Deep Convolutional Recurrent Network for Segmentation-free Offline Handwritten Japanese Text Recognition

Nam-Tuan Ly
Dept. of Computer Science
Tokyo University of A&T
Tokyo, Japan
namlytuan@gmail.com

Cuong-Tuan Nguyen
Dept. of Computer Science
Tokyo University of A&T
Tokyo, Japan
ntcuong2103@gmail.com

Kha-Cong Nguyen
Dept. of Computer Science
Tokyo University of A&T
Tokyo, Japan
congkhanguyen@gmail.com

Masaki Nakagawa
Dept. of Computer Science
Tokyo University of A&T
Tokyo, Japan
nakagawa@cc.tuat.ac.jp

***Abstract* - This paper presents a model of Deep Convolutional Recurrent Network (DCRN) for recognizing offline handwritten Japanese text lines without explicit segmentation of characters. Most of traditional offline handwritten Japanese text recognizers perform segmentation of text image into characters before individually recognizing each character. Although segmentation by recognition and context are employed to recover from segmentation errors, errors made at this stage directly make an impact on the performance of the whole system. The DCRN model consists of three parts: a convolutional feature extractor using Convolutional Neural Network (CNN) and sliding window to extract features from text image; recurrent layers using BLSTM to predict pre-frame from an input sequence; and a transcription layer using a CTC-decoder to translate the predictions into the label sequence. Experimental results on the database: TUAT Kondate database demonstrates the effectiveness of the proposed method.**

Keywords - CNN, BLSTM, CTC, sliding window, segmentation-free

I. INTRODUCTION

The handwritten Japanese text recognition is still a big challenging problem and has been receiving much attention from numerous researchers. However, the existing systems are still far from perfection because of the large character set; varieties of characters mixed of thousands of Kanji characters of Chinese origin, two sets of phonetic characters, alphabets, numerals, symbols, etc.; diversity of writing styles and multiple-touches between characters. Most of the traditional offline handwritten Japanese/Chinese text recognizers [3][4] perform segmentation of text image into characters before individually recognizing each character and integrating linguistic and geometric contexts. However, errors due to segmentation directly affect the performance of the whole system. In recent years, Deep Neural Network is demonstrating surpassing performances than the state-of-the-art accuracies on many tasks such as Convolutional Neural Network for Image recognition and feature extraction [6][15], Long Short Term Memory Recurrent Neural Networks (LSTM RNNs) for sequence prediction and labeling tasks [12][13]. Graves et al. [1] combined Bidirectional LSTM (BLSTM) and the Connectionist Temporal Classification (CTC[2]) to build a Connectionist System for unconstrained handwriting recognition. Base on Deep Neural Network, many segmentation-free methods [10][11] have been studied and have demonstrated to be powerful in image-based sequence recognition tasks. R. Messina and J. Louradour [10] combined Multi-Dimensional Long-Short Term Memory Recurrent Neural Network (MDLSTM-RNN) and the CTC to build an end-to-end trainable model for offline handwritten Chinese text recognition. However, this method does not take advantage of CNN for feature extraction and an end to end model is usually hard to converge and time-consuming (~400 epochs) when training network because of many parameters. Suryani et al. [11] proposed a method combining pretrained CNN and BLSTM followed by a Hidden Markov Model (HMM) alignment for offline handwritten Chinese text recognition. However, CTC alignment based on probability is demonstrated that achieves higher performance than HMM alignment for sequence prediction and labeling tasks [1][2].

In this paper, we propose a model of Deep Convolutional Recurrent Network (DCRN) for offline handwritten Japanese text recognition. It consists of three parts: a convolutional feature extractor using CNN and sliding window to extract features from text image; recurrent layers using BLSTM to predict pre-frame from an input sequence; and a transcription layer using a CTC-decoder to translate the predictions into the label sequence.

The rest of this paper is organized as follows: Session II presents the architecture of our proposed model, DCRN. Session III reports our experimental results and analysis. Session IV draws conclusions.

II. DEEP CONVOLUTIONAL RECURRENT NETWORK

The network architecture of DCRN consists of 3 components, including the convolutional feature extractor, the

2379-2140/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDAR.2017.357

recurrent layers, and a transcription layer, from bottom to top as shown in Fig. 1.

From the bottom of the DCRN, the convolutional feature extractor extracts a feature sequence from an input image, the recurrent layers at the top of the convolutional feature extractor predict each frame of the feature sequence output by the convolutional feature extractor. At the top of the DCRN, the transcription layer translates the pre-frame predictions by the recurrent layers into the final label sequence.

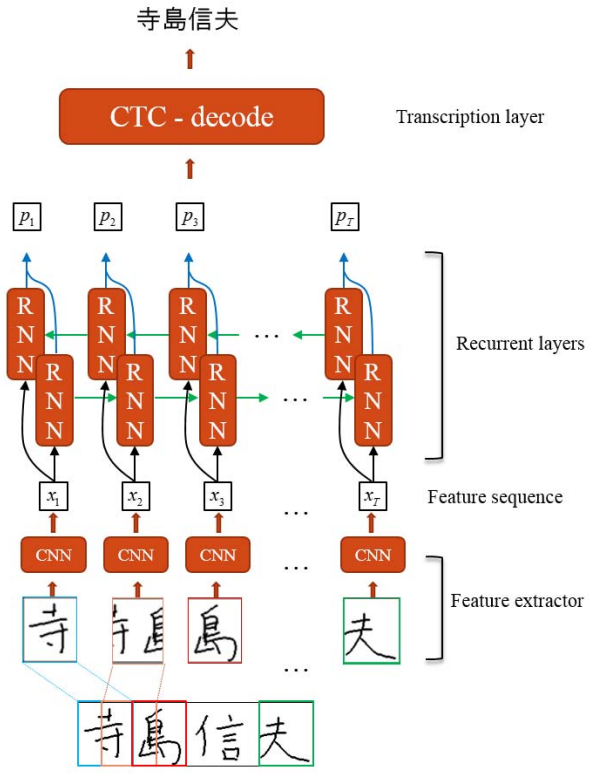

Fig. 1. Network architecture of DCRN. The network consists of three components: 1) Convolutional feature extractor, which extracts a feature sequence from an input image; 2) Recurrent layers, which predict a label distribution for each feature; 3) Transcription layer, which translates the per-frame predictions into the final label sequence.

A. Convolutional feature extractor

A Deep convolutional neural network is demonstrated that it is a powerful visual model and achieves the state-of-the-art accuracy on some tasks such as image recognition[6]. Deep Maxout [9] CNN combined with "dropout" [14] is also powerful for computer vision tasks [9].

In the DCRN model, the component of convolutional feature extraction is constructed by taking the convolutional, max-pooling and full-connected layers from a standard CNN model (softmax layer are removed). Maxout units [9] is employed after each convolutional and full-connected layer and "dropout" is only used in full-connected layers. Such components are used to extract a feature sequence from an input image by sliding a sub-window through the text image. Before being fed into the component, all the text images need to be scaled to the same height in order to have the same size of the input image for CNN. Then, the feature sequence is extracted from the text image by the convolutional feature extractor, which is the input of the recurrent layer. Each feature is 200 elements of a vector and is a time step of input for the recurrent layers. The weights of the CNN model in the convolutional feature extractor is pretrained by the TUAT Nakayosi and Kuchibue, handwritten Japanese character databases [7].

B. The Recurrent layers

Recurrent neural networks (RNNs) are connectionist models containing a self-connected hidden layer. The recurrent connection allows information of previous inputs to remain in the network's internal states; therefore it makes use of past contextual information. However, the traditional RNNs suffer from the gradient vanishing and exploding problem.

Long Short-Term Memory (LSTM) is a special kind of RNN architectures designed to address the vanishing gradient problem, which is capable of learning long-term dependencies. A LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. Each block contains a set of internal units, known as cells, whose activation is controlled by three multiplicative gate units. The effect of the gates is to allow the cells to store and access information over long periods of time. For many tasks such as handwritten text recognition, it is useful to use future as well past contextual information. However, the standard LSTM can only use past contextual information in one direction. This can be overcome by using Bidirectional LSTM (BLSTM [12]) that can learn long-range context dynamics in both input directions.

In our proposed DCRN model, the deep BLSTMs are built on top of the convolutional feature extractor, as the recurrent layers to predict a label distribution for each feature of the feature sequence extracted from the previous component.

C. Transcription layer

CTC is a specific loss function designed for the sequence labeling tasks where it is difficult to segment the input sequence to the segment that exactly matches a target sequence. CTC performs alignment of a probability output sequence to the label sequence. As a result, the system does not need to segment the input sequence for training. To avoid the difficulty of segmentation in handwritten text recognition systems, we employ CTC to be built on top of the recurrent layers, as the transcription layer in our framework.

We denote the character set as $C' = C \cup \{blank\}$, where C is a fixed set of labels and '*blank*' represents no label. For an input sequence $x = x_1, x_2, \ldots, x_T$ of length T, the conditional probability of a path π through the lattice of output labels over all the time steps is calculated by multiplying the probabilities of labels along this path:

$$p(\pi \mid x) = \prod_{t=1}^{T} p(\pi_t, t \mid x) \qquad (1)$$

where π_t is the label of the path π at time t.

A label sequence is obtained from a path by a reduction process denoted as B, which firstly removes repeated labels, then removes *blanks* in this path (e.g. B(_cc_a_tt_) = B(_c_aa_t_) = cat). The probability of a label sequence l from an input sequence x is the total probability of all the paths, where each path is reduced into this label sequence by B. It is shown as follows:

$$p(l \mid x) = \sum_{\pi : B(\pi) = l} p(\pi \mid x) \tag{2}$$

Applying the CTC forward-backward algorithm [2], $p(l \mid x)$ in (2) is obtained efficiently. For decoding, we could obtain the best label by:

$$l_{\max} = B(\pi_{\max}); \pi^t_{\max} = \arg\max_k \left(y^t_k \right), t = 1 \ldots T$$

This is obtained without explicitly segmenting the input sequence.

III. EXPERIMENTS

To evaluate the performance of the proposed DCRN model, we conducted experiments on standard benchmarks for handwritten Japanese text recognition. The information of handwritten Japanese text datasets are given in Sec. A and Sec. B, the implementation details are described in Sec. C, the results of the experiments are presented in Sec. D and the misrecognized samples are shown in Sec. E.

A. Offline Handwritten Japanese Text Datasets

TUAT Kondate database [5] is a database of online handwritten patterns mixed with text, figures, tables, maps, diagrams and so on. It was turned to offline patterns by thickening strokes by constant width. The Japanese portion of Kondate was collected from 100 Japanese writers and the horizontal Japanese text lines stored in Kondate were used in our experiments. 13,684 horizontal Japanese text lines were split into two parts: first one consisting of 12,287 text lines collected from 90 Japanese writers were used as the training set, the second one consisting of 1,398 text lines collected from 10 Japanese writers were used as the testing set. We randomly split the training set into two group, with approximately 90% for training and remainder for validation. They are summarized in Table I.

TABLE I. The detail of information of Kondate database.

	Kondate	
	Train and valid sets	**Test set**
Number of writers	90	10
Number of samples	12,287	1,398

B. Offline Handwritten Japanese Character Datasets

The weights of the CNN model in the convolutional feature extractor is pretrained by the TUAT Nakayosi and Kuchibue handwritten Japanese character databases [7]. Nakayosi contains samples of 163 writers, 10,403 character patterns covering 4,438 classes per writer. Kuchibue contains handwritten samples of 120 writers, 11,951 character patterns covering 3,345 classes per writer. The summary of the Nakayosi and Kuchibue databases are shown in Table II. They are turned to offline patterns again by thickening stroke with constant width. In this work, we experimented with 3,345 classes of JIS level-1 Kanji characters (2965 classes) and kana, alpha-numerals, symbols and so on (380 classes) for pretraining the CNN model. We used the samples of Nakayosi for training and the samples of Kuchibue for testing. We randomly split the training set into two group, with approximately 90% for training and remainder for validation.

TABLE II. Summary of Nakayosi and Kuchibue databases.

	Nakayosi	**Kuchibue**
Number of writers	163	120
Number of classes	4,438	3,345
Number of samples	1,695,689	1,435,440

C. Implementation Details

The detailed architecture of our CNN model used in the convolutional feature extractor is listed in Table III. It contains seven learned layers - four convolutional layers alternatively by four max-pooling layers, two full-connected layers and a softmax layer finally (3345 class). Each convolutional and full-connected layer is followed by Maxout units [9], using the group size of 2. Firstly, the CNN model is pretrained by using stochastic gradient descent with a batch size of 64 samples with the learning rate of 0.01 and the momentum of 0.95 on GPU. After training the CNN model, we remove just the softmax layer or both the full connected layers and the softmax layer from the CNN model to use the remaining network as the convolutional feature extractor. We call the former DCRN-s and the latter DCRN-f&s.

TABLE III. Network configuration of our CNN model. 'maps', 'k', 's' and 'p' denote the number of kernel, kernel size, stride and padding size of convolutional layers respectively. 'group' denotes the group size of Maxout units.

Type	**Configurations**
Input	96×96 image
Convolution - Maxout	#maps:32, k:5×5, s:1, p:0, group:2
MaxPooling	#window:2×2, s:2
Convolution - Maxout	#maps:32, k:3×3, s:1, p:0, group:2
MaxPooling	#window:2×2, s:2
Convolution - Maxout	#maps:64, k:3×3, s:1, p:0, group:2
MaxPooling	#window:2×2, s:2
Convolution - Maxout	#maps:64, k:5×5, s:1, p:0, group:2
MaxPooling	#window:2×2, s:2
Full-connected - Maxout	#nodes:400, group:2
Full-connected - Maxout	#nodes:400, group:2
Softmax	#nodes: 3345(number class)

At the recurrent layers, we employ Deep BLSTM network with 256 nodes of two layers. The recurrent layers and the

transcription layer are trained by using online steepest decent with the learning rate of 0.0001 and the momentum of 0.9. All of the text line images of the Kondate database are scaled to the same height before used to train the DCRN models.

D. Results of Experiments

The CNN model was pretrained for 110 epochs and the training process was stopped when the accuracy rate did not improve for 10 epochs with the result that 95.17% of accuracy rate was achieved for the test set. The results are summarized in Table IV.

TABLE IV. The accuracy of CNN model.

Model	Validation set	Testing set
CNN model	97.6%	95.17%

In order to evaluate the performance of the DCRN models, the performance is measured in terms of Label Error Rate (LER) [2] and Sequence Error Rate (SER) [2] that are defined as follows:

$$LER(h,S') = \frac{1}{Z}\sum_{(x,z)\in S'} ED(h(x),z)$$

$$SER(h,S') = \frac{100}{|S'|}\sum_{(x,z)\in S'} \begin{cases} 0 & if \quad h(x)=z \\ 1 & otherwise \end{cases}$$

Where Z is the total number of target labels in S' and ED(p, q) is the edit distance between two sequences p and q.

The recurrent layers and the transcription layer components of the DCRN-s model were trained for 50 epochs, and then the label error rate of 6.44% and the sequence error rate of 25.89% were obtained in the test set. For DCRN-f&s, the recurrent layers and the transcription layer components were trained for 40 epochs with the result of 6.95% of the label error rate and 28.04% of the sequence error rate in the test set. The results imply that the DCRN-s model, the convolutional feature extractor made by only removing the softmax layer from the CNN model, works better than DCRN-f&s, the convolutional feature extractor made by removing both the full connected layers and the softmax layer from the CNN model. Table V presents the results of our systems and the segmentation based method [3] for the Kondate database [5]. It shows that the DCRN models significantly outperform the traditional method based on segmentation with linguistic context [3] for the Kondate database in both the label error rate and the sequence error rate although they don't use any language model.

TABLE V. Label Error Rate (LER) and Sequence Error Rate (SER) on Kondate.

Model	LER		SER	
	Valid set	Test set	Valid set	Test set
DCRN-s	11.01%	6.44%	37.38%	25.89%
DCRN-f&s	11.74%	6.95%	39.33%	28.04%
Segmentation based [3]	-	11.2%	-	48.53%

For the convergence of training, our network achieves convergence after 110 epochs for CNN and about 50 epochs for BLSTM&CTC compared with approximately 400 epochs for an end to end model MDLSTM-RNN [10]. Fig. 2 presents the label error rate achieved after each epoch when training the recurrent layers and the transcription layer components of DRCN-s and DRCN-f&s.

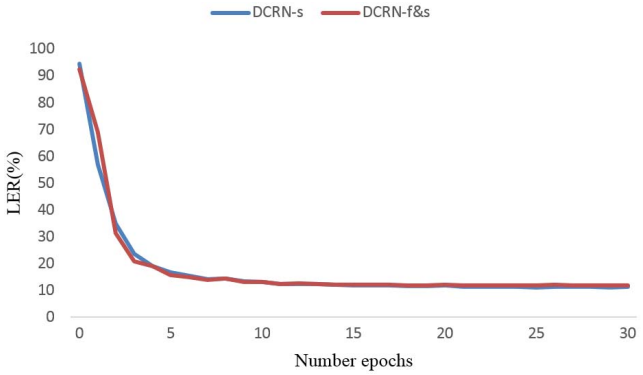

Fig. 2. Label error rates on the validation set after each epoch when training the recurrent layers and the transcription layer components of DCRN-s and DCRN-f&s.

E. Misrecognized samples

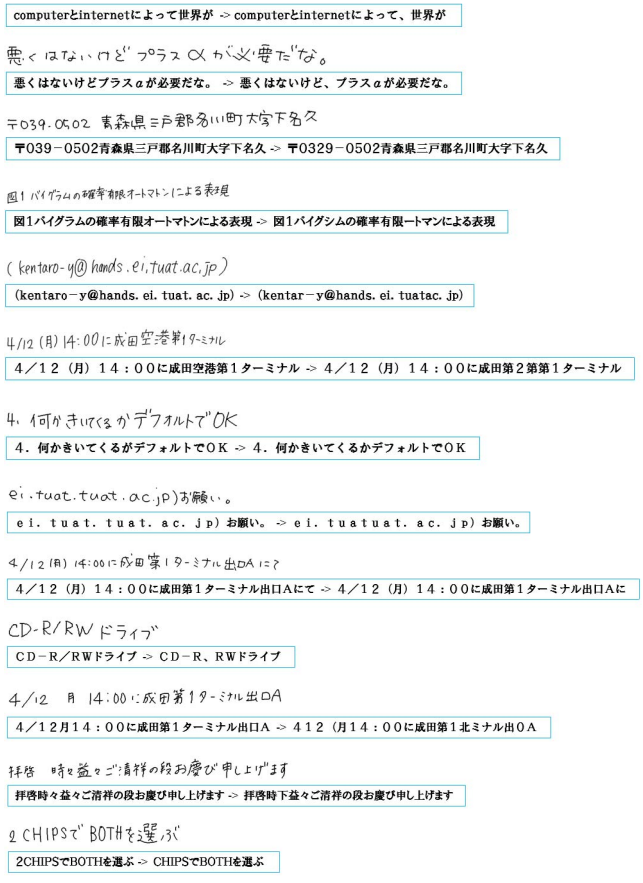

Fig. 3. Some mispredicted samples by DCRN-s.

There are a total of 362 misrecognized samples among 1398 samples. Most of them are missing some characters in the ground-truth. Fig. 3 shows some misrecognized samples by DCRN-s whose sequence error rate is 25.89%. For each sample, the upper image is an input handwritten text line image and the text bounded by the lower blue rectangular shows the ground-truth and the recognition result separated by "->".

IV. Conclusion

In this paper, we presented a novel method of Deep Convolutional Recurrent Neural Network (DCRN) for recognizing offline handwritten Japanese text. The DCRN consists of three parts: the convolutional feature extractor, the recurrent layers and the transcription layer that directly recognize offline handwritten Japanese text without segmentation. Following the experiments on the test set of handwritten Japanese text database, TUAT Kondate, the DCRN-s with the convolutional feature extractor made by removing the softmax layer from the CNN model, obtained the label error rate of 6.44% and the sequence error rate of 25.89% while the DCRN-f&s with the convolutional feature extractor made by removing both the full connected layers and the softmax layer from the CNN model, obtained the label error rate of 6.95% and the sequence error rate of 28.04% compared with the label error rate of 11.2% and the sequence error rate of 48.53% obtained by the traditional segmentation-based method. The following conclusions are drawn: 1) DCRN-s works better than DCRN-f&s; 2) the DCRN models significantly outperform the traditional segmentation-based method in both the label error rate and the sequence error rate.

References

[1] A. Graves, M. Liwicki, S. Fern´andez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 5, pp. 855-868, 2009.

[2] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," In Proceedings of the 23rd international conference on Machine learning, pp. 369-376. ACM, 2006.

[3] K. C. Nguyen and M. Nakagawa. "Text-Line Character Segmentation for Offline Recognition of Handwritten Japanese Text," IEICE Technical Report, BioX2015-50, PRMU2015-173, 2016.

[4] Q.-F. Wang, F. Yin, and C.-L. Liu, "Handwritten Chinese Text Recognition by Integrating Multiple Contexts," IEEE Trans. Pattern Anal. Mach. Intell., vol. 34, no. 8, pp. 1469-1481, 2012.

[5] T. Matsushita and M. Nakagawa. "A Database of On-line Handwritten Mixed Objects named "Kondate"," 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 369-374 , 2014

[6] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.

[7] M. Nakagawa, K. Matsumoto, "Collection of on-line handwritten Japanese character pattern databases and their analysis," Int. J. Document Anal. Recognit., vol. 7, no. 1, pp. 69-81, 2004.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," The computing research repository, 2012.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout networks," arXiv: 1302.4389, 2013.

[10] R. Messina and J. Louradour, "Segmentation-free handwritten chinese text recognition with lstm-rnn," 13th International Conference on Document Analysis and Recognition (ICDAR), pp. 171-175, 2015.

[11] D. Suryani, P. Doetsch and H. Neys, "On the Benefits of Convolutional Neural Network Combinations in Offline Handwriting Recognition," 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 193-198, 2016.

[12] S. Hochreiter, J. Schmidhuber, "Long Short-term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

[13] A. Graves, J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, vol. 18, no. 5-6, pp. 602-610, July 2005.

[14] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," The computing research repository, 2012.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., pp. 1097-1105, 2012.