Primitive Representation Learning for Scene Text Recognition

Ruijie Yan Liangrui Peng Shanyu Xiao Gang Yao
Beijing National Research Center for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing, China
{yrj17, xiaosy19, yg19}@mails.tsinghua.edu.cn, penglr@tsinghua.edu.cn

Abstract

Scene text recognition is a challenging task due to diverse variations of text instances in natural scene images. Conventional methods based on CNN-RNN-CTC or encoder-decoder with attention mechanism may not fully investigate stable and efficient feature representations for multi-oriented scene texts. In this paper, we propose a primitive representation learning method that aims to exploit intrinsic representations of scene text images. We model elements in feature maps as the nodes of an undirected graph. A pooling aggregator and a weighted aggregator are proposed to learn primitive representations, which are transformed into high-level visual text representations by graph convolutional networks. A Primitive REpresentation learning Network (PREN) is constructed to use the visual text representations for parallel decoding. Furthermore, by integrating visual text representations into an encoder-decoder model with the 2D attention mechanism, we propose a framework called PREN2D to alleviate the misalignment problem in attention-based methods. Experimental results on both English and Chinese scene text recognition tasks demonstrate that PREN keeps a balance between accuracy and efficiency, while PREN2D achieves state-of-the-art performance.

1. Introduction

In recent years, there have been increasing demands for scene text recognition in various real-world applications, such as image search, instant translation, and robot navigation. With the emergence of deep learning, there are two main scene text recognition frameworks. One is the CRNN framework [48, 14, 15, 44, 31, 17] that encodes images into hidden representations by CNNs and RNNs, and uses the connectionist temporal classification (CTC) [10] for decoding, as shown in Fig. 1 (a). The other is the attention-based encoder-decoder framework [2, 44, 24, 7, 3, 32, 43, 54, 60, 29, 40] that can learn to align output texts with feature maps, as shown in Fig. 1 (b).

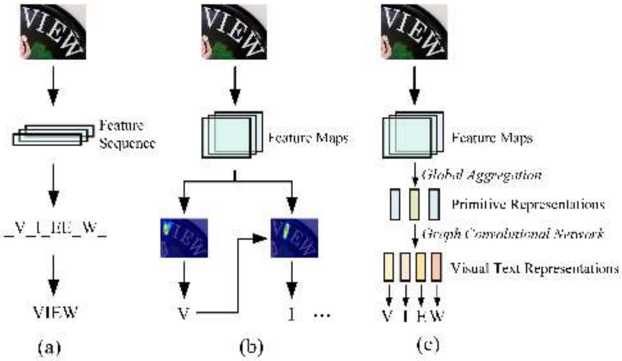

Figure 1. Illustrations of different scene text recognition frameworks. (a) CTC-based methods, where "_" denotes the blank symbol; (b) attention-based methods; (c) the proposed PREN.

However, the above methods still have room for improvement. On the one hand, for CTC-based methods, the extracted feature sequences contain redundant information that may degrade the performance on irregular text images. On the other hand, attention-based encoder-decoder methods usually suffer from the misalignment problem [7, 54], because the alignment between feature maps and texts is highly sensitive to previous decoded results, which lack global visual information. Therefore, to handle the diversity of texts in natural scenes, it is important to exploit intrinsic representations of scene text images.

In this paper, we propose a novel scene text recognition framework that learns primitive representations of scene text images. Inspired by graph representation learning methods [22, 12, 38], we model the elements in feature maps as nodes of an undirected graph. Primitive representations are learned by globally aggregating features over the coordinate space and are then projected into the visual text representation space, as shown in Fig. 1 (c).

The "primitive" representations refer to a set of base vectors that can be transformed into character-by-character vector representations in the so-called visual text representation space. The visual text representations are generated from original feature maps, which are different from character embeddings generated from ground truth or predicted texts used in an encoder-decoder model.

For the global feature aggregation, a pooling aggregator and a weighted aggregator are proposed. For the pooling aggregator, each primitive representation is learned from input feature maps through two convolutions followed by a global average pooling layer. In this way, aggregating weights are shared by all samples to learn intrinsic structural information from various scene text instances. For the weighted aggregator, input feature maps are transformed into sample-specific heatmaps, which are used as aggregating weights.

Visual text representations are generated from primitive representations by graph convolutional networks (GCNs) [22, 6]. Each visual text representation is used to represent a character to be recognized.

A primitive representation learning network (PREN) is constructed. PREN consists of a feature extraction module that extracts multiscale feature maps from input images and a primitive representation learning module that learns primitive representations and generates visual text representations. Texts are generated from visual text representations with parallel decoding.

Moreover, since visual text representations are purely learned from visual features, they can mitigate the misalignment problem [7, 54] of attention-based methods. We further construct a framework called PREN2D by integrating PREN into a 2D-attention-based encoder-decoder model with a modified self-attention network.

We conduct experiments on seven public English scene text recognition datasets (IIIT5k, SVT, IC03, IC13, IC15, SVTP, and CUTE) and a subset of the RCTW Chinese scene text dataset. Experimental results show that PREN keeps a balance between accuracy and speed, while PREN2D achieves state-of-the-art model performance.

In summary, the main contributions of the paper are as follows.

- Different from commonly used CTC-based and attention-based methods, we provide a novel scene text recognition framework by learning primitive representations and forming visual text representations that can be used for parallel decoding.
- We propose a pooling aggregator and a weighted aggregator to learn primitive representations from feature maps output by a CNN, and use GCNs to transform primitive representations into visual text representations.
- The proposed primitive representation learning method can be integrated into attention-based frameworks. Experimental results on both English and Chinese scene text recognition tasks demonstrate the effectiveness and efficiency of our method.

2. Related Work

2.1. Scene text recognition

Scene text recognition methods can be generally divided into segmentation-based methods and sequence-based methods. For segmentation-based methods [52, 53, 37, 58, 57, 23, 19, 64, 27], individual characters are segmented or localized before recognition, and character-level annotations are often required to train these models. For sequence-based methods, CTC-based methods [10, 48, 14, 44, 31, 17] and encoder-decoder frameworks with attention mechanisms [2, 44, 24, 7, 3, 32, 43, 54, 60, 29, 40] are two major techniques to recognize scene text images.

In contrast to CTC-based methods, attention-based encoder-decoder methods can learn the dependencies among the output characters, which can be regarded as using an implicit language model. However, the efficiency of attention-based methods is usually limited by the recursive decoding process. To increase the decoding speed while maintaining high recognition performance, Hu et al. [17] proposed training a CTC-based model with the guidance of an attention branch. Lyu et al. [35] developed a two-stage decoder with a relation attention module. Yu et al. [59] proposed a parallel visual attention module followed by a self-attention network with multi-way parallel transmission to learn semantic information explicitly. Different from these methods, we propose a novel scene text recognition framework with parallel decoding based on primitive representation learning.

Recently, the recognition of irregular scene texts has attracted a lot of research interests. The solutions include text rectification [45, 46, 34, 30, 56, 62], hierarchical attention mechanism [30], and multidirectional feature extraction [8]. Models with the 2D attention mechanism [55, 25, 35] have also shown strong effectiveness on irregular text recognition by retaining 2D spatial information in features. Our proposed primitive representation learning method can be integrated into 2D-attention-based frameworks to improve recognition performance.

2.2. Representation learning by feature aggregation

Representation learning has become the basis of most deep-learning methods due to its ability to learn data representations that make it easier to extract useful information when building classifiers or other predictors [4]. Feature aggregation is a commonly used method in graph representation learning tasks. GCNs [22, 6] aggregate neighboring vertex features by exploiting the graph topology. Instead of using all neighboring nodes, GraphSAGE [12] uses random walk [5] to sample several neighboring nodes, and the feature aggregation can be accomplished by a mean aggregator, a pooling aggregator, or an LSTM aggregator. Petar et al. [38] proposed the graph attention network (GAT) that

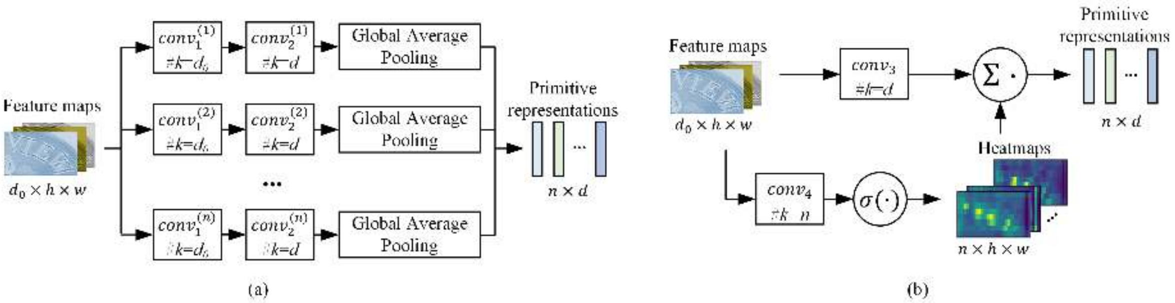

Figure 2. Two primitive representation learning methods. (a) Pooling aggregator. Two convolutional layers followed by a global average pooling layer aggregates input feature maps into a primitive representation. $\#k$ denotes the number of kernels of the convolutional layer. (b) Weighted aggregator. Input feature maps are transformed into n heatmaps. Each heatmap is used as aggregating weights of a primitive representation. $\sigma(\cdot)$ is the sigmoid activation function, and $\Sigma\cdot$ refers to scaled-dot product and summation.

learns to assign different aggregating weights to different nodes by leveraging the self-attention mechanism. Inspired by the above progress, we propose to learn primitive representations by global feature aggregations and use GCNs to transform primitive representations into visual text representations.

3. Methodology

In this section, we first describe methods for learning primitive representations and visual text representations, and then provide detailed structures of PREN and PREN2D.

3.1. Primitive representation learning

We propose learning primitive representations by using global feature aggregations over the coordinate space. In this way, primitive representations contain global information of the input image that is beneficial for the subsequent recognition process. Let $F \in \mathbb{R}^{d_0 \times h \times w}$ be feature maps extracted by a CNN, where h, w, and d_0 are the height, width, and number of channels of F, respectively. The elements in feature maps are taken as the nodes of an undirected graph, i.e., we convert F to a feature matrix $X \in \mathbb{R}^{m_0 \times d_0}$, where $m_0 = h \times w$. Let n be the number of primitive representations to learn, the feature aggregation process can be formulated as

$$Z_i = f^{(i)}(X),\ i = 1, 2, ..., n \tag{1}$$

$$\boldsymbol{p}_i = \boldsymbol{a}_i \cdot Z_i,\ i = 1, 2, ..., n \tag{2}$$

where $\boldsymbol{p}_i \in \mathbb{R}^{1 \times d}$ is the i-th primitive representation. $f^{(i)}(\cdot)$ is the mapping function of a sub-network that transforms $X \in \mathbb{R}^{m_0 \times d_0}$ into a hidden representation $Z_i \in \mathbb{R}^{m \times d}$. $\boldsymbol{a}_i \in \mathbb{R}^{1 \times m}$ is the aggregating weights of the i-th primitive representation. The n primitive representations are concatenated as $P = [\boldsymbol{p}_1; \boldsymbol{p}_2; ...; \boldsymbol{p}_n] \in \mathbb{R}^{n \times d}$.

We propose two kinds of aggregation methods with different aggregating weights $\boldsymbol{a}_i$ $(i = 1, 2, ..., n)$, i.e., a pooling aggregator and a weighted aggregator.

3.1.1 Pooling aggregator

As shown in Fig. 2 (a), a global average pooling layer is used for feature aggregation, which is equivalent to setting $a_{ij} = \frac{1}{m}$, $\forall j = 1, 2, ..., m$ in Equ. (2). The global average pooling has been proven effective for learning global information [28, 16]. In this way, the aggregating weights are shared by all samples to exploit intrinsic structural information from various scene text instances.

The function $f^{(i)}(\cdot)$ in Equ. (1) is implemented as two convolultions that conduct on the original feature maps F. Each convolution has kernel size = 3 and stride = 2. The calculation of primitive representations can be formulated as

$$\boldsymbol{p}_i = Pool(conv_2^{(i)}(\phi(conv_1^{(i)}(F)))) \tag{3}$$

where $\phi(\cdot)$ denotes an activation function. Different from the pooling aggregator used in GraphSAGE [12], we use additional convolutional layers before the pooling layer to better learn spatial information of scene text images.

3.1.2 Weighted aggregator

Due to the diversity of text instances in natural scene images, it is also important to learn sample-specific information. Therefore, we propose learning aggregating weights from input features dynamically.

As shown in Fig. 2 (b), a hidden representation $Z \in \mathbb{R}^{d \times h \times w}$ is obtained by a 3×3 convolutional layer $conv_3(\cdot)$. Another 3×3 convolutional layer $conv_4(\cdot)$ followed by a sigmoid activation function is used to convert input feature maps F to n heatmaps $H \in \mathbb{R}^{n \times h \times w}$. Aggregating weights $\boldsymbol{a}_i$ can be obtained by flattening the i-th heatmap H_i. Primitive representations can be calculated by a scale-dot product and summation operation, i.e., we have

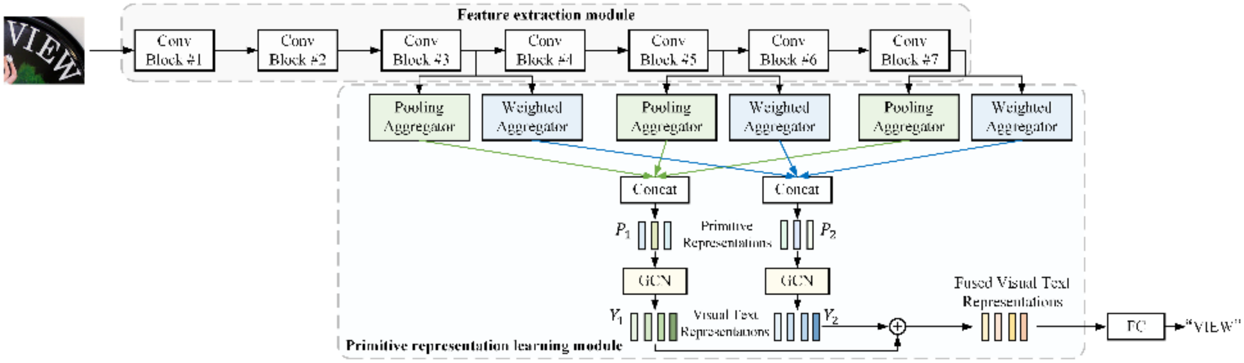

Figure 3. The system framework of PREN that consists of a feature extraction module and a primitive representation learning module. Both pooling aggregators and weighted aggregators learn primitive representations P_1 and P_2 from feature maps. Visual text representations Y_1 and Y_2 are obtained from primitive representations P_1 and P_2 by two GCNs, and are summed into fused visual text representations for parallel decoding.

$$Z = \phi(conv_3(F)) \tag{4}$$

$$H = \sigma(conv_4(F)) \tag{5}$$

$$\boldsymbol{p}_i = \sum_{x=1}^{h}\sum_{y=1}^{w} H_{i,x,y} Z_{x,y} \tag{6}$$

3.2. Visual text representation generation

Since primitive representations contain global information of the input image, textual information can be extracted from primitive representations. We propose to generate visual text representations through a linear combination of primitive representations followed by a fully-connected layer, which can be formulated as

$$Y = \phi(BPW) \tag{7}$$

where $P \in \mathbb{R}^{n\times d}$ denotes primitive representations. $B \in \mathbb{R}^{L\times n}$ is a coefficient matrix of the linear combination, and L is a maximum decoding length. $W \in \mathbb{R}^{d\times d}$ is a weight matrix, and $\phi(\cdot)$ is an activation function.

Equ. (7) can be implemented by using a GCN [22], where the coefficient matrix B plays a similar role to an adjacency matrix. Since there is no explicit graph topology for primitive representations, B is randomly initialized and learned in the training stage.

Each visual text representation $\boldsymbol{y}_i$ ($i = 1, 2, ..., L$) is used to represent a character to be recognized. For text string shorter than L, the excess part of Y corresponds to padding symbols.

3.3. Primitive representation learning network

3.3.1 Overview of PREN

As shown in Fig. 3, PREN consists of a feature extraction module and a primitive representation learning module. Three pooling aggregators and three weighted aggregators are used to learn primitive representations from multiscale feature maps. Let P_1 and P_2 denote primitive representations learned by pooling aggregators and weighted aggregators, respectively. Visual text representations Y_1 and Y_2 are obtained by two GCNs and are summed into fused visual text representations Y. A fully-connected layer is used to convert Y into logits for parallel decoding.

3.3.2 Feature extraction module

We use EfficientNet-B3 [50] as the feature extraction module, which consists of seven mobile inverted bottlenecks (MBConv blocks) [42, 49], as marked by "Conv Block #1" to "Conv Block #7" in Fig. 3.

We denote the feature maps output by the i-th convolutional block by F_i. To take advantage of multiscale features, feature maps F_3, F_5, and F_7, which are $1/8$, $1/16$, and $1/32$ the input image scale, are used as inputs for the primitive representation learning module.

3.3.3 Primitive representation learning module

For feature maps output by each selected convolutional block, both a pooling aggregator and a weighted aggregator are used to learn primitive representations. Let d denote the number of channels of F_7 and n be the number of primitive representations to learn. The output of each feature aggregator has the dimension of $\mathbb{R}^{n\times\frac{d}{3}}$. As shown in Fig. 3, the outputs of the three pooling aggregators are concatenated as $P_1 \in \mathbb{R}^{n\times d}$, and the outputs of the three weighted aggregators are concatenated as $P_2 \in \mathbb{R}^{n\times d}$.

Two GCNs are used to generate visual text representations Y_1 and Y_2 from primitive representations P_1 and P_2 respectively. Y_1 and Y_2 are summed into fused visual text representations Y. The probability of each character is computed from Y through a fully-connected layer followed by softmax. Therefore, the decoding process of PREN is fully parallel and efficient.

3.4. Incorporating the 2D attention mechanism

The visual text representations output by PREN are also flexible to integrate into attention-based encoder-decoder models to alleviate the misalignment problem [7, 54]. For attention-based methods, the alignment between texts and feature maps relies on previous decoded results. Since visual text representations are purely learned from visual features, they can provide global visual information that helps learn stable and accurate alignments.

Based on the above analysis, we construct PREN2D by combining PREN and a baseline model with the 2D attention mechanism. As shown in Fig. 4, the feature extraction module is shared by both PREN and the encoder-decoder module based on a modified Transformer model [51]. Visual text representations output by PREN are used to augment character embeddings of ground truth texts in the training stage or previous decoded texts in the inference stage, which can provide global guidance in the encoder-decoder attention calculation in the modified Transformer model.

For the feature extraction module, the outputs of the final convolutional block F_7 are upsampled and added to F_5, and the results are unsampled again and added to F_3. The obtained 2D feature maps $F \in \mathbb{R}^{d \times h \times w}$ have the same resolution as F_3 and the same number of channels as F_7.

In the original Transformer model, the encoder and decoder have $N = 6$ identical blocks. In our model, the encoder and decoder are simplified to have $N = 2$ identical blocks. For the encoder, we propose a modified self-attention mechanism that can be formulated as

$$q_i = f(\mathcal{N}(f_i)) \cdot W_Q \tag{8}$$

$$k_j = g(\mathcal{N}(f_j)) \cdot W_K \tag{9}$$

$$\alpha_{ij} = softmax(\frac{1}{\sqrt{d}} q_i k_j^T) \tag{10}$$

$$v_i = \sum_{j=1}^{m} \alpha_{ij} x_j W_V \tag{11}$$

where $f_i \in \mathbb{R}^{1 \times d}$ $(i = 1, 2, ..., m)$ is the i-th element in feature maps F, and $m = h \times w$. $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ are three learnable matrices with respect to queries, keys and values. $\mathcal{N}(f_i)$ denotes spatially adjacent elements of i. $f(\mathcal{N}(f_i))$ and $g(\mathcal{N}(f_j))$ are implemented as two 3×3 convolutional layers. In this way, the encoder can better model local spatial relationships during the computation of the attention weight α_{ij}.

A Transformer decoder [51] is used for text transcription. We use a gated unit to combine visual text representations and character embeddings. Let Y and E denote visual text representations and character embeddings, respectively. V and O are encoder outputs and decoder outputs, respectively. Formally, the calculation process of the decoder is

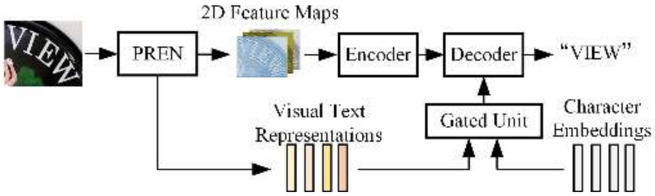

Figure 4. Illustration of PREN2D. At each decoding step t, the t-th character embedding is combined with the t-th visual text representation by a gated unit.

$$z = \sigma([Y, E] \cdot W_z) \tag{12}$$

$$E' = z \odot Y + (1 - z) \odot E \tag{13}$$

$$O = f_{dec}(E', V) \tag{14}$$

where $[\cdot]$ refers to concatenation, W_z is a learnable weight matrix, $\odot$ denotes element-wise product, and $f_{dec}(\cdot)$ is the mapping function of the decoder.

3.5. Training and inference

Both PREN and PREN2D can be trained end-to-end with cross-entropy between the prediction and ground truth. The ground truth is generated by adding an ending symbol $\langle$eos$\rangle$ after the last character and expanded to a maximum decoding length with padding symbols $\langle$pad$\rangle$. Let l denote the length of the original text, and the loss is calculated according to

$$\mathcal{L} = -\sum_{t=1}^{l+1} logp(g_t | I) \tag{15}$$

where I refers to the input image, g_t $(t = 1, 2, ..., l)$ is the t-th character, and g_{l+1} is the ending symbol $\langle$eos$\rangle$. Padding symbols $\langle$pad$\rangle$ are ignored during the loss computation.

In the inference stage, PREN predicts the whole text at one step, while PREN2D recognizes characters recursively. The presence of the first ending symbol $\langle$eos$\rangle$ in the decoding results indicates the end of decoding.

4. Experiments

We conduct both English and Chinese scene text recognition experiments. For English scene text recognition, we compare our method with previous state-of-the-art methods and conduct a series of ablation studies to explore the effect of each part of our models. For Chinese scene text recognition, we evaluate the performance of our method on a multi-oriented text recognition task.

Table 1. Word recognition accuracy (%) across methods and datasets. MJ, ST, Char, and Add denote MJSynth [18], SynthText [11], character bounding boxes, and additional training data, respectively. The method with the * symbol had its results reported in Baek et al. [1], where a reimplemented model is trained on MJ+ST. The best results of models trained on MJ+ST are marked in **bold**.

Model	Training data	Regular Test Datasets				Irregular Test Datasets		
		IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE
Mask TextSpotter (Liao et al.) [26]	MJ+ST+Char	95.3	91.8	95.0	95.3	78.2	83.6	88.5
SAR (Li et al.) [25]	MJ+ST+Add	95.0	91.2	-	94.0	78.8	86.4	89.6
SCATTER (Litman et al.) [29]	MJ+ST+Add	93.7	92.7	96.3	93.9	82.2	86.9	87.5
CRNN (Shi et al.) [44, 1]*	MJ+ST	82.9	81.6	93.1	89.2	64.2	70.0	65.5
AON (Cheng et al.) [8]	MJ+ST	87.0	82.8	91.5	-	68.2	73.0	76.8
DAN (Wang et al.) [54]	MJ+ST	94.3	89.2	95.0	93.9	74.5	80.0	84.4
ASTER (Shi et al.) [46]	MJ+ST	93.4	89.5	94.5	91.8	76.1	78.5	79.5
SE-ASTER (Qiao et al.) [40]	MJ+ST	93.8	89.6	-	92.8	80.0	81.4	83.6
AutoSTR (Zhang et al.) [63]	MJ+ST	94.7	90.9	93.3	94.2	81.8	81.7	-
RobustScanner (Yue et al.) [60]	MJ+ST	95.3	88.1	-	94.8	77.1	79.5	90.3
SRN (Yu et al.) [59]	MJ+ST	94.8	91.5	-	95.5	82.7	85.1	87.8
CNN-LSTM-CTC	MJ+ST	92.0	89.8	93.1	93.9	76.7	80.6	80.9
PREN	MJ+ST	92.1	92.0	94.9	94.7	79.2	83.9	81.3
Baseline2D	MJ+ST	95.4	93.4	95.4	95.9	81.9	86.0	89.9
PREN2D	MJ+ST	**95.6**	**94.0**	**95.8**	**96.4**	**83.0**	**87.6**	**91.7**

4.1. English scene text recognition

4.1.1 Experimental setup

For English scene text recognition, our models are trained on two commonly used public synthetic scene text datasets, i.e., MJSynth (MJ) [18] and SynthText (ST) [11]. The model performance is tested on seven public real scene text datasets: IIIT5k-Words (IIIT5k) [36], Street View Text (SVT) [52], ICDAR 2003 (IC03) [33], ICDAR 2013 (IC13) [21], ICDAR 2015 (IC15) [20], SVT-Perspective (SVTP) [39], and CUTE80 (CUTE) [41]. There are various divisions for test sets of IC13 and IC15. We follow the protocol of Yu et al. [59] where the IC13 test set consists of 857 images and the IC15 test set contains 1811 images.

For ablation studies, all models are trained for three epochs. The learning rate of the first two epochs is set to 0.5 and decreased to 0.1 at the third epoch. When compared with other state-of-the-art methods, we continue to train the models for another five epochs. The learning rate is initialized to 0.1 and decreased to 0.01 and 0.001 at the third epoch and the fifth epoch, respectively. The training batch size is set to 128, and ADADELTA [61] is adopted as the optimizer. Input images are normalized into 64 × 256 pixels. The alphabet includes all case-insensitive alphanumerics. The number of primitive representations is 5. The maximum decoding length is set to 25 since the lengths of most common English words are less than 25. Word accuracy is used as the performance evaluation index.

4.1.2 Comparison with state-of-the-art methods

The comparison of our models with previous state-of-the-art methods is shown in Table 1. To better observe the performance gain of primitive representation learning, we also train a CTC-based model (CNN-LSTM-CTC) by replacing the CNN in the CRNN [44] with an EfficientNet-B3 [50], and train a baseline model with 2D attention mechanism (Baseline2D). Baseline2D has the same feature extraction module, encoder-decoder module, and training configurations as used in PREN2D.

PREN achieves better recognition accuracy on all test sets than CNN-LSTM-CTC. By exploiting visual text representations, PREN2D outperforms Baseline2D on all test sets. In particular, accuracy gains of 1.1%, 1.6%, and 1.8% are obtained on irregular text datasets IC15, SVTP, and CUTE, respectively. PREN2D also achieves higher accuracy than previous state-of-the-art models that are trained on the MJSynth [18] and SynthText [11] datasets. The recognition performance on both regular and irregular scene text image datasets shows the effectiveness of our method.

4.1.3 Comparison of computation cost

Table 2. Comparison of the recognition speeds of various models. DL. Framework refers to deep-learning framework.

Model	DL. Framework	NVIDIA GPU	Time
CNN-LSTM-CTC	PyTorch	Tesla V100	23.6ms
PREN			22.7ms
Baseline2D			61.6ms
PREN2D			67.4ms

Table 2 shows the average recognition speeds of various models on a single image. PREN has a slightly higher recognition speed than CNN-LSTM-CTC. Compared with Baseline2D, the extra time consumption of PREN2D is only 5.8ms on average.

4.1.4 Comparison of feature aggregation methods

Table 3. Word accuracy (%) of PRENs with various feature aggregation methods.

Aggregator	IIIT5k	IC03	IC13	SVTP	CUTE
Pooling	91.1	93.5	**94.7**	79.2	77.4
Weighted	90.0	92.0	93.2	79.5	77.4
Pooling + Weighted	**91.8**	**93.9**	**94.7**	**81.7**	**81.3**

In this experiment, we study the effect of various feature aggregation methods. We compare PRENs with only pooling aggregators or only weighted aggregators, as well as with both pooling aggregators and weighted aggregators.

Table 3 lists the comparison results. Two phenomena can be observed. (1) The model with weighted aggregators has lower recognition accuracy than the model with pooling aggregators on regular text datasets (IIIT5k, IC03, and IC13), but achieves equal or higher recognition accuracy on irregular text datasets (SVTP and CUTE). (2) Combining the two aggregation methods can significantly improve recognition performance, especially on irregular text datasets.

4.1.5 Comparison of various numbers of primitive representations

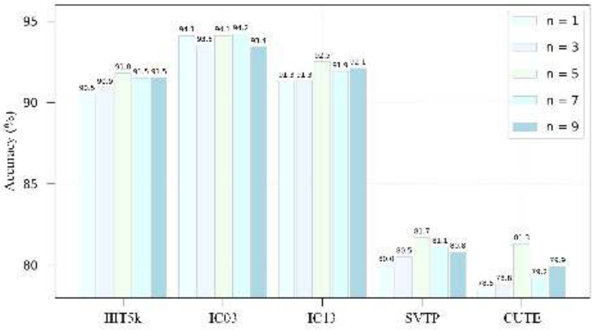

Figure 5. Comparison of PRENs with various numbers of primitive representations n.

Fig. 5 shows the comparison results of PRENs with various numbers of primitive representations. Too few or too many primitive representations will cause performance degradation. Learning five primitive representations achieves the best recognition performance on the IIIT5k, IC13, SVTP, and CUTE test sets.

4.2. Chinese scene text recognition

We further conduct a Chinese scene text recognition experiment. There are thousands of commonly used Chinese characters, and multi-oriented texts are common in Chinese scene images. Therefore, Chinese scene text recognition is a challenging task that can evaluate the robustness of scene text recognizers.

4.2.1 Experimental setup

For Chinese scene text recognition, our models are first trained on a self-built synthetic dataset, and then fine-tuned and tested on real samples. For the synthetic dataset, 1 million images are synthesized by following Gupta et al. [11] with the corpus collected from THUOCL [13]. The real samples with multiple orientations including horizontal, vertical, and skewed texts are selected from the RCTW [47] dataset. The training set for fine-tuning consists of 6000 images, and the test set includes 1000 images.

Since CTC-based models encode input images into feature sequences, a fixed normalized height is required for all images [9]. As a result, for CNN-LSTM-CTC, vertical text images are rotated 90 degrees first, and all images are normalized to 64×256 pixels. In contrast, PREN, Baseline2D, and PREN2D can handle input images with multiple orientations. We divide the whole samples into horizontal and vertical subsets according to aspect ratios of original images. Horizontal and vertical text images are normalized into 64×256 pixels and 256×64 pixels, respectively. In the training stage, data of each training iteration is randomly taken from the horizontal subset or the vertical subset.

All models are trained on synthetic samples for 6 epochs and fine-tuned on real samples for 20 epochs. The learning rate is initialized to 0.5 and decreased to 0.1 at the sixth epoch. The character set contains 5658 characters.

4.2.2 Comparison of different models

Table 4. Word accuracy (%) of different models for multi-oriented Chinese scene text recognition.

Model	Horizontal	Vertical	Average
CNN-LSTM-CTC	53.4	64.8	59.1
PREN	73.8	79.2	76.5
Baseline2D	82.2	86.8	84.5
PREN2D	**82.6**	**87.4**	**85.0**

Table 4 shows the comparison results of different models. For CNN-LSTM-CTC, the rotation of vertical text images doubles the patterns to learn, while PREN can avoid this problem and achieve significantly higher accuracy. Baseline2D and PREN2D have better performance. One possible reason is that Chinese texts contain a lot of similar characters, thus the implicit language model learned by the encoder-decoder architecture is important for accurate recognition. Compared with Baseline2D, PREN2D achieves higher recognition accuracy on both horizontal and vertical test sets, which demonstrates the effectiveness of the proposed primitive representation learning method.

4.3. Visualization and analysis

Qualitative comparison of different models. Table 5 shows the qualitative comparison of different models. Irregular text images are challenging for CNN-LSTM-CTC. For PREN, errors are mainly caused by similar characters such as "O" and "D". Baseline2D suffers from the misalignment problem, e.g., the last character is repeatedly recognized twice for the third sample. PREN2D shows better robustness than the other three models.

Table 5. Qualitative comparison of different models. For Chinese texts, characters in Unicode form are also listed. Wrongly recognized characters are marked in red.

Input Image				
Ground Truth	VACATION	BOOKSTORE	售楼部 u+552e u+697c u+90e8	品安化黑茶 u+54c1 u+5b89 u+5316 u+9ed1 u+8336
CNN-LSTM-CTC	VJACATION	BOOKSTOVE	香楼部 u+9999 u+697c u+90e8	康莱坳黑茶 u+5eb7 u+83b1 u+5773 u+9ed1 u+8336
Baseline2D	VAGATION	BOOKSTONE	售楼部部 u+552e u+697c u+90e8 u+90e8	品安化里茶 u+54c1 u+5b89 u+5316 u+91cc u+8336
PREN	VACATION	BOOKSTDGE	售楼部 u+552e u+697c u+90e8	品安化黑茶 u+54c1 u+5b89 u+5316 u+9ed1 u+8336
PREN2D	VACATION	BOOKSTORE	售楼部 u+552e u+697c u+90e8	品安化黑茶 u+54c1 u+5b89 u+5316 u+9ed1 u+8336

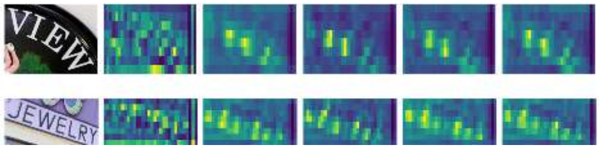

Figure 6. Visualization of heatmaps generated by the weighted aggregator that learns $n = 5$ primitive representations.

Visualization of different aggregators. Fig. 6 shows the heatmaps generated by the weighted aggregator. The first heatmap has larger values in character boundary areas, while the other heatmaps focus on character center areas.

For the pooling aggregator, the feature maps before pooling (i.e., after $conv_2(\cdot)$ and before $Pool(\cdot)$ in Eq. (3)) show the contribution of each part of feature maps to primitive representations. As shown in Fig. 7, for various input images, feature maps corresponding to the same primitive representation are similar, e.g., the first feature map generally has larger responses on the bottom parts and the second feature map focuses on the upper-left and lower-right parts.

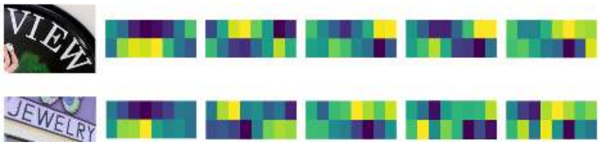

Figure 7. Feature maps before global average pooling of the pooling aggregator that learns $n = 5$ primitive representations. Values are averaged in the channel dimension for visualization.

The visualizations in Fig. 6 and Fig. 7 indicate that the pooling aggregator can learn common structural information from various text instances, and the weighted aggregator has a better ability to distinguish foreground and background areas.

Visualization of PREN2D. Fig. 8 visualizes the attention scores generated by Baseline2D and PREN2D for an input image. By utilizing visual text representations, PREN2D can generate more accurate attention areas and alleviate incorrect alignments. For example, Baseline2D wrongly aligns the right part of "N" in the image with the character "I". In contrast, the attention map of PREN2D covers the central region of "N", in which the alignment is correct.

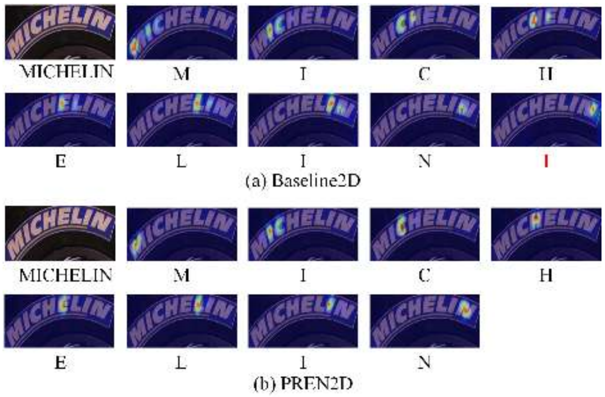

Figure 8. An example of attention scores generated by (a) the baseline model and (b) PREN2D. Texts under the input images are the ground truth, and the characters under attention maps are recognized results. The baseline model incorrectly recognizes "MICHELIN" as "MICHELINI", while PREN2D outputs correct results.

5. Conclusion

In this paper, we propose a primitive representation learning method for scene text recognition. Visual text representations generated from primitive representations can be either directly used for parallel decoding, or further integrated into a 2D-attention-based encoder-decoder framework to improve recognition performance. In future work, we will investigate more possible ways of learning primitive representations.

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, et al. What is wrong with scene text recognition model comparisons? Dataset and model analysis. In *ICCV*, pages 4715–4723, 2019. 6

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1, 2

[3] Fan Bai, Zhanzhan Cheng, Yi Niu, et al. Edit probability for scene text recognition. In *CVPR*, pages 1508–1516, 2018. 1, 2

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013. 2

[5] Perozzi Bryan, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *SIGKDD*, pages 701–710, 2014. 2

[6] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, et al. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019. 2

[7] Zhanzhan Cheng, Fan Bai, Yunlu Xu, et al. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, pages 5076–5084, 2017. 1, 2, 5

[8] Zhanzhan Cheng, Yangliu Xu, Fan Bai, et al. AON: Towards arbitrarily-oriented text recognition. In *CVPR*, pages 5571–5579, 2018. 2, 6

[9] Chankyu Choi, Youngmin Yoon, Junsu Lee, et al. Simultaneous recognition of horizontal and vertical text in natural images. In *ACCV*, pages 202–212, 2018. 7

[10] Alex Graves, Santiago Fernández, Faustino Gomez, et al. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006. 1, 2

[11] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 6, 7

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NIPS*, pages 1024–1034, 2017. 1, 2, 3

[13] Shiyi Han, Yuhui Zhang, Yunshan Ma, et al. THUOCL: Tsinghua open Chinese lexicon. 2016. http://thuocl.thunlp.org/. 7

[14] Pan He, Weilin Huang, Yu Qiao, et al. Reading scene text in deep convolutional sequences. *arXiv preprint arXiv:1506.04395*, 2015. 1, 2

[15] Pan He, Weilin Huang, Yu Qiao, et al. Reading scene text in deep convolutional sequences. In *AAAI*, pages 3501–3508, 2016. 1

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 3

[17] Wenyang Hu, Xiaocong Cai, Jun Hou, et al. GTC: Guided training of CTC towards efficient and accurate scene text recognition. In *AAAI*, pages 11005–11012, 2020. 1, 2

[18] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, et al. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014. 6

[19] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *ECCV*, pages 512–528, 2014. 2

[20] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, et al. ICDAR 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015. 6

[21] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, et al. ICDAR 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013. 6

[22] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 1, 2, 4

[23] Chen-Yu Lee, Anurag Bhardwaj, Wei Di, Vignesh Jagadeesh, and Robinson Piramuthu. Region-based discriminative feature pooling for scene text recognition. In *CVPR*, pages 4050–4057, 2014. 2

[24] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for OCR in the wild. In *CVPR*, pages 2231–2239, 2016. 1, 2

[25] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, pages 8610–8617, 2019. 2, 6

[26] Minghui Liao, Pengyuan Lyu, Minghang He, et al. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *arXiv preprint arXiv:1908.08207*, 2019. 6

[27] Minghui Liao, Jian Zhang, Zhaoyi Wan, et al. Scene text recognition from two-dimensional perspective. In *AAAI*, pages 8714–8721, 2019. 2

[28] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, pages 1–10, 2014. 3

[29] Ron Litman, Oron Anschel, Shahar Tsiper, et al. SCATTER: Selective context attentional scene text recognizer. In *CVPR*, pages 11962–11972, 2020. 1, 2, 6

[30] Wei Liu, Chaofeng Chen, and Kwan-Yee K Wong. Char-Net: A character-aware neural network for distorted scene text recognition. In *AAAI*, pages 7154–7161, 2018. 2

[31] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, et al. STAR-Net: A spatial attention residue network for scene text recognition. In *BMVC*, 2016. 1, 2

[32] Zichuan Liu, Yixing Li, Fengbo Ren, et al. Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network. In *AAAI*, pages 7194–7201, 2018. 1, 2

[33] Simon M Lucas, Alex Panaretos, Luis Sosa, et al. ICDAR 2003 robust reading competitions: Entries, results, and future directions. *IJDAR*, 7(2-3):105–122, 2005. 6

[34] Canjie Luo, Lianwen Jin, and Zenghui Sun. MORAN: A multi-object rectified attention network for scene text recognition. *Pattern Recog.*, 90:109–118, 2019. 2

[35] Pengyuan Lyu, Zhicheng Yang, Xinhang Leng, et al. 2D attentional irregular scene text recognizer. *arXiv preprint arXiv:1906.05708*, 2019. 2

[36] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012. 6

[37] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *CVPR*, pages 3538–3545, 2012. 2

[38] Veličković Petar, Guillem Cucurull, Arantxa Casanova, et al. Graph attention networks. In *ICLR*, pages 1–12, 2018. 1, 2

[39] T. Q. Phan, P. Shivakumara, S. Tian, et al. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013. 6

[40] Zhi Qiao, Yu Zhou, Dongbao Yang, et al. SEED: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, pages 13528–13537, 2020. 1, 2, 6

[41] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, et al. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.*, 41(18):8027–8048, 2014. 6

[42] Or Sharir and Amnon Shashua. On the expressive power of overlapping architectures of deep learning. *arXiv preprint arXiv:1703.02065*, 2017. 4

[43] Fenfen Sheng, Zhineng Chen, and Bo Xu. NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. *arXiv preprint arXiv:1806.00926*, 2019. 1, 2

[44] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(11):2298–2304, 2017. 1, 2, 6

[45] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, et al. Robust scene text recognition with automatic rectification. In *CVPR*, pages 4168–4176, 2016. 2

[46] Baoguang Shi, Mingkun Yang, Xinggang Wang, et al. ASTER: An attentional scene text recognizer with flexible rectification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(9):2035–2048, 2019. 2, 6

[47] Baoguang Shi, Cong Yao, Minghui Liao, et al. ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). In *ICDAR*, pages 1429–1434, 2017. 7

[48] Bolan Su and Shijian Lu. Accurate scene text recognition based on recurrent neural network. In *ACCV*, pages 35–48, 2014. 1, 2

[49] Mingxing Tan, Bo Chen, Ruoming Pang, et al. MNASNET: Platform-aware neural architecture search for mobile. In *CVPR*, pages 2820–2828, 2019. 4

[50] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 4, 6

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 5

[52] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011. 2, 6

[53] Tao Wang, David J Wu, Adam Coates, and Andrew Y Ng. End-to-end text recognition with convolutional neural networks. In *ICPR*, pages 3304–3308, 2012. 2

[54] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, et al. Decoupled attention network for text recognition. *arXiv preprint arXiv:1912.10205*, 2019. 1, 2, 5, 6

[55] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, page 2048–2057, 2015. 2

[56] Mingkun Yang, Yushuo Guan, Minghui Liao, et al. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, pages 9147–9156, 2019. 2

[57] Cong Yao, Xiang Bai, and Wenyu Liu. A unified framework for multioriented text detection and recognition. *IEEE Trans. Image Process.*, 23(11):4737–4749, 2014. 2

[58] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *CVPR*, pages 4042–4049, 2014. 2

[59] Deli Yu, Xuan Li, Chengquan Zhang, et al. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020. 2, 6

[60] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, et al. RobustScanner: Dynamically enhancing positional clues for robust text recognition. *arXiv preprint arXiv:2007.07542*, 2020. 1, 2, 6

[61] Matthew D Zeiler. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012. 6

[62] Fangneng Zhan and Shijian Lu. ESIR: End-to-end scene text recognition via iterative image rectification. In *CVPR*, pages 2059–2068, 2019. 2

[63] Hui Zhang, Quanming Yao, Mingkun Yang, et al. AutoSTR: Efficient backbone search for scene text recognition. *arXiv preprint arXiv:2003.06567*, 2020. 6

[64] Zheng Zhang, Chengquan Zhang, Wei Shen, Cong Yao, Wenyu Liu, and Xiang Bai. Multi-oriented text detection with fully convolutional networks. In *CVPR*, pages 4159–4167, 2016. 2