Computers, Materials & Continua CMC, vol.60, no.2, pp.781-794, 2019

A Novel Scene Text Recognition Method Based on Deep Learning

Maosen Wang[1], Shaozhang Niu[1, *] and Zhenguang Gao[2]

Abstract: Scene text recognition is one of the most important techniques in pattern recognition and machine intelligence due to its numerous practical applications.
Scene text recognition is also a sequence model task. Recurrent neural network (RNN) is commonly regarded as the default starting point for sequential models. Due to the non-parallel prediction and the gradient disappearance problem, the performance of the RNN is difficult to improve substantially. In this paper, a new TRDD network architecture which base on dilated convolution and residual block is proposed, using Convolutional Neural Networks (CNN) instead of RNN realizes the recognition task of sequence texts.
Our model has the following three advantages in comparison to existing scene text recognition methods: First, the text recognition speed of the TRDD network is much fast than the state-of-the-art scene text recognition network based recurrent neural networks (RNN). Second, TRDD is easier to train, avoiding the problem of exploding and vanishing, which is major issue for RNN. Third, both using larger dilated factors and increasing the filter size are all viable ways to change receptive field size. We benchmark the TRDD on four standard datasets, it has higher recognition accuracy and faster recognition speed based on the smaller model. It is hopefully used in the real-time application.

Keywords: Scene text recognition, dilated convolution, CTC, CNN, TCN.

1 Introduction

With the popularization of smart phones and the tremendous demands of text recognition in Augmentation Reality, scene text recognition is an important part for scene understanding. However, scene text recognition is much more challenging task due to the text in natural scene images is vastly variable in layout and appearance, being drawn from a lot of fonts and styles, suffering from occlusions, inconsistent lighting, noise, orientations.

Scene text recognition methods can be generally grouped into segmentation-based word recognition and holistic word recognition. There are many studies of segmentation-based methods [Yi, Huang, Hao et al. (2014); Jaderberget, Vedaldi and Zisserman (2014); Babenko and Belongie (2012)], these methods are very effective for the scanned documents. However, it is very difficult to split characters in complicated cases, especially for Asian characters, that include amount of left and right structure characters.

[1] Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, 100876, China.

[2] Department of Computer Science, Framingham State University, Framingham, MA 01772, USA.

[*] Corresponding Author: Shaozhang Niu. Email: szniu@bupt.edu.cn.

Error splitting or merging in the word segmentation almost affect the accuracy of recognition. In addition, these methods adopt isolated character classification, recognize subsequent word separately, and discard meaningful context information of text, so their reliability and robustness are reduced in text recognition. In order to solve the problem, sequence text recognition [Españaboquera, Castrobleda, Gorbemoya et al. (2016); Bissaccoet, Cummins and Netzer (2013); Xiong, Wang, Zhu et al. (2018)] is proposed. For the scene text, the segmentation of the text is no needed, the holistic text is directly recognized as a sequence. The strong sequence features are extracted through the Deep Neural Networks (DNN) network ensure robustness to various distortions text and messy background. Sequence text recognition becomes the mainstream model of scene text recognition, such as CRNN [Shi, Bai and Yao (2015)], DTRN [He, Zhang, Ren et al. (2015)], FAN [Cheng, Bai, Xu et al. (2017)], that generally use the RNN model to learn contextual information of text. RNN is almost the only choice of sequence models, but it has some disadvantages such as not parallelism, unstable gradient, and high memory requirement for training [Bai, Kolter and Koltun (2018)], researchers have been looking for better models to replace RNN.

In recent research, temporal convolutional network (TCN) is applied across all sequence tasks, the performance of TCN outperforms canonical recurrent architectures such as LSTM [Hochreiter and Schmidhuber (1997)], GRU [Jozefowicz, Zaremba and Sutskever (2015); Dey and Salemt (2017)] and RNN on 11 sequence tasks [Bai, Kolter and Koltun (2018)].TCN network is essentially a CNN network, which integrates dilated causal convolution and residual block.

Motivated by TCN design idea, this paper proposed a new network model TRDD (Text recognition based on dilation and residual block). The model uses two basic residual modules, one module is composed of dilated convolution and the other is composed of ordinary convolution. The TRDD network has the following characteristics:

First, in both training and evaluation, a long input sequence can be processed as whole in TRDD, instead of sequentially as in RNN.

Second, the receptive field size of sequence features can be increased by using larger dilation factors or increasing the filter size.

Third, the residual block is used to improve the network training speed and enrich the semantic features of the text.

Fourth, the filters in TRDD are shared across a layer, with backpropagation path depending only on network depth, in practice, gated RNNs likely to take up too much memory.

2 Related work

Before defining the network architecture, we describe the nature of the sequence modeling tasks. Suppose that we give an input sequence $x^0, x^1, \ldots, x^T$ and wish to predict some corresponding outputs $y^0, y^1, \ldots, y^T$ at each time. A sequence modeling network is any function $f: X^{T+1} \to Y^{T+1}$ that produces the mapping

$$\hat{y}^0, \hat{y}^1, \ldots, \hat{y}^T = f(x^0, x^1, \ldots, x^T) \tag{1}$$

$\hat{y}^t$ Depends only on $x^0, \ldots, x^t$ and not on any later inputs $x^{t+1}, \ldots, x^T$.

The goal of supervised network training is to find a network f that minimizes some expected loss between the predictions and the actual outputs:

$$L(y^0, y^1, \ldots, y^T, f(x^0, x^1, \ldots, x^T)). \quad (2)$$

RNN was once considered to be the only option that processes sequence data. The RNN architecture is shown in Fig. 1. x Is the input, h is the hidden layer unit, o is the output, L is the loss function, and y is the label of the training set. h^t Represents the state at time t, which is determined not only by the input of x^t, but also by $h^{t-1}, \ldots$, V, U *and* W Are weights, and the same type of connection weights are the same.

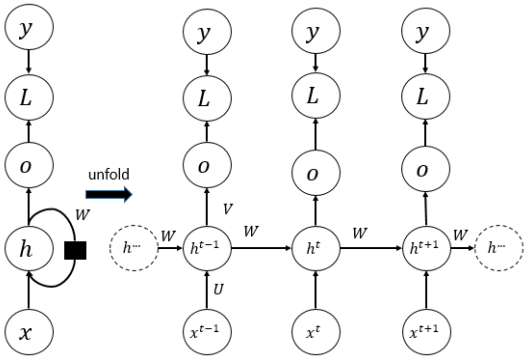


Figure 1: RNN architecture

$$h^t = tanh(Ux^t + Wh^{t-1} + b) \quad (3)$$

$$o^t = Vh^t + c \quad (4)$$

$$y^t = softmax(o^t) \quad (5)$$

$$L(\{y^0, y^1, \ldots, y^T\}, \{x^0, x^1, \ldots, x^T\}) = -\sum \log(p_{model}(y^t | \{x^1, \ldots, x^t\}) \quad (6)$$

The BPTT (back-propagation through time) algorithm is a common used method for training RNN. In fact, it is a BP algorithm based on time back propagation which continuously searches for a better path along the negative gradient direction until module convergence. The partial derivatives of W and U at time t are as follows:

$$\frac{\partial L^{(t)}}{\partial W} = \sum_{k=0}^{t} \frac{\partial L^t}{\partial o^t} \frac{\partial yo^t}{\partial h^t} \left(\prod_{j=k+1}^{t} \frac{\partial h^j}{\partial h^{j-1}} \right) \frac{\partial h^k}{\partial W} \quad (7)$$

$$\frac{\partial L^{(t)}}{\partial U} = \sum_{k=0}^{t} \frac{\partial L^t}{\partial o^t} \frac{\partial yo^t}{\partial h^t} \left(\prod_{j=k+1}^{t} \frac{\partial h^j}{\partial h^{j-1}} \right) \frac{\partial h^k}{\partial U} \quad (8)$$

Middle part of the formula：

$$\prod_{j=k+1}^{t} \frac{\partial h^j}{\partial h^{j-1}} = \prod_{j=k+1}^{t} tanh' . W_s \quad (9)$$

$$\prod_{j=k+1}^{t} \frac{\partial h^j}{\partial h^{j-1}} = \prod_{j=k+1}^{t} sigmod' . U_s \quad (10)$$

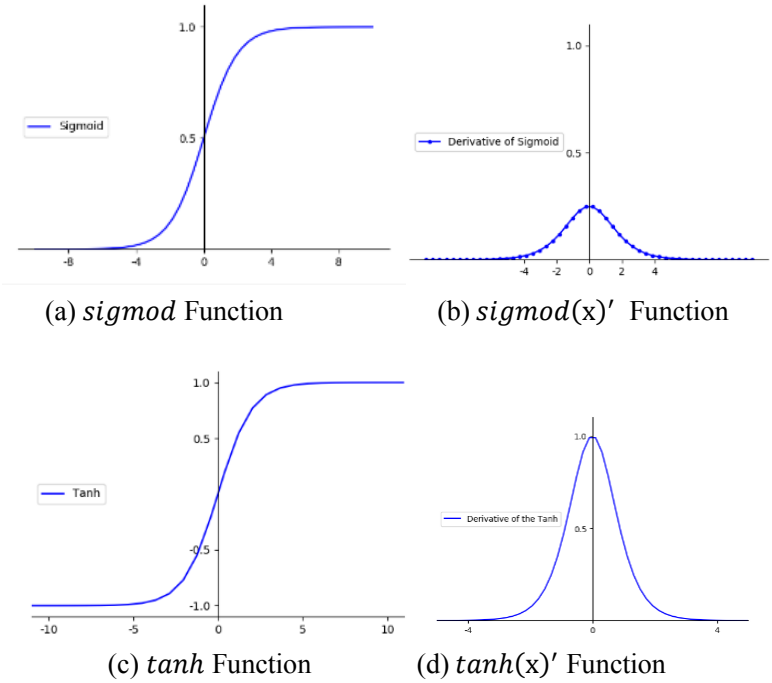
A major issue of RNN is the problem of gradient vanishing, which is caused by its architecture. The activation function of RNN is generally the $sigmod$ function or the $tanh$ function, reference formula (11, 12), the function graph is shown in Figs. 2(a), 2(b). In the back-propagation gradient calculation, it can be seen from formula (9, 10) that $\frac{\partial L^{(t)}}{\partial W}, \frac{\partial L^{(t)}}{\partial U}$ is the multiplication of the derivatives of $sigmod$ or $tanh$ over time series.

$$sigmod(\mathrm{x}) = \frac{1}{1+e^{-x}} \tag{11}$$

$$tanh(\mathrm{x}) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{12}$$

$$sigmod(\mathrm{x})' = sigmod(\mathrm{x})(1 - sigmod(\mathrm{x})) \tag{13}$$

$$tanh(\mathrm{x})' = 1 - (tanh(\mathrm{x}))^2 \tag{14}$$



(a) $sigmod$ Function (b) $sigmod(\mathrm{x})'$ Function

(c) $tanh$ Function (d) $tanh(\mathrm{x})'$ Function

Figure 2: Activation function and its derivatives

It can be seen from Figs. 2(c) and 2(d) that the range of the function derivative range of the $sigmod$ function is (0, 0.25), the derivative range of the $tanh$ function is (0, 1]. The product of multiple derivatives multiplied is getting smaller and smaller until it is close to zero, which is the phenomenon of "gradient disappearance", the calculation process is as follows:

$$.Because: |sigmoid(\mathrm{x})'| < 0.25 \ so: \prod_{j=k+1}^{t} \frac{\partial h^{j}}{\partial h^{j-1}} = \prod_{j=k+1}^{t} sigmiod'.W_s \rightarrow 0 \ Causing \ \frac{\partial L^{(t)}}{\partial W} \rightarrow 0$$

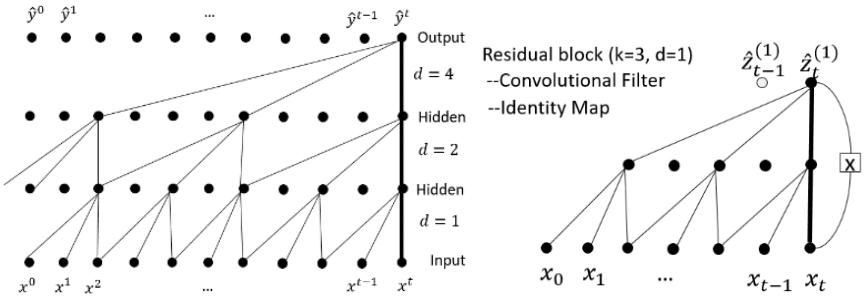
$$Because: |tanh(\mathrm{x})'| < 1 \ so: \prod_{j=k+1}^{t} tanh'.U_s \rightarrow 0 \ Causing \ \frac{\partial L^{(t)}}{\partial U} \rightarrow 0$$

The second problem RNN is that predictions for later time steps are performed sequentially, $\hat{y}^t$ depends only on $x^0, x^1, \ldots, x^t$ and not on any "future" inputs $x^{t+1}, \ldots, x^T$. The predictions for later time steps must wait for their predecessors to complete, so it cannot be done in parallel like CNN predictions.

Finally, the RNN takes up too much memory during the training process. In the case of a long input sequence, RNN can easily consume amount of memory to store the temporary and partial results. Because the backpropagation path depends not only on network depth but also on the length of sequence, RNN network requires more memory than CNN.

Jozefowicz et al. [Jozefowicz, Zaremba and Sutskever (2015)] searched through more than ten thousand different RNN architectures and evaluated their performance on various sequence modeling tasks. They concluded that if there were "architecture must better than the LSTM", then they were "not trivial to find".

Yet recent results indicate that temporal convolutional network which called TCN can outperform recurrent networks on sequence modeling task. The distinguishing characteristics of TCN are the convolutions in the architecture and map a sequence of any length to output sequence of the same length, just as with RNN. TCN architecture is shown in Fig. 3(a), a dilation causal convolution with dilation factors d=1, 2, 4 and filter size k=3. The receptive field is able to cover all values from the input sequence. Fig. 3(b) is an example of residual connection [He, Zhang, Ren et al. (2015)] in a TCN.



(a) Causal convolutional (b) Residual connection module

Figure 3: Schematic diagram of TCN network architecture

Inspired by the TCN network, the TRDD network proposed in this paper makes full use of the dilated convolution and residual modules in the network architecture. The dilated convolution expands the size of receptive field, and the residual network enhances semantic information of sequence features.

3 TRDD network for Scene text recognition

The network architecture of TRDD, as shown in Fig. 4, mainly consists two parts, the features extraction layers and transform layer. The features extraction layers use a dilated convolution and residual network to extract robust sequence features which is consistent with the order of the text in image. The transform layer translates the pre-frame predictions by the features extraction layers into a label sequence. The TRDD absorbs the design idea of TCN in the sequence modeling task, abandons the RNN network, and fuses the dilated convolution and residual module in the network, and archives large improvements.

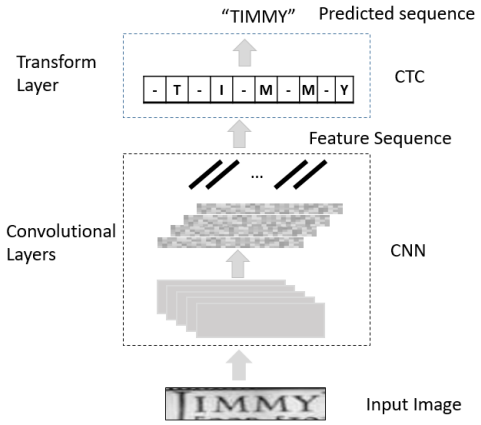


Figure 4: TRDD Model pipeline

3.1 Features extraction layers

The traditional text recognition aims at taking a cropped image of a single word and recognizing the word depict, but they can't be applied in function to scene text recognition due to the variable foreground and background texture. Scene text is no longer segmented by single characters, and features are extracted directly from text images to form sequence features. Assuming that $(x^0, \ldots, x^t, \ldots, x^T)$ are feature vectors extracted from text images through CNN. From CNN receptive field analysis, the receptive field size of the sequence feature corresponds to a range of the input text image. As shown in Fig. 5.

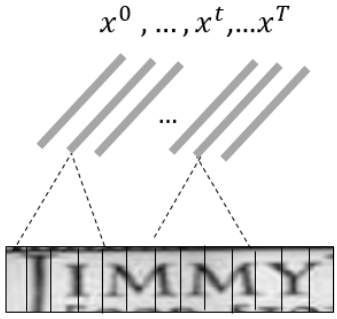


Figure 5: Receptive field of the sequence feature

RNN is one of approach to increase the size of the receptive field of sequence features. In this paper, we present a new module that uses dilated convolutions to extract sequence features from input text image, which is based on the fact that dilated convolutions support exponential expansion of the receptive field without loss of the resolution or coverage. Let $F^0, F^1, \ldots, F^{n-1}$ be discrete functions and let $k_0, k_1, \ldots, k_n$ be discrete 3×3 filters.

Define the receptive field of an element q in F^{i+1} as the set of elements in F^0 that modify the value of $F^{i+1}(q)$. Suppose the size of the receptive field of q in F^{i+1} be the number of these elements. It is clear to see that the size of the receptive filed of each element in

F^{i+1} is $\left(2^{i+2}-1\right)\times(2^{i+2}-1)$ .The receptive field is a square of exponentially increasing size. As shown Fig. 6:

Set F^1 Is produced by from F^0 by a one-dilated convolution, each element in F^1 has a receptive field of 3×3;

Set F^2 Is produced by from F^1 by a two-dilated convolution, each element in F^2 has a receptive field of 7×7;

Set F^3 Is produced by from F^2 by a four-dilated convolution, each element in F^3 has a receptive field of 15×15.

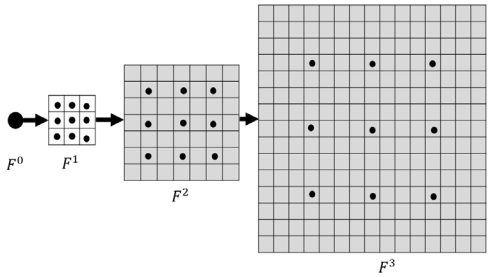
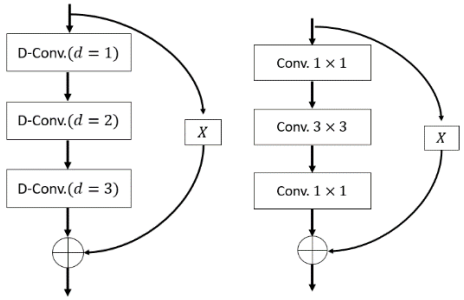


Figure 6: Dilation supports exponential expansion of receptive field

In the TRDD model, we used two basic unit modules to extract sequence features: (1) a residual module consists of three dilated convolution with a dilation factor $d = 1,2,4$, which call "A-Module" as shown in Fig. 7(a), (2) a residual module consists of three convolutions with filter sizes $k = 1\times1,3\times3,1\times1$, which call "B-Module", as shown in Fig. 7(b).



(a) A-Module (b) B-Model

Figure 7: Two basic modules of TRDD

The network architecture is shown in Fig. 8. Before text images being fed into the network, they need to be scaled to the same height. The text image is a color image with a height of 32. The feature of image is extracted through two paths, one based on "A-Module" and the other path is on "B-Module". The features are represented by $C\times W\times H$ $(W>H)$, C is the number of channels of feature map, W is the width of the feature map and H is the height of the feature map. After several pooling operations, the height H of the feature map is converted to $1(H=1)$, and the three-dimensional matrix $C\times$

$W \times H$ is converted to the two-dimensional matrix $C \times W$, which is the sequence features of the text image. For example, for a color text image with a height of 32 and a width of 280, the matrix of sequence features vector extracted by features Extraction layers is 36×512. It should be noted that each feature vector of a feature sequence is produced from left to right on feature maps by column, each column of the feature maps corresponds to a text range of input image, which termed the receptive field. Experiments show that the size of receptive fields for extracting sequence features by this method is large, and the width is generally larger than half of the image width.

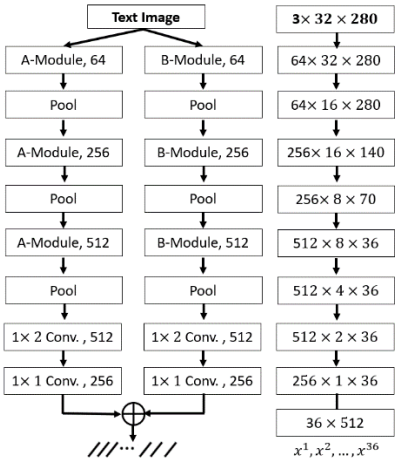


Figure 8: Two-branch feature extraction

3.2 Transform layer

Transform layer transforms the sequence features($X = (x^0, \ldots, x^T)$) extracted from the text image into a sequence of label set ($Z = (z^0, \ldots, z^T)$), including Chinese characters, punctuation, English characters, numbers, spaces and all other characters. This conversion process is shown in Fig. 9, predictions are made by select the label sequence that has the highest probability.

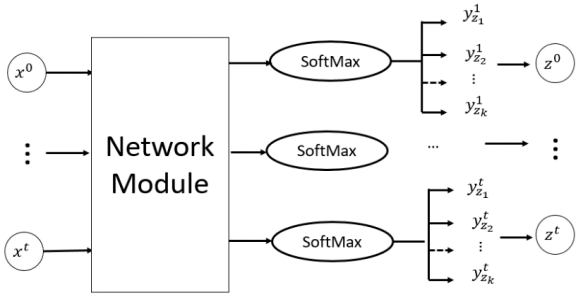


Figure 9: Conditional probability of sequence features

3.3 Calculation of the loss function

We utilize conditional probability define in Connectionist temporal classification (CTC) [Graves, Santiago and Gomez (2006); Graves (2008)] to calculate loss in the training phase. TRDD can be training with the maximum likelihood estimation of the probability as the objective function.

The input sequence $x = x_1, \ldots, x_T$ where T is the sequence length. $(y_{\pi_1}^1, y_{\pi_2}^2 \ldots y_{\pi_T}^T)$ Is a probability distribution over the set $S' = S \cup \{blank\}$ where S contains all labels in the recognition task, as well as a 'blank' label or no label. Since the probabilities of the labels at each time step are conditionally independent given x^t, the conditional probability of a path $\pi \in S'$ is given by:

$$p(\pi|y) = \prod_{t=1}^{T} y_{\pi(t)}^t \tag{15}$$

where y_π^t is the activation of output unit k at time t.

Paths of model output are mapped onto labelling $l \in S' \leq T$ by an operator F that removes first the repeated labels, then the blanks. Assume path:$\pi^1, \pi^2, \pi^3, \pi^4$ values are as follows

$$\pi^1 = (-,\ n, n, i, h, h, a, a, -, o, o)$$

$$\pi^2 = (-,\ n, i, -, i, h, a, -, -, -, o)$$

$$\pi^3 = (n,\ n, i, i, h, h, a, a, o, o)$$

$$\pi^4 = (n,\ n, i, -, i, h, a, -, -, o)$$

Then $F(\pi^1), F(\pi^2), F(\pi^3), F(\pi^4)$ yield the labelling $l = (n, i, h, a, o)$, since the paths are mutually exclusive, the conditional probability of some labelling $l \in S'$ is sum of all paths corresponding to it, since the paths are mutually exclusive.

$$p(l|y) = \sum_{\pi \epsilon F^{-1}(l)} p(\pi|y) \tag{16}$$

where the probability of π is defined as formula (15). The different paths that are mapping into the same labelling is what allows CTC to use unsegmented data, because it means that the module only to learn the order of the labels, and not to align with the input sequence one by one. A naive calculation of $p(l|y)$ is unfeasible, since there are many paths for labelling l. For example, suppose there are 30 path mapping label sequences l and length of l is 5, there are $C_{29}^5 = 120000$ possible paths. However, $p(l|y)$ can be efficiently computed using the forward-backward propagation algorithm described in Graves et al. [Graves, Santiago and Gomez (2006)].

3.4 Network training

Denote the training dataset by $D = (I_i, l_i)$, where I_i, is training text image and l_i is the ground truth label sequence. The objective function O for CTC is to minimize the negative log probability of the conditional probability of ground truth.

$$O = -\sum_{(x,z) \in S} lnp(l_i|y_i) \tag{17}$$

where y_i is the sequence vectors produced by the features extraction layers fromI_i. The objective function is calculated directly from the input image and its ground truth label sequence, so the module can be end-to-end trained on pairs of images and sequences.

4 Experiments

In this section, we perform a mass of experiments to verify the effectiveness of TRDD from three aspects: the receptive field of sequence features, the convergence speed and prediction accuracy of the network, and the speed and accuracy of network recognition.

4.1 Receptive field analysis

The concept of receptive filed is crucial for understanding and analyzing how deep network work. Since anywhere in an input text image outside the receptive field of unit does not affect the value of that unit, it is necessary to control the size of receptive field to ensure that it covers the all relevant image region.

State-of-the-art scene text modules are basically based on CNN and RNN module, such as CRNN and DTRN. The CNN extracts the sequence features from the text image and RNN learns contextual information. From the concept of receptive field, the feature extracted by CNN already contains a big receptive field, If the receptive field range of feature can meet the needs of text recognition, the LSTM module has little role and can be removed.

Feature vectors $C_1, \ldots, C_t \ldots, C_T$ are extracted from the input image through the network module. The method of calculating the receptive field of C_t in the input image is as follows:

From left to right, the pixel values of each column of the text image are set to zero, and the variation of the feature vector C_t is calculated. The magnitude of these changes reflects the intensity of each column affecting the feature vectorC_t, thereby derives C_t size of receptive field in the input image.

The input image resolution is a 3×280×32, the extracted feature vector is $(C_1, \ldots, C_t \ldots, C_{36})$. Calculate the receptive field size of C_{10} extracted by CRNN and TRDD, The X-axis represents the width of the text image and the Y-axis represents the average response intensity. As shown the Fig. 11:

(1) The size of receptive field of sequence feature extracted by TRDD and CRNN in input image is very close in the X-axis.

(2) The receptive field sensitivity of TRDD is larger than CRNN in the Y-axis, For Asian fonts such as Chinese, Japanese or Korean recognition, the local information is more important for text recognition, representation of the sequence feature extracted by TRDD is more effective than CRNN.

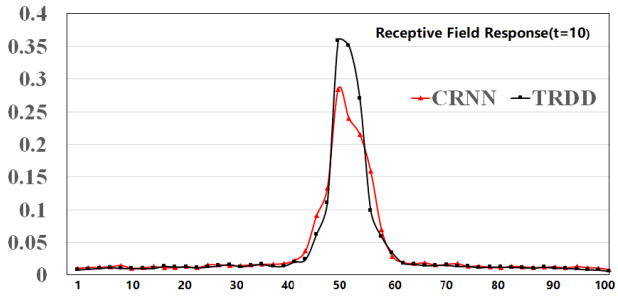


Figure 11: t=10 Respond to range on the input image

4.2 Network convergence and accuracy

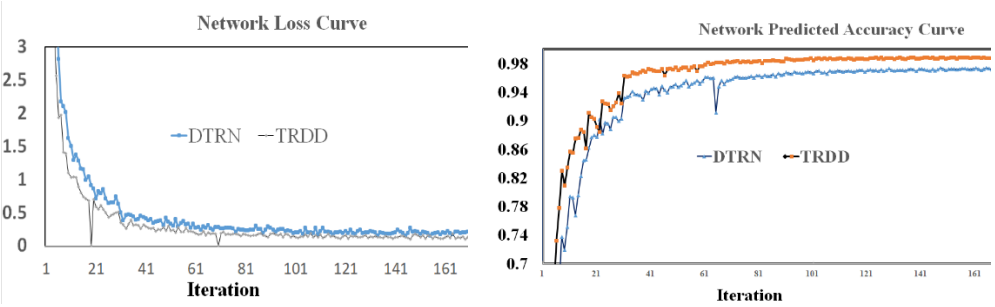
DataSets: Syn90K [Jaderberg, Simonyan, Vedaldi et al. (2016)]

Comparative model: DTRN [He, Huang, Qiao et al. (2015)]

Computer configure:

CPU: Xeon E3 1230, memory: 16G, an Nvidia 1080TI GPU graphics.

The experimental results are shown in Fig. 12. It can be seen from the Fig. 12(a), the TRDD network converges faster than the DTRN, and the network training error is reduced by 2%. As can be seen from the Fig. 12(b), the TRDD network is 3% more accurate than DTRN.



(a): Network loss curve **(b):** Network predicted accuracy curve

Figure 12: Loss curve and prediction curve during training

4.3 Network convergence and accuracy

4.3.1 Network predictive speed test

The 60,000 text images from Syn90K are used to evaluate the prediction speed, model size and prediction accuracy of the two models. The results of the comparison are shown in Tab. 1.

Table 1: Evaluation of TRDD and DTRNon synth90K

Name	Pre. Time	Mod. Size	Accuracy (%)
DTRN	9.44 ms	35.6 MB	0.976
TRDD	3.67 ms	27.5 MB	0.985

In the table, "Pre. Time" is the average prediction time of 6000, "Mod. Size" is the size of the model file, and "Accuracy" is the average test accuracy. As can be seen from the Tab. 1, Compared with DTRN network, the prediction time of TRDD network is greatly improved. The prediction time of the TRDD network is increased by 2.5 times, the model size is reduced by 27%, and the average prediction accuracy is greater than the DTRN model.

4.3.2 Network recognition accuracy experiment

DataSets

The following datasets are used in our experiments.

IIIT 5K-words (IIIT5K) [Mishra, Alahari and Jawahar (2012)] contains 5000 words patches cropped from natural scene images found by Google Image, 2000 for training and 3000 for test. Select 1000 images from the test data as test data.
Street View Text (SVT) [Babenko and Belongie (2012)] is organized from the Google Street View dataset. Selects 600 text images from this dataset.
ICDAR 2013 (IC13) [Karatzas, Shafait, Uchida et al. (2013)] has 848 cropped word patches for training and 1095 for test, Select 1000 text images from this dataset, mostly horizontal text images.
ICDAR 2015 (IC15) [Karatzas, Lu, Shafait et al. (2015)] contains 4468 patches for training and 2077 for test, Select 1800 text images as test data.
The test images were selected from the datasets IIIT5K, SVT, IC13,and IC15, and the TRDD model trained by Synth90k dataset was compared with Mishra et al. [Mishra, Alahari and Jawahar (2012)], Jaderberg et al. [Jaderberg, Simonyan, Vedaldi et al. (2016)], PhotoOCR [Bissacco, Cummins and Netzer (2013)], and CRNN [Shi, Bai and Yao (2015)] algorithms. The results are shown in Tab. 2.

Table 2: Recognition accuracies(%) on four datasets

Model	SVT	IIIT5K	IC13	IC15
Mishra	87.4	94.2	93.3	70.6
Jaderberg	80.7	93.1	90.8	80.4
PhotoOCR	78.6	96.3	90.3	90.7
CRNN	80.8	95.5	89.5	86.7
TRDD	**90.2**	**97.1**	**94.5**	**91.8**

From the experimental data on four datasets in Tab. 2, it can be seen that the recognition accuracy of TRDD is improved by 1%-2% comparing with other algorithms.

5 Conclusion

In this work we present a new TRDD network model based TCN. Comparing with the traditional sequence text recognition model, the problem of gradient vanishing and gradient exploding in the training phase can be solved due to the removal of the RNN module. Moreover, the prediction speed has a fundamental improvement comparing to other networks since it can be processed in parallel. The dilated convolution increases the receptive field range of the sequence features, and the residual network enriches the semantic expression of the sequence features. Experiments show that the convergence speed, prediction speed and network model size of the TRDD are better than other networks, especially the network prediction speed, TRDD outperforms previous state-of-the-art results in scene text recognition.

Acknowledgement: This work is supported by The National Natural Science Foundation of China (U1536121, 61370195).

References

Almazan, J.; Gordo, A.; Fornes, A.; Valveny, E. (2014): Word spotting and recognition with embedded attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2552-2566.

Babenko, B.; Belongie, S. (2012): End-to-end scene text recognition. *IEEE International Conference on Computer Vision*, pp. 1457-1464.

Bai, S.; Kolter, J. Z.; Koltun, V. (2018): An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. https://arxiv.org/abs/1803.01271.

Bissacco, A.; Cummins, M.; Netzer, Y. (2013): PhotoOCR: reading text in uncontrolled conditions. *IEEE International Conference on Computer Vision*, pp. 785-792.

Cheng, Z.; Bai, F.; Xu; Y.; Zheng, G.; Pu, S. et al. (2017): Focusing attention: towards accurate text recognition in natural images. *IEEE International Conference on Computer Vision*, pp. 5086-5094.

Dey, R.; Salemt, F. M. (2017): Gate-variants of gated recurrent unit (GRU) neural networks. *IEEE International Midwest Symposium on Circuits and Systems*, pp. 1597-1600.

Españaboquera, S.; Castrobleda, M. J.; Gorbemoya, J.; Zamoramartinez, F. (2011): Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 4, pp. 767-779.

Gers, F. A.; Schmidhuber, E. (2001): LSTM recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333-1340.

Graves, A. (2008). Offline arabic handwriting recognition with multidimensional recurrent neural networks. *Advances in Neural Information Processing Systems*, pp. 549-559.

Graves, A.; Santiago F.; Gomez, F. (2006)： Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. *IEEE International Conference on Machine Learning*, pp. 369-376.

He, P.; Huang, W.; Qiao, Y.; Loy, C.; Tang, X. (2015): Text attention convolutional neural networks for scene detection. *A Publication of the IEEE Signal Processing Society*, vol. 25, no. 6, pp. 2529-2545.

He, K.; Zhang, X.; Ren, S.; Sun, J. (2015): Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, vol. 9, no. 8, pp. 1735-1780.

Huang, W.; Lin, Z.; Yang, J.; Wang, J. (2013): Text localization in natural images using stroke feature transform and text covariance descriptors. *IEEE International Conference on Computer Vision*, pp. 1241-1248.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. (2016): Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, vol. 116, no. 1, pp. 1-20.

Jaderberg, M.; Vedaldi, A.; Zisserman, A. (2014): Deep features for text spotting. *European Conference of Computer Vision*, pp. 512-518.

Jozefowicz, R.; Zaremba, W.; Sutskever, I. (2015): An empirical exploration of recurrent network architectures. *Proceedings of International Conference on Machine Learning*, pp. 2342-2350.

Karatzas, D.; Lu, S.; Shafait, F.; Uchida, S.; Valveny, E. et al. (2015): ICDAR 2015 competition on robust reading. *International Conference on Document Analysis and Recognition*, pp. 1156-1160.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; Lluis, G. I. B. et al. (2013): ICDAR 2013 robust reading competition. *International Conference on Document Analysis and Recognition*, pp. 1484-1493.

Martens, J.; Sutskever, I. (2011): Learning recurrent neural networks with hessian-free optimization. *Proceedings of the 28th International Conference on Machine Learning*, pp. 1033-1040.

Mishra, A.; Alahari, K.; Jawahar, C. V. (2012): Scene text recognition using higher order language priors. *Proceeding of the British Machine Vision Conference*, pp. 1-11.

Novikova, T.; Barinova, O.; Kohli, P.; Lempitsky, V. (2012): Large-lexicon attribute-consistent text recognition in natural images. *Proceedings of the European Conference on Computer Vision*, pp. 752-765.

Shi, B.; Bai, X.; Yao, C. (2015): An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298-2304.

Yin, X. C.; Yin, X.; Huang, K.; Hao, H. W. (2014): Robust text detection in natural scene images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 970-983.

Xiong, Z. Y.; Shen, Q. Q.; Wang, Y. J.; Zhu. C. Y. (2018): Paragraph vector representation based on word to vector and CNN learning. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 213-227.