \in [0, 1]$ is the attention map. $A_{t,ij}$ means the attention value in location (i, j) at time step t. $p_t \in R^K$ is the prediction at time step t, where K is the classification categories. V_{ij} is the visual feature in location (i, j). M in Eq. (2) is a linear transformation layer. *FullyConnected* is the fully connected layer. Finally, after implementing an argmax layer following $[p_1, p_2, \ldots, p_N]$, the visual prediction $V_R \in R^{N\times 1}$ is sent to the following language model for linguistic rules learning. N is the max length and is set to 25 in this paper.

$$p_t = \text{FullyConnected}(\sum_{\forall i,j} Att_{t,ij} V_{ij}) \tag{1}$$

$$Att_{t,ij} = \frac{\exp(M(V_{ij})_t)}{\sum_{\forall i,j} \exp(M(V_{ij})_t)} \tag{2}$$

Benefiting from the destruction learning in Destruction Learning Module (DLM) and the word-level recognition on destructed images, vision model is able to take both visual texture and linguistic information in the visual context into account. Thus, visual predictions with high character-wise accuracy are generated to effectively reduce the rectification burden of language model (proved in Sec. IV-D.1). In the testing stage, we remove DLM and only use VRM to recognize original text images. Thus, DLM introduces zero computation cost in the testing stage.

C. Language Model

Taking the visual predictions as input, the language model aims to rectify the visual predictions based on learned linguistic rules. In this section, we first introduce the structure of the language model, and then illustrate the details of Language Rectification Modules (LRM).

1) The Structure of the Language Model: Inspired from [20] and [19], we construct a similar architecture for language modeling (shown in Fig. 5), which contains four transformer units [18]. However, there are two main differences: 1) progressive language modeling process. Different from [20] and [19] simply stacking transformer units for language modeling, we decouple the language modeling process to several steps. Each step contains one transformer unit (shown in Fig. 5), and the detailed implementation simply follows [19], [20] without special settings. 2) Progressive language modeling guidance rectification. For the 2nd, 3rd and 4th language modeling steps, we construct an LRM to rectify the word length and give an accurate guidance for language modeling. Based on these designs, we introduce a progressive rectification structure. By taking the learned

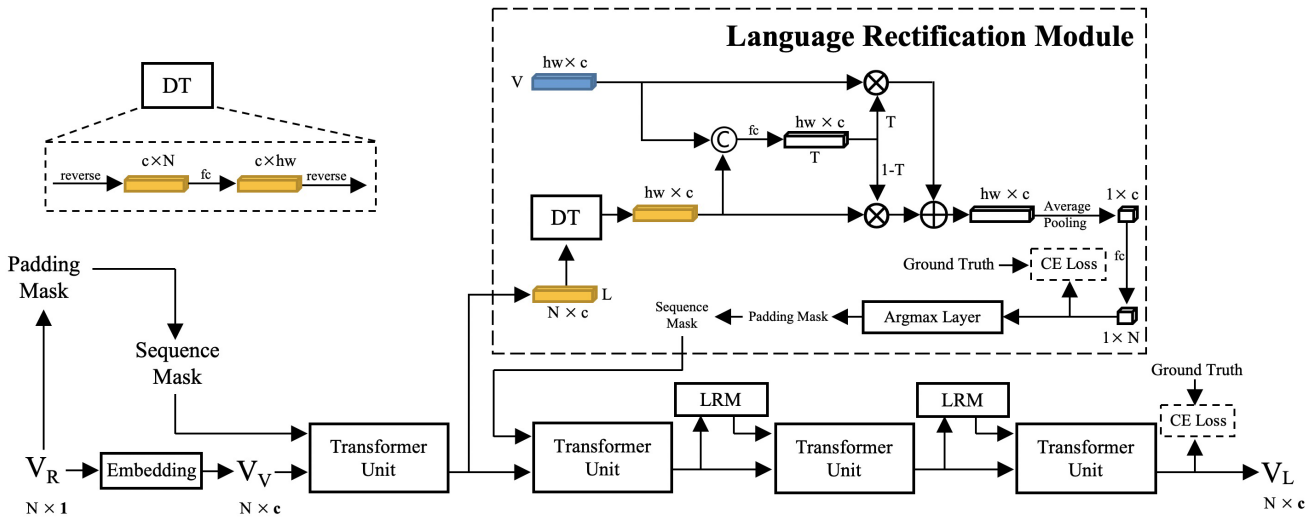

Fig. 5. The architecture of language model. V_R is short for visual results, which is predicted from the vision model. V_V is generated from V_R through an embedding layer. DT is short for dimension transformation module. The illustrated three Language Rectification Modules (LRMs) form the progressive rectification network. LRMs progressively eliminate the inaccurate language guidance caused by the visual predictions. The padding mask for V_V is generated by directly calculating the word lengh of V_R. The padding mask in LRM is generated from the predicted word length and the sequence mask is obtained from the corresponding padding mask. The ground truth used in LRM is directly generated from the word-level annotation by calculating the length of the word.

linguistic information from the last step and visual information from the vision model, the progressive rectification network effectively eliminates the wrong language guidance step-by-step.

For the input visual prediction V_R in Fig. 5, we firstly use an embedding layer to generate the character-wise embedding features V_V. Then, we send V_V and sequence mask generated from V_R to the first transformer unit for language modeling. The output linguistic feature V_L from the last transformer unit is sent to Learnable Fusion Module for final prediction, which will be introduced in Sec. III-D.

2) Language Rectification Module: As shown in Fig. 5, we construct the Language Rectification Module (LRM) in the 2nd, 3rd and 4th language modeling steps. LRM takes the visual features $V \in R^{hw \times c}$ (shown in Fig. 4) and linguistic features $L \in R^{N \times c}$ (gained from the last language modeling step) as inputs. h and w are the height and width, and c is the number of channel. N is the max length of the word, which is set to 25 in our experiment. The linguistic feature L is firstly transformed to the same dimension as V through a dimension transformation (DT) module, which consists of two transpose operations and one fully connected layer. Then, we concatenate both features to calculate the attention map T, which is used to balance the visual and linguistic information. Next, we aggregate the visual and linguistic information based on the attention map T through element-wise product. After that, a fully connected (fc) layer is used to predict the length of the word in the input image I. We regard the word length prediction as a pure classification task, and use a cross-entropy loss to supervise LRM. Finally, the word length is obtained after an argmax layer and is leveraged to generate padding and sequence masks.

With only three fully connected layers and one average pooling layer, the Language Rectification Module (LRM) effectively provides an accurate guidance for language modeling with little extra computation cost (shown in Sec. IV-G). Furthermore, the proposed LRM is also demonstrated to be beneficial for network convergency in training stage (detailed in Sec. IV-H.1).

D. Learnable Fusion Module

Through aggregating the visual and linguistic information, robust text recognition can be obtained based on visual and linguistic cues. Instead of simply integrating the multimodal features along the time dimension N [19], [21], we propose a Learnable Fusion Module (LFM) to adaptively aggregate the visual and linguistic features in a common semantic space.

As shown in Fig. 6, the visual feature $V_F \in R^{N \times c}$ (generated from Eq. (1) before fully connected layer) and linguistic feature $V_L \in R^{N \times c}$ (in Fig. 5) are firstly mapped to the vectors with dimension $R^{q \times c}$. Then, by concatenating the visual and linguistic features along "q" dimension, LFM effectively aggregates the multimodal information in a common semantic space and eliminates the mis-alignment issue [19]. Next, the fused features are mapped to the vector V_f with dimension $R^{N \times c}$. In the implementation, V_f is enhanced by introducing the features from VSFD [19] through element-wise addition. The final recognition result is obtained through a fully connected layer. The dimension q is discussed in our ablation studies.

E. Loss Function

We add supervision in the vision model, language model and Learnable Fusion Module (LFM). The loss function is formulated as follows:

$$L = L_V + L_L + \lambda_F L_F, \tag{3}$$

where L_V, L_L and L_F mean the loss for the vision model, language model and LFM respectively. Following [19], we set λ_F to 2.

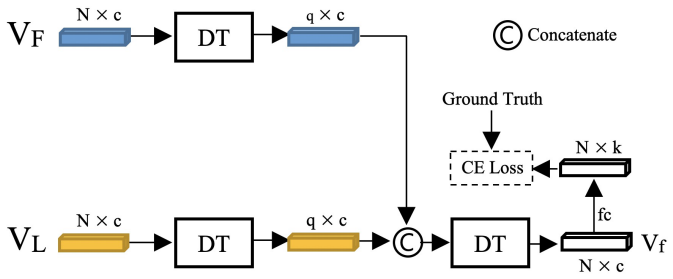

Fig. 6. The architecture of Learnable Fusion Module (LFM). DT is the same as the operation illustrated in Fig. 5, but we change the output channel of fully connected layer to q. fc is short for fully connected layer.

1) Loss for Vision Model: The loss in vision model includes three parts: the reconstruction loss L_r, the destructed classification loss L_d and the recognition loss L_{recV}. L_V is formulated as following:

$$L_V = \lambda_1 L_r + \lambda_2 L_d + \lambda_3 L_{recV}, \tag{4}$$

where we set $\lambda_1, \lambda_2, \lambda_3$ to 0.1, 1 and 1 respectively. We choose the L1 loss for reconstruction loss L_r. L_r is used to supervise the first branch in Destruction Learning Module (DLM) and is formulated as following:

$$L_r = \frac{1}{M}\sum_{m=1}^{M} |p_m - l_m|, \tag{5}$$

where p_m is the predicted order and l_m is the ground truth (detailed in Sec. IV-B). m is the index of image patch and M is the total number of patches.

The destructed classification loss L_d is formulated in Eq. (6), and the binary cross-entropy loss is utilized to supervise the classification branch.

$$L_d = g_d \log(p_d) + (1 - g_d)\log(1 - p_d), \tag{6}$$

where p_d is the predicted category and g_d is the ground truth (1 for destructed image and 0 for original image).

For the recognition loss L_{recV} in vision model, we use the cross-entropy loss in Eq. (7), which is similar to those in recent STR methods [7], [12].

$$L_{recV} = \frac{1}{N}\sum_{t=0}^{N} \log(p_t|g_t), \tag{7}$$

where p_t is the predicted character at time step t, and g_t is the ground truth. N is the max length of the word, we set it to 25 in our experiment.

2) Loss for Language Model: The loss in language model includes two parts: the rectification loss L_p for Language Rectification Module (LRM) and the recognition loss L_{recL}. L_L is formulated as following:

$$L_L = \gamma L_p + \lambda_L L_{recL}, \tag{8}$$

where γ is used to balance the rectification loss L_p and the recognition loss L_{recL}, which is discussed in ablation study in our experiment section. Following [19], we set λ_L to 0.15.

The rectification loss L_p is formulated as following:

$$L_p = L_{p1} + L_{p2} + L_{p3}, \tag{9}$$

$$L_{pi} = \log(p_r|g_r), \tag{10}$$

where L_{pi} $(i = 1, 2, 3)$ are the corresponding loss function for LRM in the 2nd, 3rd and 4th language modeling steps. p_r and g_r are the predicted word length and ground truth respectively. For the recognition loss L_{recL}, we choose the same formulation as Eq. (7).

3) Loss for Learnable Fusion Module (LFM): We choose the same recognition loss function as Eq. (7) to formulate L_F.

IV. Experiments

A. Datasets

SynthText (ST) [46] contains 80k synthetic images. We use the word-level bounding-box annotations to crop the word instances to train our model.

SynthText90K (90K) [27] is another synthetic dataset, which consists of 9 million images for training. We combine this dataset and SynthText to train our model.

ICDAR2013 (IC13) [47] contains 1095 testing images. We discard images containing less than three characters or including non-alphanumeric characters by using [31].

ICDAR2015 (IC15) [48] recognition task provides 500 scene images. 1811 cropped word images are finally kept by filtering some extremely distorted images.

IIIT 5K-Words (IIIT5K) [49] is a dataset collected from the website containing 3000 word images for testing.

Street View Text (SVT) [31] contains 647 images cropped from 250 scene images, which is collected from Google Street View.

Street View Text-Perspective (SVTP) [50] is also cropped from Google Street View images containing 639 test images. Many images in this set are heavily distorted.

CUTE80 (CT) [51] is a dataset proposed for curved text recognition. It contains 288 cropped testing images.

Following recent STR methods [13], [19], we train our method on the two synthetic datasets (ST and 90K), and test our model on 6 benchmarks including IC13, IC15, IIIT5K, SVT, SVTP and CT. According to the shape characteristics of word sequences, these datasets can be divided into regular datasets (IIIT5K, IC13 and SVT) and irregular datasts (IC15, SVTP and CT). Due to distorted shape characteristics existing in irregular datasets, the text recognition on irregular datasets is more difficult than the text recognition on regular datasets.

B. Label Generation

We do not introduce additional annotations in our method, which makes the proposed PETR be easily finetuned on other datasets. The labels for Destruction Learning Module (DLM) and Language Rectification Module (LRM) are detailed in the following.

1) Labels for Destruction Learning Module: As detailed in Sec. III-B.1, we destruct the text image along the horizontal direction. Inspired by [23], we firstly cut the text image into M patches evenly. Then, a random vector l of size M is generated,

where the m^{th} element $l_m = \frac{m}{M} - \frac{1}{2}, \{m = 1, 2, \ldots, M\}$. We randomly scramble the sequence $l = [l_1, l_2, \ldots, l_M]$ and the image patches. Finally, we generate the corresponding reestablishment sequence l' and destructed text image I'. By doing these, we regard the reconstruction learning as a pure distance regression task. For the classification branch, we set the label to 0 for original image and 1 for the destructed image.

2) Labels for Language Rectification Module: Benefiting from the word-level annotations in recent benchmarks, we directly calculate the word length as the label for Language Rectification Module (LRM). Thus, the task in LRM is a pure classification task, which predicts the number of the input word images.

C. Implementation Details

Following recent STR methods [11], [19], we choose ResNet45 as our backbone and set the stride to 2 in the stage 2, 3, 4. As input image with size 256×64 or 128×32 obtains almost the same results, following the most recent approaches [13], [19], we resize the input images to 256×64. Data augmentation contains color jittering, random rotation and perspective distortion. We implement our model on 4 NVIDIA V100 GPUs with batch size 384. The total network is trained end-to-end using Adam optimizer. We set the learning rate to 1e-4. The recognition includes 0-9, 26 characters and an end-of-sequence (EOS) token.

In the training stage, we concatenate the destructed and original text images in a batch and send them to the vision model. Following [19], two unidirectional transformer units are used in the language model.

D. Ablation Study

We conduct several experiments to demonstrate the effectiveness of our proposed modules.

1) The Effectiveness of Providing Visual Predictions With High Character-Wise Accuracy: To evaluate the effectiveness of Destruction Learning Module (DLM) to generate visual predictions with high character-wise accuracy, we conduct several experiments in Table I. Firstly, we demonstrate the effectiveness of DLM in recognition enhancement when only visual information is considered. As shown in Table I, the baseline model implemented with only vision model obtains a low accuracy on IC15 and CT datasets. This is because the text images in these datasets are usually with low-quality (*e.g.* blur, noise, etc.). Benefiting from the destructed learning and word recognition on destructed images, the vision model is able to directly capture linguistic information in the visual space. To be specific, baseline implemented with DLM gains 2% and 0.9% improvement on IC15 and CT datasets respectively. Then, we implement the language model in the baseline to prove that proposed DLM can reduce the rectification burden of language model. As shown in Table I, our DLM effectively improves the recognition results on SVTP and CT datasets by 1.3% and 2.1% respectively. For regular datasets, the improvement is also considerable, which outperforms the baseline by 0.2%, 1.0% and 0.1% on IIIT5K, IC13 and SVT datasets respectively. Though we are **the first one** to introduce the recognition of destructed text images in STR task, the significant improvement demonstrates its effectiveness of concurrently considering linguistic information in the visual context. Thus, DLM effectively enhances the visual predictions and reduces the rectification burden of language model. The qualitative analysis about DLM will be detailed in Sec. IV-H.2.

TABLE I

THE ABLATION STUDY ABOUT DESTRUCTION LEARNING MODULE (DLM). * MEANS ONLY VISION MODEL IS USED. † MEANS LANGUAGE MODEL IS CONSTRUCTED

Methods	IIIT5K	IC13	SVT	IC15	SVTP	CT	total
Baseline *	94.7	94.0	88.7	79.5	81.1	87.1	88.8
DLM *	**94.9**	**94.5**	**88.9**	**81.5**	**81.6**	**88.0**	**89.5**
Baseline †	95.3	95.2	90.9	82.1	83.7	87.8	90.3
DLM †	**95.5**	**96.2**	**91.0**	**82.2**	**85.0**	**89.9**	**90.7**

TABLE II

THE ABLATION STUDY ABOUT THE NUMBER OF PATCHES (M) USED IN DESTRUCTION LEARNING MODULE (DLM)

M	IIIT5K	IC13	SVT	IC15	SVTP	CT	total
4	**94.9**	94.2	88.9	80.8	81.6	87.2	89.3
8	**94.9**	94.5	88.9	**81.5**	81.6	**88.0**	**89.5**
16	94.6	**94.6**	**89.1**	80.7	**81.7**	87.5	89.2

TABLE III

THE ABLATION STUDY ABOUT THE NUMBER OF LANGUAGE RECTIFICATION MODULES (LRMS) USED IN THE LANGUAGE MODEL

Number	IIIT5K	IC13	SVT	IC15	SVTP	CT	total
-	95.3	95.2	90.9	82.1	83.7	87.8	90.3
1	95.4	95.2	90.9	82.2	84.4	87.8	90.4
2	95.5	95.9	91.3	82.3	84.6	**88.2**	90.6
3	**95.7**	**96.7**	**91.6**	**82.8**	**84.7**	**88.2**	**91.0**

To study the relationship between the recognition performance and the number of patches used in DLM, we set the number of patches used in DLM as 4, 8 and 16. As shown in Tab. II, though using more patches in DLM can capture stronger linguistic information in the visual space, a large number of patches also increase the recognition burden on the destructed images, where exists a 0.3% decrease in the average accuracy from 8 patches to 16 patches. Thus, we set the number of patches to 8 in DLM.

2) The Effectiveness of Language Modeling Guidance Rectification: We study the relationship between the number of Language Rectification Module (LRM) implemented in the language model and the recognition performance. The language model implemented with 1 LRM means that we only construct LRM in the 2nd language modeling step. Thus, the language model implemented with 3 LRMs means that we construct LRM in 2nd, 3rd and 4th language modeling steps (shown in Fig. 5). As shown in Table III, the language model implemented with 1 LRM has limited capability for the rectification of language modeling guidance, which further demonstrates that there exists a heavily wrong guidance in the language modeling process. When we construct LRM in the language model step-by-step, the significant improvement proves the effectiveness of our progressive rectification structure. When we construct 3 LRMs in the language model,

TABLE IV
THE ABLATION STUDY ABOUT γ

γ	IIIT5K	IC13	SVT	IC15	SVTP	CT	total
0.05	95.4	95.4	**92.1**	82.7	85.3	89.2	90.8
0.1	**95.7**	**96.7**	91.6	**82.8**	84.7	88.2	**91.0**
0.5	95.5	96.0	91.7	82.3	**85.7**	**89.9**	90.8

TABLE V
THE ABLATION STUDY ABOUT THE DIMENSION IN LEARNABLE FUSION MODULE (LFM)

q	IIIT5K	IC13	SVT	IC15	SVTP	CT	total
Baseline	95.3	95.2	90.9	82.1	83.7	87.8	90.3
128	**95.5**	95.7	91.5	**82.5**	**85.4**	**89.6**	**90.8**
256	95.4	**95.8**	**92.0**	82.2	**85.4**	88.9	90.7

the relative improvement is 0.4%, 1.5%, 0.7%, 0.7%, 1.0% and 0.4% on IIIT5K, IC13, SVT, IC15, SVTP and CT datasets respectively. Though further increasing the number of transformer units and LRMs will obtain better results (91.2% in average accuracy when using 5 transformer units and 4 LRMs), we think it is very important to conduct a fair comparison with other methods (both using 4 transformer units in language model in ABINet [21], SRN [19]). Thus, all the experiments are conducted on the language model with 4 transformer units in our method. Based on above observations, LRM effectively helps the network to learn robust linguistic rules and significantly enhances the recognition results in both regular and irregular datasets. The qualitative analysis about LRM will be introduced in Sec. IV-H.1.

Furthermore, we study how the ratio γ in Eq. (8) influences the rectification performance in Table IV. Higher value of γ means that PETR focuses more on the word length optimization while paying less attention to the linguistic rules learning. In contrast, lower value of γ means that PETR focuses more on the linguistic rules learning while paying less attention to the word length optimization. As shown in Table IV, $\gamma = 0.1$ obtains the best results, which keeps a better balance between the ability of language modeling guidance rectification and linguistic rules learning.

3) The Effectiveness of Sufficient Linguistic Information Expression: We study the relationship between dimension q and the recognition performance to demonstrate the effectiveness of Learnable Fusion Module (LFM). As shown in Table V, the proposed LFM effectively handles the feature heterogeneity problem and sufficiently expresses linguistic information in multimodal characteristics. Through mapping the two independent information into a common semantic space, LFM improves the recognition results on irregular datasets significantly (1.7% and 1.8% on SVTP and CT respectively). For regular datasets, the improvement is also considerable (0.2% on IIIT5K, 0.5% on IC13 and 0.6% on SVT datasets respectively). When we increase the dimension to 256, there exists a performance drop (0.1% in the average accuracy). We infer that the larger dimension may cause the under fitting problem. Though the large amounts of synthetic training images reduce the influence of this problem to a certain extent, there still exists the different distribution between real-word datasets and synthetic datasets [7], [55].

TABLE VI
ABLATION STUDY ABOUT STEP-BY-STEP IMPLEMENTING OUR PROPOSED MODULES

DLM	LRM	LFM	total
-	-	-	90.3
-	✓	-	91.0
-	✓	✓	91.2
✓	✓	✓	**91.4**

4) Step-by-Step Evaluation: We add the proposed modules step-by-step to demonstrate their effectiveness. As shown in Table VI, the proposed Destruction Learning Module (DLM), Language Rectification Module (LRM) and Learnable Fusion Module (LFM) progressively enhance the recognition performance. To better reflect the overall accuracy improvement, the average accuracy on 6 benchmarks is shown. Benefiting from the progressive rectification structure and sufficient integration of multimodal characteristics, the language model effectively learns robust linguistic rules and sufficiently expresses the linguistic information. As shown in Table VI, baseline implemented with LRM and LFM significantly obtains 0.9% improvement in average accuracy. Finally, thanks to the reduction of correction burden by DLM, PETR achieves 91.4% in the average accuracy.

E. Comparison With Related Methods

The comparisons of PETR with previous outstanding approaches are shown in Table VII. In order to conduct a fair comparison with the most recent methods (VisionLAN [22] and ABINet [21]), we reproduce these two methods and using the same vision model as ours (2 transformer units). As the lexicon is always not available before practical use, we only compare the recognition results without any lexicon. Compared with visual enhancing methods, Aster [11] implements a rectification module to eliminate the distortion in the visual space. Different from Aster [11], our method enhances the visual recognition by guiding the visual model to learn linguistic information in the visual context. Thus, PETR eliminates the control point detection for distorted rectification and is more robust for irregular texts. Compared with [11], the relative improvement of our "Our Vision" model on the irregular text image recognition is 5.4%, 3.1% and 8.5% on IC15, SVTP and CT respectively. Though ESIR [37] further iteratively uses the rectification module to reduce the influence of distorted images, our "Our Vision" model also obtains a much better result on irregular text recognition (4.6%, 2.0% and 4.7% improvement on IC15, SVTP and CT respectively). To enhance the contextual linguistic information, SEED [13] uses an additional FastText to supervise the generated word embedding. Benefiting from the correct language modeling guidance provided by Language Rectification Module (LRM), our method is able to learn more robust linguistic information. Compared with SEED, PETR obtains 2.5% and 3.9% improvement in average accuracy on regular (Avg-R) and irregular (Avg-IR) datasets respectively. The language model in parallel transformer-based methods [19], [20], [21] suffers from the inaccurate language modeling guidance and great correction burden. Benefiting from the progressive language modeling

TABLE VII

RESULTS ON IIIT5K, IC13, SVT, IC15, SVTP AND CUTE DATASETS. FOLLOWING [13], [19], ALL THE RESULTS ARE UNDER NONE LEXICON. "CHAR" AND "WORD" MEAN CHARACTER-LEVEL AND WORD-LEVEL ANNOTATIONS USED IN THE TRAINING STAGE. AVG-R AND AVG-IR ARE SHORT FOR AVERAGE ACCURACY ON REGULAR DATASETS (IIIT5K, IC13 AND SVT) AND IRREGULAR DATASETS (IC15, SVTP, CT). "OUR VISION" MEANS THE VISION MODEL IMPLEMENTED WITH DESTRUCTION LEARNING MODULE (DLM). LAN IS SHORT FOR LANGUAGE

	Methods	Training Data	Annos	IIIT5K	IC13	SVT	Avg-R	IC15	SVTP	CT	Avg-IR
Vision Only	FCN [28]	ST	word,char	91.9	91.5	86.4	91.0	-	-	-	-
	CTC [35]	90K	word	81.2	89.6	82.7	83.0	-	-	-	-
	ACE [34]	90K	word	82.3	89.7	82.6	83.8	68.9	70.1	82.6	70.6
Vision + Language	FAN [17]	90K+ST	word	87.4	93.3	85.9	88.3	70.6	-	-	-
	AON [39]	90K+ST	word	87.0	-	82.8	-	68.2	73.0	76.8	70.2
	Aster [11]	90K+ST	word	93.4	91.8	89.5	92.5	76.1	78.5	79.5	77.0
	Fang *et al.* [5]	90K+ST	word	86.7	93.3	86.7	88.0	71.2	-	-	-
	ESIR [37]	90K+ST	word	93.3	91.3	90.2	92.5	76.9	79.6	83.3	78.2
	ScRN [52]	90K+ST	word,char	94.4	93.9	88.9	93.5	78.7	80.8	87.5	80.1
	SAR [36]	90K+ST	word	91.5	91.0	84.5	90.4	69.2	76.4	83.3	72.4
	Lyu *et al.* [20]	90K+ST	word	94.0	92.7	90.1	93.2	76.3	82.3	86.8	78.8
	Liao *et al.* [53]	90K+ST	word	93.9	95.3	90.6	93.7	77.3	82.2	87.8	79.5
	TextScanner [7]	90K+ST	word,char	83.9	92.9	90.1	86.5	79.4	84.3	83.3	81.0
	DAN [12]	90K+ST	word	94.3	93.9	89.2	93.5	74.5	80.0	84.4	76.8
	SRN [19]	90K+ST	word	94.8	95.5	91.5	94.5	82.7	85.1	87.8	83.8
	Wang *et al.* [54]	90K+ST	word	94.4	93.7	89.8	93.6	75.1	80.2	86.8	77.5
	SEED [13]	90K+ST	word	93.8	92.8	89.6	93.0	80.0	81.4	83.6	80.7
	Yue *et al.* [9]	90K+ST	word	95.3	94.8	88.1	94.2	77.1	79.5	**90.3**	79.0
	VisionLAN [22]	90K+ST	word	95.4	95.0	91.4	94.7	81.8	83.7	88.2	82.9
	ABINet [21]	90K+ST	word	95.3	96.7	91.7	95.0	83.1	**86.2**	88.9	84.4
Ours	Baseline (Vision)	90K+ST	word	94.7	94.0	88.7	93.7	79.5	81.1	87.1	80.7
	Our Vision	90K+ST	word	94.9	94.5	88.9	94.0	81.5	81.6	88.0	82.2
	Baseline (Vision + Lan)	90K+ST	word	95.3	95.2	90.9	94.6	82.1	83.7	87.8	83.1
	Our PETR	90K+ST	word	**95.8**	**97.0**	**92.4**	**95.5**	**83.3**	**86.2**	89.9	**84.6**

guidance rectification and reduction of correction burden, PETR effectively enhances the capability of transformer-based language model and improves the recognition accuracy. Compared with [19], [20], and [21], PETR significantly outperforms these methods and achieves a new state-of-the-art result on both regular and irregular datasets (95.5% vs 93.2%, 94.5% and 95.0% in Avg-R and 84.6% vs 78.8%, 83.8% and 84.4% in Avg-IR). Furthermore, as PETR uses the same language model as SRN [19], PETR only introduces 1.7M extra parameters (Table IX) while obtaining 1% and 0.8% improvement in Avg-R and Avg-IR respectively. The accuracy of VisionLAN [22] relies on the quality of occlusion maps, which are generated from the weakly-supervised learning. Thus, its linguistic learning process suffers from the inaccurate language modeling guidance problem (*e.g.* inaccurate occlusion maps), which will limit its capability in accurate recognition. Compared with VisionLAN [22], the destructed images of PETR are randomly generated rather than predicted, and LRM further introduces the accurate language modeling guidance for accurate recognition. Thus, PETR is able to achieve better performance in both Avg-R and Avg-IR (95.5% vs 94.7% in Avg-R and 84.6% vs 82.9% in Avg-IR). Though ScRN [52] and TextScanner [7] introduce additional annotations (character-level labels) in the training stage, the proposed PETR can obtain much better results with only original word-level annotations.

F. The Generalization on Long Chinese Dataset

To evaluate the performance on long non-Latin word images, we conduct experiments on the TRW15 dataset [56]. This dataset contains 484 test images. Following SRN [19], we crop 2997 word images for testing, and we set the max length to 50.

The results are shown in Table VIII. As 2D-Attention only considers the visual textures in the vision model, the captured linguistic information in the visual context effectively improves the recognition accuracy of PETR. Compared with 2D-Attention, the proposed PETR obtains 15.9% improvement in recognition accuracy. Benefiting from the progressive language modeling guidance rectification, our PETR is more robust to the long word recognition. Thanks to the accurate language modeling guidance for long text, the proposed method effectively outperforms parallel transformer-based method [19], [21] by 2.6% and 1.0% in accuracy respectively. Finally, the proposed PETR obtains a new state-of-the-art result on TRW15 dataset (88.1% in accuracy). The significant improvement on the Long non-Latin word images further demonstrates the generalization ability of our method.

Furthermore, we conduct an additional experiment on ReCTS [57] in Tab. IX. Specially, we reproduce the popular methods [11], [21], [35] to show our significance. Compared with RNN-based methods [11], [35], PETR outperforms these methods by at least 12.3% in accuracy. Though Shi *et al.* [35] achieves more efficient recognition than our method, the simple network structure seriously limits its recognition performance. In the end, PETR achieves 92.8% in accuracy, outperforming existing the most popular method [21] by 0.4%.

G. Speed and Parameter Size

We compare the speed and parameter size between PETR and recent outstanding methods. Specially, RNN-based methods [11], [35] and transformer-based methods [19], [21] are

TABLE VIII

THE RESULTS ON TRW15. THE RECOGNITION RESULTS OF CTC AND 2D-ATTENTION ARE REFERRED TO SRN [19]

Method	Accuracy
CASIA-NLPR [56]	72.1
SCCM [58]	81.2
2D-Attention [19]	72.2
CTC [19]	73.8
SRN [19]	85.5
ABINet [21]	87.1
PETR	**88.1**

TABLE IX

THE RESULTS ON RECTS AND IC13. TRAINING IS THE TRAINING TIME ON RECTS. "OUR VISION" MEANS THE VISION MODEL IMPLEMENTED WITH DESTRUCTION LEARNING MODULE (DLM)

	ReCTS		IC13		
Method	Accuracy Params(10^6)	Speed(ms)	Accuracy Params(10^6)	Speed(ms)	Training(h)
Shi *et al.* [35]	76.8 **11.2**	**4.0**	86.7 **8.3**	**5.8**	**49**
Aster [11]	80.5 26.6	170.0	94.1 22.3	74.1	65
SRN [19]	92.1 56.8	48.6	95.2 45.4	31.2	240
ABINet [21]	92.4 44.2	34.8	96.7 32.8	30.7	204
Our Vision	83.4 22.4	26.8	94.5 19.6	20.1	124
Our PETR	**92.8** 58.5	52.3	**97.0** 47.1	35.0	250

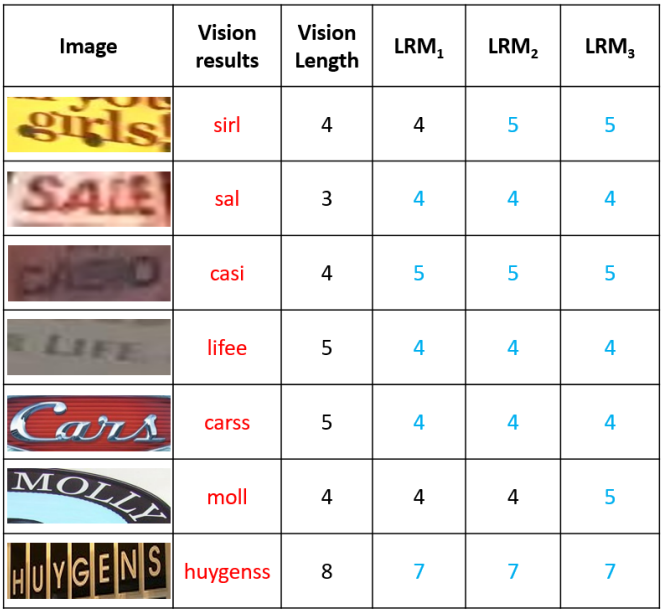

Fig. 7. The qualitative analysis of Language Rectification Module (LRM). Vision results are predicted by the vision model. Vision length is directly calculated from the vision results by computing the word length, which is used to generate masks for the first language modeling step. LRM_1, LRM_2 and LRM_3 are the predicted word length from the first, second and third LRM. The numbers in blue match to the ground-truth.

included. For fair comparison, we refer to their official codes and carefully re-implement their methods in our environment. As shown in Table IX, compared with Serially Reasoning Methods (SRMs) [11], Parallelly Reasoning Methods (PRMs) [19], [21] show significance in high inference speed. Though the concise structure helps [35] to obtain the best performance in efficiency, its simple pipeline shows limited recognition capability in the real implementation. Compared with existing PRMs, though additional computation cost is caused by LRM and LFM in testing phase, PETR obtains the significant improvement in accuracy (92.8% vs 92.1% and 92.4% on ReCTS and 97.0% vs 95.2% and 96.7% on IC13) with impressive efficiency. As DLM is only implemented in the training stage, the performance of "Our Vision" model is impressive, which obtains 94.5% in accuracy with speed of 20.1ms per image on IC13 dataset.

H. Qualitative Analysis of PETR in Rectification

In this section, we provide some qualitative results to demonstrate the effectiveness of PETR.

1) The Effectiveness of Language Rectification Module (LRM) in Language Modeling Guidance Rectification: We visualize the rectification process by LRM in Fig. 7. LRM_i means the predicted word length from i^{th} LRM. For occluded images (the first row), the proposed LRM effectively rectifies the word length and provides an accurate guidance for reasoning the occluded character. For the text images with confusing visual cues (the second to fifth rows), LRM successfully infers the word length from the confused image (*e.g.* blur, noise etc.) by taking the visual and linguistic information into account together. For the distorted word images (the sixth and seventh rows), LRM can also handle these cases and provide accurate language modeling guidance. Furthermore, the visualization of word images "girls" and "molly" illustrates that LRM sometimes can not provide an accurate guidance in the initial steps. Benefiting from the progressive rectification structure, PETR is able to optimize the word length by considering the learned linguistic rules from the last language modeling step, providing an accurate guidance for these difficult cases. As LRM adaptively considers the visual and linguistic information for prediction, the number of excessive rectification cases is very little in our experiment, which is less than 3% of the rectified samples. Thus, LRM is proved to effectively introduce accurate language modeling guidance for robust linguistic learning.

To better understand how LRM works, we using CAM to visualize the activated areas. As shown in Fig. 8, compared with feature maps introducing linguistic information, the activated area are dispersive. However, the activated areas provide a more focused response after taking the linguistic information into account, which confirms that the linguistic information can help LRM to correct the language modeling guidance.

It is worth mentioning that the proposed LRM also helps the network to achieve better convergency in the training stage. As shown in Fig. 9, we visualize the best average accuracy on the testing datasets (calculated based on 6 benchmarks) during the training stage. The accurate guidance for language modeling effectively helps the network to achieve better convergency in the training stage, which realizes more accurate recognition results on testing datasets at each step. Benefiting from the accurate language modeling guidance in both training and testing stages, PETR is able to learn more robust linguistic rules and provide more accurate recognition results.

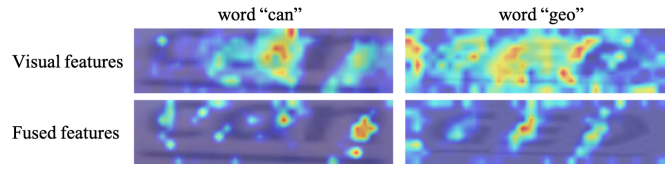

Fig. 8. The visualization based on CAM. Fused features means we introduce the linguistic information.

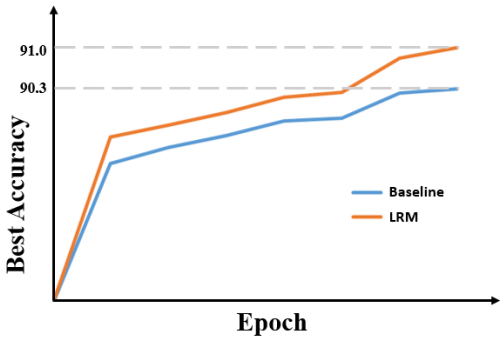

Fig. 9. The best average accuracy on testing datasets during the training stage.

2) The Effectiveness of Destruction Learning Module (DLM) in Visual Prediction Enhancement: We visualize some recognition results to prove the effectiveness of DLM. As shown in Fig. 10, the proposed DLM significantly handles the confusing characters (the first and second rows). For example, the character "e" has the similar visual cues to the character "c" in the word "valerie". The vision model without DLM wrongly gives the prediction "c", while the model guided by DLM correctly infers the character "e" through capturing the linguistic information in the visual space. For occluded images and complicated background (the third row), vision model guided by DLM can also eliminate the background interference. Furthermore, the correct recognition of blur and poor visibility images (the fourth row) also proves the effectiveness of our DLM. For example, due to the poor visibility, the character "a" has similar visual cues to the character "r" in the word "man". The model guided by DLM accurately predicts the character "a" and gives a correct word recognition.

There might be a concern that how can the local convolution operator capture the long-range dependencies to rank the image patches. As it is difficult to visualize the receptive field, we provide a rigorous theoretical derivation to support our method. The last down-sampled features with size 8×32 are sent to 15 residual blocks in the backbone, where each block contains a 3×3 convolutional layer. Thus, the receptive field is 31×31 in these 15 blocks. Furthermore, the average pooling layer (AP in Fig. 4) can further help DLM to perceive long-range information. Thus, DLM is able to rank the image patches.

To better understand how the recognition of destructed images helps the vision model to learn the linguistic information in the visual space, we visualize the recognition process of destructed images in Fig. 11. Benefiting from introducing the recognition of the destructed images in the training stage, the vision model is able to capture the inner relationship between patches, which correctly focuses on the character-wise visual cues in each time step. Thus, in the testing stage, the vision model is able to capture the linguistic information in the visual space to assist the recognition on confusing visual cues (shown in Fig. 10).

Fig. 10. The qualitative analysis of Destruction Learning Module (DLM). Characters in red are wrongly predicted. Top string: the predictions from the vision model without implementing DLM. Bottom string: the recognition from model constructed with DLM. The DLM effectively helps the vision model to concurrently capture the linguistic information in the visual context, which gives more accurate predictions without additional computation cost.

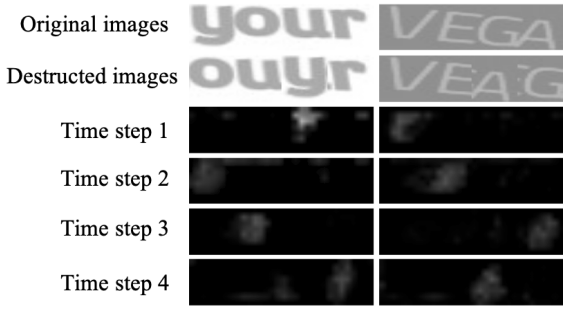

Fig. 11. The recognition process on destructed images. 4 patches are used in this visualization.

Fig. 12. The examples of failure cases. Characters in red are wrongly predicted. Top string: the predictions from PETR. Bottom string: the corresponding ground-truth.

I. Analysis of Failure Cases

The failure cases of PETR are shown in Fig. 12. These cases can be divided into two categories: 1) for the special fonts that are rare in training (the first row of Fig. 12). It is difficult for PETR to capture the visual context, which also increases the

rectification burden of language model. 2) For the images with very poor visibility (the second row of Fig.12), our PETR fails to capture both visual and linguistic information for recognition. It is worth mentioning that these two problems also exist in recent other methods [12], [13], [19].

V. Conclusion

In this paper, we essentially explore the language modeling process in the transformer-based language model, and propose a progressive scene text recognizer (PETR) to take a further step toward accurate scene text recognition. The proposed PETR effectively enhances the capability of language model by progressively rectifying the language modeling guidance and effectively reducing the correction burden. Furthermore, we also visualize the relationship between the low-quality visual predictions and the correction burden of language model. To sufficiently express the linguistic information in multimodal characteristics, a new Learnable Fusion Module is proposed to aggregate visual and linguistic information in a common semantic space. The extensive experiments demonstrate the effectiveness of our method and the exhaustive ablation studies are also beneficial to other related researches. In the future, we will implement our method in end-to-end text spotting task to further exploit its potential.

Acknowledgment

The authors acknowledge the support of GPU cluster built by MCC Laboratory of the Information Science and Technology Institution, USTC.

References

[1] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, Nov. 2014.

[2] X. Bai, C. Yao, and W. Liu, "Strokelets: A learned multi-scale mid-level representation for scene text recognition," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2789–2802, Jun. 2016.

[3] P. Dai, H. Zhang, and X. Cao, "SLOAN: Scale-adaptive orientation attention network for scene text recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 1687–1701, 2021.

[4] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Semi-supervised scene text recognition," *IEEE Trans. Image Process.*, vol. 30, pp. 3005–3016, 2021.

[5] S. Fang, H. Xie, Z.-J. Zha, N. Sun, J. Tan, and Y. Zhang, "Attention and language ensemble for scene text recognition with convolutional sequence modeling," in *Proc. 26th ACM Int. Conf. Multimedia*, Oct. 2018, pp. 248–256.

[6] Z. Xie, Z. Sun, L. Jin, H. Ni, and T. Lyons, "Learning spatial-semantic context with fully convolutional recurrent network for online handwritten Chinese text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1903–1917, Aug. 2018.

[7] Z. Wan, M. He, H. Chen, X. Bai, and C. Yao, "TextScanner: Reading characters in order for robust scene text recognition," 2019, *arXiv:1912.12422*.

[8] C. Zhang, A. Gupta, and A. Zisserman, "Adaptive text recognition through visual matching," in *Proc. ECCV*, 2020, pp. 51–67.

[9] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "RobustScanner: Dynamically enhancing positional clues for robust text recognition," in *Proc. ECCV*, 2020, pp. 135–151.

[10] F. Sheng, Z. Chen, and B. Xu, "NRTR: A no-recurrence sequence-to-sequence model for scene text recognition," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 781–786.

[11] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[12] T. Wang *et al.*, "Decoupled attention network for text recognition," in *Proc. AAAI*, 2020, pp. 12216–12224.

[13] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "SEED: Semantics enhanced encoder–decoder framework for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13528–13537.

[14] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*.

[15] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for OCR in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2231–2239.

[16] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild: A survey," *ACM Comput. Surveys*, vol. 54, no. 2, pp. 1–35, 2021.

[17] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5076–5084.

[18] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[19] D. Yu *et al.*, "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12113–12122.

[20] P. Lyu, Z. Yang, X. Leng, X. Wu, R. Li, and X. Shen, "2D attentional irregular scene text recognizer," 2019, *arXiv:1906.05708*.

[21] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7098–7107.

[22] Y. Wang, H. Xie, S. Fang, J. Wang, S. Zhu, and Y. Zhang, "From two to one: A new scene text recognizer with visual language modeling network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 14194–14203.

[23] Y. Chen, Y. Bai, W. Zhang, and T. Mei, "Destruction and construction learning for fine-grained image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5157–5166.

[24] C. Yi and Y. Tian, "Scene text recognition in mobile applications by character descriptor and structure configuration," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 2972–2982, Jul. 2014.

[25] X. Yin, Z. Zuo, S. Tian, and C. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[26] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from web videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 542–554, Mar. 2018.

[27] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *Proc. NIPS*, 2014, pp. 1–10.

[28] M. Liao *et al.*, "Scene text recognition from two-dimensional perspective," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8714–8721.

[29] L. Neumann and J. Matas, "Real-time scene text localization and recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3538–3545.

[30] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 512–528.

[31] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1457–1464.

[32] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 4042–4049.

[33] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[34] Z. Xie, Y. Huang, Y. Zhu, L. Jin, Y. Liu, and L. Xie, "Aggregation cross-entropy for sequence recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6538–6547.

[35] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[36] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8610–8617.

[37] F. Zhan and S. Lu, "ESIR: End-to-end scene text recognition via iterative image rectification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2059–2068.
[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[39] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "AON: Towards arbitrarily-oriented text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5571–5579.
[40] Q. Wang *et al.*, "ReELFA: A scene text recognizer with encoded location and focused attention," in *Proc. Int. Conf. Document Anal. Recognit. Workshops (ICDARW)*, Sep. 2019, pp. 71–76.
[41] P. Sun *et al.*, "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14454–14463.
[42] C. Zhang *et al.*, "Look more than once: An accurate detector for text of arbitrary shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 10552–10561.
[43] S. Waqas Zamir *et al.*, "Multi-stage progressive image restoration," 2021, *arXiv:2102.02808*.
[44] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3937–3946.
[45] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020, *arXiv:2005.12872*.
[46] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2315–2324.
[47] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1484–1493.
[48] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2015, pp. 1156–1160.
[49] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proc. BMVC*, 2012, pp. 1–12.
[50] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 569–576.
[51] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.
[52] M. Yang *et al.*, "Symmetry-constrained rectification network for scene text recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9147–9156.
[53] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 532–548, Feb. 2021.
[54] Y. Wang and Z. Lian, "Exploring font-independent features for scene text recognition," in *Proc. ECCV*, 2020, pp. 1900–1920.
[55] Z. Wan, J. Zhang, L. Zhang, J. Luo, and C. Yao, "On vocabulary reliance in scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11425–11434.
[56] X. Zhou, S. Zhou, C. Yao, Z. Cao, and Q. Yin, "ICDAR 2015 text reading in the wild competition," 2015, *arXiv:1506.03184*.
[57] R. Zhang *et al.*, "ICDAR 2019 robust reading challenge on reading Chinese text on signboard," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1577–1581.
[58] F. Yin, Y.-C. Wu, X.-Y. Zhang, and C.-L. Liu, "Scene text recognition with sliding convolutional character models," 2017, *arXiv:1709.01727*.

Yuxin Wang received the B.S. degree from Xidian University in 2018. He is currently pursuing the Ph.D. degree with the School of Information Science and Technology, University of Science and Technology of China. His research interests include computer vision and signal processing.

Hongtao Xie received the Ph.D. degree in computer application technology from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2012. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include multimedia content analysis and retrieval, deep learning, and computer vision.

Shancheng Fang received the Ph.D. degree in computer software and theory from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China, in 2020. He is currently a Postdoctoral Fellow with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include multimedia analysis and computer vision.

Mengting Xing received the M.S. degree in information and communication engineering from the University of Science and Technology of China, Anhui, China, in 2022. She is currently a Computer Vision Algorithm Engineer with Baidu Intelligent Cloud (Chengdu) Technology Company Ltd. Her research interests include computer vision and signal processing.

Jing Wang received the bachelor's degree from the University of Science and Technology of China in 2010 and the Ph.D. degree from Nanyang Technological University, Singapore, in 2016. He has published more than ten articles related to artificial intelligence and data analysis, and provides more than 30 suggestions on CVE security vulnerabilities. His research interests include anomaly detection, machine learning related to finance, and deep learning-based image detection and recognition.

Shenggao Zhu received the bachelor's degree in electronic engineering and information science from the University of Science and Technology of China (USTC) in 2011 and the Ph.D. degree in computer science from National University of Singapore (NUS) in 2017. In 2017, he joined Huawei Cloud, where he is currently a Technical Expert. His research interests include computer vision and AI applications.

Yongdong Zhang (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from Tianjin University, Tianjin, China, in 2002. He is currently a Professor with the School of Information Science and Technology, University of Science and Technology of China, Hefei, China. He has authored more than 100 refereed journals and conference papers. His research interests include multimedia content analysis and understanding, multimedia content security, video encoding, and streaming media technology. He was a recipient of the Best Paper Awards in ICME 2010, PCM 2013, and ICIMCS 2013, and the Best Paper Candidate in ICME 2011. He is the Editorial Board Member of the *Multimedia Systems Journal* and the IEEE TRANSACTIONS ON MULTIMEDIA.