Journal of Artificial Intelligence and Technology, 2024, 4, 169-178
https://doi.org/10.37965/jait.2024.0473

ISTP
RESEARCH ARTICLE

Segmentation-Free Recognition Algorithm Based on Deep Learning for Handwritten Text Image

Ge Peng
School of Big Data, Baoshan University, Baoshan, Yunnan 678000, China

(Received 22 October 2023; Revised 15 February 2024; Accepted 15 February 2024; Published online 05 March 2024)

Abstract: Segmentation-based offline handwritten character recognition algorithms suffered from the segmenting difficulty of interleaving and touching in handwritten manuscripts. To tackle the problem, a segmentation-free recognition algorithm based on deep learning network is proposed in this paper. The network consists of four neural layers, including input layer for image preprocessing, convolutional neural networks (CNNs) layer for feature extraction, bidirectional long-short term network (BDLSTM) layer for sequence prediction, and connectionist temporal classification (CTC) layer for text sequence alignment and classification. Besides, a novel data processing method is performed for data length equalization. Based on this, groups of experiments, based on six typical databases, involved in evaluation indicators of character correct rate, training time cost, storage space cost, and testing time cost are carried out. The experimental results show that the proposed algorithm has better performances in accuracy and efficiency than other classical algorithms.

Keywords: deep learning; image processing; segmentation-free handwritten image recognition; sequence labeling

I. INTRODUCTION

The conventional offline optical character recognition (OCR) algorithm has been a hotspot in the field of pattern recognition. It consists of four steps, including character image preprocessing, text line segmentation, character segmentation, and character recognition. With the development of machine learning, some brilliant classification algorithms [1–6] are good enough for single character recognition. Thus, the key step for good recognition results is text segmenting in segmentation-based algorithms. Several researches were proposed to settle the character segmentation problem. The connected-component algorithm [7] has good performances in the non-torched field but cannot nicely split the torched character images. Viterbi-based algorithm [8] and the dripping segmenting algorithm [9] had done well in this task, but both were prone to unsolvable over-segmentation problems. Some other attempts have been made to address this problem by using language model-based recognition algorithms [10–13]. They tried to improve the recognition performances by integrating predefined language models into recognition algorithms. However, the predefined language models usually only considered the short-distance history characters ignoring the long term. The torched-stroke and over-segmentations still affect the performances of the existing segmentation-based OCR algorithms.

Given that, an idea of "segmentation-free" recognition, called sequence labeling as shown in Fig. 1, was given significant attention. In sequence labeling, a text line image can be directly converted to a label sequence. T. H. Su [14] proposed 36DGVS algorithm based on a hidden Markov model (HMM). It used a sliding window to extract the image features of the handwritten text line and predicted the optimal recognition result by using the HMM. Afterward, with the development of deep learning, some sequence labeling architectures [15,16] based on deep learning were used for sequence prediction. However, these algorithms assumed the pre-extraction features as network inputs which became the new problem of the sequence labeling algorithm.

Thereby, some end-to-end algorithms [17–21] were proposed to convert inputs of text line images into outputs of label sequences without the pre-extraction features. The convolutional recurrent neural network (CRNN) [19] succeeded in English text sequence labeling. However, it was prone to the over-fitting problem in some situations with a large alphabet or small databases.

Thus, a sequence labeling algorithm is proposed with the following contributions.

(a) An improved network based on the CRNN, consisting of four neural network layers, is proposed to deliver better performances and more efficiency in handwritten sequence recognition.

(b) A novel databases uniform method, based on data filling, is proposed to normalize the databases into uniform size for better model training.

(c) Handwritten recognition experiments in multiple languages are carried out in order to test the robustness network.

The rest of the paper is organized as follows. Section II presents the proposed algorithm. Section III introduces the experimental environment and databases used in this paper. Four networks based on the proposed algorithm with different parameters and experimental results are discussed in detail in section IV. Section V gives a conclusion.

II. THE PROPOSED ALGORITHM

The architecture of the proposed network consists of four neural network layers, including preprocessing, CNNs, BDLSTM, and CTC loss layers as shown in Fig. 2. The preprocessing layer is used

Corresponding author: Ge Peng (e-mail: pg_memf@126.com).

© The Author(s) 2024. This is an open access article published under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

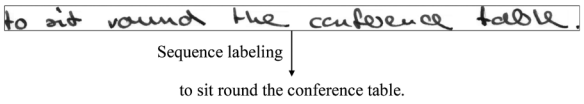

Fig. 1. Sequence labeling.

to convert the image into a normalized binary image, which will be further discussed in the experimental databases section. The designed CNNs are used for spontaneous feature extraction and image dimensionality reduction in the second layer of the proposed algorithm. The BDLSTM layer follows the CNNs to predict the sequence output based on the contextual information. The CTC loss layer is used for the alignment between sequences and labels, and the backward of the loss at the end of the network.

A. CNNs LAYER

The CNNs layer, used for feature extraction and image dimensionality reduction, includes several convolutional layers and pooling layers but excludes the fully connected layers that are used in the common CNNs to retain the structure information of the input images as completely as possible. Thus, the architecture of the CNNs layer should be designed reasonably by meeting the following two requirements.

(1) The height of CNNs layer outputs: $H = 1$.

(2) The width of CNNs layer outputs: $W \geq 5 \times$ label length.

On that foundation, the outputs of CNNs layer can fit the input dimensional requirements of the BDLSTM layer and retain the structure information at the same time. Suppose that the height of the preprocessed image is $H = 64$, the architecture of CNNs and pooling layers in this paper should be designed as shown in Fig. 3.

B. BDLSTM LAYER

The designed CNNs layer will produce a feature vector with structure semantics, which is invaluable for label prediction. Some networks are good at label prediction. Recurrent Nueral networks (RNNs) [6] do well in sequence prediction using the information about the long past inputs. Moreover, the bidirectional recurrent neural networks (BDRNNs) [22] can make good use of both past and later inputs. Based on BDRNNs, it is possible to predict the labels with a suitable feature vector produced by the CNNs layers mentioned above.

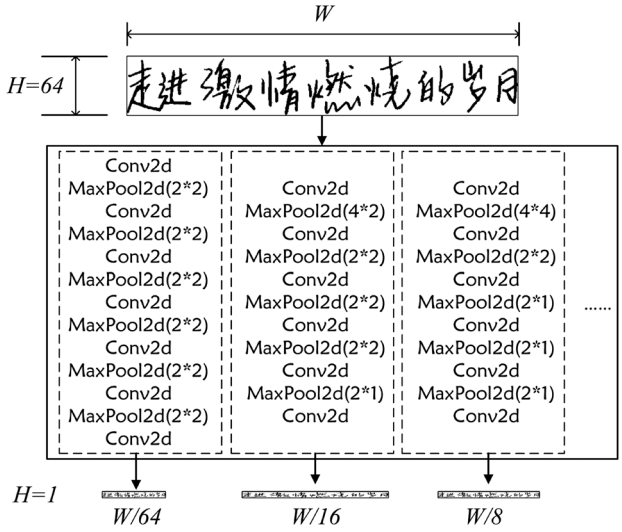

Fig. 3. Parameters setting of CNNs and pooling layers.

However, it is difficult to train the BDRNNs, and it is impossible to retain information for a long period because of the evaporation of gradient information or large increase along the time delay, called the vanishing gradient or the exploding gradient problem [23,24] as shown in Fig. 4(a) in BDRNNs network. Meanwhile, the long-short term network (LSTM) [25] is a variant of RNNs, which can tackle the problem in RNNs effectively.

The main idea of LSTM is to replace the basic unit in RNNs with a memory unit containing one or more self-connected memory cells and three gates. It is quite feasible to use the long-term information about input sequences to predict labels. The gradient transfer in RNNs and LSTM are shown in Fig. 4(b) corresponding to Fig. 4(a).

Moreover, BDLSTM network, as shown in Fig. 5, are good at using long-term information of past and later input sequences.

From then on, suppose that a neural network consists of CNNs, BDLSTM layers described above, is with a part of input image shown in Fig. 1, it will produce results of "*rroouunndd*" or "*roundd,*" or other possible outputs for the ground truth "*round*" for the reason that the output of the current network cannot align with the ground truth correctly.

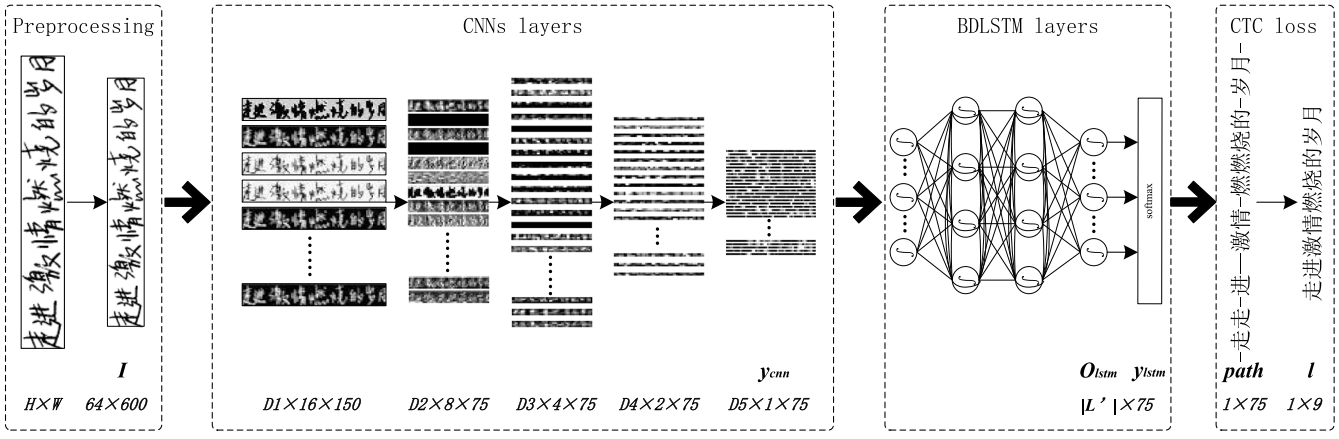

Fig. 2. Architectures of the proposed network.

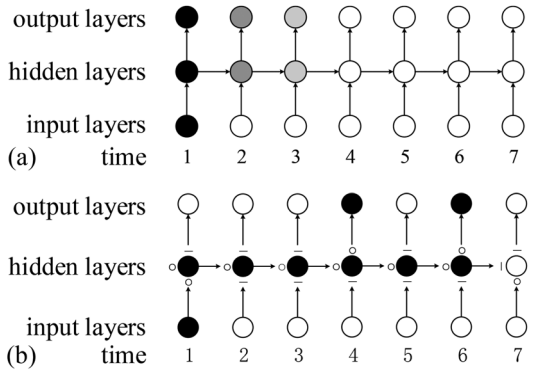

Fig. 4. Vanishing gradient problem: (a) gradient transfer of RNNs and (b) gradient transfer of LSTM.

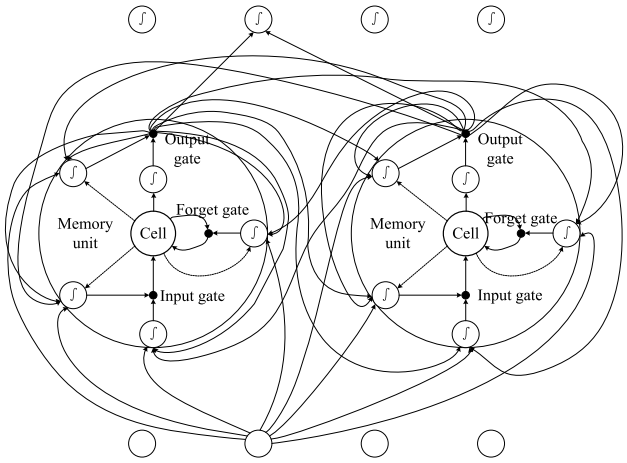

Fig. 5. BDLSTM network.

C. CTC LOSS LAYER

CTC [26] proposed by Graves for labeling the unsegmented sequences can solve the alignment problem of sequence prediction in this paper. Suppose that there is a letter set $\boldsymbol{L}$, for instance, the letter set for English letters is $\boldsymbol{L} = \{A, \ldots, Z, a, \ldots, z\}$ and length of the set is $|\boldsymbol{L}| = 52$. The letter set for the CTC algorithm should be $\boldsymbol{L}' = \boldsymbol{L} \cup \{-\}$ where "−" represents a blank character and $|\boldsymbol{L}'| = |\boldsymbol{L}| + 1$. The relationship between the inputs and labels of the CTC algorithm can be expressed by f_{ctc} as shown in formula (1):

$$\mathrm{f_{ctc}}(-\mathrm{AABB}-) = \mathrm{f_{ctc}}(-\mathrm{AA}-\mathrm{B}-) = \mathrm{AB} \tag{1}$$

From that, the inputs of the CTC algorithm will be compressed with designed rules. The blank character "−" plays a key role in the rule. It makes correct results in some situations as shown in formulas (2) and (3).

(a) If there is a "−", the result is shown in formula (2):

$$\mathrm{A} - \mathrm{A} = \mathrm{AA} \tag{2}$$

(b) If there is no "−", the result should be what is shown in formula (3):

$$\mathrm{AA} = \mathrm{A} \tag{3}$$

In this way, the outputs of the CTC layer can align with the label sequences.

D. THE FORWARD AND BACKWARD PROCESS

Suppose that there is an input image $\boldsymbol{I}$ with a size of 64×600, the label sequence corresponding to $\boldsymbol{I}$ is $\boldsymbol{l}$ and the letter set is $\boldsymbol{L}' = \boldsymbol{L} \cup \{-\}$. The forward and backward of the proposed network can be expressed as follows:

1) FORWARD.

(a) Image preprocessing. The input image $\boldsymbol{I}$ should be converted to a binary image and the height of the image should be resized into $\mathrm{H} = 64$ for the stability and efficiency of the network. By the way, we suppose that there is one grayscale image $\boldsymbol{I}$ with a size of 64×600 as the input of the CNNs in this part.

(b) CNNs layers. The CNNs layer is represented with a symbol f_{cnn}. The forward pass of CNNs layers is expressed by formula (4):

$$\mathbf{y_{cnn}} = \mathrm{f}_{\underset{\boldsymbol{\theta}_1}{\mathrm{cnn}}}(\mathbf{I}) \tag{4}$$

where $\boldsymbol{\theta}_1$ represents the parameters in CNNs layers and $\boldsymbol{y_{cnn}}$ represents the feature vector outputs as the output of the CNNs layer. The CNNs layer shown in Fig. 2 is corresponding to the third structure described in Fig. 3. Through CNNs layer, the image $\boldsymbol{I}$ will be transformed into a feature vector $\boldsymbol{y_{cnn}}$ with a size of $\mathrm{D5} \times 1 \times 75$ by the CNNs layers as shown in Fig. 2.

(c) BDLSTM layers. The BDLSTM layer is represented by f_{lstm}. The main task of the BDLSTM layers is to predict the label sequence $\boldsymbol{y_{lstm}}$ based on the feature vector $\boldsymbol{y_{cnn}}$. The forward pass of BDLSTM layers can be expressed as formulas (5) and (6):

$$\mathbf{O_{lstm}} = \mathrm{f}_{\underset{\boldsymbol{\theta}_2}{\mathrm{lstm}}}(\mathbf{y_{cnn}}) \tag{5}$$

$$\mathbf{y_{lstm}} = \mathrm{Softmax}(\mathbf{O_{lstm}}) \tag{6}$$

where $\boldsymbol{\theta}_2$ represents the parameters in BDLSTM layers and $\boldsymbol{O_{lstm}}$, with a size of $75 \times |\boldsymbol{L}'|$, determines the output of the BDLSTM networks. The output should be sent to a *Softmax* layer to obtain the posterior probability vector $\boldsymbol{y_{lstm}}$ of the label sequence prediction, with a size of $75 \times |\boldsymbol{L}'|$, as given in formula (6).

(d) CTC loss layer. The optimal label sequence prediction $\boldsymbol{l}'$ of the input image can be estimated by the posterior probability vector $\boldsymbol{y_{lstm}}$ based on CTC loss layers (represented by f_{ctc}) using formula (7):

$$\boldsymbol{l}' = \mathrm{Softmax}(\mathbf{O_{lstm}}) \tag{7}$$

2) BACKWARD.

Backpropagation through time algorithm (BPTT) by Werbos [27] is used to train the proposed networks in this paper. In addition, to finish the training of networks in an end-to-end manner and tackle the alignment problem of sequence prediction, the CTC loss is provided behind the *Softmax* loss layer. Suppose that there is a ground truth dataset composed of a group of handwritten images $\boldsymbol{I}$ labeled correctly with label sequences $\boldsymbol{l}$. The dataset is determined by $\boldsymbol{D}$ as given in formula (8):

$$\mathbf{D} = \{(\mathbf{I}_1, \mathbf{l}_1), (\mathbf{I}_2, \mathbf{l}_2), \cdots, (\mathbf{I_N}, \mathbf{l_N})\} \tag{8}$$

The CTC loss represented by ***loss*** can be calculated using formula (9):

$$\mathbf{loss} = -\ln p(\mathbf{l}|\mathbf{y}_{\mathbf{lstm}}) = -\ln \sum_{\mathbf{path}} \prod_{\mathrm{T}} \mathbf{y}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}} \tag{9}$$

Then, the loss of backpropagation to BDLSTM layers can be calculated with the chain rule based on the back propagation through time (BPTT) algorithm as shown in formula (10). Each partial derivative in formula (10) can be calculated based on the formula derivation by Graves [26]:

$$\frac{\partial \mathbf{loss}}{\partial \boldsymbol{\theta}_2} = \frac{\partial \mathbf{loss}}{\partial \mathbf{y}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}}} \frac{\partial \mathbf{y}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}}}{\partial \mathbf{O}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}}} \frac{\partial \mathbf{O}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}}}{\partial \boldsymbol{\theta}_2} \tag{10}$$

Similarly, the loss of backpropagation to CNNs layers can also be calculated with the chain rule as shown in formula (11):

$$\frac{\partial \mathbf{loss}}{\partial \boldsymbol{\theta}_1} = \frac{\partial \mathbf{loss}}{\partial \mathbf{O}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}}} \frac{\partial \mathbf{O}_{\mathbf{lstm}_{\mathbf{path}}^{\mathrm{t}}}}{\partial \mathbf{h}_{\mathrm{N}}} \frac{\partial \mathbf{h}_{\mathrm{N}}}{\partial \mathbf{h}_{\mathrm{N}-1}} \cdots \frac{\partial \mathbf{h}_1}{\partial \mathbf{y}_{\mathbf{cnn}}} \frac{\partial \mathbf{y}_{\mathbf{cnn}}}{\partial \boldsymbol{\theta}_1} \tag{11}$$

where $\boldsymbol{h}_N$ represents the parameters of the hidden layers N of BDLSTM networks. Once the iteration of network training based on one input image is done, the network will converge after some epoch of iterations in the best case.

III. EXPERIMENTAL ENVIRONMENT AND DATABASES

A. EXPERIMENTAL ENVIRONMENT

The experimental results in this paper are all based on a desktop PC with an Intel(R) Core(TM) i9-10850K CPU, 16GB memory, and an NVIDIA GeForce GTX 1080 with 8GB memory.

B. EXPERIMENTAL DATABASES

To evaluate the effectiveness of the proposed algorithm, six typical offline handwritten databases were used for the experiments, including the handwriting database by Institut für Informatik und angewandte Mathematik (IAM) [28] for English, the handwriting database by Institute of Automation Chinese Academy of Sciences (CASIA-HWDB) (including unconstrained text lines (HWDB2.0-2.2) [29] and the Harbin Institute of Technology-Multiple Writers Database (HIT-MW) [30] for Chinese, the DigitalPeter database [31] for Russian, the Rodrigo database [32] for Spanish, and the Arabic Offline Handwritten Text Database (KHATT) [33] for Arabic. All of the databases were correctly labeled. Figure 6 shows some samples from the databases, while Table I lists the properties of the six databases.

For better training and testing processing, the images and labels used in this paper were normalized as follows:

(a) *normalization of the image height.* The height of the input image H was resized to 64 pixels for better efficiency, while the information was retained as much as possible.

(b) *normalization of image width and label length*. While the image height has been normalized, the image width and label length of each individual sample are different. With this, another contribution of this paper is to propose a data processing method to normalize the database into uniform size.

Taking the IAM database as an example, the label length of the samples ranges from 1 to 94 characters. In this paper, every label is normalized to the max label length of 94 characters. The common

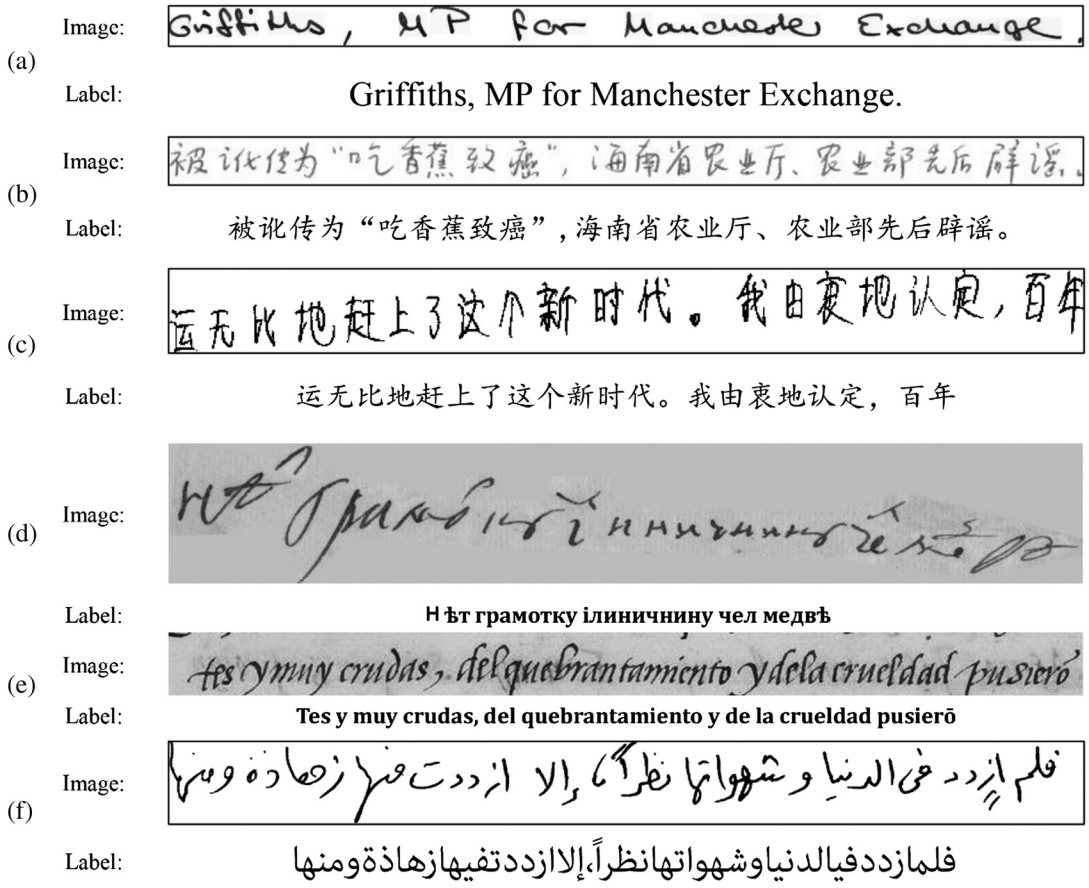

Fig. 6. Samples of the six databases. (a) IAM, (b) CASIA-HWDB, (c) HIT-MW, (d) DigitalPeter, (e) Rodrigo, and (f) KHATT.

Table I. Properties of the six offline handwritten databases

Databases		Writers	Pages	Text line images	Alphabets
IAM		657	1,539	13,353	80
CASIA-HWDB	HWDB2.0	419	2,092	20,495	2073
	HWDB2.1	300	1,500	17,292	
	HWDB2.2	300	1,488	14,443	
HIT-MW		780	853	8,673	3047
DigitalPeter		1	662	9,694	76
Rodrigo		1	853	15,010	111
KHATT		1,000	2,000	13,435	94

way for this task is padding "–". For instance, the labels of "A MOVE to stop Mr. Gaitskell from," with the length of 34, will be normalized to new labels of "A MOVE to stop Mr. Gaitskell from –" with length of 94 characters, by padding 60 "–" characters at the end of the old labels. In the more extreme cases, a label with length of 1, there are 93 "–" should be padded. It will pull in too much useless information which is disastrous for training. Therefore, the main idea of the data processing method in this paper is to fill the databases with existing samples instead of "–".

Step 1. *Collect padding samples*. Collecting some samples with a label length ranging from 1 to 51 characters, called padding samples.

Step 2. *Get the original data size*. Suppose the size of the original image $\boldsymbol{I}$ is (h,w), the length of the original label $\boldsymbol{l}$ is $|\boldsymbol{l}|$.

Step 3. *Find a suitable padding sample for the original data*. Find a padding sample in which image $\boldsymbol{I}'$ with a size of (h,w') and label $\boldsymbol{l}'$ with a length of $|\boldsymbol{l}'|$ let $|\boldsymbol{l}| + |\boldsymbol{l}'| = 52$.

Step 4. *Pad the original data*. Produce a new image with a size $(h, w + w^{'})$ by splicing the original image $\boldsymbol{I}$ with the chosen padding samples $\boldsymbol{I}'$ in the horizontal direction and the new label will be produced simply by string slicing.

Step 5. *Normalize the image*. The image is resized to a uniform size.

The data with a label length of 34 characters mentioned above can be normalized more reasonably, and the new labels of the data should be "A MOVE to stop Mr. Gaitskell from Commonwealth Relations Secretary, that they wish to be kept." The rest of the databases used in this paper were normalized in a similar manner. Moreover, the databases were divided into three parts, including 60%, 20%, and 20% as the training, validation, and testing sets, respectively.

Table II. Parameters of the proposed networks

Layers	Parameters	Size of outputs
Preprocessing	*Data preprocessing*	$16 \times 1 \times 64 \times 2133$
CNNs	*Conv(1,3*3,1,1)*	$16 \times 1 \times 64 \times 2133$
	*Conv(64,1*1,1,0)*	$16 \times 64 \times 64 \times 2133$
	*Pool(2*2,(2,2),(0,0))*	$16 \times 64 \times 32 \times 1066$
	*Conv(64,3*3,1,1)*	$16 \times 64 \times 32 \times 1066$
	*Conv(128,1*1,1,0)*	$16 \times 128 \times 32 \times 1066$
	*Pool(2*2,(2,2),(0,0))*	$16 \times 128 \times 16 \times 533$
	*Conv(128,3*3,1,1)*	$16 \times 128 \times 16 \times 533$
	*Conv(256,1*1,1,0)*	$16 \times 256 \times 16 \times 533$
	*Pool(2*2,(2,1),(0,1))*	$16 \times 256 \times 8 \times 534$
	*Conv(256,3*3,1,1)*	$16 \times 256 \times 8 \times 534$
	*Conv(256,1*1,1,0)*	$16 \times 256 \times 8 \times 534$
	*Pool(2*2,(2,1),(0,1))*	$16 \times 256 \times 4 \times 535$
	*Conv(256,3*3,1,1)*	$16 \times 256 \times 4 \times 535$
	*Conv(512,1*1,1,0)*	$16 \times 512 \times 4 \times 535$
	*Conv(512,3*3,1,1)*	$16 \times 512 \times 4 \times 535$
	*Conv(512,1*1,1,0)*	$16 \times 512 \times 4 \times 535$
	*Pool(2*2,(2,1),(0,1))*	$16 \times 512 \times 2 \times 536$
	*Conv(512,2*2,1,0)*	$16 \times 512 \times 1 \times 535$
	*Conv(512,1*1,1,0)*	$16 \times 512 \times 1 \times 535$
BDLSTM	*BDLSTM(512,512,512)*	$535 \times 16 \times 1024$
	BDLSTM(512,512,nClass)	
	Softmax	$8560 \times \lvert L' \rvert$
	CTC loss	832×1

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

This section explains and discusses the details of the proposed network implementation. The parameters and the sizes of the output of each layer of the proposed network are described in Table II where batch size is 16 for network training.

First, the preprocessing layer will transform the original image into a binary image with a size of 64×2133 and slice the original labels to the max label length. The size of outputs would be $16 \times 1 \times 64 \times 2133$ while the $batchsize = 16$.

Then the batches from the above layers will be processed into a feature vector $\boldsymbol{y_{cnn}}$ with the size of $16 \times 512 \times 1 \times 513$ by CNNs layers. Conv and Pool mentioned in Table I represent convolutional and max pooling layers, respectively, where $Conv(1,3*3,1,1)$ represents a convolutional layer with a convolutional kernel of $3*3$, the stride of 1, and the zero padding dimension of 1. $Pool(2*2,(2,1),(0,1))$ represents a max pooling layer with pooling kernel of $2*2$, the stride vertical of 2, the stride horizontal of 1, the vertical zero padding dimension of 0, and the horizontal zero padding dimension of 1.

Next, the prediction vector $\boldsymbol{O_{lstm}}$ sized $535 \times 16 \times 1024$ is produced from BDLSTM layers with an input of feature vector $\boldsymbol{y_{cnn}}$, and the posterior probability vector $\boldsymbol{y_{lstm}}$ sized $8560 \times |\boldsymbol{L}'|$,

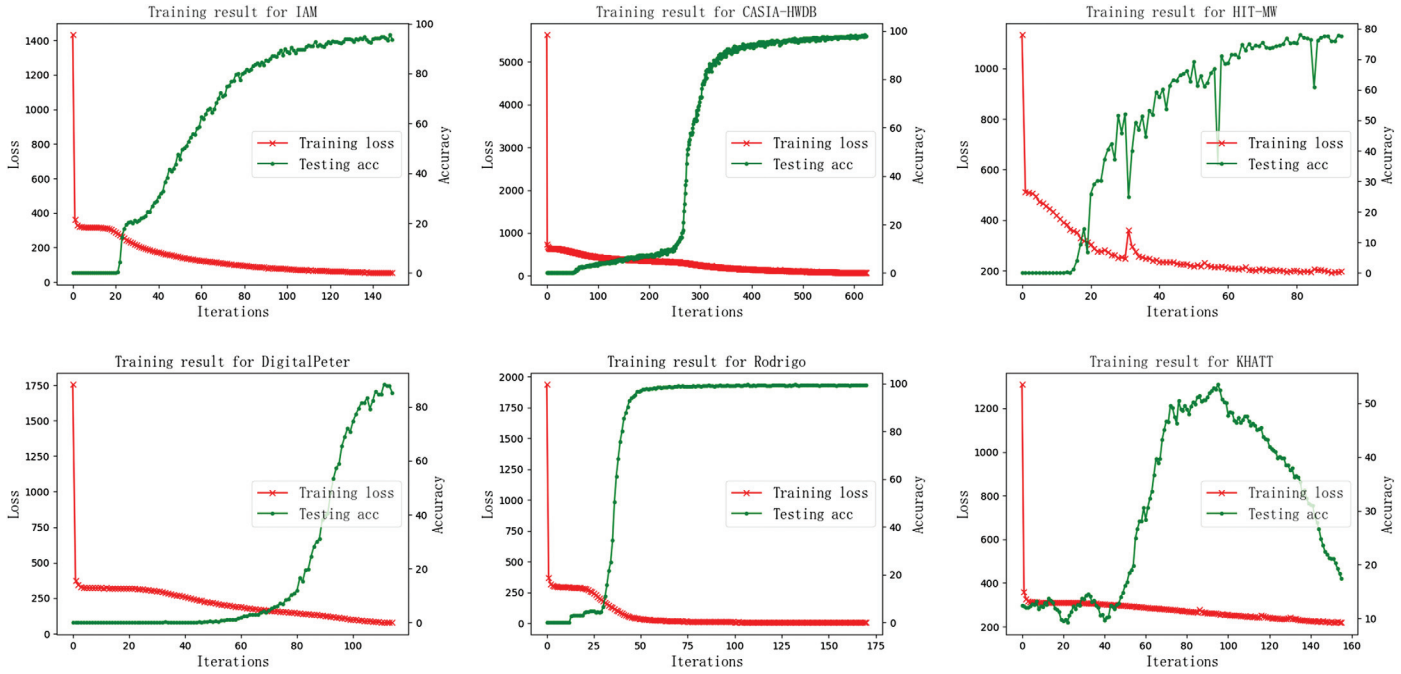

Fig. 7. Analysis of experimental results on six databases.

where $|\boldsymbol{L}'|$ represents the length of Chinese letters set in the databases and can be obtained through a *Softmax* layer based on $\boldsymbol{O}_{\boldsymbol{lstm}}$. $BDLSTM(512, 512, 512)$ represents a BDLSTM network with 512, 1024, and 1024 units in the input, hidden, and output layers because of its directionality, respectively.

Finally, the CTC layers would produce a label prediction sized 832×1, where 832 is the product of the label length $|\boldsymbol{l}|$ and *batchsize* 16.

The experimental results of training loss and testing accuracy of character correct rates (CCRs) on the six databases are shown in Fig. 7 and Table III. After 20 epochs training, the proposed algorithm obtains good performances on all databases except for the KHATT database which should be caused by the unique typography of the Arabic. Moreover, the differences in the converge speeds in Fig. 8 are produced by the variant sizes for different databases.

Furthermore, three additional variant networks, namely Proposal 2, Proposal 3, and Proposal 4, are presented as described in Table IV based on the network in Table I called Proposal 1 hereinafter. Proposal 3 uses more suitable pooling kernels in CNNs layers, and Proposal 1 retains the structure information of the input images as much as possible and simultaneously shortens the time for each iteration. Proposal 2 and Proposal 4 use more complex BDLSTM layers by increasing the memory units based on Proposal 1 and Proposal 3 separately to improve the prediction ability of the network.

Figure 8 compares the experimental results of CRNN and Proposal 1, 2, 3, and 4 within 20 epochs. It can be seen that the proposed algorithm has proven more useful than the CRNN in terms of convergence time and CCR. Moreover, Proposal 2 and Proposal 4 converge earlier and make better CCR performances than the other two proposed algorithms due to the more complex BDLSTM layers.

Table V and Fig. 9 provide more detailed comparative results upon segmentation-based and segmentation-free algorithms, including the sequence labeling CCR, model size, and testing time cost per character on CPU on the six databases. Generally, the segmentation-based algorithms have nearly equal performances with segmentation-free algorithms on the CASIA-HWDB and HIT-MW databases, while the character images have little interleaving and touching situation in the two databases. Meanwhile, the segmentation-based algorithms provide worse performances resulting in bad character segmenting results on IAM, DigitalPeter, and Rodrigo databases. Moreover, the average CCR on the KHATT database is 43.18% which is caused by the unique typography of Arabic whose letter has several variant shapes. However, the proposed algorithm achieves more excellent results, 88.67% CCR, and 11.43 ms character recognition time on average than others. Specifically, Proposal 1, 2, 3, and 4 achieve average CCRs of 86.23%, 91.13%, 86.87%, and 90.44%, respectively. The best CCRs for the six databases are 96.69%, 98.92%, 98.60%, 97.22%, 99.42%, and 58.02% based on the Proposal 2, 2, 4, 2, 4, and 1, respectively. Proposal 2 and Proposal 4 perform better than the others in sequence labeling accuracy.

On the side of efficiency, the 36DGVS algorithm has the worst performance with a testing time cost of 1.81612 s per character and

Table III. Character correct rates (CCRs) on the six databases

Results	IAM (%)	CASIA-HWDB (%)	HIT-MW (%)	DigitalPeter (%)	Rodrigo (%)	KHATT (%)
CCR	96.135	97.908	77.791	88.214	99.317	58.024

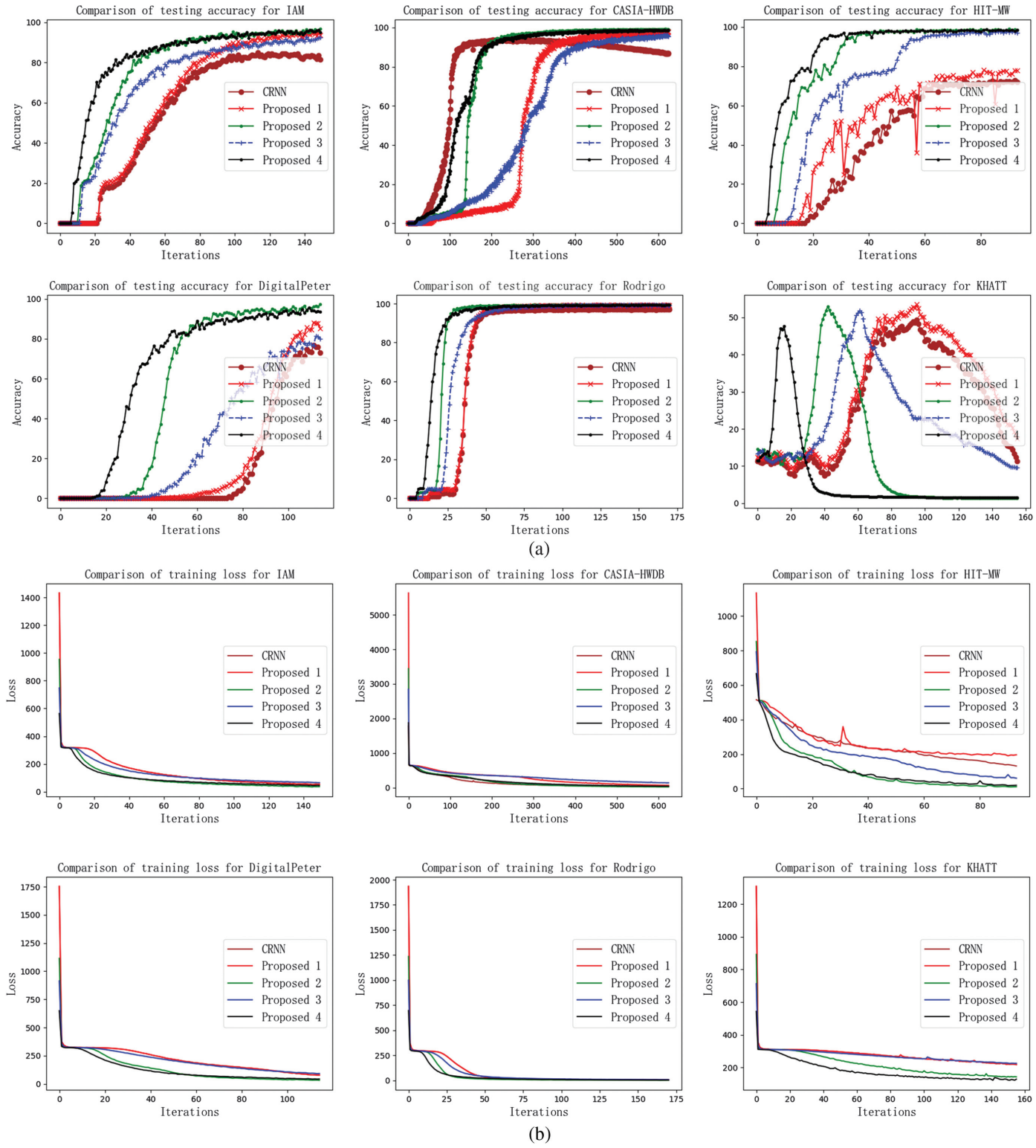

Fig. 8. Comparative analysis of experimental results for the six databases. (a) Comparison of training loss and (b) comparison of testing accuracy.

a mode size of 30.96 M. From Fig. 9, Proposal 2 and Proposal 4 occupy 161 M and 46.7 M more spaces than Proposed 1 and Proposal 2, respectively, due to the complexity in BDLSTM layers. However, it makes no significant difference in testing time. Proposal 1, 2, 3, and 4 algorithms take 10.2 ms, 13.1 ms, 10.8 ms, and 11.6 ms for one letter recognizing on CPU, respectively. Thus, Proposal 2 and Proposal 4 perform better in efficiency than the others except for the longer training time cost.

Overall, Proposal 2 and Proposal 4 prove useful in the task of sequence labeling with good accuracy and efficiency.

Table IV. Parameters of the proposed networks

Layers	Proposal 2	Proposal 3	Proposal 4
CNNs	*Conv(1,3*3,1,1)*	*Conv(1,3*3,1,1)*	*Conv(1,3*3,1,1)*
	*Conv(64,1*1,1,0)*	*Conv(64,1*1,1,0)*	*Conv(64,1*1,1,0)*
	*Pool(2*2,(2,2),(0,0))*	*Pool (4*4,(4,4),(0,0))*	*Pool (4*4,(4,4),(0,0))*
	*Conv(64,3*3,1,1)*		
	*Conv(128,1*1,1,0)*	*Conv(64,3*3,1,1)*	*Conv(64,3*3,1,1)*
	*Pool(2*2,(2,2),(0,0))*	*Conv(128,1*1,1,0)*	*Conv(128,1*1,1,0)*
	*Conv(128,3*3,1,1)*	*Conv(128,3*3,1,1)*	*Conv(128,3*3,1,1)*
	*Conv(256,1*1,1,0)*	*Conv(256,1*1,1,0)*	*Conv(256,1*1,1,0)*
	*Pool(2*2,(2,1),(0,1))*	*Pool (4*2,(4,2),(0,1))*	*Pool (4*2,(4,2),(0,1))*
	*Conv(256,3*3,1,1)*		
	*Conv(256,1*1,1,0)*	*Conv(256,3*3,1,1)*	*Conv(256,3*3,1,1)*
	*Pool(2*2,(2,1),(0,1))*	*Conv(256,1*1,1,0)*	*Conv(256,1*1,1,0)*
	*Conv(256,3*3,1,1)*	*Conv(256,3*3,1,1)*	*Conv(256,3*3,1,1)*
	*Conv(512,1*1,1,0)*	*Conv(512,1*1,1,0)*	*Conv(512,1*1,1,0)*
	*Conv(512,3*3,1,1)*	*Conv(512,3*3,1,1)*	*Conv(512,3*3,1,1)*
	*Conv(512,1*1,1,0)*	*Conv(512,1*1,1,0)*	*Conv(512,1*1,1,0)*
	*Pool(2*2,(2,1),(0,1))*	*Pool(2*2,(2,1),(0,1))*	*Pool(2*2,(2,1),(0,1))*
	*Conv(512,2*2,1,0)*	*Conv(512,2*2,1,0)*	*Conv(512,2*2,1,0)*
	*Conv(1024,1*1,1,0)*	*Conv(512,1*1,1,0)*	*Conv(1024,1*1,1,0)*
BDLSTM	*BDLSTM (1024, 1024, 1024)*	*BDLSTM (512,512,512)*	*BDLSTM (1024, 1024, 1024)*
	BDLSTM(1024,1024,nClass)	*BDLSTM(512,512,nClass)*	*BDLSTM(1024,1024,nClass)*
	Softmax		
	CTC loss		

Table V. Comparisons with state-of-the-arts

Algorithms		IAM (%)	CASIA-HWDB (%)	HIT-MW (%)	DigitalPeter (%)	Rodrigo (%)	KHATT (%)	Model size (M)	Testing time (ms/character)
Segmentation-based algorithms	A. T. Sahlol [1]	57.23	93.72	92.39	52.31	53.55	54.25	—	36.2
	Q. F. Wang [10]	58.23	92.72	91.39	54.31	56.55	30.25	—	230
	Q. F. Wang [11]	53.65	92.19	86.56	56.98	59.54	43.56	—	38
	Y. C. Wu [13]	57.22	91.80	78.68	53.67	61.23	33.66	—	65
	MQDF3-2 [14]	37.89	38.59	35.49	55.68	45.25	30.15	—	271.4
Segmentation-free algorithms	36DGVS [14]	45.67	47.67	44.33	41.26	36.63	33.67	30.96	1816.2
	Y. C. Wu [17]	66.53	95.04	92.33	58.23	60.23	33.89	21.7	367
	Z. R. Wang [18]	67.53	95.93	93.12	59.64	60.03	40.12	107	23
	CRNN [19]	81.89	92.95	64.35	77.68	97.25	50.34	18.8	8
	Proposal 1	96.14	97.91	77.79	88.21	99.31	**58.02**	47.6	10.2
	Proposal 2	**96.69**	**98.92**	98.03	**97.22**	99.36	56.55	161	13.1
	Proposal 3	92.55	96.25	97.59	81.34	99.34	54.13	47.6	10.8
	Proposal 4	96.35	98.86	**98.60**	95.35	**99.42**	53.85	161	11.6

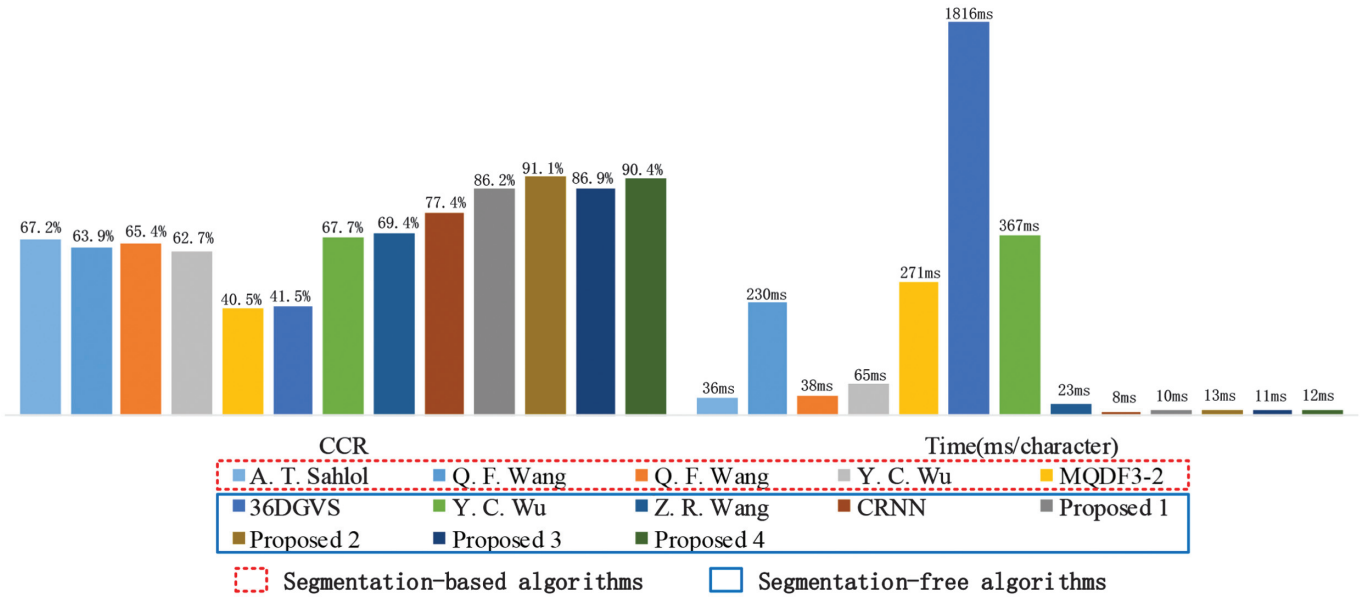

Fig. 9. Comparison of testing accuracy. Histograms of average testing CCR and average testing time cost.

V. CONCLUSION

A segmentation-free handwritten text recognition algorithm based on deep networks is proposed to tackle the challenges of sequence labeling for handwritten manuscripts with interleaving and touching in this paper. The proposed framework consisted of preprocessing, CNNs, BDLSTM, and CTC loss layers. And six representative handwritten databases of different languages, including English, Chinese, Russian, Spanish, and Arabic, are used for comparison experiments on a group of typical segmentation-based and segmentation-free algorithms. The proposed networks were proven to be useful in the task of sequence labeling, where Arabic with several variant shapes need improved algorithm for better recognition results in the future works.

ACKNOWLEDGMENTS

This work was funded by Yunnan Province Local Undergraduate University Basic Research Joint Special Fund Project (No. 202101BA070001-016).

CONFLICT OF INTEREST STATEMENT

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

[1] A. T. Sahlol, M. A. Elaziz, M. A. Al-Qaness, and S. Kim, "Handwritten Arabic optical character recognition approach based on hybrid whale optimization algorithm with neighborhood rough set," *IEEE Access*, no. 8, pp. 23011–23021, 2020. DOI: 10.1109/ACCESS.2020.2970438.

[2] J. Tian, Y. Zhang, J. Lei, C. Sun, and G. Hu, "Lightweight classification network for pulmonary tuberculosis based on CT images," *J. Artif. Intell. Technol.*, vol. 3, no. 1, pp. 25–31, 2023. DOI:10.37965/jait.2023.0153.

[3] J. Serin, K. T. Vidhya, I. S. Mary Ivy Deepa, V. Ebenezer, and A. Jenefa, "Gender classification from fingerprint using hybrid CNN-SVM," *J. Artif. Intell. Technol.*, vol. 4, no. 1, pp. 82–87, 2023. DOI: 10.37965/jait.2023.0192.

[4] D. Li, P. F. Yu, H. Li, and P. Ge, "Printed New Tai Lue character recognition based on BP neural network," in *2016 IEEE Int. Conf. Signal Image Process. (ICSIP)*, IEEE, 2016, pp. 339–342. DOI: 10.1109/SIPROCESS.2016.7888280.

[5] J. Bouvrie, "Notes on convolutional neural networks," *neural nets*, 2006. URL: https://web-archive.southampton.ac.uk/cogprints.org/5869/1/cnn_tutorial.pdf

[6] W. Zaremba, I. Sutskever, and O. Vinyals. "Recurrent neural network regularization," arXiv preprint, 2014. DOI: 10.48550/arXiv.1409.2329.

[7] R. G. Casey and E. Lecolinet, "A survey of methods and strategies in character segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 7, pp. 690–706, 1996. DOI: 10.1109/34.506792.

[8] P. Ge, P. Yu, H. Li, and L. He, "Text line segmentation using Viterbi algorithm for the palm leaf manuscripts of Dai," in *2016 Int. Conf. Audio, Lang. Image Process. (ICALIP)*, IEEE, 2017, pp. 336–340. DOI: 10.1109/ICALIP.2016.7846561.

[9] G. Congedo, G. Dimauro, S. Impedovo, and G. Pirlo, "Segmentation of numeric strings," in *3rd Int. Conf. Doc. Anal. Recogn.*, IEEE, 1995, pp. 1038–1041. DOI: 10.1109/ICDAR.1995.602080.

[10] Q. F. Wang, "Handwritten Chinese text recognition by integrating multiple contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1469–1481, 2012. DOI: 10.1109/TPAMI.2011.264.

[11] Q. F. Wang, F. Yin, and C. L. Liu, "Improving handwritten Chinese text recognition by unsupervised language model adaptation," in *2012 10th IAPR Int. Workshp. Doc. Anal. Syst.*, IEEE, 2012, pp. 110–114. DOI: 10.1109/DAS.2012.46.

[12] Y. Zhu, J. Sun, and S. Naoi, "Sub-structure learning based handwritten Chinese text recognition," in *Int. Conf. Doc. Anal. Recogn.*, IEEE, 2013. DOI: 10.1109/ICDAR.2013.66.

[13] Y. C. Wu, F. Yin, and C. L. Liu, "Evaluation of neural network language models in handwritten Chinese text recognition," in *2015 13th Int. Con. Doc. Anal. Recogn. (ICDAR)*, IEEE, 2015, pp. 166–170. DOI: 10.1109/ICDAR.2015.7333745.

[14] T. Su, "Chinese handwriting recognition: an algorithmic perspective," *Springerbriefs Electr. Comput. Eng.*, 2013. DOI: 10.1007/978-3-642-31812-2.

[15] A. Graves, *Supervised Sequence Labeling with Recurrent Neural Networks*, Heidelberg, 2013. DOI: 10.1007/978-3-642-24797-2.

[16] D. Cireşan and U. Meier, "Multi-column deep neural networks for offline handwritten Chinese character classification," in *2015 Int. Joint Conf. Neural Netwks (IJCNN)*, IEEE, 2015, pp. 1–6. DOI: 10.1109/IJCNN.2015.7280516.

[17] Y. C. Wu, F. Yin, Z. Chen, and C. L. Liu, "Handwritten Chinese text recognition using separable multi-dimensional recurrent neural network," in *2017 14th IAPR Int. Conf. Doc. Anal. Recogn. (ICDAR)*, 2017, pp. 79–84. DOI: 10.1109/ICDAR.2017.22.

[18] Z. R. Wang, J. Du, J. S. Hu, and Y. L. Hu, "Deep convolutional neural network based hidden Markov model for offline handwritten Chinese text recognition," in *2017 4th IAPR Asian Conf. Pattern Recogn. (ACPR)*, IEEE, 2017, pp. 816–821. DOI: 10.1109/ACPR.2017.65.

[19] B. Shi, B. Xiang, and Y. Cong, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2016. DOI: 10.1109/TPAMI.2016.2646371.

[20] P. Kumar and A. Sharma, "Segmentation-free writer identification based on convolutional neural network," *Comput. Electr. Eng.*, vol. 85, p. 106707, 2020. DOI: 10.1016/j.compeleceng.2020.106707.

[21] H. Zhang, C. Xu, C. Shi, H. Bi, Y. Li, and S. Mian, "HSCA-net: a hybrid spatial-channel attention network in multiscale feature pyramid for document layout analysis," *J. Artif. Intell. Technol.*, vol. 3, no. 1, pp. 10–17, 2022. DOI: 10.37965/jait.2022.0145.

[22] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: 10.1109/78.650093.

[23] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *30th Int. Conf. Mach. Learn.*, ICML, 2013, 2012, DOI: 10.1007/s12088-011-0245-8.

[24] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netwks.*, vol. 5, no. 2, pp. 157–166, 1994. DOI: 10.1109/72.279181.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[26] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: labeling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376. DOI: 10.1145/1143844.1143891.

[27] J. P. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990. DOI: 10.1109/5.58337.

[28] U. V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *Int. J. Doc. Anal. Recogn.*, vol. 5, no. 1, pp. 39–46, 2002. 10.1007/s100320200071.

[29] C. L. Liu, F. Yin, D. H. Wang, and Q. F. Wang, "CASIA online and offline Chinese handwriting databases," in *2011 Int. Conf. Doc. Anal. Recogn.*, IEEE, 2011, pp. 37–41. DOI: 10.1109/ICDAR.2011.17.

[30] T. Su, T. Zhang, and D. Guan, "Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text," *Int. J. Doc. Anal. Recogn.*, vol. 10, no. 1, p. 27, 2007. DOI: 10.1007/s10032-006-0037-6.

[31] M. Potanin, D. Dimitrov, A. Shonenkov, V. Bataev, D. Karachev, and M. Novopoltsev, "Digital Peter: dataset, competition and handwriting recognition methods," 2021. DOI: 10.48550/ARXIV.2103.09354.

[32] N. Serrano, F. Castro, and A. Juan-Císcar. "The RODRIGO database," in *Proc. of the 7th Int. Conf. Lang. Resources Eval. (LREC).*, 2010, pp. 2709–2712. URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/477_Paper.pdf

[33] S. A. Mahmoud et al., "KHATT: Arabic offline handwritten text database," in *2012 Int. Conf. Front. Handwrit. Recogn.*, 2012, pp. 449–454. DOI: 10.1109/ICFHR.2012.224.