electronics

MDPI

Article

An Algorithm for Natural Images Text Recognition Using Four Direction Features

Min Zhang [1], Yujin Yan [2], Hai Wang [1,*] and Wei Zhao [2]

[1] School of Aerospace Science and Technology, Xidian University, Xi'an 710071, China
[2] Key Laboratory of Electronic Equipment Structure Design, Ministry of Education, Xidian University, Xi'an 710071, China
* Correspondence: wanghai@mail.xidian.edu.cn; Tel.: +86-029-8820-3115

Received: 25 July 2019; Accepted: 27 August 2019; Published: 31 August 2019

Abstract: Irregular text has widespread applications in multiple areas. Different from regular text, irregular text is difficult to recognize because of its various shapes and distorted patterns. In this paper, we develop a multidirectional convolutional neural network (MCN) to extract four direction features to fully describe the textual information. Meanwhile, the character placement possibility is extracted as the weight of the four direction features. Based on these works, we propose the encoder to fuse the four direction features for the generation of feature code to predict the character sequence. The whole network is end-to-end trainable due to using images and word-level labels. The experiments on standard benchmarks, including the IIIT-5K, SVT, CUTE80, and ICDAR datasets, demonstrate the superiority of the proposed method on both regular and irregular datasets. The developed method shows an increase of 1.2% in the CUTE80 dataset and 1.5% in the SVT dataset, and has fewer parameters than most existing methods.

Keywords: text recognition; convolutional neural network; long short-term memory

1. Introduction

In a natural image, text may appear on various objects, such as advertising boards, road signs, etc. By recognizing the text, people can acquire the context and semantic information for image understanding. Therefore, scene text recognition attracts great interest in the field of computer vision. Although the optical character recognition (OCR) [1], which is widely studied for decades, can successfully deal with the text in documents, it is still a challenge to identify the text in natural images.

In recent years, with the development of artificial intelligent technology, there emerge a large number of text recognition methods [2–6] based on neural network. These methods can achieve good performance when dealing with the regular text. However, in practical applications, the scene text is usually irregular (such as slant, curved, perspective, multiple lines, etc.), as shown in Figure 1, and the performance of the above methods is not satisfying.

Figure 1. Examples of irregular (slant, curved, perspective, multiple lines, etc.) text in natural images.

All of the mentioned methods can be divided into two categories: one for predicting irregular text, and another for predicting regular text. In practical applications, the scene text is usually irregular (such as slant, curved, perspective, multiple lines, etc.), as shown in Figure 1, and the performance of the above methods is not satisfying. At present, there are many methods [7–12] to predict irregular text, which can be divided into two types. A common method [7–9] is to correct and then identify; for example, Shi et al. [7] propose a robust text recognizer with automatic rectification (RARE) consisting of a spatial transformer network (STN) [13] and a sequence recognition network (SRN). The irregular text is corrected into approximately regular text by using STN, and after that SRN recognizes the corrected text through a sequence recognition technique from left to right. RARE can recognize some natural images, but fails to recognize the vertical text. In addition, it is difficult to optimize STN to fit various kinds of irregular text in natural images, such as curved or arbitrarily oriented text. Another method [10,11] predicts final results directly, such as when Yang et al. [10] introduce an auxiliary dense character detection task that uses a fully convolutional network (FCN) [14] to learn text visual representations and an alignment loss to guide the training of an attention model. This method recognizes two-dimensional text images effectively, but the text must be presented from left to right. Moreover, character-level bounding box annotations are required during training for utilization of the attention block.

Usually, most text recognition methods treat the text in natural images as a one-dimensional sequence and recognize it from left to right, such as [6–8]. However, in reality the text can proceed in any direction, often leading to unsatisfactory recognition results. In this paper, we propose a robust end-to-end method that utilizes four direction features to predict scene text. To represent a character in any direction, the method learns the four direction features and combines them together. Moreover, the structure of the four direction features shares the neural network weights, which provides the whole network with high accuracy without an increase in the amount of data.

As shown in Figure 2, first, we use a basic convolutional neural network (BCN) to extract the low-level features. Then, we obtain the four direction feature sequences and the character placement possibility through a multidirectional convolutional neural network (MCN). The four direction feature sequences work in four directions: horizontal vectors $\overrightarrow{l}$ (left → right) and $\overrightarrow{r}$ (right → left), vertical vectors $\overrightarrow{t}$ (top → bottom) and $\overrightarrow{b}$ (bottom → top). Horizontal and vertical features are extracted by sampling the height and width of the feature map down to 1. Then, characters in the natural images can be represented as a sequence named feature code through the encoder, which combines the four direction feature sequences and corresponding placement possibility. Finally, the decoder predicts the result of the text from the feature code. In this paper, our method is not only effective at recognizing regular text, but also succeeds at predicting irregular text in complex natural images.

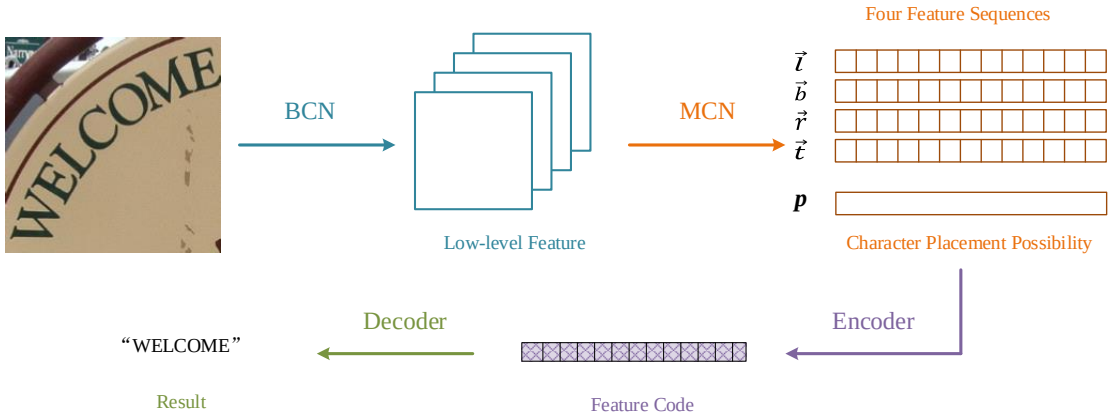

Figure 2. Flowchart of our method.

The rest of the paper is organized as follows: In Section 2, theories related to the proposed algorithm are introduced. The details of our method are described in Section 3, and the experimental results are given in Section 4. In Section 5, overall conclusions are given.

2. Related Works

2.1. Text Recognition

In recent years, a large number of papers related to text recognition have been published. Ye and Doermann's survey [15] analyzes and compares the technical challenges, methods, and performance of text recognition research in color imagery. Most traditional methods follow a bottom-up pipeline and the others work in the top-down style. The methods based on bottom-up [1,5,6,16–18] first detect and recognize single characters, then integrate them into words based on a language model. However, these methods require character-level labeling, and consume a large amount of resources.

The top-down pipeline approach directly predicts the whole text from the images without additional character-level labels. The method used in [19] can recognize 90k words, where each word corresponds to one classification in the CNN output layer. Therefore, the model can only recognize 90k words. Many recent works, such as [2,4,6,7,10,20] make use of a recurrent neural network (RNN). The authors of [6,20] proposed an end-to-end neural network, and regarded the natural images recognition as a sequence labeling problem. The network uses CNN for sequence feature extraction, RNN for per-frame prediction, and CTC [21] loss for final label sequence prediction. The method in [4] presents a recursive recurrent neural network with attention modeling (R^2AM). The model consists of two blocks: recursive CNN for image feature extraction, and RNN with attention-based mechanism for recognition. The method in [2] develops a focusing network (FN) to eliminate the attention drift of the attention network (AN). However, these methods transform the image into a feature sequence from left to right and cannot recognize irregular text (e.g., curved text) in natural images. In order to tackle this problem, the authors of [7] proposed a solution named a space transformer network (STN), which can transform the input image into a rectified image for a sequence recognition network (SRN) to recognize. The method in [10] uses a FCN [14] to learn text visual patterns and recognizes the text through the attention-based RNN.

In this paper, our method uses a top-bottom pipeline approach, which does not need to detect individual characters. Different from existing top-down methods, our method first uses BCN and MCN to extract the four feature sequences and their character placement possibility, then encodes them by RNN, and finally predicts the results. At the same time, we can only use word-level labels to train the whole network end-to-end.

2.2. BiLSTM

RNN, used in the top-down methods, is most probably a bidirectional long short-term memory neural network (BiLSTM) [2,6,20]. RNN can distinguish ambiguous characters easily by utilizing the context within the sequence. In addition, RNN can handle any length of sequence, which is very helpful in text recognition. Meanwhile, RNN can return the back propagation error to the previous layer and make it possible to train the whole network end-to-end.

The RNN layer consists of multiple RNN units with the same parameters. These units are arranged in sequence. For the i th (i = 1, ... T) unit, its input data contain the i th element of the sequence $[x_1, \cdots, x_i, \cdots, x_T]$ and the output of the (i-1) th unit h_{i-1}. Based on the input data, the i th unit outputs the result $h_i = f(x_i, h_{i-1})$. In that way, RNN units can obtain past information for prediction. However, RNN often suffers from the vanishing gradient problem [22], which makes it difficult to transmit the gradient information consistently over a long time.

LSTM [23,24], as a special RNN, consists of a memory cell and three multiplicative gates: the input gate, forget gate, and output gate. The memory cell allows the LSTM to store past contexts, while the input and output gates allow LSTM to store contexts for the future. However, in the prediction

of natural images, both types of context are important. Therefore, we refer to [25] and combine two LSTMs into BiLSTM. As shown in Figure 3, one of the LSTMs utilizes past context and another utilizes future context, and after that the final prediction sequence is obtained.

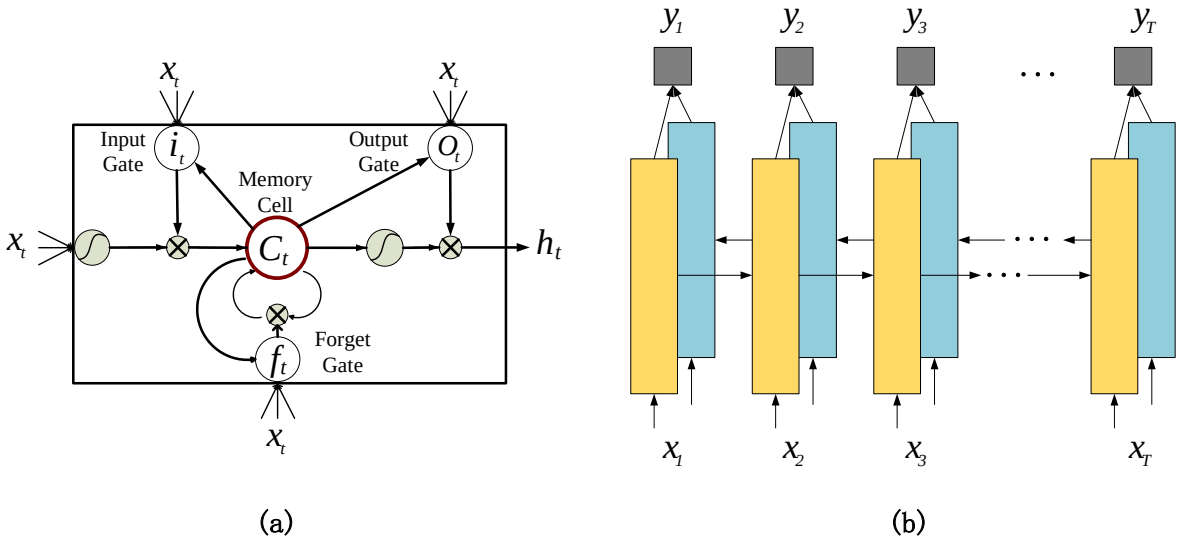

Figure 3. (**a**) The structure of LSTM unit: a memory cell and three gates, namely the input gate, the forget gate, and the output gate. (**b**) The structure of BiLSTM used in our method. It combines two LSTMs, one for past context (left to right) and one for future context (right to left).

3. Framework

In order to recognize irregular text, we identify the four direction features separately, and then fuse the four direction features to get the final recognition result. Each direction feature is acquired using the method developed in [6], which is proven to be effective for regular text recognition. As shown in Figure 4, the four direction features can be defined as follows:

$$\begin{aligned} \text{the left-to-right direction feature}: & \ \overrightarrow{t}, \overrightarrow{t'} \\ \text{the bottom-to-top direction feature}: & \ \overrightarrow{b}, \overrightarrow{b'} \\ \text{the right-to-left direction feature}: & \ \overrightarrow{r}, \overrightarrow{r'} \\ \text{the top-to-bottom direction feature}: & \ \overrightarrow{t}, \overrightarrow{t'}, \end{aligned} \tag{1}$$

where the feature sequences $\overrightarrow{l}, \overrightarrow{b}, \overrightarrow{r}, \overrightarrow{t}$ are the outputs of MCN, and the feature vectors $\overrightarrow{l'}, \overrightarrow{b'}, \overrightarrow{r'}, \overrightarrow{t'}$ are the outputs of the first BiLSTM. Specifically, our method includes four parts: (1) BCN for low-level features extraction; (2) MCN for extracting four direction feature sequences and character placement possibility; (3) encoder for combing four direction feature sequences with the character placement possibility; (4) decoder for predicting the final character sequence.

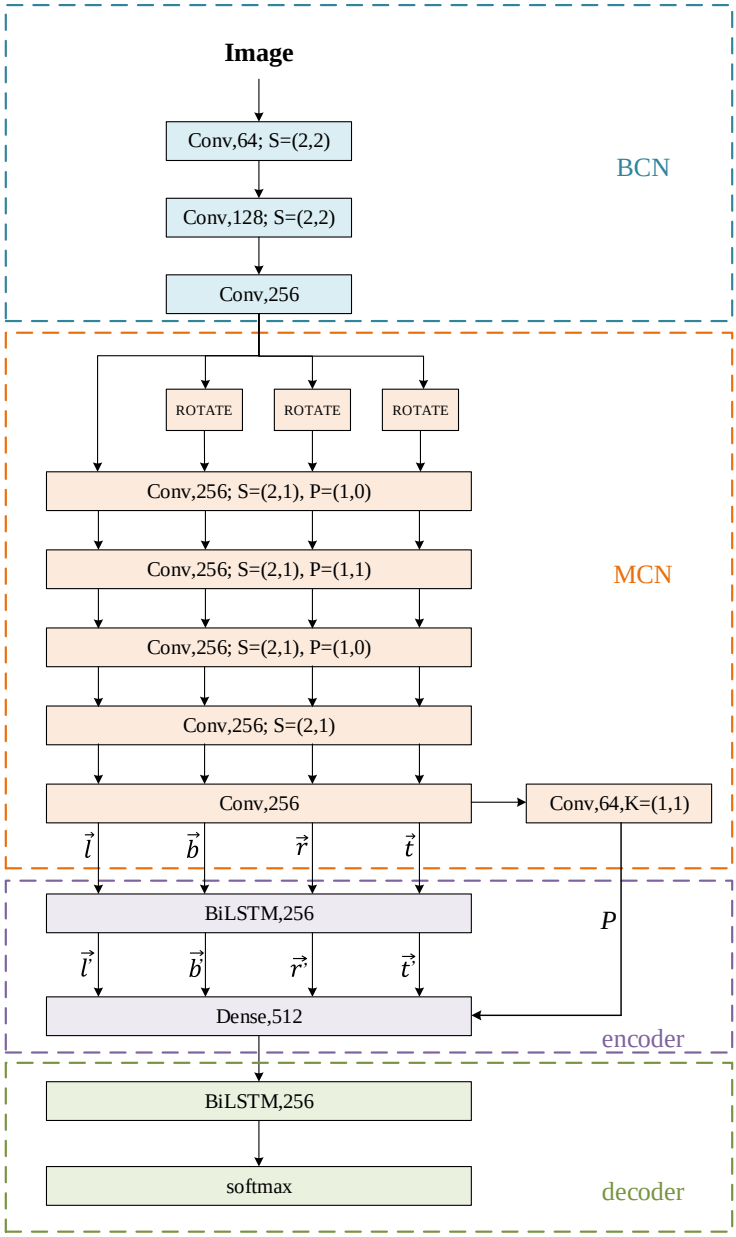

Figure 4. The network structure of our method. These parts are shown in blue, orange, purple, and green rectangles, respectively. The convolutional layers are represented as [name, channel, K = kernel size; S = pooling layer strides, P = pooling layer padding]. The dense and BiLSTM layers are represented as [name, channel].

3.1. BCN and MCN

In our model, feature extraction is performed by a basic convolutional neural network (BCN) and a multidirectional convolutional neural network (MCN).

BCN is evolved from the standard CNN model. As shown in Figure 4, the convolutional layers and max-pooling layers are used to extract the low-level features of the image. Before inputting into the network, all natural images should be scaled to 100×100 pixels. Since the input image is a square, the feature graph obtained after convolution and pooling is also a square. On the one hand, using BCN can obtain the low-level features of the image quickly. On the other hand, by reducing the size of the feature map through pooling, the computational cost can be reduced effectively.

In this work, characters in natural images are represented by multiple perspectives, just like vectors represented by the cross coordinate. Therefore, we use MCN to extract features from multiple

perspectives of a BCN feature graph. The MCN network consists of five convolutional layers, and represents the feature graph as a sequential feature. We use the rotator to rotate the feature maps of BCN 90°, 180° and 270°, respectively. MCN and BCN are connected by the rotator, and MCN can get the four direction feature sequences. As shown in Figure 5, the four direction feature sequences represent four directions: left → right, bottom → top, right → left, and top → bottom. Multidirectional feature sequences can be represented as follows:

$$\begin{aligned} \overrightarrow{l} &: (l_1, l_2, \cdots, l_L), left \rightarrow right \\ \overrightarrow{b} &: (b_1, b_2, \cdots, b_L), bottom \rightarrow top \\ \overrightarrow{r} &: (r_1, r_2, \cdots, r_L), right \rightarrow left \\ \overrightarrow{t} &: (t_1, t_2, \cdots, t_L), top \rightarrow bottom. \end{aligned} \tag{2}$$

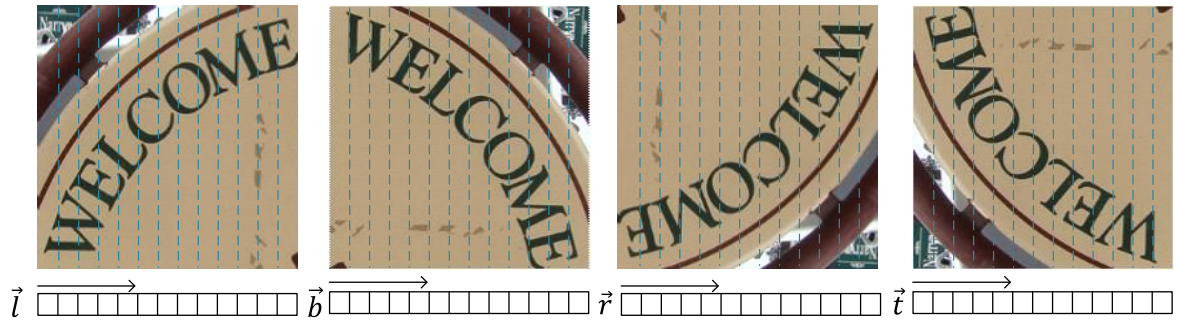

Figure 5. The input image can be divided into a series of areas, and these areas are associated with the four direction feature sequences.

The four direction feature sequences $\overrightarrow{l}, \overrightarrow{b}, \overrightarrow{r}, \overrightarrow{t}$ have a size of $L \times D$, where L and D represent the length of the four feature sequences and the channel number, respectively. By using MCN to extract sequential features in multiple directions, the network can not only reduce computational resources, but also achieve high accuracy with a small amount of training data.

3.2. Encoder

Encoder is the process of combining four direction feature sequences into one. By using BiLSTM, we can encode the output feature sequences $\overrightarrow{l}, \overrightarrow{b}, \overrightarrow{r}, \overrightarrow{t}$ of MCN into the four direction feature vectors $\overrightarrow{l'}, \overrightarrow{b'}, \overrightarrow{r'}, \overrightarrow{t'}$. By this means, the four direction feature vectors contain the contextual information, which also represents the four directions' features. Each character in a natural image can be described by $\overrightarrow{l'}, \overrightarrow{b'}, \overrightarrow{r'}, \overrightarrow{t'}$; therefore, we use MCN to calculate the possibility of corresponding characters in each direction, which is termed the character placement possibility p. For each text image, the character placement possibility is used as the weight of each feature vector, and $\sum_{j=1}^{4} p_{ij} = 1$ with a size of p equal to $L \times 4$. As shown in Figure 6, we use the corresponding p to combine the four direction feature vectors as follows:

$$\overrightarrow{h'_i} = \left[\overrightarrow{l'}, \overrightarrow{b'}, \overrightarrow{r'}, \overrightarrow{t'}\right] p_i \tag{3}$$

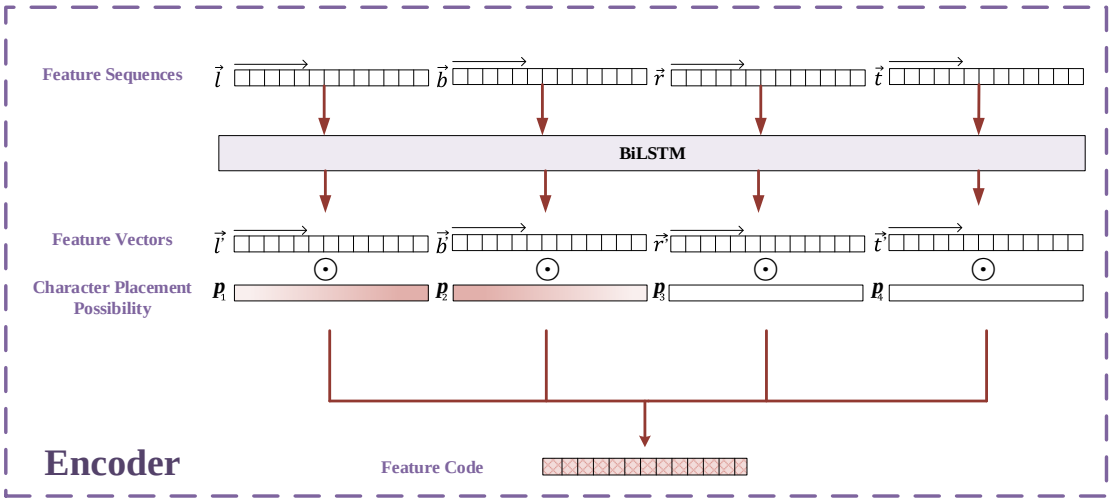

Figure 6. The structure of the encoder.

The combination of four direction feature sequences is the feature code $\overrightarrow{h'} = \left(\overrightarrow{h'_1}, \overrightarrow{h'_2}, \cdots, \overrightarrow{h'_L}\right)$.

3.3. Decoder

Decoder is the process of converting feature code into character sequence. In the decoder block, we use BiLSTM to handle the feature code $\overrightarrow{h'}$ by taking the contextual information into consideration, and the feature sequence $\overrightarrow{h}$ is obtained. The probability of each area in the text image is calculated by the formula $y_i = softmax\left(\overrightarrow{h_i}\right)$, and then the probabilities obtained are input to the connectionist time classification (CTC) layer proposed by Graves et al. [21]; finally, the character sequence is achieved.

The probability y has the size of $L \times C$, where L is the length of the prediction sequence and C is the classification of characters. We use the probability y to predict the character sequence, but the length of text in images is not equal to L. The CTC is specifically designed for tasks in which the lengths of the input sequence and target sequence are unequal and it is difficult to segment the input sequence to match the target sequence. Therefore, the CTC introduces the sequence-to-sequence mapping function B as follows:

$$B\left(\underset{\pi}{argmax} P(\pi|p)\right) \tag{4}$$

where B removes the repeated labels and non-character labels. The CTC looks for an optimized path (π) with maximum probability through the input sequence. For example, $B(-hh-e-l-ll-oo-) = hello$, where '-' represents the non-character label.

4. Experiments

4.1. Datasets

The regular and irregular benchmarks are as follows:

CUTE80 (CT80 for short) [26] is collected for evaluating curved text recognition. It contains 288 cropped natural images for testing. No lexicon is associated.

ICDAR 2015 (IC15 for short) [27] contains 2077 cropped images, of which more than 200 are irregular (arbitrarily-oriented, perspective, or curved). No lexicon is associated.

IIIT5K-Words (IIIT5K for short) [28] is collected from the Internet, containing 3000 cropped word images in its test set. Each image specifies a 50-word lexicon and a 1k-word lexicon, both of which contain the ground truth words as well as other randomly picked words.

Street View Text (SVT for short) [17] is collected from Google Street View, and consists of 647 word images in its test set. Many images are severely corrupted by noise and blur, or have very low resolutions. Each image is associated with a 50-word lexicon.

ICDAR 2003 (IC03 for short) [29] contains 251 scene images, labeled with text bounding boxes. Each image is associated with a 50-word lexicon defined by Wang et al. [17]. For fair comparison, we discard images that contain non-alphanumeric characters or have fewer than three characters [17]. The resulting dataset contains 867 cropped images. The lexicons include the 50-word lexicons and the full lexicon that combines all lexicon words.

ICDAR 2013 (IC13 for short) [16] is the successor to IC03, from which most of its data are inherited. It contains 1015 cropped text images. No lexicon is associated.

4.2. Details

Network: In order to ensure that the network can recognize regular and irregular text, we input 100×100 pixel images, a size that mostly ensures we retain the character of the images. All convolutional layers have 3×3 size of kernels, and the pooling layers have 2×2 size kernels. We use batch normalization (BN) [30] and RELU activation behind each convolution layer. In the decoding process, we designed the BiLSTM and 37 outputs (including 26 letters, 10 digits, and an EOS symbol).

Training: We trained our model using synthetic data [3] and real-world training datasets [17,27–29] by the ADADELTA [31] optimization method. Meanwhile, we conducted data augmentation by randomly rotating each image range from 0° to 360° once. Our method was implemented under the Tensorflow framework [32], running on a computer with an 8-core CPU, 32G RAM, TitianXP GPU, and Ubuntu 16.04.

4.3. Comparative Evaluation

As shown in Table 1, we compared the recognition results of our method with 17 related methods in each regular dataset. Our method had the advantages of SVT and achieved a similar performance to other top algorithms in the IIIT5K, IC03, and IC13 datasets. The proposed method had 0.2% accuracy improvement over SVT-50 and 1.5% accuracy improvement over SVT-None compared to the most effective method [2]. Compared with [2] and [19], we achieved a simpler structure. The method in [2] is composed of two main structures: one is an attention network (AN) for recognizing characters, and the other is a focusing network (FN) for adjusting attention by evaluating whether AN properly pays attention to the target areas in the image. Additional position correction can make the network achieve better results on the basis of recognition, but this step needs to obtain the position information of each character during training. However, our network does not need additional character-level labeling in the training stage, and using word-level labeling can train the network end-to-end. Our method is more accurate than Cheng's baseline [2], which removes FN that requires additional character-level labeling. The output of the method in [19] is the whole word, so text beyond the scope of its vocabulary cannot be recognized. The proposed algorithm, which can handle strings of arbitrary length and random arrangement, is more suitable for natural scenes.

Table 2 shows that our method outperforms the other methods in the CT80 and IC15 datasets. For example, our method can achieve 0.5% accuracy improvement in the IC15 dataset and 1.2% accuracy improvement in the CT80 dataset compared with the most effective method [2]. Figure 7 shows some pictures of dataset IC15 and CT80, which represent the problems of slant, curved, perspective, multiple lines, and so on. Since the MCN can effectively obtain multidirectional text information, the proposed method can well identify slant and perspective text. The encoding algorithm enables the network to process curved text. Moreover, our network has strong robustness against noise and can recognize a whole line of text accurately. In conclusion, this work achieves better results on irregular datasets and higher robustness on regular datasets with less loss of accuracy.

Table 1. Results in regular datasets. '50', '1k' and 'Full' in the second row refer to the lexicon used, while 'None' means recognition without a lexicon.

Method	IIIT5K			SVT		IC03			IC13
	50	1k	None	50	None	50	Full	None	None
ABBYY [17]	24.3	-	-	35.0	-	56.0	55.0	-	-
Wang et al. [17]	-	-	-	57.0	-	76.0	62.0	-	-
Mishra et al. [25]	64.1	57.5	-	73.0	-	81.8	67.8	-	-
Wang et al. [33]	-	-	-	70.0	-	90.0	84.0	-	-
Goel et al. [34]	-	-	-	77.3	-	89.7	-	-	87.6
Bissacco et al. [1]	-	-	-	90.4	78.0	-	-	-	-
Alsharif [35]	-	-	-	74.3	-	93.1	88.6	-	-
Almazan et al. [36]	91.2	82.1	-	89.2	-			-	-
Yao et al. [37]	80.2	69.3	-	75.9	-	88.5	80.3	-	-
Jaderberg et al. [16]	-	-	-	86.1	-	96.2	91.5	-	-
Su and Lu [38]	-	-	-	93.0	-	92.0	82.0	-	-
Gordo [39]	93.3	86.6	-	91.8	-	-	-	-	-
Jaderberg et al. [19]	97.1	92.7	-	95.4	80.7	98.7	**98.6**	93.1	90.8
Jaderberg et al. [16]	95.5	89.6	-	93.2	71.7	97.8	97.0	89.6	81.8
Shi et al. [6]	97.6	94.4	78.2	96.4	80.8	98.7	97.6	89.4	86.7
Shi et al. [7]	96.2	93.8	81.9	95.5	81.9	98.3	96.2	90.1	88.6
Cheng's baseline [2]	98.9	96.8	83.7	95.7	82.2	98.5	96.7	91.5	89.4
Cheng et al. [2]	**99.3**	**97.5**	**87.4**	97.1	85.9	**99.2**	97.3	**94.2**	**93.3**
Ours	99.0	96.9	87.2	**97.3**	**87.4**	97.8	96.4	91.0	90.1

Table 2. Results in irregular datasets. 'None' means recognition without a lexicon.

Method	CT80	IC15
	None	None
Shi et al. [6]	54.9	-
Shi et al. [7]	59.2	-
Cheng et al. [2]	63.9	63.3
ours	**65.1**	**63.5**

Figure 7. Correct recognition (**left**); incorrect samples (**right**).

To better illustrate the advantages of our structure, we provide Table 3, which includes properties named end-to-end, CharGT, dictionary, and model size. End-to-end: this column is to show whether the model can deal with the image directly without any handcrafted features. As can be observed from Table 3, only the model in [6,16,19] and ours can be trained end-to-end. CharGT: this column is to indicate whether the model needs character-level annotations in training. The output of ours is a sequence, and our method only uses word-level labeling. Dictionary: this column is to indicate

whether the model depends on a special dictionary and so is unable to handle out-of-dictionary words or random sequences. Our algorithm can handle strings of arbitrary length and random arrangement. Model size: this column is to report the storage space of the model. As shown in Table 3, there are fewer parameters in our model than in the models in [16,19], which means our model is smaller. Ours has only 15.4 million parameters, slightly more than the model in [6], which can only recognize regular text. Table 3 clearly shows that our model is more practical.

Table 3. Comparison of various methods. Attributes for comparison include: (1) End-to-end: the model being end-to-end trainable; (2) CharGT: the model needs character-level annotation beyond training; (3) dictionary: the model can only recognize words belonging to a special dictionary; (4) model size: the number of model parameters. M stands for millions.

Method	End-to-End	CharGT	Dictionary	Model Size
Wang et al. [17]	×	NEEDED	NOT NEED	-
Mishra et al. [25]	×	NEEDED	NEEDED	-
Wang et al. [33]	×	NEEDED	NOT NEED	-
Goel et al. [34]	×	NOT NEED	NEEDED	-
Bissacco et al. [1]	×	NEEDED	NOT NEED	-
Alsharif [35]	×	NEEDED	NOT NEED	-
Almazan et al. [36]	×	NOT NEED	NEEDED	-
Yao et al. [37]	×	NEEDED	NOT NEED	-
Jaderberg et al. [16]	×	NEEDED	NOT NEED	-
Su and Lu [38]	×	NOT NEED	NOT NEED	-
Gordo [39]	×	NEEDED	NEEDED	-
Jaderberg et al. [19]	√	NOT NEED	NEEDED	490 M
Jaderberg et al. [16]	√	NOT NEED	NOT NEED	304 M
Shi et al. [6]	√	NOT NEED	NOT NEED	8.3 M
ours	√	NOT NEED	NOT NEED	15.4 M

4.4. Comparison with Baseline Models

To evaluate the efficiency of the proposed method, we also trained two baseline models: Without_4 and Without_P. Without_4 combines the four direction feature maps of the rotator rotation into one through a convolutional layer. Without_P combines these features through a dense layer. We conducted experiments on both regular and irregular text recognition datasets.

From Table 4, we can see that the baseline models do not recognize regular or irregular text well. Unlike the proposed model and the Without_P model, the Without_4 model achieves only 39.8% accuracy in the CT80 dataset containing curved text. These comparison results indicate that the four direction features are necessary for recognizing curved text. In both regular and irregular text datasets, the accuracy of the Without_P model is lower than that of the proposed model. These results demonstrate the necessity of an encoder, which can effectively combine the four direction features with character placement possibility.

Table 4. Results from regular and irregular datasets.

Method	Regular Dataset				Irregular Dataset	
	IIIT5K	SVT	IC03	IC13	CT80	IC15
Without_4	73.3	77.4	82.4	78.9	39.8	50.4
Without_P	79.6	81.2	85.7	82.5	54.5	54.6
ours	**87.2**	**87.4**	**91.0**	**90.1**	**65.1**	**63.5**

5. Conclusions

In this paper, we present a novel neural network for scene text recognition by (1) designing BCN and MCN to extract four direction features and the corresponding character placement possibility, (2) using an encoder to integrate the four direction features and their placement possibility and thus gain a feature code, and (3) applying a decoder to recognize the character sequence. Different from most existing methods, our method can recognize both regular and irregular text from scene images. The proposed algorithm outperforms other methods in irregular datasets and achieves a similar performance to the top methods in regular datasets. Our method can achieve 87.4% accuracy in the SVT dataset, 65.1% in the CT80 dataset, and 63.5% in the IC15 dataset. The experiments on regular and irregular datasets demonstrate that our method achieves superior or highly competitive performance. Moreover, we trained two baseline models, and the results prove that the proposed method with four direction features and character placement possibility is effective. In the future, we plan to make it more practical for real-world applications.

Author Contributions: Conceptualization, M.Z. and Y.Y.; methodology, H.W. and W.Z.; software, Y.Y. and W.Z.; validation, M.Z. and H.W.; formal analysis, M.Z.; investigation, Y.Y.; resources, H.W.; data curation, W.Z.; writing—original draft preparation, Y.Y.; writing—review and editing, W.Z.; visualization, Y.Y.; supervision, M.Z.; project administration, M.Z.; funding acquisition, H.W.

Funding: This research was funded by the China Postdoctoral Science Foundation (2018M633471), the National Natural Science Foundation of Shaanxi Province under grant 2019JQ-270, the AeroSpace T.T. and C. Innovation Program, and the China Scholarship Council (201806965054).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bissacco, A.; Cummins, M.; Netzer, Y.; Neven, H. PhotoOCR: Reading Text in Uncontrolled Conditions. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013.
2. Cheng, Z.; Bai, F.; Xu, Y.; Zheng, G.; Pu, S.; Zhou, S. Focusing Attention: Towards Accurate Text Recognition in Natural Images. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
3. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition. *arXiv*. 2014. Available online: https://arxiv.org/abs/1406.2227 (accessed on 9 December 2014).
4. Osindero, S.; Lee, C.Y. Recursive Recurrent Nets with Attention Modeling for OCR in the Wild. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
5. Neumann, L.; Matas, J. Real-time scene text localization and recognition. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012.
6. Shi, B.; Bai, X.; Yao, C. An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Trans. Pattern Anal. & Mach. Intell.* **2016**, *39*, 2298–2304.
7. Shi, B.; Lyu, P.; Wang, X.; Yao, C.; Bai, X. Robust Scene Text Recognition with Automatic Rectification. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016.
8. Shi, B.; Yang, M.; Wang, X.; Lyu, P.; Yao, C.; Bai, X. ASTER: An Attentional Scene Text Recognizer with Flexible Rectification. *IEEE Trans. Pattern Anal. & Mach. Intell.* **2019**, *41*, 2035–2048.
9. Gao, Y.; Chen, Y.; Wang, J.; Lei, Z.; Zhang, X.Y.; Lu, H. Recurrent Calibration Network for Irregular Text Recognition. *arXiv*. 2018. Available online: https://arxiv.org/abs/1812.07145 (accessed on 18 December 2018).
10. Yang, X.; He, D.; Zhou, Z.; Kifer, D.; Giles, C.L. Learning to Read Irregular Text with Attention Mechanisms. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), Melbourne, Australia, 19–25 August 2017.

11. Wang, P.; Yang, L.; Li, H.; Deng, Y.; Shen, C.; Zhang, Y. A Simple and Robust Convolutional-Attention Network for Irregular Text Recognition. *arXiv*. 2019. Available online: https://arxiv.org/abs/1904.01375 (accessed on 2 April 2019).
12. Bai, F.; Cheng, Z.; Niu, Y.; Pu, S.; Zhou, S. Edit Probability for Scene Text Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
13. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial transformer networks in Advances in neural information processing systems. *arXiv*. 2016. Available online: https://arxiv.org/abs/1506.02025 (accessed on 4 February 2016).
14. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. , Boston, MA, USA, In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2015, Boston, MA, USA, 8–10 June 2015.
15. Ye, Q.; Doermann, D. Text detection and recognition in imagery: A survey. *IEEE Trans. pattern Anal. Mach. Intell.* **2014**, *37*, 1480–1500. [CrossRef] [PubMed]
16. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Deep structured output learning for unconstrained text recognition. *arXiv*. 2015. Available online: https://arxiv.org/abs/1412.5903 (accessed on 10 April 2015).
17. Wang, K.; Babenko, B.; Belongie, S. End-to-end scene text recognition. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011.
18. Wang, K.; Belongie, S. Word spotting in the wild. In Proceedings of the European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010.
19. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vis.* **2016**, *116*, 1–20. [CrossRef]
20. He, P.; Huang, W.; Qiao, Y.; Loy, C.C.; Tang, X. Reading scene text in deep convolutional sequences. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
21. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In Proceedings of the 23rd international conference on Machine learning, Pittsburgh, PA, USA, 25–29 June 2006.
22. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
23. Gers, A.F.; Schraudolph, N.N.; Schmidhuber, J. Learning Precise Timing with LSTM Recurrent Networks. *J. Mach. Learn. Res.* **2002**, *3*, 115–143.
24. Graves, A. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780.
25. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013.
26. Risnumawan, A.; Shivakumara, P.; Chan, C.S.; Tan, C.L. A robust arbitrary text detection system for natural scene images. *Expert Syst. Appl.* **2014**, *41*, 8027–8048. [CrossRef]
27. Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V.R.; Lu, S.; et al. ICDAR 2015 competition on robust reading. In Proceedings of the 2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015.
28. Mishra, A.; Alahari, K.; Jawahar, C. Scene text recognition using higher order language priors. In Proceedings of the BMVC-British Machine Vision Conference, Surrey, UK, 3–7 September 2012.
29. Lucas, S.M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; Young, R. ICDAR 2003 robust reading competitions. In Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh, UK, 6 August 2003.
30. Ioffe, S.; Szegedy, C. Batch Normalization: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv*. 2015. Available online: https://arxiv.org/abs/1502.03167 (accessed on 2 March 2015).
31. Zeiler, M.D. ADADELTA: An Adaptive Learning Rate Method. *arXiv*. 2012. Available online: https://arxiv.org/abs/1212.5701 (accessed on 22 December 2012).

32. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, USA, 2–4 November 2016.
33. Wang, T.; Wu, D.J.; Coates, A.; Ng, A.Y. End-to-end text recognition with convolutional neural networks. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012.
34. Goel, V.; Mishra, A.; Alahari, K.; Jawahar, C.V. Whole is greater than sum of parts: Recognizing scene text words. In Proceedings of the 2013 12th International Conference on Document Analysis and Recognition, Washington, DC, USA, 25–28 August 2013.
35. Alsharif, O.; Pineau, J. End-to-End Text Recognition with Hybrid HMM Maxout Models. *arXiv*. 2013. Available online: https://arxiv.org/abs/1310.1811 (accessed on 7 October 2013).
36. Almazán, J.; Gordo, A.; Fornés, A.; Valveny, E. Word Spotting and Recognition with Embedded Attributes. IEEE Trans. *Pattern Anal. Mach. Intell.* **2014**, *36*, 2552–2566. [CrossRef] [PubMed]
37. Yao, C.; Bai, X.; Shi, B.; Liu, W. Strokelets: A Learned Multi-scale Representation for Scene Text Recognition. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014.
38. Su, B.; Lu, S. Accurate scene text recognition based on recurrent neural network. In Proceedings of the Asian Conference on Computer Vision, Singapore, 1–5 November 2014.
39. Gordo, A. Supervised mid-level features for word image representation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 8–10 June 2015.