Random Blur Data Augmentation for Scene Text Recognition

DEGUO MU, WEI SUN, GUOLIANG XU, AND WEI LI
1National Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China
2Software College, Beihang University, Beijing 100083, China
Corresponding author: Deguo Mu (mudg@buaa.edu.cn)

ABSTRACT In this paper, we propose to apply data augmentation approaches that provide more diverse training images, thus helping train more robust deep models for the Scene Text Recognition (STR) task. The data augmentation methods are Random Blur Region (RBR) and Random Blur Units (RBUs). Specifically, we first introduce RBR designed for the STR task. In training, RBR randomly selects a region and sets the pixels in this region with an average value. However, when RBR provides more various training samples for STR, it may make the samples ambiguous and reduce the recognition accuracy. To address the above problem, we also propose RBUs that divides the blur region into several units. Note that the pixels of one unit share the same value. In this way, RBUs can provide additional readable training samples and help train more robust deep models. Extensive experiments on several STR datasets show that RBUs achieve highly competitive performance. Besides, RBUs are complementary to commonly used data augmentation techniques.

INDEX TERMS Deep learning, computer vision, text recognition, random blurring.

I. INTRODUCTION

Text has rich semantic details and has been utilized in many artificial intelligence applications, such as autonomous driving, travel translator, and image retrieval. This paper focuses on Scene Text Recognition (STR) [1], [2], which has become a critical task in many applications. In a typical STR scenario, given an image patch that contains texts from natural scenes, STR methods attempt to predict the sequence of characters. The maturity of the Optical Character Recognition (OCR) system [3], [4] has been successfully applied to clean documents. However, due to various text appearances in the real world and the imperfect conditions where these scenes are captured, most traditional OCR methods fail in STR tasks.

One reason causing the difficulties in STR tasks is the limited data, which leads to the failure of training robust deep models. To feed more data into the deep networks, one possible way is to manually annotate additional text images. With these labelled images, the STR models are expected to achieve a higher prediction accuracy. However, extra labelled images require huge consumption of human and financial resources. Thus, we utilize synthetic data to train deep neural networks in this paper. Synthetic data is generated by synthetic engines and has several benefits, e.g., low cost and variety. On the one hand, the generation process of synthetic data is simple and requires no manual annotations. On the other hand, synthetic engines can provide text images with various styles and shapes for the STR task.

However, synthetic data and natural images have a different distribution, making the models trained with synthetic data achieve unsatisfied prediction accuracy in the STR task. Data augmentation has the capability to change the data distribution and increase the diversity of images, which is beneficial to STR. Furthermore, data augmentation can improve the generalization ability of deep models in many computer vision applications. Thus, many efforts have focused on data augmentation to alleviate the problem caused by different distributions. For example, Inoue [5] proposes an effective augmentation strategy that mixes images together by averaging their pixel values. Then, Summers and Dinneen [6] utilize non-linear methods to combine images into new training instances. Unlike the above approaches [5], [6] that augment data applied at the image level, DeVries and Taylor [7] discuss augmentation in the feature space by adding noise, interpolating, and extrapolating as common forms of feature space augmentation. In terms of the STR task, data augmentation is...
also essential to train more robust deep models. In this paper, we adopt data augmentation directly at image-level to provide diverse samples.

We first propose the Random Blur Region (RBR) to blur one image region randomly. RBR is a data augmentation approach that seeks to prevent overfitting by altering the input space. More specifically, RBR randomly selects a region in the image that contains scene text. Then, RBR calculates the average value of the pixels in the region to obtain the blur value. Finally, the pixels in this region are set to the blur value. However, the deep model that adopts RBR cannot achieve better recognition accuracy. The main reason is that RBR makes the processed images more ambiguous. As shown in Figure 1, the original example in the first column can be easily recognized as “DB”. However, when the image is blurred by RBR, the first character “D” is difficult to be identified and can be regarded as “R”, “D” or “B”. In this case, the training makes it difficult for the network to converge, thereby affecting the recognition accuracy of the deep model. Thus, a data augmentation method suitable for the STR task should keep the processed images readable.

Based on RBR, we propose Random Blur Units (RBUs) to solve the ambiguity problem. The difference between RBR and RBUs is that RBUs utilize a more elaborate blur strategy by dividing the blur region into several small units. RBUs can produce diverse training samples while keeping the blurred areas readable. On the one hand, similar to RBR, RBUs can be also regarded as a data augmentation method that produces more training samples. On the other hand, although some image parts are blurred by RBUs, the characters in the image still remain readable. As illustrated in the first example of Figure 1, when the character “D” is processed by RBUs, we can easily recognize this character.

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

- We propose Random Blur Region (RBR), which is a simple and effective data augmentation approach for scene text recognition. Although RBR can improve the robustness of deep STR models, it brings ambiguity for the input image simultaneously.
- Based on RBR, we propose a more reasonable data augmentation method, named Random Blur Units (RBUs), that can solve the ambiguity problem caused by RBR.
- Experiments illustrate that RBR and RBUs can both improve the text recognition accuracy for the STR task.

As a more reasonable approach, RBUs can help output more robust deep STR models.

The rest of this paper is organized as follows: related work is viewed in Section II. The workflow containing four stages is described in Section III-A to give a whole picture of the STR task. Then, the proposed RBR and RBUS are detailed in Section III-B and Section III-C, respectively.

II. RELATED WORK

Many research interests have been attracted to scene text recognition (STR) [8]–[11], which is an essential process in computer vision tasks. There exist multiple characters in scene text images, making STR a more difficult task than single character recognition. Recently, STR benefits from Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). CNN is a feed-forward neural network and extracts deep semantic features by means of stacking convolutional layers. Different from CNN, RNN is a type of artificial neural network designed to recognize sequence data, i.e., text and video. Generally, recent arts that solve the STR problem mainly consist of four steps: transformation, feature extraction, sequence modeling and prediction. Thus, in this section, we describe the methods focusing on the four stages. In addition, we also describe some data augmentation methods that are critical to avoid overfitting in training deep neural networks.

A. TRANSFORMATION

Transformation is applied to ease the downstream stages by normalizing the input text images. As texts captured in natural scenes have diverse shapes, it is difficult for the subsequent stage to extract features directly from curved and tilted texts. Inspired by spatial transformation network (STN) [12], many works [13], [14] apply Thin-Plate-Spline (TPS) transformation to solve the problem of irregular text shapes, which are caused by perspective distortion, curved character placement, etc. STN proposes a spatial transformer to allow spatial data manipulation explicitly within the neural network. In this way, the network has the ability to spatially transform feature maps without any extra training supervision. Liu et al. [13] propose STAR-Net that utilizes the TPS transformation to remove the distortions of texts in natural images. Shi et al. [14] propose a robust text recognizer with automatic rectification to process irregular texts. In testing, this recognizer [14] first adopts the TPS transformation to rectify an input image into a more “readable” one. Then, the transformed image is fed into a sequence recognition network. In the proposed method, we adopt the TPS transformation to preprocess the input image that contains scene text.

B. FEATURE EXTRACTION

Feature extraction is the process that maps an input image to a representation. In STR, the feature representation aims to suppress irrelevant features, i.e., font, color, size and
background. More formally, in the feature extraction stage, an input image \( X \) is fed into a deep model to generate a visual feature map \( V = \{v_i\} \), where \( i = 1, \ldots, l \) and \( l \) is equivalent to the channels of the feature map. Then, the features are utilized to estimate the characters of scene texts. To extract features for STR, some works have studied several deep architectures, e.g., VGG [15], RCNN [16] and ResNet [17]. VGG [15] is an early deep model and consists of multiple convolutional layers and several fully connected layers. Lee and Osindero [16] present recursive recurrent neural networks (RCNN) with attention mechanism for OCR in natural scene images. In RCNN [16], the method adopts RCNN to extract features efficiently and effectively. ResNet [17] introduces residual connections into CNN network to ease the training of relatively deeper CNNs. In this paper, we apply the ResNet-50 [17] architecture to our framework as the feature extractor.

C. SEQUENCE MODELING

Sequence modeling, as a bridge between visual features and predictions, reshapes the extracted features of the last stage into a sequence of features. In this stage, the sequence contains contextual information and is utilized to predict characters in the next stage. In [18], Graves et al. introduce multiple bidirectional long short term memory to model sequence and capture long-range dependencies. Similar to [18], many works [14], [19], [20] also introduce BiLSTM into the STR framework. Litman et al. [21] propose to apply a deeper BiLSTM model to improve the encoding of contextual and deploy intermediate supervisions along the network layers. Different from [21], some works [22]-[24] do not use BiLSTM structure, and consider that BiLSTM is computationally intensive and time consuming. Yin et al. [22] propose to simultaneously detect and recognize characters by sliding the text line image with character models, which are learned end-to-end on text line images labeled with text transcripts. In [22], the outputs of the character classifier on the sliding windows are normalized and decoded with the algorithm based on connectionist temporal classification. Instead of adopting the chain structure of BiLSTM, Gao et al. propose an end-to-end fully convolutional network with the stacked convolutional layers to effectively capture the long-term dependencies among elements of scene text image. Fang et al. [23] propose an attention-based architecture which is completely based on CNNs.

D. PREDICTION

The objective of the prediction stage is to estimate the output character sequence from the identified features of an image. Connectionist temporal classification (CTC) [25] and the attention mechanism [26] are generally utilized in predicting characters. Graves et al. propose CTC to label unsegmented sequences directly by training RNNs. Specifically, CTC is used in STR as a prediction module, i.e., the transcription layer that converts the input features made by CNNs or RNNs into a target string sequence by calculating the conditional probability. Bahdanau et al. [26] propose an attention mechanism in the field of neural machine translation, which automatically searches for the predicted word that is relevant to parts of a given source sentence. In recognizing scene text, the attention mechanism [26] is generally applied together with RNN nets to perform text prediction.

E. DATA AUGMENTATION

Deep neural networks perform remarkably well on many computer vision tasks, i.e., classification, detection and object re-identification. However, existing deep networks are heavily reliant on large-scale data to avoid overfitting and achieve robust deep representations. To provide more training data to the deep models, more and more research interests have focused on data augmentation. Existing data augmentation approaches include geometric transformations, color space augmentations, random erasing and adversarial training. Some geometric augmentation operations, i.e., flipping, scaling and cropping, have been applied in many computer vision applications like classification [27], detection [28] and re-identification [29], [30]. Performing augmentations in the color channels space is another strategy that is very practical to implement. One way to perform color augmentation is to isolate a single color channel such as R, G or B. Cubuk et al. [31] utilize reinforcement learning algorithm to search the policy for data augmentation.

Deep STR models also suffer from overfitting caused by insufficient text data. However, only a few data augmentation works have been studied in STR. The widely utilized synthetic datasets [32] provide more than 10 million samples that can be involved to augment the training set. As for handwritten text, existing training data can hardly cover various writing styles. Thus, data augmentation is also critical for the handwritten text recognition task [33]. However, generating additional handwritten data is also challenging. In the STR task, more reliable training data can bring benefits to the training process of deep models. In this paper, we utilize random blur units to increase the diversity of characters in the natural scene and thus train more robust deep STR models.

III. METHOD

In this section, the proposed method is introduced from three aspects. First, we describe the workflow of the prevalent Scene Text Recognition (STR) framework, which is simple but effective. There are four relatively independent stages in this framework, and each can utilize various advanced modules. Then, we describe the Random Blur Region (RBR), which is a data augmentation method. Specifically, RBR randomly selects a rectangular region in the input image, and the contents in this region are blurred. Finally, based on RBR, we propose the Random Blur Units (RBUs) that divide RBR into multiple small units.

A. SYSTEM WORKFLOW

As illustrated in Figure 2, the workflow consists of four stages: transformation, feature extraction, sequence modeling, and prediction.
In the transformation stage, we feed an original image into the transformation module to achieve a more regular and readable output [14]. Note that extra labels and pretraining models are not required in this stage. Specifically, there exist even number of key points around the text. As shown in Figure 2, the green “+” symbols denote the key points. Based on these key points, the parameters of Thin-Plate-Spline (TPS) transformation are estimated by a grid generator. As the grid generator can back-propagate gradients in the training process, it can be incorporated into the main structure in an end-to-end manner. Finally, a rectified image containing readable output [14]. In this paper, we choose ResNet-50 [17] as the backbone network.

In the sequence modeling stage, we utilize a Recurrent Neural Network (RNN) to extract more robust sequential features. Introducing the sequence modeling component can enhance the contextual interactions that exist in the sequential features. Specifically, we introduce Bidirectional Long Short Term Memory (BiLSTM) [36] to capture bidirectional dependencies. Note that the sequence length is not changed in this stage.

In the prediction stage, the feature sequence is input to a deep prediction model to recognize text labels. As shown in Figure 2, there are two alternative prediction methods: Connectionist Temporal Classification (CTC) [37] and Attention-based prediction (Attn) [38]. CTC mainly contains three steps. At first, CTC builds a lexicon that contains a “blank” label and all characters appearing in the text recognition task. Second, CTC predicts all the probable character symbols. Finally, adjacent prediction characters are merged by CTC. Due to the introduction of the “blank” symbol, CTC can handle the neighbored identical characters. Attn can automatically predict the sequence conditioned on the input sequential representations. In the Attn component, attention on contextual dependencies is realigned to give more accurate sequence prediction.

Note that only the feature extraction and prediction stage are necessary, while the other stages are designed for higher recognition accuracy at the expense of efficiency. In this paper, we first deploy the proposed augmentation method into a two-stage framework in order to determine the optimal values of hyper-parameters. Then, we conduct experiments by using the whole framework that consists of all four stages. Detailed comparison results are presented in Section IV-D.

**FIGURE 2.** The workflow of the STR system contains four stages: transformation, feature extraction, sequence modeling, and prediction. Specifically, the input image is first rectified during the transformation stage. Key points are denoted by green “+” symbols. Then, sequential features are generated after feature extraction and sequence modeling stage. The input image is converted into a sequence of feature vectors. Finally, in the prediction stage, the model predicts the texts by means of CTC or Attn. Compared with CTC, Attn considers contextual dependencies and achieves higher prediction accuracy.

In the transformation stage, we feed an original image into the transformation module to achieve a more regular and readable output [14]. Note that extra labels and pretraining models are not required in this stage. Specifically, there exist even number of key points around the text. As shown in Figure 2, the green “+” symbols denote the key points. Based on these key points, the parameters of Thin-Plate-Spline (TPS) transformation are estimated by a grid generator. As the grid generator can back-propagate gradients in the training process, it can be incorporated into the main structure in an end-to-end manner. Finally, a rectified image containing regular texts is obtained by interpolating new pixels utilizing the grid generator.

In the feature extraction stage, the transformed image is embedded into a sequence of feature maps. In image recognition and classification applications, the feature extraction models generally contain multiple convolutional layers, pooling layers, and a fully connected layer. The difference between these applications and STR is that the fully connected layer is not involved in the feature extraction stage for STR. Note that the output of this stage is a 3-Dimensions feature map. Then, the feature map is split into a sequence of feature vectors, and the sequence length is equal to the feature map columns. In this way, the image is transformed into a sequential form. The feature extraction model has a backbone, which has many alternative structures including VGG [34], ResNet [17], SENet [35], etc. In this paper, we choose ResNet-50 [17] as the backbone network.

In the sequence modeling stage, we utilize a Recurrent Neural Network (RNN) to extract more robust sequential features. Introducing the sequence modeling component can enhance the contextual interactions that exist in the sequential features. Specifically, we introduce Bidirectional Long Short Term Memory (BiLSTM) [36] to capture bidirectional dependencies. Note that the sequence length is not changed in this stage.

In the prediction stage, the feature sequence is input to a deep prediction model to recognize text labels. As shown in Figure 2, there are two alternative prediction methods: Connectionist Temporal Classification (CTC) [37] and Attention-based prediction (Attn) [38]. CTC mainly contains three steps. At first, CTC builds a lexicon that contains a “blank” label and all characters appearing in the text recognition task. Second, CTC predicts all the probable character symbols. Finally, adjacent prediction characters are merged by CTC. Due to the introduction of the “blank” symbol, CTC can handle the neighbored identical characters. Attn can automatically predict the sequence conditioned on the input sequential representations. In the Attn component, attention on contextual dependencies is realigned to give more accurate sequence prediction.

Note that only the feature extraction and prediction stage are necessary, while the other stages are designed for higher recognition accuracy at the expense of efficiency. In this paper, we first deploy the proposed augmentation method into a two-stage framework in order to determine the optimal values of hyper-parameters. Then, we conduct experiments by using the whole framework that consists of all four stages. Detailed comparison results are presented in Section IV-D.

**B. RANDOM BLUR REGION**

In this section, we first introduce some necessary notations and parameters in the STR task. Each input text sample is initialized as a gray image which has only one channel. The size of the input image $I$ is $W \times H$, and $I(i,j)$ denotes the pixel located at $(i,j)$, where $1 \leq i \leq W$ and $1 \leq j \leq H$. Note that the pixel value $I(i,j)$ ranges from 0 to 255.

In this paper, we propose the Random Blur Region (RBR), which is a rectangular area randomly selected in the image. $w$ and $h$ denote the width and height of this blur region, respectively. Thus, the blur region can be written as $((x_1, y_1), (x_2, y_2))$. $(x_1, y_1)$ and $(x_2, y_2)$ are the coordinates of
the endpoints, which are distributed on the same diagonal line of this rectangle area, respectively. The coordinates of two endpoints meet that \(1 \leq x_1 < x_2 \leq W\) and \(1 \leq y_1 < y_2 \leq H\).

The calculated blur value of RBR can be written as

\[
pix_{RBR} = \left[ \frac{1}{w \cdot h} \sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} I(i,j) \right]. \tag{1}
\]

Then, the pixel value of the augmented image \(I'\) can be obtained by

\[
I'(i,j) = \begin{cases} 
    pix_{RBR} & x_1 \leq i \leq x_2, y_1 \leq j \leq y_2 \\
    I(i,j) & \text{otherwise.}
\end{cases} \tag{2}
\]

In this way, all pixels in this random region are set to the same value \(pix_{RBR}\), achieving the goal of region blurring. Some example pairs processed by RBR are illustrated in Figure 3.

In RBR, we introduce 3 additional hyperparameters, i.e., blur probability \(p\), blur ratio \(r_a\), and height-width ratio \(r_{hw}\). First, we set a blur probability \(p\) that determines whether an input image is processed by RBR. Note that the blur process is not dispensable. Second, the parameter \(r_a\) represents the ratio of the RBR area to the overall input image. Third, the parameter \(r_{hw}\) denotes the height-width ratio of the selected rectangle.

As shown in Figure 3, RBR makes the processed images ambiguous. For example, due to random blurring, the text image “cut” on the bottom left can be recognized as “cut” or “cat”, making this image have two possible labels. This kind of training examples with uncertain labels may lead to network non-convexity, which influences the recognition accuracy in testing severely.

Although blurring some parts of the image can achieve data augmentation, RBR cannot effectively improve the recognition accuracy for the STR task. The main reason is that details in the STR task are crucial to recognize characters. In other words, sometimes one specific character only exists in one small region of the image. However, when RBR blurs the critical or full region of this character, it is almost impossible to identify this character. Example pairs that RBR may bring ambiguity include ‘0-8’, ‘O-Q’, ‘E-F’, etc.

C. RANDOM BLUR UNITS

In this part, we propose the Random Blur Units (RBUs) to solve the ambiguity problem. RBUs derive from RBR, and the process of generating RBUs is shown in Figure 4. More specially, we first divide the RBR into several equal areas, and the pixels of one unit share the same value. As shown in Figure 5, the RBUs usually have different pixel values.

We define the width of the blur unit as \(s\). The width and height of the blur area are denoted as \(w\) and \(h\), respectively. Note that in the process of random blurring, the values of \(w\) and \(h\) are integral multiples of \(s\). Therefore, there are \(w/s\) units in each horizontal line and \(h/s\) units in each vertical column. We use \(num_w\) and \(num_h\) to represent the number of units counted in the horizontal and vertical directions, respectively. In other words, there are \(num_w \cdot num_h\) units in the blurred area.

One random blur unit can be represented by \((a, b)\), satisfying that \(0 \leq a < num_w\) and \(0 \leq b < num_h\). The pixel value in the unit \((a, b)\) is:

\[
pix(a, b) = \left[ \frac{1}{s^2} \sum_{i=x_1+a \cdot s}^{x_2+(a+1) \cdot s-1} \sum_{j=y_1+b \cdot s}^{y_2+(b+1) \cdot s-1} I(i,j) \right]. \tag{3}
\]

Blurred image is denoted as \(I'\), and blurred pixel is

\[
I'(i,j) = \begin{cases} 
    pix(i-x_1, j-y_1) & x_1 \leq i \leq x_2, y_1 \leq j \leq y_2 \\
    I(i,j) & \text{otherwise},
\end{cases}
\tag{4}
\]

where \(s\) controls the size of blur unit and affects the clarity of the characters. When \(s\) is set to 1, the image remains original resolution, and the blurring operation is not deployed. With the increase of \(s\), the clarity of blur region decreases gradually. In RBR and RBUs, we introduce four hyper parameters totally. Detailed parameters analysis is presented in Section IV-C. Examples processed by RBUs are shown in Figure 5.

IV. EXPERIMENTS

To verify the effectiveness of the proposed methods, we conduct detailed experiments on public datasets. In this section, we first introduce the datasets used in training and testing. Then, implementation details of our methods are pre-
sent. Besides, we conduct an ablation study to demonstrate the effectiveness of our approach compared to the baseline. Finally, we show the performance of our methods on benchmarks.

A. DATASETS

1) TRAINING DATASETS

In this paper, we use two large-scale synthetic datasets: MJSynth (MJ) [39] and SynthText (ST) [32] to train deep models. There exist several reasons for introducing synthetic datasets during the training process. First of all, synthetic datasets contain quite massive text images. Meanwhile, generating synthetic data does not require manual annotations that consume a lot of labor costs. Furthermore, synthetic data can also provide various training data for different OCR applications. Training with synthetic data is one of the available methods to enhance the generalization ability of the STR model.

The process of generating data in MJ consists of 6 stages: font rendering, border/shadow rendering, base coloring, projective distortion, natural data blending, and noise. The synthetic engine randomly selects a font and renders an inset border, an outset border, and shadow to generate colored text. When stimulating the 3D environment, a random and full-projective distortion transformation is performed on the colored text. Then, synthetic texts are blended with natural data. Finally, the synthetic engine adds some noises, \textit{i.e.}, Gaussian noise, JPEG compression, \textit{etc}. The height of synthetic texts is fixed at 32 pixels, and the width is variable.

Unlike MJ, the ST dataset adopts a different synthetic data generation strategy: text characters are added over a specific region from a real scene image. Firstly, nearly 8,000 background images are sampled from Google Image Search, and the images that contain texts are eliminated manually. Then, with the help of segmentation and geometry estimation, a region that has unified color and texture is selected to distribute text characters. In contrast to arranging text randomly on the real scene images, this operation prevents the text from crossing obvious image discontinuities.

MJ has 8.9 million images with word boxes, and ST contains 5.5 million images. As shown in Figure 6, examples from MJ and ST are realistic enough to satisfy the training requirements in the STR task.

2) EVALUATION DATASETS

For evaluation, we use real scene text images to test the practical performance of the STR model. In total, seven datasets are involved, which are all cropped from actual pictures.

The IIT 5K-word (IIT) [40] dataset is collected from Google Image Search. Query words contain signboard, house name plates, movie posters, \textit{etc}. Furthermore, all words in images are manually annotated with bounding boxes and corresponding ground truth images. As a result, a training set containing 2,000 word images and a testing set containing 3,000 word images are obtained. In our experiments, only the testing set is used.

Street View Text (SVT) [41] dataset is collected from Google Street View, and consists of outdoor images annotated with a list of texts. In SVT, texts come from business signage, which is easily obtained by geographic business searches. After labeled with bounding boxes and annotated by texts, a subset that contains 647 word images is obtained for evaluation.

ICDAR 2003 [42] is a robust reading competition, which breaks down the original problem into three sub-problems. We only consider the dataset for word recognition, named IC03. There are two versions: one contains 860 word images, and another consists of 867 word images. Both two versions are used in our experiments. Similarly, ICDAR 2013 (IC13) [43] and ICDAR 2015 (IC15) [44] are also used in evaluation. According to the number of images in the testing set, each dataset has two versions. The two testing sets of IC13 have 857 and 1,015 word images, respectively. As for IC15, two sets are composed of 1,811 and 2,077 word images, respectively.

The StreetViewText-Perspective (SP) [45] dataset consists of texts in street images with a great variety of viewpoints. SP is built on the original SVT dataset and is specifically designed for perspective text recognition. The number of text images for testing is 645, and the heights of images vary largely due to the variety of viewpoints and orientations.

The CUTE80 (CT) [46] dataset is proposed for curved texts recognition. Word images in CT are collected from natural scenes. In addition, CT has 288 images for evaluation.

Text images for evaluation are randomly sampled as shown in Figure 7. A total of 12,067 images are used in the testing phase. Under the setting of our method for STR, only uppercase or lowercase English letters and Arabic numerals, \textit{i.e.}, 0 to 9, are considered. For a query text image, if the corresponding prediction is correct, the length of the result is correct, and all characters are right completely. Otherwise, the prediction result is wrong. We use prediction accuracy to quantitatively evaluate the performance of STR models.
In this paper, completely different datasets are used for training and testing. On the one hand, the datasets for training are both synthetic datasets in our work. It is difficult to label enough scene text images. Therefore, most existing methods choose to use synthetic text images instead of using real data in training. We also adopt this scheme. On the other hand, 7 real-world datasets are used during testing. For a fair comparison, the total accuracy is calculated on all testing datasets. In a word, synthetic data is used for training because of its easy availability as well as large-scale quantity, while real-world data is used for testing.

B. IMPLEMENTATION DETAILS

We follow the framework described in [47]. However, we omit the Transformation and Sequence Modeling stages for simplicity in the experiments about parameters analysis. At feature extraction stage, we utilize ResNet-50 [17], and CTC is chose at the final prediction stage. The learning rate is 0.1, and the optimizer is Adadelta. Involved characters include 10 Arabic numerals and 26 English letters. Furthermore, for a fair comparison, we also use the parameter values proposed in [47] during training deep neural networks. In a batch, the image ratio between MJ and ST is 0.5 : 0.5, following the prevalent setting [47]. The height and width of input images are 32 and 100 pixels, respectively. Note that random blurring is only carried out during training. In the testing phase, the input images keep the original resolution. The code is based on Pytorch [48], which is a library for Python and an open source machine learning framework. To reduce the accidental influence of training, we repeat each set of experiments more than three times. Displayed results are averaged from the experiments under the same setting.

To find the optimal batch size setting, we test different batch sizes: 64, 128, 192, and 256. Experiments are reported in Table 1. Note that the number of iterations is fixed at 300,000 in our experiments. Therefore, a larger batch size leads to a higher computational cost compared with the 192 setting. Therefore, we set the batch size to 192 in our method.

| batch size | 64  | 128 | 192 | 256 |
|------------|-----|-----|-----|-----|
| total accuracy | 78.3 | 82.2 | 84.0 | 84.2 |

TABLE 1. Analysis of batch size. With the increase of batch size, the higher total accuracy is achieved. Note that the setting when batch size is 256 has a small accuracy improvement but a higher computational cost than the 192 setting. For this reason, the batch size is set to 192 in our experiments.

C. PARAMETERS ANALYSIS

In this section, we analyze the importance of parameters introduced in our method and find out the optimal setting. For simplicity, we omit the transformation and sequence modeling stages. At the feature extraction stage, we use ResNet-50, and CTC is adopted at the prediction stage. At first, we compare the effects of different blur approaches, i.e., RBR and RBUs. Next, we conduct detailed experiments to analyze the introduced parameters: blur probability $p$, the area ratio of blur region, the height-width ratio of the blur rectangle, and the size of blur unit $s$. Particularly, for each experimental case, $p$ and $s$ are real numbers that are set in advance. The area ratio and the height-width ratio of the blur region are randomly sampled from a default interval.

1) RBR AND RBUs

In this paper, we propose two blur patterns: RBR and RBUs. RBR is a blur rectangle region, in which all pixels share the same value. RBUs denote improved RBR, which is composed of multiple blur units. Although RBR can achieve data augmentation, important local information is lost in the blurring process. RBUs could alleviate this problem by establishing different units.

Comparison results between RBR and RBUs are shown in Table 2. Specifically, the total accuracy of RBR is higher than baseline by 1.4%. However, it is lower than RBUs by 0.4% under the same setting, which demonstrates the effectiveness of the proposed RBUs. Furthermore, the results also prove that RBUs alleviates the ambiguity problem brought by RBR and thus achieves higher prediction accuracy.

2) BLUR PROBABILITY

We set the blur probability $p$ to control the number of blurred images in the training phase. To find the optimal value of $p$, we conduct additional experiments on three datasets: IC03, IC13 and IC15. Four values are tested: 0.5, 0.6, 0.7, and 0.8. We take no account of extraordinarily higher or lower probabilities, intuitively. Experimental results are presented in Table 3. The highest accuracy evaluated on the test datasets is achieved when $p = 0.5$ or $p = 0.6$.

3) THE AREA OF BLUR REGION

In this work, we adopt an area ratio parameter to determine the size of the blur region. The area ratio provides adaptability for different input images and ensures that larger input images are efficiently augmented.
TABLE 2. Comparison results. Our methods are denoted as “+RBR” and “+RBUs”, respectively. It is clearly shown that our method achieves the highest recognition accuracy on 9 out of 10 benchmarks. Compared with RBR, RBUs can help train more robust deep STR models and obtain higher accuracy. Note that the methods are divided into two categories according to the choice in the prediction stage: CTC and Attn. Experimental results demonstrate that the proposed methods achieve higher accuracy when Attn is used. The main reason is that Attn utilizes sufficient contextual information and is conditioned on the sequential input representations, which is more suitable for blur augmentation.

| Prediction | Method   | Hit  | SVT  | IC03 | IC13 | IC15 | SP | Total accuracy | CT |
|------------|----------|------|------|------|------|------|----|----------------|----|
| CTC        | CRNN [36] | 82.9 | 81.6 | 93.1 | 92.6 | 91.1 | 89.2 | 69.4           | 46.2 | 70.0 | 65.5 | 78.4 |
|            | CRNN + RBR | 81.4 | 79.9 | 92.7 | 92.8 | 90.1 | 88.8 | 67.4           | 46.7 | 68.4 | 64.1 | 78.5 |
|            | CRNN + RBUs | 81.9 | 80.1 | 93.3 | 93.4 | 90.9 | 89.5 | 67.1           | 46.7 | 65.9 | 62.7 | 78.4 |
|             | Rosetta [49] | 84.3 | 84.7 | 93.4 | 92.9 | 90.9 | 89.0 | 71.2           | 66.0 | 73.8 | 69.2 | 80.0 |
|             | Rosetta + RBR | 83.4 | 84.4 | 93.5 | 92.6 | 90.9 | 89.7 | 68.3           | 65.8 | 70.5 | 66.9 | 79.7 |
|             | Rosetta + RBUs | 83.7 | 82.8 | 93.6 | 93.3 | 91.9 | 90.5 | 70.0           | 67.4 | 71.8 | 70.7 | 80.6 |
|             | ORCNN [50] | 84.2 | 83.7 | 93.5 | 93.0 | 90.9 | 88.8 | 71.4           | 63.8 | 73.6 | 68.1 | 79.8 |
|             | GRCNN + RBR | 82.2 | 81.6 | 93.6 | 92.6 | 89.5 | 88.4 | 67.5           | 64.9 | 71.0 | 65.9 | 78.8 |
|             | GRCNN + RBUs | 81.3 | 80.4 | 92.9 | 92.6 | 90.7 | 89.6 | 68.5           | 65.9 | 71.5 | 66.2 | 79.0 |
| Attn       | STAN-Net [13] | 87.0 | 86.9 | 94.4 | 94.0 | 92.8 | 91.5 | 76.1           | 70.3 | 77.3 | 71.7 | 82.9 |
|            | STAR + RBR | 85.4 | 86.1 | 94.4 | 94.2 | 91.5 | 90.6 | 74.0           | 71.3 | 76.4 | 73.9 | 82.8 |
|            | STAR + RBUs | 85.5 | 85.8 | 94.4 | 94.1 | 90.8 | 90.0 | 75.0           | 72.3 | 77.7 | 72.1 | 83.1 |
| RARE [44]  | 86.2 | 83.8 | 95.9 | 95.0 | 94.9 | 93.0 | 91.4 | 74.5           | 71.9 | 76.9 | 68.3 | 83.1 |
|            | RARE + RBR | 85.3 | 85.9 | 95.0 | 95.0 | 94.9 | 93.0 | 91.4 | 74.5           | 71.9 | 76.9 | 68.3 | 83.1 |
|            | RARE + RBUs | 85.1 | 86.1 | 94.8 | 94.8 | 93.7 | 92.6 | 75.4 | 72.6 | 77.4 | 71.1 | 83.6 |
| R2AM [16]  | 83.4 | 82.4 | 92.2 | 92.0 | 90.2 | 88.1 | 68.9 | 63.6 | 72.1 | 64.9 | 78.5 |
|            | R2AM + RBR | 82.2 | 83.5 | 92.8 | 92.3 | 90.2 | 87.7 | 69.1 | 66.6 | 71.5 | 63.1 | 79.3 |
|            | R2AM + RBUs | 82.9 | 82.2 | 92.9 | 92.8 | 90.7 | 88.6 | 68.7 | 66.3 | 74.7 | 63.8 | 79.6 |
| Baseline [47] | 87.9 | 87.5 | 94.9 | 94.4 | 93.6 | 92.3 | 77.6 | 71.8 | 79.2 | 74.0 | 84.0 |
| Baseline + RBR | 87.0 | 87.3 | 95.6 | 95.4 | 93.8 | 92.3 | 77.5 | 74.8 | 80.5 | 73.9 | 83.1 |
| Baseline + RBUs | 87.5 | 87.9 | 94.1 | 94.0 | 93.5 | 92.5 | 78.3 | 75.4 | 80.9 | 80.1 | 85.5 |
| SATRN [10] | 92.8 | 91.3 | 96.7 | 94.1 | 91.0 | 89.4 | 79.0 | 85.5 | 87.8 | 88.5 |
|            | SATRN + RBR | 92.1 | 91.5 | 97.1 | 94.4 | 91.1 | 94.4 | 79.2 | 87.2 | 88.2 | 88.6 |
|            | SATRN + RBUs | 92.3 | 91.6 | 97.1 | 94.7 | 91.4 | 94.7 | 79.7 | 87.4 | 88.8 | 89.1 |

TABLE 3. Parameter analysis of $p$, which controls the number of the blurred images in the training phrase. We select four values: $0.5, 0.6, 0.7$ and $0.8$. Experiments show that the highest accuracy is achieved when $p = 0.5$ or $p = 0.6$.

| $p$ | IC03 | IC13 | IC15 |
|-----|------|------|------|
| 0.5 | 850  | 867  | 857  |
| 0.6 | 93.0 | 92.7 | 91.0  |
| 0.7 | 92.9 | 92.5 | 91.6  |
| 0.8 | 92.6 | 92.4 | 90.2  |

4) THE HEIGHT-WIDTH RATIO OF BLUR REGION
The height-width ratio determines the shape of blur region. Similar to the area ratio, we also test several intervals: [0.1, 0.4], [0.4, 0.7], [0.7, 1.0], [1.0, 1.3], [1.3, 1.6], and [1.6, 1.9]. Experiments are also conducted on the IC15 dataset, and results are presented in Figure 9. When the height-width ratio is less than 1, the accuracy is better than the cases with a height-width ratio greater than 1. The reason is that texts are mainly distributed horizontally in the STR task. Therefore, a blur region whose height-width ratio is greater than 1 may completely cover some characters. On the contrary, the blur region whose height is smaller than width only covers part of texts, and this case does not cause the text to be entirely unrecognizable.

5) THE SIZE OF RBU
In this work, $s$ denotes the size of the random blur unit that affects the clarity of the blurred picture. Specifically, the blurred region becomes more unclear with the growth of $s$. When $s = 1$, the blur process is not carried out. Therefore, we start with $s = 2$ to find out the optimal value of $s$. Experimental results are reported in Table 4. A total of three datasets are tested: IC03, IC13 and IC15.

According to the Table 4, the accuracy of the cases when $s = 3$ and $s = 4$ is higher than that of $s = 2$. Furthermore, the results of $s = 1$ are from a reproduced baseline. Note that images have larger blur regions. In this part, we choose the IC15 dataset, which is divided into two small datasets: IC15_1811 and IC15_2077. For each image in the experiments, the area ratio is randomly sampled from a fixed interval. We intuitively select four intervals: [0.1, 0.2], [0.2, 0.3], [0.3, 0.4] and [0.4, 0.5].

Experimental results are shown in Figure 8. As the blur region ratio increases, the accuracy rises first and then decreases. The reason for the decrease in accuracy is that larger blur region covers too much useful information, making the STR model difficult to recognize characters correctly. Note that the highest accuracy is obtained near the interval [0.2, 0.3].
Parameter analysis of the height-width ratio. The height-width ratio controls the shape of the blurring region. In the blurring process, the height-width ratio is randomly selected from a predetermined interval. We test 6 intervals totally: [0.1, 0.4], [0.4, 0.7], [0.7, 1.0], [1.0, 1.3], [1.3, 1.6] and [1.6, 1.9]. When the height-width ratio is less than 1, the accuracy is better than the cases with a height-width ratio greater than 1.

When $s > 1$, the results of RBUs are better than the baseline, which proves the effectiveness of random blur.

**6) BACKBONE**

We also conduct experiments to find the optimal backbone. Experimental results are reported in Table 5. The involved backbones are Inception-V1, Xception, ResNet-50, DenseNet-121, and ResNeXt-50. The evaluation datasets are IC13 and IC15. As shown in Table 5, the accuracy of ResNet-50, DenseNet-121 and ResNeXt-50 is close and higher than Inception-V1 and Xception consistently. Therefore, we choose ResNet-50 as the backbone in the feature extraction stage.

**D. PERFORMANCE ON BENCHMARKS**

In this section, we give the accuracy comparison on 10 public benchmarks. Involved methods include CRNN [36], R2AM [16], GRCNN [50], Rosetta [49], RARE [14], STAR-Net [13] and the baseline [47] in this paper. For a fair comparison, we use the accuracy of these approaches under the same setting published in [47]. In addition, we add Transformation and Sequence Modeling components described in Section III-A to our approach. Comparison results are presented in Table 2. Our methods are denoted as RBR and RBUs, respectively. The experiments clearly show that our method achieves the highest recognition accuracy on 9 out of 10 benchmarks. Furthermore, our method also matches the baseline on the rest benchmark. Note that blurring by RBUs has the highest total accuracy, outperforming the baseline and RBR by a 1.5% margin and a 0.4% margin, respectively. SATRN [10], which is one of the state-of-the-art models, achieves higher accuracy with the proposed augmentation strategy. The total accuracy is improved by 0.6%. To sum up, the results in Table 2 strongly demonstrates the effectiveness of the proposed method.

Moreover, as illustrated in Table 2, we also apply RBR and RBUs on many existing STR methods. For example, STAR-Net + RBR indicates that we apply the RBR method when training the STAR-Net model, and STAR-Net + RBUs indicates that RBUs are involved in the training process of STAR-Net. According to the module in the prediction stage, the STR methods can be divided into two categories, i.e., Connectionist Temporal Classification (CTC) and Attention-based prediction (Attn). On the one hand, CRNN, Rosetta, GRCNN, and STAR-Net utilize the CTC module. On the other hand, RARE, R2AM, and the baseline adopt the Attn module.

**RBR and RBUs can effectively improve the recognition accuracy of the Attn methods.** For example, compared with RARE, the total accuracy of RARE + RBR increases by +1.1, and the total accuracy of RARE + RBUs has an improvement of +1.6. The above results clearly show that RBR and RBUs both can improve test recognition accuracy. However, when we apply RBR and RBUs to the CTC methods, there is no significant improvement in terms of total accuracy. The main reason is that Attn and CTC utilize different prediction mechanisms. On the one hand, the Attn methods utilize sufficient contextual information and are conditioned on the input sequential representations. On the other hand, the CTC approaches only consider local features to predict probable character symbols, meaning that little contextual information is involved in the prediction process. In summary, the proposed approaches, including RBR and RBUs, can be beneficial for deep models that utilize contextual information in the prediction stage.

**Compared with RBR, RBUs can help train more robust deep STR models.** Compared with Baseline, Baseline + RBR and Baseline + RBUs improve total accuracy by +1.1 and 1.5, respectively. This result clearly shows that RBUs can achieve higher recognition accuracy. The main reason is that RBUs can alleviate the ambiguity problem caused by RBR and produce more reasonable training samples.

**V. CONCLUSION**

As a widely studied technique, data augmentation has been deployed in many applications to gain sample diversity and improve the performance of a specific task. In this paper, we focus on Scene Text Recognition (STR) and propose two data augmentation approaches, named Random Blur Region (RBR) and Random Blur Units (RBUs).
To demonstrate the effectiveness of data augmentation in the STR task, we conduct experiments based on a baseline that has four stages: transformation, feature extraction, sequence modeling, and prediction. With data enhancement technology, our goal is to obtain more training data to solve the problem of insufficient data faced by the deep learning model. Finally, the accuracy of model recognition is improved.

RBR selects a rectangular region randomly, and the pixels in the region are set to an average value when generating an augmented image. In RBR, we introduce three additional hyperparameters: blur probability, blur ratio, and height-width ratio to make the data augmentation. In this way, RBR can provide more various training images. However, letter recognition in some images of augmentation may cause ambiguity errors because of random blurring. For example, if the letters in the middle of the words cat and cut are fuzzy, it is difficult to tell which is the correct word. Experimental results show that the text recognition accuracy based on the RBR cannot improve significantly.

Based on RBR, we propose RBUs to provide samples with less ambiguity. Specifically, we divide the blur region into several small units, and the pixels in each unit are replaced by the corresponding averaged value. Different from RBR, RBUs produce diverse training samples while keeping the blurred area readable. Experimental results on multiple public datasets demonstrate the effectiveness of RBUs. It is shown that our method achieves the highest recognition accuracy on 9 out of 10 benchmarks. Meanwhile, the CTC and Attention model are chosen in the prediction stage. The proposed methods that utilize the Attention model can achieve higher accuracy. The main reason is that the Attention model utilizes sufficient contextual information and is conditioned on the sequential input representations, which is more suitable for the blur data augmentation. We also include additional experiments to discuss the blur probability, the area of blur region, the height-width ratio of the blur region, and the size of RBUs. We have completely reported the experimental comparison results of different parameter settings. We also conduct the experiment to find the optimal backbone.

In future work, we will further study data augmentation approaches for the STR task. Especially in the scene with insufficient data, we will further experiment on whether we can create more data through data augmentation to support deep learning model training. In addition, we will apply the proposed data augmentation approaches to other CNN tasks, such as classification and detection. Maybe our methods can also achieve the effect of improvement.

REFERENCES

[1] Z. Huang, J. Lin, H. Yang, H. Wang, T. Bai, Q. Liu, and Y. Pang, “An algorithm based on text position correction and encoder-decoder network for text recognition in the scene image of visual sensors,” Sensors, vol. 20, no. 10, p. 2942, May 2020.

[2] S. Ablahat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, “Improved handwritten digit recognition using convolutional neural networks (CNN),” Sensors, vol. 20, no. 12, p. 3344, Jun. 2020, doi: 10.3390/s20123344.

[3] H. Michalak and K. Okarma, “Improvement of image binarization methods using image preprocessing with local entropy filtering for alphanumeric character recognition purposes,” Entropy, vol. 21, no. 6, p. 562, Jun. 2019.

[4] R. Ahmed, M. Gogate, A. Tahir, K. Dashtipour, B. Al-tamimi, A. Havalah, M. El-Affendi, and A. Hussain, “Novel deep convolutional neural network-based contextual recognition of Arabic handwritten scripts,” Entropy, vol. 23, no. 3, p. 340, Mar. 2021.

[5] H. Inoue, “Data augmentation by pairing samples for images classification,” 2018, arXiv:1801.02929. [Online]. Available: http://arxiv.org/abs/1801.02929

[6] C. Summers and M. J. Dinneen, “Improved mixed-example data augmentation,” in Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV), Jan. 2019, pp. 1262–1270.

[7] T. DeVries and G. W. Taylor, “Dataset augmentation in feature space,” 2017, arXiv:1702.05538. [Online]. Available: http://arxiv.org/abs/1702.05538

[8] U. Yaseen, J. H. Shah, M. A. Khan, G. J. Ansari, S. U. Rehman, M. Sharif, S. Kadry, and Y. Nam, "Text detection and classification from low quality natural images," Intell. Autom. Soft Comput., vol. 26, no. 4, pp. 1251–1266, 2020.

[9] Y. Mou, L. Tan, H. Yang, J. Chen, L. Liu, R. Yan, and Y. Huang, “PlugNet: Degradation aware scene text recognition supervised by a pluggable super-resolution unit,” in Proc. 16th Eur. Conf. Comput. Vis. (ECCV), in Lecture Notes in Computer Science, vol. 12360. Glasgow, U.K.: Springer, Aug. 2020, pp. 158–174.

[10] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding, “Towards accurate scene text recognition with semantic reasoning networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 12113–12122.

[11] L. Han, P. Yang, and F. Zhang, “Review of scene text detection and recognition,” Arch. Comput. Methods Eng., vol. 27, no. 2, pp. 433–454, Apr. 2020.

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, “Spatial transformer networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 2017–2025.

[13] W. Liu, C. Chen, K.-Y. Wong, Z. Su, and J. Han, “STAR-Net: A Spatial attention residue network for scene text recognition,” in Proc. Brit. Mach. Vis. Conf., 2016, p. 7.

[14] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, “Robust scene text recognition with automatic rectification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4168–4176.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Represent., Y. Bengio and Y. LeCun, Eds., 2015.

[16] W. Lee and S. Ovadia, “Recursive recurrent nets with attention modeling for OCR in the wild,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 2231–2239.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[18] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 5, pp. 855–868, May 2009.

[19] 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. 2016.

[20] W. Liu, C. Chen, and K. K. Wong, “Char-Net: A character-aware neural network for distorted scene text recognition,” in Proc. AAAI Conf. Artif. Intell. Palo Alto, CA, USA: AAAI Press, 2018, pp. 7154–7161.

[21] R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor, and R. Manmatha, “SCATTER: Selective context attentional scene text recognizer,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 11962–11972.

[22] F. Yin, Y.-C. Wu, X.-Y. Zhang, and C.-L. Liu, “Scene text recognition with sliding convolutional character models,” 2017, arXiv:1709.01727. [Online]. Available: http://arxiv.org/abs/1709.01727

[23] 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.
[24] Y. Gao, Y. Chen, J. Wang, M. Tang, and H. Lu, "Reading scene text with fully convolutional scene modeling," *Neurocomputing*, vol. 339, pp. 161–170, Apr. 2019.

[25] 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. (ICML)*, 2006, pp. 369–376.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, arXiv:1409.0473. [Online]. Available: http://arxiv.org/abs/1409.0473

[27] H. Li, X. Dou, C. Tao, Z. Wu, J. Chen, J. Peng, M. Deng, and L. Zhao, "RSI-CB: A large-scale remote sensing image classification benchmark using crowdsourced data," *Sensors*, vol. 20, no. 6, p. 1594, Mar. 2020, doi: 10.3390/s20061594.

[28] H. T. Nguyen, E.-H. Lee, C. H. Bae, and S. Lee, “Multiple object detection based on clustering and deep learning methods,” *Sensors*, vol. 20, no. 16, p. 4424, Aug. 2020, doi: 10.3390/s20164424.

[29] K. Lv, H. Du, Y. Hou, W. Deng, H. Sheng, J. Jiao, and L. Zheng, "Vehicle re-identification with location and time stamps," in *Proc. CVPR Workshops*, 2019, pp. 399–406.

[30] K. Lv, H. Sheng, Z. Xiong, W. Li, and L. Zheng, "Pose-based view synthesis for vehicles: A perspective aware method," *IEEE Trans. Image Process.*, vol. 29, pp. 5163–5174, 2020.

[31] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "AutoAugment: Learning augmentation strategies from data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 113–123.

[32] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2315–2324.

[33] C. Luo, Y. Zhu, L. Jin, and Y. Wang, "Learn to augment: Joint data augmentation and network optimization for text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13746–13755.

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556

[35] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[36] 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.

[37] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, vol. 148, 2006, pp. 369–376.

[38] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[39] M. Iaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," 2014, arXiv:1406.2227. [Online]. Available: http://arxiv.org/abs/1406.2227

[40] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proc. Brit. Mach. Vis. Conf.*, Surrey, U.K., 2012, pp. 1–11.

[41] K. Wang and S. J. Belongie, "Word spotting in the wild," in *Proc. 11th Eur. Conf. Comput. Vis. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 6311, Crete, Greece. Berlin, Germany: Springer, Sep. 2010, pp. 591–604.

[42] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 robust reading competitions," in *Proc. Int. Conf. Document Anal. Recognit.*, Edinburgh, U.K., Aug. 2003, pp. 682–687.

[43] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. I. Bigorda, S. H. Heidrech, J. Mas, D. F. Mota, J. A. Almazan, and L. P. de las Heras, “ICDAR 2013 robust reading competition,” in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Washington, DC, USA, Aug. 2013, pp. 1484–1493.

[44] D. Karatzas, L. Gomez-Bigorda, A. Nicolau, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, “ICDAR 2015 competition on robust reading,” in *Proc. 13th Int. Conf. Document Anal. Recognit. (ICDAR)*, Nancy, France, Aug. 2015, pp. 1156–1160.

[45] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. T. Tan, “Recognizing text with perspective distortion in natural scenes,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 569–576.

[46] 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. 5027–5048, 2014.

[47] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 4714–4722.

[48] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. D. Alché-Buc, E. Fox, and R. Garnett, Eds. Red Hook, NY, USA: Curran Associates, 2019, pp. 8024–8035.

[49] F. Borisuyk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 71–79.

[50] J. Wang and X. Hu, “Gated recurrent convolution neural network for OCR,” in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 335–344.

**DEGUO MU** was born in Ju County, Shandong, China, in 1980. He received the M.S. degree in software engineering from the College of Software, Beihang University, where he is currently pursuing the Ph.D. degree in computer science and artificial intelligence with the School of Computer.

Since 2011, he has been an Assistant Professor with the College of Software, Beihang University, and the Director of iYuba Laboratory. He had led the iYuba team to create more than 50 apps for learning language. In China there are about 50 million users learning English by iYuba apps until now. His research interests include deep learning on language education, mobile internet, and social networks.

Prof. Mu’s awards and honors include the First Prize of Innovation and Entrepreneurship of MIIT, in 2018, and the Ten Years Outstanding Graduates of Beihang Software College, in 2012.

**WEI SUN** was born in Sichuan, China, in 1961. He received the M.S. degree in computer science from Beihang University, and the Ph.D. degree from the University of Illinois, Chicago.

Since May 2003, he has been the Dean and a Professor with the Software College, Beihang University, training tens of thousands of software engineering masters in more than 10 years. He has published and edited nine books and more than 70 academic papers. He is an Executive Chairperson of Beijing Software Industry Association and a Professor of the FIU School of Computer Science, Florida State University, Miami, USA.

Dr. Sun served as a member of more than 30 international conference program committees and served as the Chairperson for the conference or program committee at the four major international conferences of ACM and IEEE.

**GUOLIANG XU**, photograph and biography not available at the time of publication.

**WEI LI**, photograph and biography not available at the time of publication.