A Sequence Labeling Based Approach for Character Segmentation of Historical Documents

Liangcai Gao, Xiaode Zhang, Zhi Tang✉
ICST, Peking University, Beijing, China
{*glc, zhangxiaode, tangzhi*}*@pku.edu.com*

Yaoxiong Huang, Lianwen Jin
SCUT
h.yaoxiong@foxmail.com
eelwjin@scut.edu.cn

***Abstract*—As an important prerequisite step of historical document image analysis, character segmentation is fundamental but challenging. In this paper, we propose a novel approach for the handwritten character segmentation of historical documents by treating it as a sequence labeling problem. In more detail, the proposed model first segments document image into lines, then each column in the line image is given a label to indicate it is a segmentation position or not. The segmentation labeling is achieved by a neural model, which combines a CNN for feature extraction, a LSTM for sequence modeling and a CRF for sequence labeling. The performance of our methods has been evaluated on a 300-page dataset including 96,479 characters. The experimental results demonstrate that the proposed methods achieve superior or highly competitive performance compared with other methods.**

***Keywords*-historical documents; character segmentation; sequence labeling;**

I. Introduction

In recent years, a trend towards digitalization of historical documents can be observed [1]. Digitalization is a great way to protect historical documents from direct manipulation for consulting, exchanging and distant accessing purposes. Among all kinds of these processes, character segmentation is very challenging due to the poor quality of historical document images. Thus, many efforts are made to develop systems working solely on the rough input images without any priori segmentation [2]. For example, Bluche et al. [3] proposed a model based on multi-dimensional long short-term memory (LSTM) recurrent neural networks to perform end-to-end processing of line segmentation and multi-line recognition. Ghorbel et al. [4] achieved a segmentation-free word spotting system for handwritten documents using two filtering modules.

However, there is still a strong demand for reliable and robust character segmentation methods. The widely used standard Optical Character Recognition (OCR) systems and word spotting systems still use segmented lines or character images as input, and the overall performance of the systems could heavily depend on the quality of segmentation. Besides, scholars and researchers may wish to consult the original historical images by searching several keywords. The exact bounding boxes of characters generated by segmentation can help them locate the search term rapidly. What's more, the location information obtained by segmentation can provide great convenience for storage, management and research of historical documents.

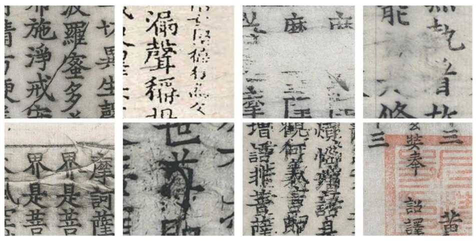
Figure 1: Some example images of Buddhist sutras.

In this paper, we address the problem of line segmentation and character segmentation of Buddhist sutras, a kind of handwritten historical documents. The goal is to give the exact bounding box to each character in Buddhist sutras. The line segmentation is relatively simple because of the regular arrangement of lines, but the character segmentation is a challenging task. As shown in Fig.1, Buddhist sutras suffers from many problems including stains, torn pages, ink seeping, etc. due to aging and degradation. Besides, touched characters and variation in the size of characters also make it difficult for segmentation.

Traditional methods of line and character segmentation can be roughly divided into three categories [5] (see Section II for details): projection-based, block-based and granular-based. The drawback of these methods is that they are sensitive to the quality of input images. To overcome this problem, some deep models are proposed for segmentation [5, 6], which use raw image pixels as input. Inspired by the recent advances in end-to-end OCR [7, 8] and Named Entity Recognition [9], we treat the character segmentation as a sequence labeling problem. The proposed neural network combines a convolutional neural network (CNN) for feature extraction and a LSTM for sequence modeling, and the sequence labeling is achieved by a Conditional Random Field (CRF).

The distinctive properties of this paper are summarized as follows. First, by converting the segmentation to a sequence labeling problem, we provide a new perspective to handle this problem, which also can be applied to other problems (such as musical score segmentation). Second, the contributions of different parts in the proposed network are discussed in detail. In addition, the proposed methods achieve superior or highly competitive performance, showing the effectiveness of our methods.

The remainder of this paper is organized as follows: Section II reviews the related works. Section III introduces

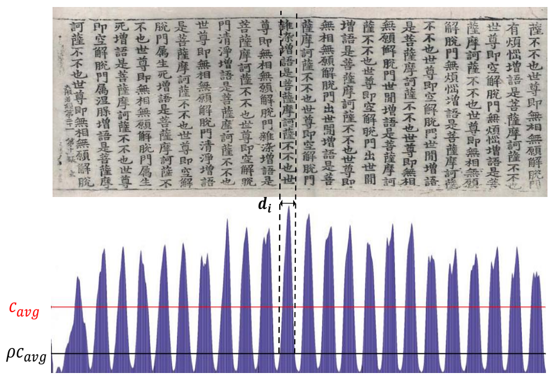

Figure 2: Projection profile for a Buddhist sutras image.

the proposed method for line segmentation and character segmentation. Experimental results are presented in Section IV, and we conclude this paper in Section V.

II. Related Works

The existing methods for line and character segmentation can be roughly classified into three categories: projection-based, block-based and granular-based.

Projection-based methods. In these methods, local minimums of projection function are used to indicate the positions for segmentation [10]. Projection-based methods are commonly applied to binary images [11, 12]. In [11], projection profiles along X axis and Y axis were utilized to obtain text lines and word borders respectively. In [12], the text lines were segmented by projection profiles and the characters were separated by contour tracing algorithm. However, binarization process for some images is a real challenge, therefore projection-based methods are adapted for gray scale images [10, 13]. Manmatha at al. [10] used projection profile during the line segmentation stage and the projection function was smoothed with a Gaussian filter. Kesiman at al. [13] applied an average block projection profile of gray level images to acquire segmentation results.

Block-based methods. In these methods, page images are cut into small areas, and these areas are then merged or split to produce homogeneous regions [5]. Chen et al. [14] proposed a multi-plane approach for text segmentation, they first decomposed the image into distinct object blocks and then a text extraction procedure was applied on the resultant blocks. Ouwayed et al. [15] extracted multi-oriented text lines based on the local orientation estimation, the orientation in each mesh was estimated, extended and corrected, and finally the text lines were extracted and separated

Granular-based methods. In these methods, basic elements of images, like connected components or even pixels, are merged into words, lines and others. In [16], an iterative method was proposed to group the connected components into lines. To avoid manually designed rules, classification was also introduced into these methods. Moll et al. [17] achieved page segmentation by classifying individual pixels into predefined classes. In [18], Bukhari et al. used shape and context information of each connected component as a feature vector and then classified them into text and non-text with a multi-layer perception. Le et al. [19] developed an approach similar to [18], but extracted features without component rescaling and used decision tree to label connected components.

The character segmentation is very similar to the object detection in natural scene images. Therefor this problem also can be solved by object detection models like R-fcn [20], SSD [21], Yolo9000 [22], etc. However, characters are commonly small objects and appear densely in an image, thus these models do not achieve a high performance in our experiments.

III. Proposed Method

The proposed method mainly consists of two stages, namely, line segmentation and character segmentation. Line segmentation is relatively simple, but character segmentation is so challenging because of the image degradations, character touching, overlapping and multi-connected components as shown in Fig.1 and Fig.3. Thus, we propose a neural network combining CNN, LSTM and CRF to address this problem.

A. Line Segmentation

Considering the regular arrangement of the lines in Buddhist sutras images, the projection profile technique is utilized here to segment the image lines. Let $f(x, y)$ be the value of a pixel (x, y) in a binary image, $f(x, y) = 0$, if (x, y) is a background pixel, otherwise $f(x, y) = 1$. The vertical projection function can be defined as

$$P(x) = \sum_{y=0}^{H} f(x, y)$$

where H is the height of the image. Fig.2 shows a Buddhist sutras image and its projection profile.To avoid the local extremum caused by noise, we adopt the same Gaussian filter mentioned in [10] to smooth the projection function. With this profile, we can get a series of intervals $\{d_i\}$ which satisfy the following condition

$$P(x_i^j) > \rho C_{avg} \qquad \forall x_i^j \in d_i$$

where $C_{avg} = \frac{1}{W}\sum_{x=0}^{W} P(x)$ is the average pixel density, W is the width of the image, ρ is set to 0.3 empirically. Then the text line can be determined by the corresponding interval d_i whose length $L(d_i)$ satisfies the following condition

$$|L(d_i) - L(d_{avg})| < \beta L(d_{avg})$$

where $L(d_{avg})$ is the average length of the intervals $\{d_i\}$, β is set to 0.3 empirically.

B. Character Segmentation

With the segmented line images, the problem of character segmentation can be converted into a sequence labeling problem. We try to label whether each column of the line image is a correct segmentation position. The source, $\boldsymbol{I} \in \mathcal{I}$, consists of an image with height H and width

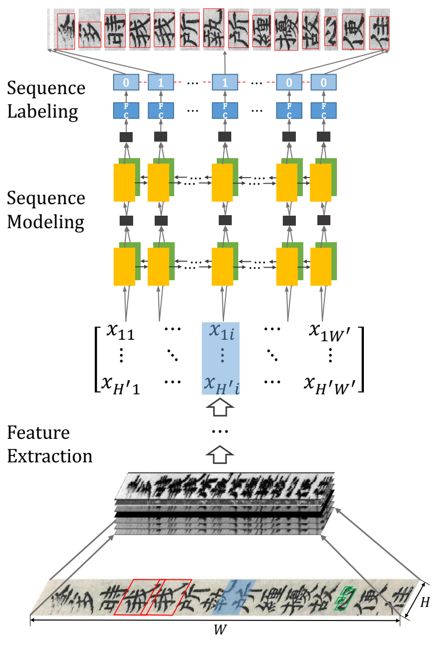

Figure 3: Network structure of character segmentation.The red rectangle in original image shows the overlap between characters and the green rectangle shows the multi-connected components within a character. The blue region represents the receptive filed corresponding to the vector x_i in last feature map. The red dotted line across output sequence is only used for CRF.

W, e.g. $R^{H\times W}$ for gray scale inputs. The target, $\boldsymbol{y} \in \mathcal{Y}$, consists of a sequence of labels $\{y_1, y_2, \cdots, y_{W'}\}$, where y_i is the label which equals to 1 when corresponding columns are the correct segmentation positions, otherwise equals to 0. $W^{'}$ is the length of the output, which means $W/W^{'}$ columns share the same label. The map from $\boldsymbol{I}$ to $\boldsymbol{y}$ is implemented by combining a CNN for feature extraction and LSTMs for sequence modeling, then based on the output of LSTMs, sequence labeling is performed with a linear classifier or a CRF, as illustrated in Fig.3.

1) Feature Extraction: The visual features of the line images are extracted with a multi-layer CNN. Given an input image of size $H \times W$, the CNN produces an output of size $D \times H^{'} \times W^{'}$, where D denotes the number of channels, $H^{'}$ and $W^{'}$ are the height and width of feature map. To feed the output of CNN into LSTMs, Stephen et al. [8] passed i-th columns of all the maps (of size $D \times H^{'}$) to a fully-connected layer to reduce the dimension, Baoguang et al. [7] concatenated these columns to form a single vector. While in this paper, we perform the 1×1 convolution proposed in [23] before the last convolutional layer to reduce the channels, resulting in a single feature map with the size of $H^{'} \times W^{'}$. 1×1 convolution has two main advantages. One is that it has much less parameters compared with fully-connected layer, thus can greatly avoid overfitting. The other is that it is more capable of feature representation compared with direct concatenation of different channels.

When designing a CNN for our model, the following criteria are taken into consideration [8]:(1) The pooling operation should be applied as few as possible. Since a label is assigned to each column of the feature map, which means every $W/W^{'}$ columns in original images share the same label, where W and $W^{'}$ are the width of original image and last feature map respectively. Too much pooling operations will lead to a small $W^{'}$ and too much columns in original images share the same label, which is not accurate enough for segmentation. (2) The width of receptive field in last feature map should not be too small. Each column of the feature map is associated with a rectangle region in the original image, which is termed as receptive field (highlighted in blue in Fig.3). Intuitively, to decide whether a column in original image is a correct segmentation position, the context information from its left and right plays an important role, thus a proper receptive field is necessary.

2) Sequence Modeling: Built on the top of last convolutional layer, a deep bidirectional Recurrent Neural Network takes as input the columns of feature map $\{x_1, x_2, \cdots, x_{W'}\}$ and produces another sequence $\{z_1, z_2, \cdots, z_{W'}\}$ which represents some information about the input at every time step. Here, we choose LSTM as the basic RNN unit for its capability of capturing long range dependencies within a sequence. To efficiently use the information from past (left) and future (right) input features, we follow [7] and combine two LSTMs, one forward and one backward, into a bidirectional LSTM, and the output of LSTMs from two directions are concatenated. Furthermore, two bidirectional LSTMs are stacked to form a deep bidirectional LSTM, as illustrated in Fig.3.

3) Sequence Labeling: Based on the output sequence $\{z_1, z_2, \cdots, z_{W'}\}$ of LSTM, we try to assign a label y_i to each z_i, $y_i = 1$ if the corresponding columns in original image are the correct segmentation positions, otherwise $y_i = 0$. Two approaches are introduced here, one is linear classification and the other is CRF.

Linear Classification. It is simple but effective to make independent labeling decisions for each output z_i using a linear classifier. The dimension of z_i is reduced to the number of target labels by a fully connected linear layer. Finally it is fed into a softmax to generate a probability vector and the index of higher probability represents the correct label.

CRF. The linear classification makes labeling decisions independently, however, there are still strong dependencies across output labels. For example, if we have observed the end of a segmentation position, there is little probability to observe another segmentation position in the next predictions within the width of a character. Thus we follow the work in [9] and use a CRF to classify them jointly. Here a fully connected layer is still performed on $\{z_1, z_2, \cdots, z_{W'}\}$, resulting in a score matrix $\boldsymbol{P}$. $\boldsymbol{P}$ is of size $W^{'} \times k$, where $k = 2$ is the number of labels, and $P_{i,j}$ corresponds to the score of j-th label of the i-th column in the feature map. For a sequence of labels

Table I: Detailed Information about Training set and Testing set.

	Pages	Lines	Characters
Training set	700	16351	226701
Testing set	300	6990	96479

$\{y_1, y_2, \cdots, y_{W'}\}$, we define its score to be

$$s(\boldsymbol{X}, \boldsymbol{y}) = \sum_{i=1}^{W'} \boldsymbol{A}_{y_i, y_{i+1}} + \sum_{i=1}^{W'} \boldsymbol{P}_{i, y_i}$$

where $\boldsymbol{X}$ is the feature map, $\boldsymbol{A}$ is a matrix of transition scores such that $A_{i,j}$ represents the score of a transition from label i to label j. A softmax over all possible label sequences $\boldsymbol{Y}_{\boldsymbol{X}}$ yields a probability for the sequence $\boldsymbol{y}$:

$$p(\boldsymbol{y}|\boldsymbol{X}) = \frac{e^{s(\boldsymbol{X},\boldsymbol{y})}}{\sum_{\boldsymbol{y'} \in \boldsymbol{Y}_{\boldsymbol{X}}} e^{s(\boldsymbol{X},\boldsymbol{y'})}}$$

Then, we predict the output sequence that obtain the maximum score given by:

$$\boldsymbol{y}^* = argmax_{\boldsymbol{y'} \in \boldsymbol{Y}_{\boldsymbol{X}}} s(\boldsymbol{X}, \boldsymbol{y}^{'})$$

4) Sequence Label to Bounding box: With the sequence labels, the bounding boxes of the characters can be easily obtained. As illustrated in Fig.3, the sequence labels, where 1 represents the segmentation position, are mapped onto the original image according to the scaling ratio ($W/W^{'}$ columns share the same label as mentioned above, here $W/W^{'} = 2$). We can get a coarse bounding box between every two neighbouring segmentation positions, and then it can be refined to a tight box by searching for the non blank area in binary image.

IV. Experiments

A. Dataset

We collect 1,000 pages of Buddhist sutras, which contain 23,341 lines in total, from Tripitaka Koreana in Han [1] to perform our experiments. We randomly select 700 images for training and 300 images for testing. Detailed information about the training set and testing set is shown in Table I. The bounding boxes of characters in this dataset are annotated by searching the connected components in binary images and refined manually. After line segmentation with projection profile, the height of line images is resized to 60, because a fixed-dimensional input is necessary for LSTMs. In fact, there is no need to resize the width, but for the convenience of batch training, it is resized to 500. Within each line, the ground truth of segmentation positions for characters is calculated using the annotated bounding boxes.

B. Implementation Details

The network (denoted as CNN_5_BiLSTM_CRF in Table III) configurations are presented in Table II. We follow the criteria discussed in Section III to design our CNN, and the max pooling is only performed once to keep the width of last feature map. 1×1 convolution is used after Conv4 to reduce the dimension. Each convolution layer is followed by a Rectify Linear Unit (ReLU). Two bidirectional LSTMs are stacked to form a deep bidirectional LSTM. The sequence labeling is finished by a linear classifier or a CRF.

The network is implemented with PyTorch [2] and it is optimized by Adam [24] with a learning rate of 0.001 and a batch size of 10. To avoid overfitting, the models are selected when the loss curve becomes plain during training. All the experiments are carried out on a workstation with the Intel(R) Xeon(R) E5-2640 2.50GHz CPU, 64GB RAM and one NVIDIA Titan X GPU.

Table II: Network Configurations.

Sequence Labeling	Softmax or CRF output: 250×1				
FC	output: 250×2				
LSTM output: 250×128					
	Input dim	Hidden dim	Bidirectional		Step
LSTM2	128	64	True		250
LSTM1	30	64	True		250
CNN output: $30 \times 250 \times 1$					
	In dim	Out dim	Kernel size	Stride	Padding
Conv5	8	1	1×1	1×1	0
Conv4	16	8	3×3	1×1	1
Conv3	16	16	3×3	1×1	1
Max pool	16	16	2×2	2×2	0
Conv2	8	16	3×3	1×1	1
Conv1	1	8	5×5	1×1	2
Input	$60 \times 500 \times 1$				

C. Evaluation Metrics

After converting the sequence labels to bounding boxes, Precision (P), Recall (R), and F-score (F) with respect to different Intersection over Union (IoU) ratio are adopted to evaluate the proposed method. The predicted bounding boxes that have IoU overlap with the ground truth higher than a specified value are considered correct. For the object detection in natural scene images, IoU higher than 0.5 is commonly used for evaluation [22]. However, it is not sufficient for the character segmentation in historical document images because of the strict requirements for accurate bounding boxes. It is crucial to the performance of other systems which take as input the segmented characters. Thus in the following experiments, we only report the results measured with IoU higher than 0.7. Especially, we are more concerned about the performance with respect to the IoU of 0.8.

D. Experimental Results

1) Comparisons with other methods: We compare our methods with three traditional methods including Projection [13], Grouping [16], CCS [19], and two deep learning based methods including R-fcn [20] and SSD [21]. As shown in Table III, all the traditional methods preform poorly. Even our baseline model (denoted as

[1]http://kb.sutra.re.kr/ritk/index.do

[2]http://www.pytorch.org/

Table III: Recall (R), Precision (P) and F-score (F) of different methods with respect to IoU ranging from 0.7 to 0.8.

Method	IoU 0.7			IoU 0.75			IoU 0.8		
	R	P	F	R	P	F	R	P	F
Projection [13]	0.3261	0.3434	0.3345	0.2223	0.2316	0.2269	0.1538	0.1579	0.1558
Grouping [16]	0.2660	0.2107	0.2351	0.1573	0.1093	0.1290	0.1502	0.1081	0.1087
CCS [19]	0.7551	0.7642	0.7596	0.7063	0.7148	0.7105	0.6221	0.6296	0.6258
R-fcn [20]	0.8854	0.9732	**0.9272**	0.8385	0.9217	0.8781	0.7115	0.7822	0.7452
SSD [21]	0.5956	**0.9854**	0.7424	0.5746	**0.9519**	0.7166	0.5231	**0.8660**	0.6523
LSTM_Linear (ours)	0.7745	0.7843	0.7794	0.7368	0.7461	0.7414	0.6549	0.6632	0.6590
BiLSTM_Linear (ours)	0.8858	0.7849	0.8323	0.8077	0.7157	0.7589	0.6249	0.5537	0.5872
CNN_3_BiLSTM_Linear (ours)	0.9227	0.8731	0.8972	0.8820	0.8345	0.8576	0.7934	0.7507	0.7715
CNN_5_BiLSTM_Linear (ours)	**0.9244**	0.8747	0.8989	**0.8866**	0.8390	0.8621	0.8032	0.7600	0.7810
CNN_3_BiLSTM_CRF (ours)	0.9063	0.9291	0.9176	0.8726	0.8945	0.8833	0.7944	0.8143	0.8043
CNN_5_BiLSTM_CRF (ours)	0.9205	0.9320	0.9262	0.8847	0.8959	**0.8903**	**0.8035**	0.8136	**0.8085**

LSTM_Linear) outperforms the three traditional methods, showing the superiority of deep models. The LSTM_Linear mainly consists of two stacked unidirectional LSTMs and a linear classifier, and the line images resized to 30×250 are directly fed into the LSTMs.

SSD achieves high Precision at the expense of Recall, resulting in a low F-score, which means lots of characters are ignored by this method. R-fcn yields a good result at the IoU of 0.7, but its performance drops faster than our main model (CNN_5_BiLSTM_CRF) with the increase of IoU. Our method can still reach a high F-score about 0.81 at the IoU of 0.8, showing the effectiveness of the proposed method.

2) Contributions of different parts in the proposed network: In this section, we gradually modify the baseline model (LSTM_Linear) to the final model (CNN_5_BiLSTM_CRF) to investigate the contributions of different parts in the proposed network.

Contribution of bidirectional LSTM. The only modification from LSTM_Linear to BiLSTM_Linear is to add an opposite directional LSTM. This small change leads to a significant improvement of the performance, suggesting the importance of the information obtained from past and future when labeling sequence. Since there are many touched characters and multi-connected component characters in the dataset, the context information plays an important role.

Contribution of the features extracted by CNN. Instead of feeding the image into LSTMs directly, the CNN_3_BiLSTM_Linear applies a CNN with three convolutional layers (Conv1-Max pool-Conv4-Conv5 in Table II, the output dimension of Conv1 is changed to 16) to extract features and the last feature map is sent to LSTM. This modification brings another substantial improvements. It shows that the features extracted by CNN are more expressive than the raw image pixels.

Contribution of the depth of CNN. If we go deep with CNN layers from three to five (CNN_3_BiLSTM_Linear to CNN_5_BiLSTM_Linear, CNN_3_BiLSTM_CRF to CNN_5_BiLSTM_CRF), a slightly improvement can be observed. It is attributed to two aspects. One is that deeper models are more capable of feature extraction, and the other is that deep models have wider receptive fields which can capture more context information in original images.

Contribution of CRF. Replacing the linear classifier with CRF (CNN_3_BiLSTM_Linear to CNN_3_BiLSTM_CRF, CNN_5_BiLSTM_Linear to CNN_5_BiLSTM_CRF) yields the best results among the proposed methods. This shows that considering the dependencies across output labels is crucial to help make decisions.

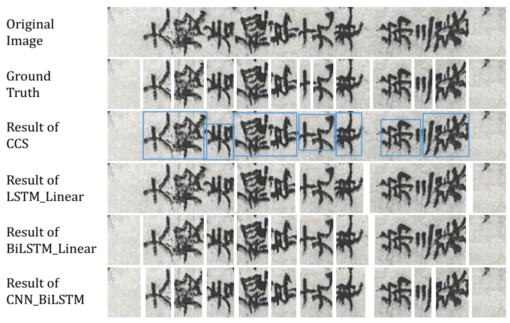

Figure 4: Segmentation results of different methods.

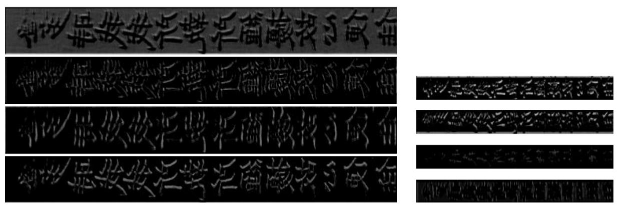
(a) Output of Conv1 (b) Output of Conv4

Figure 5: Output of different convolutional layers.

3) Case study: We give an example of testing image and results generated by differen methods in Fig.4. This line image is difficult for the traditional method CCS because of the touched characters. Unidirectional LSTM seems not sensible to the beginning of a character because it runs form left to right, while bidirectional LSTM performs better. Our main model (CNN_5_BiLSTM_CRF) yields a pretty good result (denoted as CNN_BiLSTM in Fig.4) for this image, although it is not completely correct. The results of other three CNN_BiLSTM based models are similar to the main model, thus we do not display them.

We can observe from Fig.4. that the best result also contains a mistake, '十' and '九' are not segmented, because they are extremely like a whole character from visual feature and size, and it is a typical error for our method.

4) Visualization of convolutional layers: To further understand what the CNN have learned from an input image, we visualize the output of different convolutional layers and display some channels of Conv1 and Conv4 in Fig.5. In the early stage, the filter seems to capture the edge of characters. With the increase of depth, the output of CNN becomes sparse, and only "important" information is remained.

V. Conclusion

In this paper, a neural network is proposed to perform the character segmentation in historical documents based on the line images obtained by projection profile. By converting the segmentation problem to sequence labeling problem, this approach provides a new perspective to deal with segmentation. The network mainly consists of three parts: a CNN for feature extraction, a LSTM for sequence modeling and a CRF for sequence labeling. A detailed discussion has been given to show the contributions of these three different parts. Experimental results demonstrate that our methods achieve superior or highly competitive performance, compared with traditional methods as well as other deep learning based methods.

A future direction would be to apply the proposed method to other segmentation problems such as musical score segmentation.

Acknowledgment

This work is supported by the projects of National Natural Science Foundation of China (No. 61573028 and 61472014), which is also a research achievement of Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology). We also thank the anonymous reviewers for their valuable comments.

References

[1] K. R. Michael Stauffer, Andreas Fischer, "Ensembles for graph-based keyword spotting in historical handwritten documents," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on.* IEEE, 2017, pp. 714–720.

[2] T. S. R. L. Tobias Gruuening, Gundram Leifert, "A robust and binarization-free approach for text line detection in historical documents," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on.* IEEE, 2017, pp. 236–241.

[3] T. Bluche, "Joint line segmentation and transcription for end-to-end handwritten paragraph recognition," in *Advances in Neural Information Processing Systems*, 2016, pp. 838–846.

[4] A. Ghorbel, J.-M. Ogier, and N. Vincent, "A segmentation free word spotting for handwritten documents," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on.* IEEE, 2015, pp. 346–350.

[5] F. Y. C.-L. L. Yue Xu, Wenhao He, "Page segmentation for historical handwritten documents using fully convolutional networks," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on.* IEEE, 2017, pp. 236–241.

[6] T. M. Breuel, "Robust, simple page segmentation using hybrid convolutional mdlstm networks," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on.* IEEE, 2017, pp. 733–740.

[7] B. Shi, X. Bai, and C. Yao, "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*, 2016.

[8] S. K. P. N. Stephen Rawls, Huaigu Cao, "Combining convolutional neural networks and lstms for segmentation-free ocr," in *Document Analysis and Recognition (ICDAR), 2017 14th International Conference on.* IEEE, 2017, pp. 155–160.

[9] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[10] R. Manmatha and N. Srimal, "Scale space technique for word segmentation in handwritten documents," *Lecture notes in computer science*, pp. 22–33, 1999.

[11] R. P. dos Santos, G. S. Clemente, T. I. Ren, and G. D. Cavalcanti, "Text line segmentation based on morphology and histogram projection," in *Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on.* IEEE, 2009, pp. 651–655.

[12] R. Chamchong and C. C. Fung, "Character segmentation from ancient palm leaf manuscripts in thailand," in *Proceedings of the 2011 Workshop on Historical Document Imaging and Processing.* ACM, 2011, pp. 140–145.

[13] M. W. A. Kesiman, J.-C. Burie, and J.-M. Ogier, "A new scheme for text line and character segmentation from gray scale images of palm leaf manuscript," in *Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on.* IEEE, 2016, pp. 325–330.

[14] Y.-L. Chen and B.-F. Wu, "A multi-plane approach for text segmentation of complex document images," *Pattern Recognition*, vol. 42, no. 7, pp. 1419–1444, 2009.

[15] N. Ouwayed and A. Belaïd, "A general approach for multi-oriented text line extraction of handwritten documents," *International Journal on Document Analysis and Recognition*, pp. 1–18, 2012.

[16] L. Likforman-Sulem and C. Faure, "Extracting text lines in handwritten documents by perceptual grouping," *Advances in handwriting and drawing: a multidisciplinary approach*, pp. 117–135, 1994.

[17] M. A. Moll and H. S. Baird, "Segmentation-based retrieval of document images from diverse collections." in *DRR*, 2008, p. 68150L.

[18] S. S. Bukhari, A. Azawi, M. I. Ali, F. Shafait, and T. M. Breuel, "Document image segmentation using discriminative learning over connected components," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems.* ACM, 2010, pp. 183–190.

[19] V. P. Le, N. Nayef, M. Visani, J.-M. Ogier, and C. De Tran, "Text and non-text segmentation based on connected component features," in *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on.* IEEE, 2015, pp. 1096–1100.

[20] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision.* Springer, 2016, pp. 21–37.

[22] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[24] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.