Recognition of Text Segments in Complex Scenes (Postprint)

Authors: Wang Xiaonan, Zhang Li, He Sinan

Date: 2018-06-19T00:00:00+00:00

Abstract

This work investigates the problem of inaccurate segmentation of text segment images under complex backgrounds or character adhesion, and proposes a recognition method for text segments in complex scenarios. The method leverages the correlation between images and text sequences to design a bidirectional recurrent neural network for encoding image feature sequences, followed by an integrated Connectionist Temporal Classification (CTC) and attention module for decoding the encoded features to produce output. The algorithm is evaluated on multiple datasets (public datasets ICDAR2013 and ICDAR2003, as well as a CAPTCHA dataset), achieving recognition accuracies of 90.2%, 87.4%, and 92.5% respectively, thereby validating the effectiveness of the proposed algorithm. The experimental results are of significant importance for text segment recognition and its applications.

Full Text

Preamble

Text Segment Recognition in Complex Scenes

Wang Xiaonan, Zhang Li, He Sinan
(Department of Electronic Engineering, Tsinghua University, Beijing 100084, China)

Abstract: This paper addresses the challenge of accurately segmenting text segment images with complex backgrounds or character adhesion, proposing a novel method for complex scene text segment recognition. The method leverages the correlation between images and text sequences by designing a bidirectional recurrent neural network to encode image feature sequences, followed by an integrated Connectionist Temporal Classification (CTC) and attention module for decoding the encoded features. The algorithm was evaluated on multiple datasets (public datasets ICDAR2013 and ICDAR2003, as well as a CAPTCHA

dataset), achieving recognition accuracies of 90.2%, 87.4%, and 92.5% respectively, thereby demonstrating its effectiveness. The experimental results hold significant importance for text segment recognition and its practical applications.

Keywords: text segment recognition; CTC; attention; integration

0 Introduction

Text recognition refers to the process of extracting and identifying text characters from an image containing textual content, converting them into a character string [1]. Typically, this problem can be divided into two stages: a text detection stage and a text segment recognition stage. Text detection identifies text regions in an image and bounds them with rectangular boxes, while text recognition reads the text segment images and identifies their content.

Traditional approaches can incorporate language models using optimization methods such as Bayesian inference [7–9], Markov random fields [10,11], conditional random fields [12,13], and probabilistic graphical models [14,15]. However, the overall architecture that introduces language models is not end-to-end, requiring separate training of individual modules. Jaderberg et al. [16] achieved image serialization and sequence feature extraction by cascading convolutional neural network features with recurrent neural networks. This architecture replaced the strong linguistic correlation constraints of language models with recurrent neural networks based on image sequences, thereby enabling end-to-end training. Lee et al. [17] introduced an attention-based encoder-decoder model on this foundation, allowing the model to select specific sequences from image feature sequences for decoding. Shi et al. [18] introduced the CTC module from speech recognition, which calculates the probability of ground-truth sequences appearing through dynamic programming and trains by maximizing the log-loss function corresponding to this probability. Breuel [19] improved the convolutional-recurrent framework by adding geometric normalization to the input, further enhancing the overall recognition performance of the network. Bolan et al. [20] added Histogram of Oriented Gradients (HOG) features to the input layer of the convolutional module, adding several specific feature maps to the input channels, which also yielded some performance improvements.

This paper proposes to recognize text segments by integrating CTC and attention mechanisms. The attention decoding mechanism can fully utilize encoded feature sequences for feature selection, while CTC achieves alignment between predictions and ground truth through probability calculation. By having both networks share a common encoding module and then performing weighted accumulation of loss functions on the decoding layers, module integration is achieved. Experimental results demonstrate that the integrated network can improve text segment recognition accuracy, while the visualization of attention weights reveals the internal mechanism of text segment recognition.

1 Joint CTC-Attention Mechanism

1.1 Attention Encoder-Decoder Model

The encoder-decoder model encodes input feature sequences through a bottom-layer RNN and then decodes the output through a top-layer RNN decoding network. The attention-based encoder-decoder model can capture specific parts of the input sequence that correspond to output labels. There are two types of attention mechanisms: hard attention and soft attention. Hard attention selects a specific region through weights at particular time steps, resulting in discontinuous loss functions that make network training difficult. Soft attention takes a weighted average of all regions at specific time steps, making it more suitable for end-to-end training. This paper adopts the soft attention mechanism.

We define two RNN modules as the encoding and decoding modules: a bidirectional RNN serves as the encoding module to encode the input image feature sequence, and a double-layer RNN serves as the decoding module to generate or decode the output sequence. The hidden states of the encoding layer are defined as $(h_1, h_2, \dots, h_{T_A})$, and the hidden states of the decoding layer are defined as $(d_1, d_2, \dots, d_{T_B}) \coloneqq (h_{T_A+1}, h_{T_A+2}, \dots, h_{T_A+T_B})$. At each time step t, the attention vector is calculated based on the encoding hidden state vectors:

$$u_i^t = v^T \tanh(W_1 h_i + W_2 d_t)$$

where v and weight matrices W_1 and W_2 are trainable parameters in the model. The vector u_t has length T_A, where the i-th element corresponds to the score value for focusing attention on the i-th encoding hidden state h_i. These score values are normalized through a softmax function to generate attention masks based on the encoding hidden state vectors:

$$a_i^t = \mathrm{softmax}(u_i^t) = \frac{\exp(u_i^t)}{\sum_j \exp(u_j^t)}$$

Finally, d_t' is obtained by weighting the encoding hidden states:

$$d_t' = \sum_i a_i^t h_i$$

The concatenation of d_t' and d_t serves as the new hidden state input to the model at the next time step and as the output prediction vector for the current moment. For input sequence x and ground-truth sequence l, the corresponding loss function is:

$$loss_{attention} = -\ln P(l|x) = -\sum \ln P(l_u|x, l_{1:u-1})$$

where $l_{1:u-1}$ represents all characters before the current ground-truth label.

1.2 CTC Sequence Alignment

CTC introduces specific mapping rules to calculate the generation probability of label sequences in the mapping space through inverse mapping, thereby achieving automatic alignment between feature sequences and label sequences. CTC ignores the specific positions of each label in the label sequence and instead calculates the posterior probability of the entire label sequence l in the prediction results $y = (y_1, y_2, \dots, y_T)$. When using the negative logarithm of this probability value as the training objective, only the input image and the corresponding label sequence are needed, avoiding the need to label the position of each character.

The principle of CTC' s conditional probability calculation is as follows: Input the label sequence $y = (y_1, y_2, \dots, y_T)$, where T corresponds to the sequence length and $y_t \in \mathbb{R}^{|I'|}$ ($I' = I \cup \text{blank}$, where I contains all character labels and blank is an additional space label). A sequence mapping function β is defined on $\pi \in {I'}^T$, where β includes two operations: first removing repeated characters, then removing space characters. The function β maps π to l. For example, β maps "-hh-e-l-ll-o-o" to "hello" . The conditional posterior probability is defined as the sum of posterior probabilities of all π mapped to l through β:

$$p(l|y) = \sum_{\pi:\beta(\pi)=l} p(\pi|y)$$

where the posterior probability of π appearing is defined as $p(\pi|y) = \prod_{t=1}^{T} y_{\pi_t}^t$, with $y_{\pi_t}^t$ being the probability of label π_t at time stamp t. Direct calculation of equation (5) has high time complexity, which can be effectively reduced through dynamic programming methods that compute forward-backward vectors.

For input sequence x and ground-truth sequence l, the corresponding loss function is:

$$loss_{ctc} = -\ln P(l|x) = -\ln P(y|l)$$

1.3 Joint CTC-Attention

Joint CTC-Attention integrates the attention module and CTC module, where CTC and attention share a common encoding RNN module. The network structure is shown in [Figure 1: see original paper]. This integrated approach can not only utilize all feature information before the current position through the encoder-decoder mechanism in the attention architecture but also leverage feature information after the current position through CTC's method of calculating global probabilities, thereby making full use of feature information. Simultaneously, CTC-Attention accelerates network convergence speed and improves recognition performance by incorporating the CTC module into the attention

module. The visualization of attention vectors can reveal the internal mechanism of network text segment recognition.

The network' s loss function is designed as a weighted accumulation of the loss functions of the two modules:

$$loss_{all} = (1 - \alpha) * loss_{attention} + \alpha * loss_{ctc}$$

where α is a learnable parameter with range $0 \leq \alpha \leq 1$.

2 Experimental Design

2.1 Datasets

This paper conducts algorithm experiments on three datasets: ICDAR2003 dataset, ICDAR2013 dataset, and YZM dataset, using the Synth90k dataset as the training set for the first two.

ICDAR2003 (IC03): A street text recognition dataset, where the test set includes 251 panoramic images and 860 cropped field images.
ICDAR2013 (IC13): A street text recognition dataset, where the test set includes 1010 cropped field images, with most images sourced from the ICDAR2003 dataset.
YZM: A CAPTCHA dataset, where the training set contains 20,000 CAPTCHA images and the test set contains 1,000 test CAPTCHA images. The image content includes digits and English letters with relatively blurred backgrounds.
Synth90k: The dataset contains 9 million synthetic cropped field images. This synthetic dataset consists of highly realistic text segment images that can be used as a training set for text segment recognition.

2.2 Implementation Details

The experiments adopt the network structure design shown in [Figure 1: see original paper], with the CNN portion using the architecture shown in [Figure 2: see original paper]. The encoding layer is a bidirectional LSTM with 256 hidden layer nodes per LSTM unit. The decoding layer is an integration of two modules: a CTC module and an attention-based double-layer LSTM decoding module, with the hidden layer nodes of the decoding LSTM units both set to 128. The attention vector calculation method follows equation (1).

Due to the diversity of input image sizes, images are first normalized. The aspect ratio is maintained while resizing images to a height of 32 and width W. The resized images, when input to the CNN, output feature maps with length exactly 1, which can be directly serialized and input to the upper modules. Based on the size of W, images are assigned to specified intervals using corresponding decoding lengths. The interval list is: [64,108], [108,140], [140,256], [256,W_{MAX}]

(where W_{MAX} represents the maximum image width), with corresponding decoding lengths of 11, 17, 19, 22, and 32 for each interval. During testing, the same strategy is adopted, with zero-padding for $W < 64$ and scaling to width W_{MAX} for $W > W_{MAX}$.

Network training employs stochastic gradient descent, with gradients for each layer computed through backpropagation. The RNN portion uses Backpropagation Through Time (BPTT) [21] to compute corresponding differential errors. The CTC portion uses the forward-backward algorithm [22] to compute differential errors. Network parameters are initialized using truncated Gaussian initialization with mean 0.0 and truncated standard deviation 0.05. The network optimization method uses the Adam algorithm with an initial learning rate of 0.001, beta1 of 0.9, and beta2 of 0.999. The Adam algorithm can adaptively compute the learning rate at each time step.

The evaluation metric is accuracy, calculated as:

$$accuracy = \frac{M}{N}$$

where M represents the number of text segment images recognized completely correctly in the dataset, and N represents the total number of samples in the dataset.

2.3 Experimental Results

Experiments are conducted on three datasets: IC03, IC13, and YZM. The first two datasets contain natural scene text segment images, while the last contains artificial CAPTCHA images, all characterized by complex backgrounds and blurred distinctions between text foreground and background. For parameter α, we select values of 0.2, 0.5, and 0.7 for experimental comparison, with recognition accuracies shown in .

Text Segment Dataset Experimental Results (Metric: Accuracy, Unit: %)

Method	IC03	IC13	YZM
Attention	87.1	84.3	90.2
CTC-Attention ($\alpha = 0.2$)	**90.2**	**87.4**	**92.5**
CTC-Attention ($\alpha = 0.5$)	89.5	86.8	91.7
CTC-Attention ($\alpha = 0.7$)	88.9	86.1	91.2
CTC	88.3	85.5	91.0

As shown in , integrating CTC and attention at the decoding stage can improve the overall accuracy of text segment recognition, with the greatest improvement when α is 0.2, achieving the best results on all three datasets. As α increases, the overall recognition accuracy shows a downward trend.

The following table compares our algorithm with current mainstream algorithms on the two public datasets ICDAR2003 and ICDAR2013, with results shown in .

Text Segment Recognition Results on IC03 and IC13 (Metric: Accuracy, Unit: %)

Method	IC03	IC13
CRNN(CTC) [18]	86.0	82.0
Literature [16]	87.0	84.0
PhotoOCR [7]	88.2	85.0
Literature [17] (Attention)	87.1	84.3
CTC-Attention ($\alpha = 0.2$)	**90.2**	**87.4**

As can be seen, our algorithm outperforms some mainstream algorithms on these two public datasets.

To evaluate algorithm complexity and model convergence speed, the test accuracy on the IC13 test set is output every training batch during training on the Synth90k dataset. The experiment selects the model with parameter α of 0.2 and compares the test results with CTC and attention methods, with results shown in [Figure 3: see original paper].

[Figure 3: see original paper] Model Training Curve Comparison (Horizontal axis: training batches, Vertical axis: IC13 test accuracy)

As shown in [Figure 3: see original paper], under converged conditions, the attention method performs better than CTC on the test set. However, the attention method suffers from high training time complexity and slow convergence. The CTC method converges relatively faster during training but has lower test accuracy. CTC-Attention accelerates the training speed of attention by integrating the CTC module into the attention decoding layer, while also improving overall test performance, achieving better performance on the test set than both methods at convergence.

2.4 Visualization Analysis

To better reveal the network's working principle, we analyze the network's mechanism by visualizing the weights of the attention decoding layer. In equation (2), a_i^t represents the weight of the i-th encoding vector in the encoded feature sequence acting on the t-th decoding vector. By visualizing the magnitude of a_t at each effective time step, we can analyze which regions play a major role for the current decoding time step.

Based on this analysis, the feature sequence obtained after CNN feature extraction and bidirectional LSTM encoding of the input image is passed through the attention layer, which performs weighted accumulation of the encoded feature

sequence before feeding it into the decoding LSTM. By visualizing the weight vectors, we can observe which feature regions play a major role in the above process for specific decoding positions. In [Figure 4: see original paper], when the encoded feature sequence of the corresponding region contributes more, the corresponding pixel values in the image are higher, and the region appears brighter. The visualization results reveal the internal mechanism of text segment recognition: for each character in the recognition result, it is precisely the image region at its corresponding position that plays the major role. Attention demonstrates "attention" to specific label characters by assigning different weights to different regions.

[Figure 4: see original paper] Attention Weight Visualization

3 Conclusion

Traditional text segment recognition methods first perform segmentation and then combine language models for post-processing, which causes error accumulation at each step and affects overall recognition accuracy. This paper analyzes the necessity of designing end-to-end networks for text segment recognition. For cases where characters are adhered to each other or text images have complex backgrounds, traditional character segmentation methods have high error rates, and adopting holistic end-to-end recognition can better solve this difficult segmentation problem. This paper analyzes the principles, advantages, and disadvantages of current two end-to-end methods (CTC and attention) and designs an integrated network for end-to-end text segment recognition. Experimental results on multiple datasets demonstrate that by sharing the encoding layer and integrating different decoding layers, the network can improve the overall accuracy of text segment recognition. This paper also conducts visualization analysis of the network' s working mechanism for recognizing text segment images, providing intuitive understanding of the network decoding process.

References

[1] Wang Kejun. Chinese printed document recognition technology [M]. Beijing: Science Press, 2010.
[2] Wu Rui. Research and implementation of text recognition technology in natural scenes [D]. Harbin: Harbin Institute of Technology, 2010.
[3] Zhang Dangzhong. An overview of chinese character recognition technology [J]. Linguistic Writing, 1997 (2): 79-88.
[4] Ye Q, Doermann D. Text detection and recognition in imagery: a survey [J]. IEEE Trans on Pattern Analysis & Machine Intelligence, 2015, 37 (7): 1480-1500.
[5] Pan Weishen, Jin Lianwen, Feng Ziyong. Chinese character recognition based on multi-scale gradient and deep neural network [J]. Journal of Beijing University of Aeronautics and Astronautics, 2015, 41 (4): 751-756.
[6] He Xin. Research on text segmentation and text line recognition in natural

scenes [D]. Beijing: University of Chinese Academy of Sciences, 2016.
[7] Bissacco A, Cummins M, Netzer Y, et al. PhotoOCR: reading text in uncontrolled conditions [C]// Proc of IEEE International Conference on Computer Vision. 2013: 785-792.
[8] Zhang Dongqing, Chang Shihfu. A Bayesian framework for fusing multiple word knowledge models in videotext recognition [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2003: 528-533.
[9] Weinman J, Learned-Miller E. Improving recognition of novel input with similarity [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2006: 308-315.
[10] Chen Datong and Odobez J. Video text recognition using sequential monte carlo and error voting methods [J]. Pattern Recognition Letter, 2005, 26 (9): 1382-1392.
[11] Weinman J, Learned-Miller E, Hanson A. A discriminative semi-Markov model for robust scene text recognition [C]// Proc of International Conference on Pattern Recognition. 2008: 1-5.
[12] Jawahar C V. Top-down and bottom-up cues for scene text recognition [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2012: 2687-2694.
[13] Shi Cunzhao, Wang Chunheng, Xiao Baihua, et al. Scene text recognition using part-based tree-structured character detection [C]// Proc of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2013: 2961-2968.
[14] Wachenfeld S, Klein H U, Jiang Xiaoyi. Recognition of screen-rendered text [C]// Proc of International Conference on Pattern Recognition. 2006: 1086-1089.
[15] Lee S H, Kima J H. Complementary combination of holistic and component analysis for recognition of low-resolution video character images [J]. Pattern Recognition Letters, 2008, 29 (4): 383-391.
[16] Jaderberg M, Simonyan K, Vedaldi A, et al. Deep structured output learning for unconstrained text recognition [J]. Eprint Arxiv, 2015, 24 (6): 603–611.
[17] Lee C Y, Osindero S. Recursive recurrent nets with attention modeling for ocr in the wild [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2231-2239.
[18] Shi Baoguang, Bai Xiang, Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2017, 39 (11): 2298-2304.
[19] Breuel T M. High performance text recognition using a hybrid convolutional-LSTM implementation [C]// Proc of IAPR International Conference on Document Analysis and Recognition. Washington DC: IEEE Computer Society, 2017: 11-16.
[20] Su Bolan, Lu Shijian. Accurate recognition of words in scenes without character segmentation using recurrent neural network [J]. Pattern Recognition, 2016, 63: 397-405.

[21] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures [J]. Neural Netw, 2005, 18 (5): 602-610.
[22] Graves A, Gomez F. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks [C]// Proc of International Conference on Machine Learning. 2006: 369-376.

Note: Figure translations are in progress. See original paper for figures.