Text Segmentation for Japanese Historical Documents using Fully Convolutional Neural Network

HUNG TUAN NGUYEN[†1] CUONG TUAN NGUYEN[†1]
MASAKI NAKAGAWA[†1] ASANOBU KITAMOTO[†2,3]

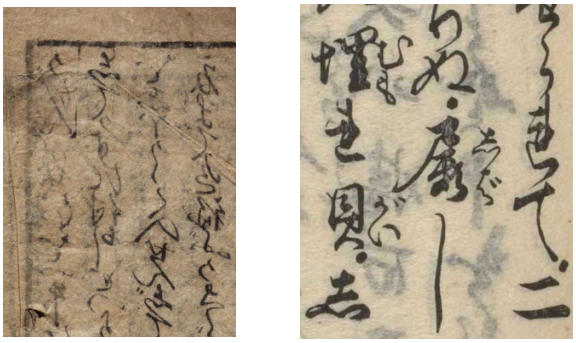
Abstract: We propose to use a Fully Convolutional Network (FCN) for text segmentation from Japanese historical document images. The trained FCN model has the ability to segment text pixels from raw document images with various background styles and image sizes. However, the demerit of FCN is the requirement of pixel-level ground-truth, which is expensive, especially for historical documents. By employing the Otsu local binarization method on each isolated character, we label every pixel of all document images belonging to the Pre-Modern Japanese Text (PMJT) database. Another problem is the imbalance between the number of background pixels and the number of text pixels. Thus, we multiply a weighted parameter to gradients based on the ratio between the number of background pixels and the number of text pixels during the training process.

Keywords: Text Segmentation, Fully Convolutional Neural Network, FCN, Japanese Historical Documents

1. Introduction

In recent years, many large historical document databases have been annotated and published in order to answer the demand for preserving historical documents and availing them for research without damaging physical documents [1]–[8]. This has raised several new challenges in document analysis and handwriting recognition field. For historical documents, some typical challenges such as damage, fade, show-through, various backgrounds add extra difficulties on top of anomalous deformation as well as a limited resource as shown in Fig. 1.

In the Edo period (1603-1868), Japanese historical documents were vertically written with brush or wood-block printed. There are other difficulties in analyzing them, for instance, vertical/horizontal guidelines and overlapping/touching between characters as shown in Fig. 2 and 3. Even experts have difficulties and take a long time to read these documents. The usual Optical Character Recognition (OCR) or Handwriting Text Recognition (HTR) systems cannot be used directly on the historical documents due to the above difficulties. Hence, our research focuses on the deep neural network based text segmentation method as a preprocessing step for OCR or HTR systems for historical documents.



a) Fade b) Show-through

Fig. 1. Samples of the anomalous deformation.

2. Methodology

We propose a method consisting of two parts: Fully Convolutional Network (FCN) and a pixel-level labeled database from character bounding boxes. FCN has been proposed to solve the semantic segmentation problem of general image [9]. It has inspired several studies in deep neural network based layout analysis for historical documents, which proved the efficiency of the deep neural network for solving the pixel-level classification tasks [10]–[12].

The output and input of the FCN are of the same size, the objective of FCN in pixel labeling is to create a pixel-level dense feature map. Thus, FCN could be considered as a convolution-based encoder-decoder network, where the encoder consists of multiple convolutional layers with pooling layers between them, and the decoder consists of multiple deconvolutional layers. The encoder usually comes from one of the well-known convolutional neural network for image classification task. The output of the FCN has the same spatial shape of the input image, and the depth of FCN output equals the number of classes/categories, for example, two if there are only two categories (background and text) as in this paper.

a) Vertical guide lines b) Horizontal guide lines

Fig. 2. Samples of guidelines.

†1 Tokyo University of Agriculture and Technology, Department of Computer and Information Sciences.
†2 ROIS-DS Center for Open Data in the Humanities
†3 National Institute of Informatics

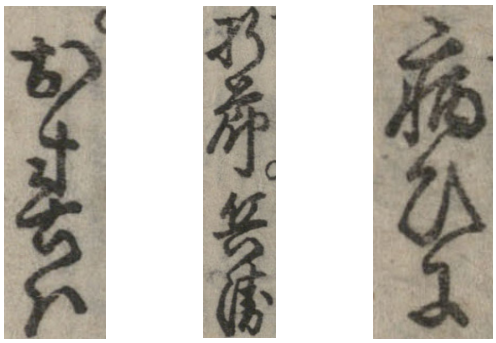


Fig. 3. Overlapping/touching samples.

Fig. 4 visualizes the structure of our proposed network, where the input is an RGB image of 500x500x3. Each Block Type 1 is a block of two convolutional layers with a kernel size of 3x3 followed by one max-pooling layer with a kernel size of 3x3 and a stride of 2x2. Each Block Type 2 consists of four convolutional layers with a kernel size of 3x3 followed by one max-pooling layer with a kernel size of 3x3 and a stride of 2x2. The first two blocks are Block Type 1 which aims to extract edges and details from an input image. Then, the next three blocks are Block Type 2 which are composed of more convolutional layers as well as a higher number of feature maps to capture high-level features such as the shape of ink strokes or even characters. After the above blocks, we employ two convolutional layers (CONV) with a kernel size of 3x3 and the last convolutional layers with a kernel size of 1x1x(#classes) which could be considered as a pixel type classifier. In this paper, #classes is two since each pixel will be a text or a background pixel. All of the above blocks and layers are known as an encoder.

Next, we employ deconvolutional layers (DE-CONV) to obtain the pixel classification at the same spatial size of the input images. The deconvolutional layers are also named as transposed convolutional layers. They play a role of upsampling layers to enlarge the feature maps through the transposed convolutional operators. In network structure, there are residual connections, which are the element-wise add operators (yellow arrows) between the convolution layers of the encoder and the deconvolution layers of the decoder in order to avoid the zero mapping by the identity mapping. As shown in Fig. 4, the encoder extracts the features by multiple levels from fine to coarse, and the decoder reconstructs the features from coarse to fine, where the high-level semantic information is still retained. Moreover, the weights of the encoder might be initialized and fixed by any pre-trained models such as VGG16 [13] or ResNet [14] on the ImageNet database which reduces the training time to converge.

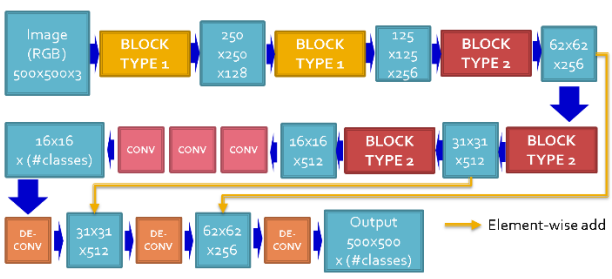


Fig. 4. The structure of the proposed network.

The previous researches required the pixel-level labeled databases in order to achieve the state-of-the-art performance on text/non-text segmentation. This kind of databases requires a considerable effort in both time and cost to label every pixel of document images. Moreover, it is unreasonable for historical documents due to the lack of experts in this field. Fortunately, there is a Japanese historical documents database with separate character bounding boxes [15]. We propose to generate the pixel-level labels by employing the Otsu binarization method on every character bounding box.

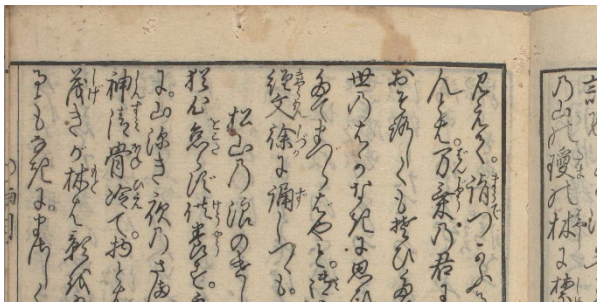
3. Experiments

3.1 Preparation of pixel-level labels

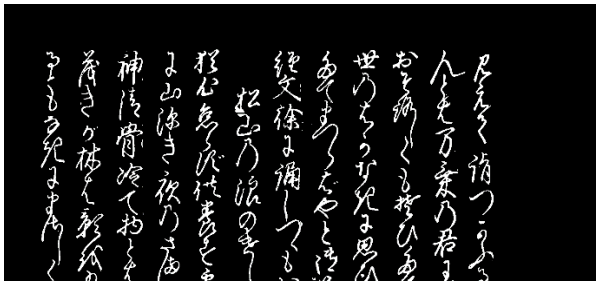
Inspired by the HBA database, where every single pixel is labeled as a text /figure/background pixel [8], we propose the following process to assign a label for every single pixel as background (0) or text (1). First of all, the pixels not covered by any bounding box are labeled as background pixels. Next, we employ the Otsu method to binarize every bounding box. The binarized black pixels is assigned as text pixels while other pixels are assigned as background pixels, as shown in Fig. 5.

3.2 Training FCN

First of all, we split the Pre-Modern Japanese Text (PMJT) database into two disjoint subsets: training and testing set consisting of 1,603 and 401 images, respectively. Due to the limitation of GPU memory, the input images have the spatial shape of 500-by-500. The size of input images does not cover the whole page of historical documents in the PMJT database. Thus, we randomly crop the squared regions of 500-by-500 from every page and feed them to the FCN as input images. During the training process with 100.000 epochs, we compute the cross-entropy losses at pixel-level and optimize them by Adam algorithm [16]. We train the proposed FCN four times with different initialization weights.



a) Raw image (RGB)



b) Pixel-level labels generated by Otsu local binarization.

Fig. 5. An example of pixel-level labels.

In our database, the number of background pixels is on average 28 times larger than the number of text pixels, which is a common challenge of pixel-level labeling task. Thus, we employ the different weights on the learning rate for the text and background pixels.

3.3 Evaluating and Visualizing results of FCN

To evaluate the trained model, we employ the following metrics: pixel accuracy, frequency weighted pixel accuracy, mean Intersection over Union (IoU) and frequency weighted IoU which measures the accuracy of prediction at pixel-level in several aspects.

$$\text{PixelAccuracy} = \left(\sum_{i\in\{0,1\}} pixel(i,i)\right) / \sum_{i\in\{0,1\}} total\_pixel(i) \quad (1)$$

$$\text{Freq. W. Accuracy} = \left(\sum_{i\in\{0,1\}} pixel(i,i)/total\_pixel(i)\right) \Big/ (\#num\_classes) \quad (2)$$

$$\text{Mean IoU} = \frac{\sum_{i\in\{0,1\}} pixel(i,i)/\left(total\_pixel(i) + \sum_{j\in\{0,1\}} pixel(j,i) - pixel(i,i)\right)}{(\#num\_classes)} \quad (3)$$

$$\text{Frequency Weighted Mean IoU} = \frac{\sum_{i\in\{0,1\}} total\_pixel(i) * pixel(i,i)/\left(total\_pixel(i) + \sum_{j\in\{0,1\}} pixel(j,i) - pixel(i,i)\right)}{\sum_{k\in\{0.1\}} total\_pixel(k)} \quad (4)$$

where *pixel*(*i, j*) is the number of pixels of class *i* predicted to belong to class *j*. Besides, *total_pixel*(*i*) is the total number of pixels belonging to class *i*. Since the number of text and background pixels are highly unbalanced, the weighted frequency measurements are used to compute the accuracy without unbalancing.

Table I shows the average results from four training times with different initialization of network parameters. The pixel accuracy is 91.93% while the frequency weighted pixel accuracy is slightly lower at 90.15%, which means that the noises in prediction are low. According to these results, the proposed FCN achieves more than 90% in pixel accuracy, which is a promising result in text segmentation task. In order to measure the ratio of correct prediction over the total number of pixels belonging to a class. The mean IoU is 82.31%, and the frequency weighted IoU is 85.45%. There are more than 80% of the pixel-level predictions matched with the binarization based labels.

Table I. FCN results of pixel segmentation task on PMJT database.

Metric	Result (%)
Pixel accuracy	91.93
Frequency Weighted pixel accuracy	90.15
Mean IoU	82.31
Frequency Weighted IoU	85.45

In order to visualize a test image having larger shape than 500-by-500, we split it into multiple non-overlapped sub-regions so that the largest shape should be 500-by-500. After binalization, we concatenate the sub-regions again to obtain the final output with the shape of the test image. Note that the FCN is not dependent on the shape or size of input images, which means that FCN can process an input image with the shape of 500-by-500 (constrained by the GPU memory) or smaller.

Fig. 6 shows the result of an image with common background and typical vertical writing style. Although almost document images are the same as that in Fig. 6, there are some exceptions as shown in the following figures. Fig. 7 shows the result on an image with a table-like layout, in which the location of characters, as well as the structure of the table, are not fixed. The proposed network performs well on both the vertical writing style and the table-like layout. Even though the performance on pixel accuracy is not completely perfect as shown in these two figures, these predictions seem good enough for further analyses and recognition tasks. Before applying the recognizer to these predictions, they need to be processed by the post-processing as presented in the next subsection.

a) Input image. b) Prediction.

Fig. 6. Result of FCN on a typical background image.

a) Input image b) Prediction

Fig. 7. Result of a table-like layout image.

Fig. 8 shows the result on an image with graphics/drawing. It is also interesting that our model segments some characters that are even not marked by people as shown in Fig. 9. These predictions suggest that the trained model converges at the general optimal solution, which means that it does not over-fit on the training set.

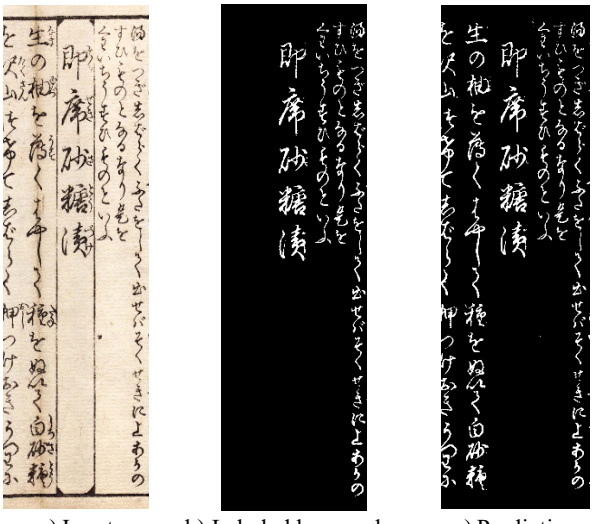
3.4 Post-processing

This section presents our simple post-processing method to segment the vertical text-lines based on the pixel-level text segmentation predictions. The x-projection pixel histogram is computed using the prediction as shown in Fig. 10. For the vertical text-line image, this histogram is useful. However, it is not easy to apply it to images with table-like structure. The x-projection pixel histogram is not robust for all cases in the PMJT database. Thus, we propose the following connected component based method.

First, the connected components among the prediction pixels are computed. Second, the components are grouped in case they are overlapped. The results from these two steps are the vertical text blocks as shown in Fig. 11, which are entirely recognized by our proposed recognizer in [15].

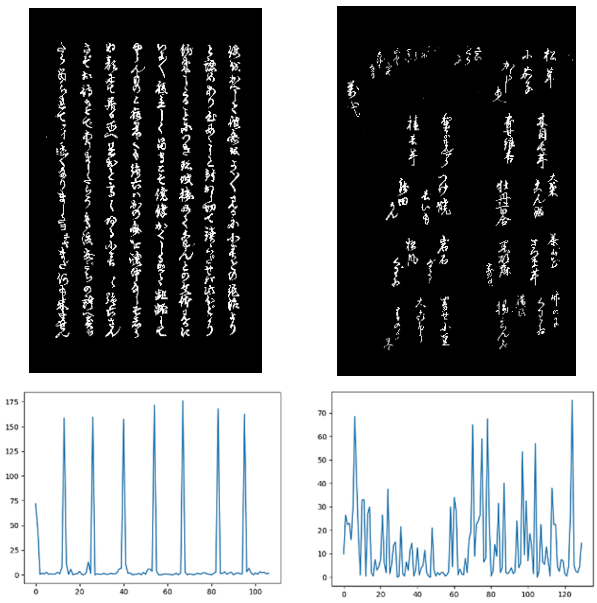
a) Input image. b) Prediction.

Fig. 8. Results of images with graphics.



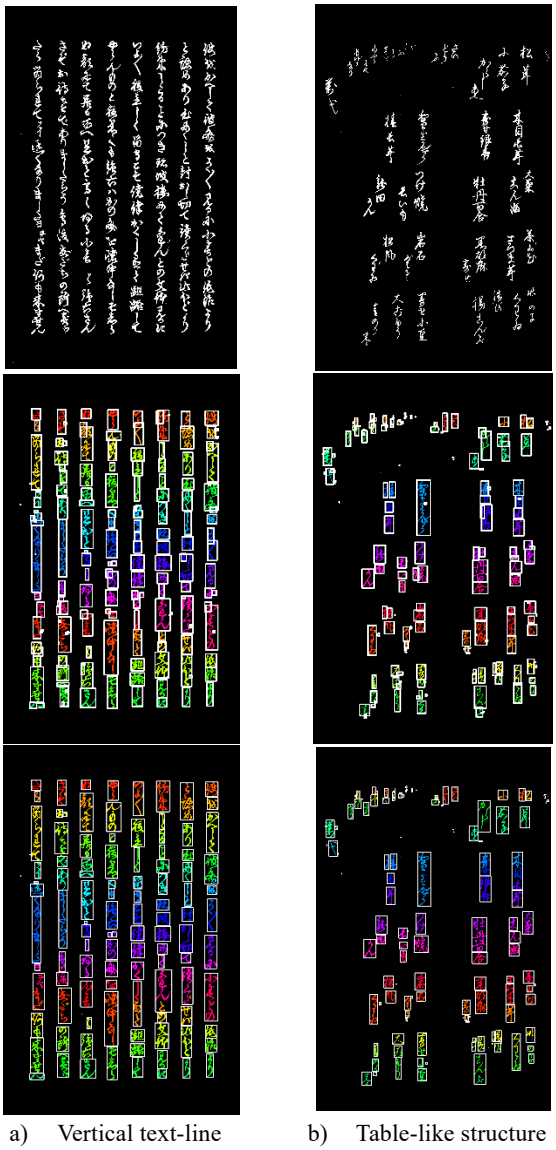
a) Input b) Labeled by people c) Prediction

Fig. 9. Prediction of unmarked characters.



a) Vertical text-line b) Table-like structure

Fig. 10. The x-projection pixel histogram.



a) Vertical text-line b) Table-like structure

Fig. 11. Connected components grouping method.

4. Discussions and Conclusions

Even the high accuracy on text/non-text segmentation at pixel-level, our model still requires the post-processing steps before employing an optical character recognition due to the lack of high-level semantic segmentation during training. We plan to employ multi-resolution FCN to predict the high-level segmentation as well as the pixel-level segmentation.

Shortly, we are extending our FCN model to cope with more categories such as graphics, annotation, guidelines, kana, and kanji. Then, the text segmented regions (kana or kanji) are recognized by our historical text-line recognizer [15]. It should be useful for researchers in historical document processing areas since the trained model could be used to process a large number of scanned images with less human effort.

Reference

[1] S. Nicolas, T. Paquet, and L. Heutte, "Enriching Historical Manuscripts: The Bovary Project," in *Document Analysis Systems VI*, S. Marinai and A. R. Dengel, Eds. Springer, Berlin, Heidelberg, 2004, pp. 135–146.

[2] B. Gatos, K. Ntzios, I. Pratikakis, S. Petridis, T. Konidaris, and S. J. Perantonis, "An efficient segmentation-free approach to assist old Greek handwritten manuscript OCR," *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 305–320, Feb. 2006.

[3] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *Int. J. Doc. Anal. Recognit.*, vol. 9, no. 2–4, pp. 139–152, 2007.

[4] A. Kitadai, J. Takakura, M. Ishikawa, M. Nakagawa, H. Baba, and A. Watanabe, "Document Image Retrieval to Support Reading Mokkans," in *Proceedings of the 8th IAPR International Workshop on Document Analysis Systems*, 2008, pp. 533–538.

[5] A. Fischer, H. Bunke, N. Naji, J. Savoy, M. Baechler, and R. Ingold, "The HisDoc project. automatic analysis, recognition, and retrieval of handwritten historical documents for digital libraries," in *Proceedings of the InterNational and InterDisciplinary Aspects of Scholarly Editing*, 2012, pp. 81–96.

[6] C. Papadopoulos, S. Pletschacher, C. Clausner, and A. Antonacopoulos, "The IMPACT dataset of historical document images," in *Proceedings of the 2nd International Workshop on Historical Document Imaging and Processing*, 2013, pp. 123–130.

[7] T. Van Phan, K. Cong Nguyen, and M. Nakagawa, "A Nom historical document recognition system for digital archiving," *Int. J. Doc. Anal. Recognit.*, vol. 19, no. 1, pp. 49–64, Mar. 2016.

[8] M. Mehri, P. Héroux, R. Mullot, J.-P. Moreux, B. Coüasnon, and B. Barrett, "HBA 1.0: A Pixel-based Annotated Dataset for Historical Book Analysis," in *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, 2017, pp. 107–112.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 28th IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[10] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, "Page segmentation of historical document images with convolutional autoencoders," in *Proceedings of 13th International Conference on Document Analysis and Recognition*, 2015, pp. 1011–1015.

[11] G. Renton, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, "Handwritten Text Line Segmentation Using Fully Convolutional Network," in *Proceedings of the 14th IAPR International Conference on Document Analysis and Recognition*, 2017, pp. 5–9.

[12] Y. Xu, F. Yin, Z. Zhang, and C.-L. Liu, "Multi-task Layout Analysis for Historical Handwritten Documents Using Fully Convolutional Networks," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 1057–1063.

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] H. T. Nguyen, N. T. Ly, K. C. Nguyen, C. T. Nguyen, and M. Nakagawa, "Attempts to recognize anomalously deformed Kana in Japanese historical documents," in *Proceedings of the 4th International Workshop on Historical Document Imaging and Processing*, 2017, pp. 31–36.

[16] D. P. Kingma and J. L. Ba, "Adam: a Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Acknowledgments

This research is partially supported by ROIS-DS-JOINT 027RP2018.