Segmenting Text in Japanese Historical Document Images using Convolutional Neural Networks

Hung Tuan Nguyen[1], Cuong Tuan Nguyen[1], Asanobu Kitamoto[2] and Masaki Nakagawa[1]
[1]Tokyo University of Agriculture and Technology
[2]National Institute of Informatics

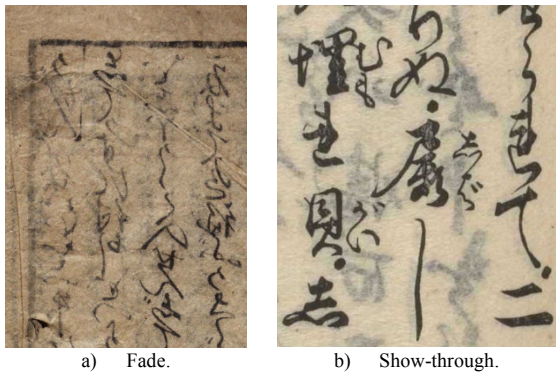
For historical document analysis and recognition, there exist many challenges such as damage, fade, show-through, anomalous deformation, various backgrounds, limited resources and so on. These challenges raise the demand for preprocessing historical document images. In this paper, we propose deep neural networks, named Pixel Segmentation Networks (PSNet) for text segmentation from Pre-Modern Japanese text (PMJT) historical document images. The proposed networks are used to segment pixels of text from raw document images with various background styles and image sizes, which is helpful for the later steps in historical document analysis and recognition. For preparing training patterns, we applied the Otsu local binarization method on every single character and extracted the pixel-level labels of all training document images. To evaluate the proposed networks, we used following two metrics: pixel-level accuracy (PlA) and the ratio of intersection over a union of the true test region and its detected region (IoU). Since there is the great imbalance between the number of background pixels and that of text pixels, we normalize the measurements by a weighted parameter based on the frequency of background and text pixels. Then, we made experiments on the PMJT database, which is randomly split into the training set of 1,556 images, validation set of 333 images and testing set of 333 images. The experiments show the best PlA of 98.75%, the frequency-weighted PlA of 95.27%, IoU of 87.89%, and the frequency-weighted IoU of 97.68% when 1,556 images are uses for training. Moreover, the performance of CED-PSNet12 is only degraded as little as around 2 percentage points even when under 100 images, 1/16 of the original training set are used.

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In recent years, many large historical document databases have been annotated and published in order to answer the demand for preserving historical documents and availing them for research without damaging physical documents [1]–[8]. Nguyen et al. applied deep neural networks to recognize Japanese historical text that were vertically written with brush or woodblock printed in the Edo period (1603-1868) [9]. They achieved the best results in a recognition contest[1] on short text of three deformed Kana characters and multiple text-lines of them, where deformed Kana is a set of 46 phonetic characters deformed from Chinese characters. They applied the Otsu binarization preprocessing on each character bounding box and improved the recognition rates since it reduces noises and other deformation effects.

However, it is difficult to employ the Otsu method without character bounding boxes to binarize a historical document due to several challenges such as damage, fade, show-through, anomalous deformation, different backgrounds, limited resources and so on. These challenges are common in historical documents, which add extra difficulties to document analysis and recognition as shown in Fig. 1. There are other challenges in analyzing them which are not found in contemporary documents, such as vertical or horizontal guidelines, complex layouts of characters and cursive writing through an entire text-line, as shown in Fig. 2 and 3. Even experts face difficulties and take a long time to read these documents. Obviously, the usual Optical Character Recognition (OCR) or Handwriting Text Recognition (HTR) systems cannot be used directly on the historical documents due to these problems.

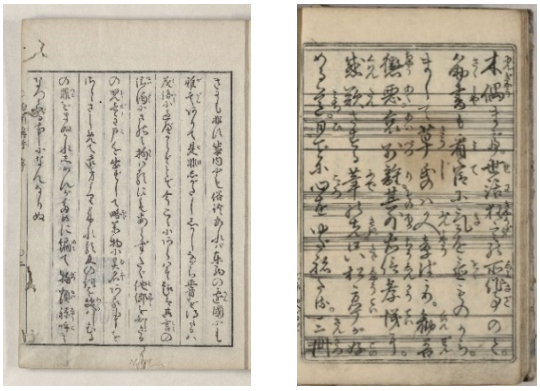
In this paper, we focus on the deep neural network-based text segmentation as a preprocessing step of OCR or HTR for historical documents. We present a method that classifies text pixels from other pieces of information such as background, noise, figure pixels, so that the process of text recognition can concentrate only on text pixels.



a) Fade. b) Show-through.

Figure 1. Samples of the anomalous deformation.

[1] https://sites.google.com/view/alcon2017prmu (in Japanese)



a) Vertical guidelines. b) Horizontal guidelines.

Figure 2. Samples of guidelines.

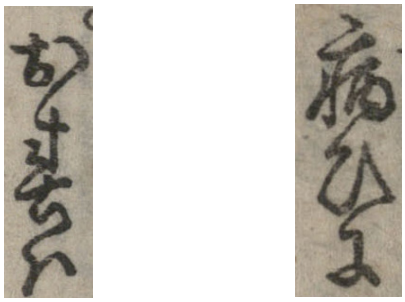


Figure 3. Overlapping/touching samples.

2. Related works

During the last decade, there have been many studies for text segmentation on historical document images, which usually consist of binarization and connected component analysis to extract text areas based on prior knowledge, heuristic rules or machine learning methods. Phan et al. applied the Voronoi diagram for connected-component boundary detection and neighborhood representation [10]. They used prior knowledge and heuristic rules to gather adjacent Voronoi areas that form character areas in historical document images. Bukhari et al. used a multi feedforward neural network to classify connected components by their extracted features such as orientation, height, width, foreground, relative distance with neighborhood [11]. For Japanese historical woodblock printed document, Panichkriangkrai et al. proposed a two-step method consisting of vertical text-line segmentation and character segmentation [12]. Their text-line segmentation is based on vertical projection on binarized images. They applied heuristic rules to segment characters from the segmented vertical text-lines. Those studies were successfully used for printed historical document images without various backgrounds or heavy noise [13]. However, their methods need to redesign the heuristic rules for different databases because they used data-dependent information to make their classifiers. Thus, there is still a demand for robust text segmentation from historical document images.

In recent years, deep neural networks have been studied and applied for semantic segmentation tasks, which solve the segmentation problem for general images. A few studies have also been reported for historical documents using deep neural networks, with their effectiveness of the deep neural networks shown for solving the pixel-level classification tasks [14]–[16]. Chen et al. presented an unsupervised method using a convolutional autoencoder (CAE) to learn and extract features from pixels of historical document images [14]. They used a Support Vector Machine (SVM) to classify those extracted features into four categories: periphery, background, text block or decoration. It replaced the traditional handcrafted features by automatically learned features, but it still depends on a simple classifier. Renton et al. and Xu et al. proposed end-to-end deep neural networks for segmenting text-lines and characters from historical document images [15], [16]. Their networks outperformed the previous research using traditional handcrafted features. However, those methods were designed to achieve the best accuracy at text-line level segmentation. In our research, we focus on exploring the performance of deep neural networks for pixel-level segmentation on historical document images.

3. Proposed Method

In order to solve the problem of text segmentation at pixel-level in historical documents, we propose deep neural networks, named Pixel Segmentation Networks (PSNet) of two types. The input of PSNet is a scanned image from historical documents and the output is a segmented image of the same size where each pixel is labeled as text, figure or background. The training scheme of PSNet is shown in Fig. 4. For PSNet, pixel-level labels are required for supervised training. However, the Pre-Modern Japanese Text (PMJT) database contains only character bounding boxes instead of pixel-level labels. Therefore, we apply the Otsu binarization on every character bounding box to obtain the text pixels while other pixels are labeled as background. Note that the character bounding boxes are not required during the testing phase. Although the Otsu method does not provide the perfect ground-truth for the image of a whole page, it is quite reliable for an area within each character bounding box. In the following subsections, we present the details of each network architecture.

3.1. Convolutional Encoder-Decoder based PSNet

The basic type of PSNet is a Convolutional Encoder-Decoder model (CED-PSNet), which is inspired by the Fully Convolutional Network [17], to solve the semantic segmentation problem for general images. A CED-PSNet consists of four components: encoder, feature aggregator, classifier and decoder.

The encoder extracts features from a raw image (input image) by multiple convolutional layers with pooling layers between them. The first architecture of PSNet is shown as CED-PSNet16 in Table I, inspired by

VGGNet-16 [18]. Since this architecture consists of more than 47.43 million parameters, it may overfit to a small database as PMJT. Moreover, it has five max pooling layers to reduce the spatial size of an input image by 2^5 times, which also may cut off important information. The following two architectures, CED-PSNet12 and CED-PSNet9, are derived from the first one by reducing the number of parameters and keeping more information. The denotations for convolution blocks (CONV Block) in Table I are formatted as [kxkxN_{fm}]xN_l where kxk is the kernel size, N_{fm} is the number of feature maps and N_l is the number of stacked layers. At the end of each CONV Block, there is a Max Pooling layer (MaxPool) with the kernel size of 2x2 and the stride of 2.

The feature aggregator follows the encoder. It consists of a large receptive field convolution layer with the kernel size of 8x8 to capture wide context information and another convolution layer with the kernel size of 1x1 to transform context features. The classifier is a convolution-based classifier with the kernel size of 1x1 and the depth of #classes, where #classes is the number of classes that pixels belong to, which uses the aggregated features as input. Here, #classes is two if there are only two categories (text and background) and three if there are three categories (text, figure and background). The classified output is at a low resolution that should be upsampled back to the original resolution.

The decoder reconstructs the output back to the spatial shape of the input image. It consists of multiple deconvolutional layers (DECONV) to produce a pixel classification at the same spatial size as the input image. The deconvolutional layers are also named transposed convolutional layers. They play a role of up-sampling to enlarge the feature maps through transposed convolutional operators. The denotations of DECONV in Table I are formatted as kxkxN_{fm},u:u_r, where u_r is the upsampling rate. CED-PSNet has residual connections, which are element-wise addition operators (green arrows) between the convolution layers of the encoder and the deconvolution layers of the decoder in order to detect text pixels by features from multiple scales.

As shown in Fig. 5, the encoder extracts features by multiple levels from fine to coarse, and the decoder reconstructs features from coarse to fine, where the high-level semantic information is still retained.

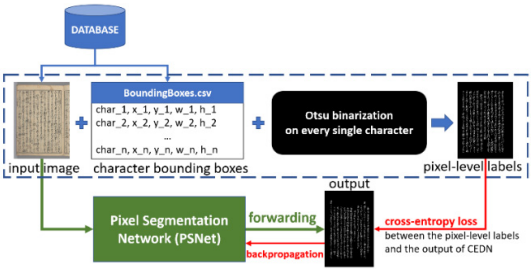


Figure 4. The overflow of PSNet training.

TABLE I. DIFFERENT ARCHITECUTURES OF CED-PSNET.

	CED-PS Net16	CED-PS Net12	CED-PSNet9
Input shape	*512x512x3*		
CONV Block 1 *Output shape*	[3x3x64]x2 *256x256x64*	[3x3x64]x3 *256x256x64*	[3x3x64]x2 *256x256x64*
CONV Block 2 *Output shape*	[3x3x128]x2 *128x128x128*	[3x3x128]x3 *128x128x128*	[3x3x128]x2 *128x128x128*
CONV Block 3 *Output shape*	[3x3x256]x3 *64x64x256*	[3x3x128]x3 *64x64x128*	[3x3x128]x2 *64x64x128*
CONV Block 4 *Output shape*	[3x3x512]x3 *32x32x512*		
CONV Block 5 *Output shape*	[3x3x512]x3 *16x16x512*		
Encoded feature maps shape	***16x16x512***	***64x64x128***	***64x64x128***
Feature aggregator	[8x8x1024] [1x1x1024]		
Classifier	[1x1x#classes]		
Output shape	*16x16x*#class	*64x64x*#classes	
DECONV 1	3x3x512,u:2	3x3x128,u:2	
DECONV 2	3x3x256,u:2	3x3x64,u:2	
DECONV 3	3x3x#classes, u:8	3x3x#classes, u:2	
Output shape	*512x512x#classes*		
Total #parameters	>47.43M	>10.17M	>9.95M

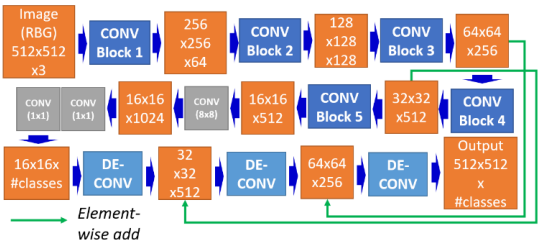


Figure 5. Structure of the proposed CED-PSNet16.

3.2. Dilated Convolution based PSNet

As mentioned above for CED-PSNet, MaxPool at the end of each CONV Block reduces the spatial size of feature maps to produce scale and orientation invariant results. It is useful for the image classification task because the produced feature maps are generalized. For pixel segmentation, however, these max pooling layers may cause a loss of information through each CONV Block. This is the reason for introducing element-wise addition operators in CED-PSNet. We replace the CONV Block using convolution layers and MaxPool by dilated convolution layers, which is proposed for semantic segmentation [19].

A dilated convolution block (Dilated CONV) is denoted as [kxkxN_{fm};d:d_r]xN_l where d_r is the dilation rate. Dilated convolution is considered as a general form of the usual convolution, which corresponds to the dilation rate of 1 (shown on the left in Fig. 6). As the dilation rate increases, a larger receptive field is

obtained with the same kernel-size for convolution as shown in Fig. 6. There are two main merits: the receptive field is enlarged at the original resolution with the same number of parameters and the depth of the network is reduced by eliminating the DECONV layers. Table II shows details of the two network architectures DC-PSNet12 and DC-PSNet9. However, the number of computing operators of a dilated convolutional network is much larger than that of an equivalent common convolutional network, because the spatial size of feature maps in a dilated convolutional network is not reduced by MaxPool. Fig. 7 presents our Dilated Convolution PSNet12 (DC-PSNet12) and DC-PSNet9, which are designed to capture sub-regions of the same sizes as the above CED-PSNet12 and CED-PSNet9, respectively.

3.3. Training patterns preparation

During the training process, there are two subprocesses: pixel-level ground-truth preparation (dashed-line rectangle) and PSNet training by input images and generated ground-truth as shown in Fig. 4. The previous research required the pixel-level labeled databases in order to achieve the state-of-the-art performance on text/non-text segmentation. Such databases require a considerable effort in both time and cost to label every pixel of the historical document image. Moreover, this is an unreasonable task for historical documents due to a lack of experts.

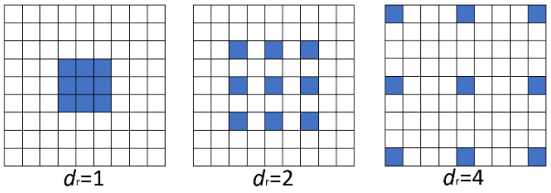


Figure 6. Dilation rates in DC-PSNet12.

TABLE II. DIFFERENT ARCHITECTURES OF DC-PSNET.

	DC-PSNet12	DC-PSNet9
Input shape	*512x512x3*	
Dilated CONV 1 *Output shape*	[3x3x64;d:1]x3 *512x512x64*	[3x3x64;d:1]x2
Dilated CONV 2 *Output shape*	[3x3x128;d:2]x3 *512x512x128*	[3x3x128;d:2]x2
Dilated CONV 3 *Output shape*	[3x3x128;d:4]x3 *512x512x128*	[3x3x128;d:4]x2
Feature aggregator	[8x8x1024] [1x1x1024]	
Classfier	[1x1x#classes]	
Output shape	*512x512x#classes*	
Total #parameters	>10.10M	>9.88M

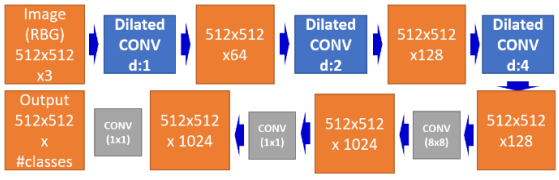


Figure 7. Structure of the proposed DC-PSNet12.

The PMJT database has separate character bounding boxes [9], which are useful to generate the pixel-level ground-truth.

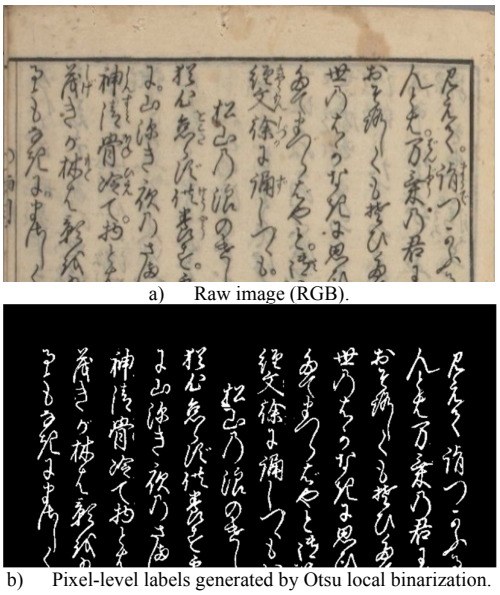
We employ the Otsu binarization method on every character bounding box [20]. First, all pixels not covered by any bounding box are labeled as background pixels. Secondly, the binarized pixels are assigned as text pixels (white pixels) while other pixels are assigned as background pixels (black pixels), as shown in Fig. 8. Thus, every single pixel is labeled background (0) or text (1), which is used later as the pixel-level ground-truth. Although the Otsu method does not provide the perfect ground-truth, it is quite reliable as it is applied to each character bounding box, which makes the method work very well without a large imbalance of black and white pixels. We confirmed the generated pixel-level ground-truth visually.

4. Experiments

In this section, we present the database and two main metrics to evaluate PSNet. In the following subsections, we describe the details of our experiments to evaluate the performance of different proposed network architectures for text segmentation at the pixel-level. In Section 4.3, we classify pixels into two classes: text and non-text pixels where non-text pixels consist of background and figure. In Section 4.4, we classify pixels into three classes: text, figure and background. In Section 4.5, we limit the size of the original training set by 1/2 to 1/16 to show the robustness of the proposed networks to smaller sizes of the training set.

4.1. Database

Since late 2016, Center for Open Data in the Humanities (CODH) has published an open access database of Japanese historical documents named Pre-Modern Japanese Text (PMJT). The PMJT database



a) Raw image (RGB).

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

Figure 8. An example of pixel-level labels.

contains 2,222 scanned images with 403,242 character bounding boxes. In this research, we applied our proposed networks to the PMJT database. We randomly split the database into three disjoint subsets: training, validation and testing sets, which consist of 1,556, 333 and 333 images, respectively. To make the experiments on the three-class segmentations, we inspected and drew bounding boxes of figures in 2,222 images in PMJT. Then, all the bounding boxes for figures were used to label the figure pixels. For the experiments in Section 4.5, we randomly selected 778, 389, 194, and 97 samples from the training set (1,556 images) with the ratio of 1/2, 1/4, 1/8 and 1/16 of the training set, respectively.

4.2. Settings

Due to the limitation of GPU memory, the size of the input images is set to 512x512. This size of input, however, cannot cover the whole page of historical documents in the PMJT database. Therefore, squared regions of 512x512 are randomly cropped from every page and fed to PSNet as input images. During the training process with 100,000 epochs, the cross-entropy loss is computed at the pixel level between the predicted output and the generated pixel-level ground-truth. The cross-entropy loss is optimized by the Adam algorithm [21]. Each proposed network architecture is trained four times with different initialization weights to obtain the best result. In the PMJT database, the number of background pixels is around 18.5 times larger than that of text pixels, which is a common challenge for the pixel-level labeling task. Therefore, we apply different learning rates for the text and background pixels, which are 0.005 and 0.00025, respectively.

4.3. Performance from different networks on two classes

To evaluate segmentation performance on two classes (text and background), we use two metrics: pixel-level accuracy (PIA) as shown in Eq. (1) and intersection over union (IoU) as shown in Eq. (2), which is the percentage of overlap between true test regions and detected regions.

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

$$\text{IoU} = \frac{1}{2}\sum_{i\in\{0,1\}} \frac{pixel(i,i)}{total\_pixel(i)+\sum_{j\in\{0,1\},j\neq i} pixel(j,i)} \quad (2)$$

where *pixel*(*i*, *j*) is the number of pixels of class *i* predicted to belong to class *j*. Also, *total_pixel*(*i*) is the total number of pixels belonging to class *i*.

There is an imbalance between the numbers of background pixels and text pixels, which is eliminated by normalizing these two metrics by the frequency.

$$\text{Frequency} - \text{Weighted PIA} = \left(\sum_{i\in\{0,1\}} pixel(i,i)/total\_pixel(i)\right)\Big/2 \quad (3)$$

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

Table III shows the results of different network architectures with two main metrics (Eq. (1) and (2)), as well as their normalized forms (Eq. (3) and (4)). The high accurate results from the proposed networks prove that CED- and DC-based PSNets extract not only local information but also context information to classify every pixel. CED-PSNet12 and DC-PSNet12 perform better than CED-PSNet9 and DC-PSNet9, respectively, through all metrics. Their performances have only some slight differences, which shows that DC-PSNet12 is equivalent with CED-PSNet12. On the other hand, the shallower CED-PSNet12 achieved better performance than the deeper network CED-PSNet16 probably because the former unlikely overfit to a small database as PMJT. For CED-PSNet9 and DC-PSNet9, their PIAs are almost the same but CED-PSNet9 is better on frequency-weighted PIA, IoU and frequency-weighted IoU, which means that the CED-PSNet9 predicts text pixels more accurately than DC-PSNet9.

4.4. Performance on three classes

To evaluate the segmentation performance for the three classes (text, figure and background), we use PIA. Eq. (5) defines the general formula to compute PIA for the i^{th} class:

$$\{i^{th}\} - \text{PIA} = \left(\sum_{i\in output\_set} pixel(i,i)\right)/\sum_i total\_pixel(i) \quad (5)$$

where output_set={Text, Figure, Background}. To simplify the denotation, the text, figure and background PIAs are abbreviated as T-PIA, F-PIA and B-PIA, respectively.

TABLE III. TWO-CLASS SEGMENTATION RESULTS OF DIFFERENT NETWORK ARCHITECTURES ON PMJT DATABASE.

Metrics	Result (%) by different networks				
	CED-PSNet16	CED-PSNet12	CED-PSNet9	DC-PSNet12	DC-PSNet9
Pixel-level Accuracy (PIA)	92.89	**98.75**	97.45	98.60	97.78
Frequency-Weighted PIA	59.09	94.58	93.19	**95.27**	91.94
Intersection over Union (IoU)	46.05	**87.89**	82.66	84.63	79.84
Frequency-Weighted IoU	84.44	**97.68**	95.79	97.45	96.11

TABLE IV. THREE-CLASS SEGMENTATION RESULTS OF DIFFERENT NETWORK ARCHITECTURES ON PMJT DATABASE.

Metrics	Result (%) by different networks				
	CED-PSNet16	CED-PSNet12	CED-PSNet9	DC-PSNet12	DC-PSNet9
Pixel-level Accuracy (PIA)	91.38	**96.79**	96.08	96.73	96.84
Text pixel-level accuracy (T-PIA)	21.75	88.31	83.53	**93.64**	94.05
Figure pixel-level accuracy (F-PIA)	5.62	49.82	33.35	**50.62**	40.74
Background pixel-level accuracy (B-PIA)	98.31	98.21	**98.40**	98.25	98.34

Text segmentation results of different network architectures on PMJT database. Table IV presents the results of three-class segmentation by different network architectures. PlAs of CED-PSNet12, CED-PSNet9, DC-PSNet12 and DC-PSNet9 are almost the same. However, the three metrics (T-PlA, F-PlA and B-PlA) present differences in their performances. First, T-PlAs and F-PlAs of DC-PSNet12 and DC-PSNet9 are higher than those of CED-PSNet12 and CED-PSNet9, respectively. The DC-PSNet architectures seem to be better at text segmentation when there is another class besides the two classes: text and background. For figure pixels, all network architectures do not predict well, i.e., only from 33.35% to 50.62%, which might be due to the fact that there are not many figure pixels compared with text or background pixels. The number of figure pixels is 0.173×10^9 while those of text and background pixels are 0.642×10^9 and 11.886×10^9, respectively. Moreover, the average size of figures in the PMJT database is much larger than the average size of characters. The receptive field size of PSNet designed for text pixel segmentation might not be large enough to cover the figure regions.

4.5. Performance on different sizes of training set

In order to evaluate the efficiency of PSNet with a smaller number of training samples, we trained CED-PSNet12 and DC-PSNet12 with smaller sizes of training sets, the results of which are shown in Table V and VI, respectively. Those networks are selected because they achieved the best performances compared with other networks. Note that the following results are on the two classes (text and background). For CED-PSNet12, PlA slightly decreases by 1.21 percentage points while the number of training samples is reduced from 1,556 images to only 97 images.

TABLE V. TEXT SEGMENTATION RESULTS BY CED-PSNET12 USING DIFFERENT SIZES OF TRAINING SET.

Metrics	Result (%) by training with different number of training samples				
	97 (1/16)	194 (1/8)	389 (1/4)	778 (1/2)	1,556
Pixel-level Accuracy (PlA)	96.12	97.50	98.17	98.31	98.75
Frequency-Weighted PlA	90.07	91.52	93.67	93.83	94.58
Intersection over Union (IoU)	80.21	83.33	85.17	86.51	87.89
Frequency-Weighted IoU	94.86	95.27	96.88	97.05	97.68

TABLE VI. TEXT SEGMENTATION RESULTS BY DC-PSNET12 USING DIFFERENT SIZES OF TRAINING SET.

Metrics	Result (%) by training with different number of training samples				
	97 (1/16)	194 (1/8)	389 (1/4)	778 (1/2)	1,556
Pixel-level Accuracy (PlA)	95.51	97.66	98.15	98.24	98.60
Frequency-Weighted PlA	90.57	92.11	94.74	94.82	95.27
Intersection over Union (IoU)	73.13	79.52	81.21	81.66	84.63
Frequency-Weighted IoU	93.05	95.91	96.82	96.94	97.45

IoUs are from 87.73% to 94.27%, while frequency-weighted IoUs are from 96.18% to 98.33%, which means that more than 87% of the pixel-level predictions matched with the ground-truth labels. DC-PSNet12 has the same performance as CED-PSNet12 when it is trained on the whole training set. Its PlA decreases by 3.09 percentage points compared with 2.63 percentage point of decrease by CED-PSNet12 when only 1/16 training samples are used. The frequency-weighted PlA of DC-PSNet12, however, is higher than that of CED-PSNet12, which means that DC-PSNet12 correctly predicts more text pixels than CED-PSNet12. Thus, CED-PSNet12 and DC-PSNet12 are appropriate for segmenting text even when only a small number of training images are available.

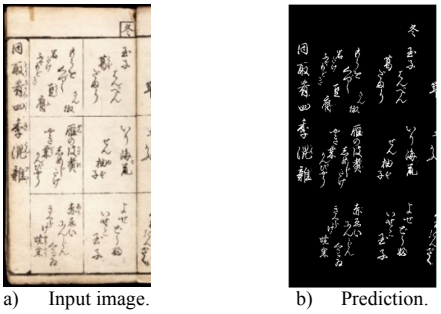
4.6. Visualization results

In order to visualize a test image having a larger shape than 512x512, we split it into multiple non-overlapping sub-regions so that the largest shape should be 512x512. After binarization, we concatenated the sub-regions again to obtain the final output with the shape of the test image. Note that PSNet is not dependent on the shape or size of input images, which means that it can process an input image of the size 512x512 (constrained by the GPU memory) or smaller. Fig. 9 shows the result of an image with a common background and a typical vertical writing style. Although most document images are the same as in Fig. 9, there are some exceptions, as shown in Figs. 10 and 11. Fig. 10 shows the result on an image with a table-like layout in which the character locations, as well as the table structure, are not fixed. The proposed networks perform 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 reasonable for further analyses and recognition tasks. Fig. 11 shows the result of an image with graphics/drawing. It is also interesting that our model even segments some characters that are not marked by people, as shown in Fig. 12. These predictions suggest that the trained model converges at the general optimal solution and does not over-fit on the training set.

a) Input image. b) Prediction.

Figure 9. Result of CED-PSNet9 on a typical background image.

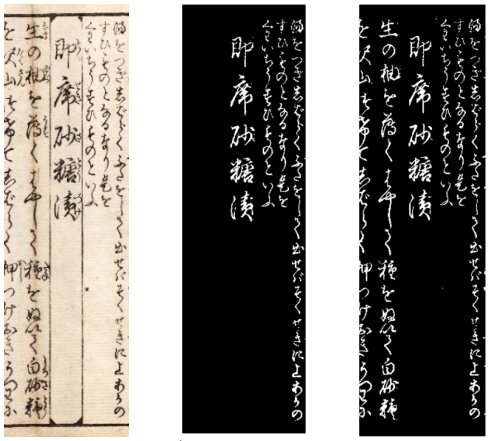


a) Input image. b) Prediction.

Figure 10. Result of CED-PSNet12 on a table-like layout image.

a) Input image. b) Prediction.

Figure 11. Result of CED-PSNet12 on an image with graphic layout.

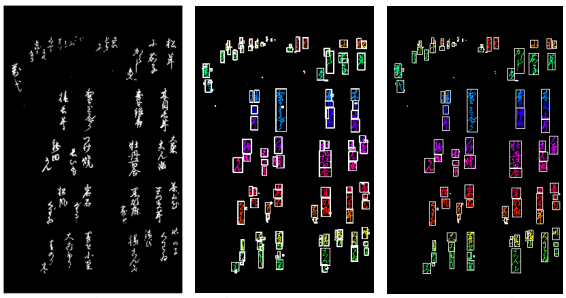


a) Input image. b) Labeled from PMJT. c) Prediction.

Figure 12. Result of CED-PSNet12 on an example where some characters are unlabeled in PMJT.

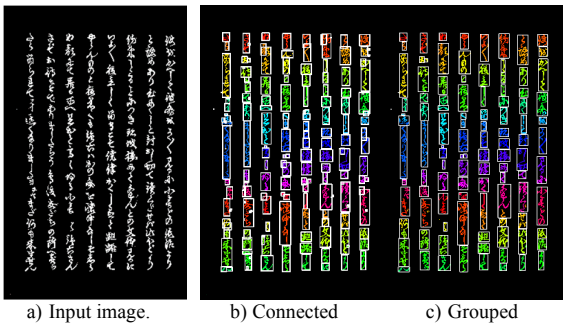
5. Discussions

As mentioned in the above Introduction, the text segmentation should be useful for OCR and HTR systems. It requires the vertical text-line segmentation since the OCR and HTR systems perform better on single text-line rather than on multiple text-line [9]. For segmenting the vertical text-lines based on the pixel-level text segmentation predictions, we employ the connected-component analysis (a bottom-up approach). First, the connected components among the prediction pixels are computed. Secondly, the components are grouped in case they overlap. The connected component-based method seems appropriate for both vertical text-line and table-like layouts. The results of the connected component and grouping are shown in Fig. 13 and 14 as the vertical text blocks, which are entirely appropriate to be recognized by the previous recognizers. Fig. 13 shows the connected components extracted from the text segmentation predictions even these text blocks are vertical unaligned. These results seem to be linked with high-level segmentation, which is related to character, sentence or text-line, and so on. Even when a high accuracy on text/non-text segmentation at pixel-level is achieved, our model still requires the text-line segmentation process before employing an optical character recognition due to a lack of high-level semantic segmentation during training. For future work, we will use end-to-end text-line recognizers to recognize text regions without character segmentation. It should be useful for researchers in the historical document processing area since a trained model could be used to process an enormous number of scanned images without requiring large human effort.



a) Input image. b) Connected components. c) Grouped components.

Figure 13. Connected-components grouping method on an image with table-like layout.



a) Input image. b) Connected components. c) Grouped components.

Figure 14. Connected-components grouping method on an image with vertical text-line layout.

6. Conclusions

This paper presented Pixel Segmentation Networks (PSNet) to segment text at the pixel level from Japanese historical document images stored in the Pre-Modern Japanese Text (PMJT) database. We compared two promising types of networks: Convolutional Encoder-Decoder (CED-PSNet) and Dilated

Convolution (DC-PSNet). Since the PMJT database did not contain pixel labels for text segmentation, we generated them using the Otsu binarization method on each character bounding box. First, we evaluated the performances of different network architectures on two categories (text and background pixels) by two main metrics: pixel-level accuracy (PlA) and Intersection over Union (IoU), as well as their normalized forms: frequency-weighted PlA and frequency-weighted IoU. The experiments showed the best PlA of 98.75% by CED-PSNet12 and the frequency-weighted PlA of 95.27% by DC-PSNet12. The highest mean IoU is 87.89%, and the frequency-weighted IoU is 97.68% by CED-PSNet12. CED-PSNet12 and DC-PSNet12 have similar accuracies probably because their structures are designed to capture the same size of receptive field, which proves that CED- and DC-based PSNets extract not only local information but also context information to classify every pixel. On the other hand, the shallower CED-PSNet12 achieved better performance than the deeper network CED-PSNet16 probably because the former unlikely overfit to a small database as PMJT.

Secondly, we evaluated different PSNet architectures on three categories (text, figure and background pixels) using Text-PlA (T-PlA), Figure-PlA (F-PlA) and Background-PlA (B-PlA). CED-PSNet12 and DC-PSNet12 achieved the same PlA but DC-PSNet12 achieved the best performance on T-PlA (93.64%) and F-PlA (50.62%). The F-PlA was low because the receptive field size of PSNet designed for text pixel segmentation might not be large enough to cover the figure regions which are much larger than the text regions. Thirdly, we evaluated the dependency on the training set size of PSNet because historical document databases often do not store a large number of training samples. The performance of CED-PSNet12 slightly drops around 2 percentage points even with only a small number of training samples under 100 training samples. Thus, CED-PSNet12 is expected to be applicable to other low-resource historical documents.

Acknowledgments

We would like to express our thanks to Prof. Bipin Indurkhya for discussing the classification method and improving the presentation. This research is partially supported by the Grant-in-Aid for Scientific Research (S)-18H05221 and (A)-18H03597 as well as ROIS-DS-JOINT 027RP2018.

References

[1] S. Nicolas, T. Paquet, and L. Heutte, “Enriching Historical Manuscripts: The Bovary Project,” in Document Analysis Systems VI, 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 Proc. of the 8th IAPR Int'l 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 Proc. 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 Proc. of the 2nd International Workshop on Historical Document Imaging and Processing, 2013, pp. 123–130.

[7] T. Van Phan, K. C. 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 Proc. of the 4th International Workshop on Historical Document Imaging and Processing, 2017, pp. 107–112.

[9] 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 Proc. of the 4th International Workshop on Historical Document Imaging and Processing, 2017, pp. 31–36.

[10] T. Van Phan, B. Zhu, and M. Nakagawa, “Development of Nom character segmentation for collecting patterns from historical document pages,” in Proc. of the 1st Workshop on Historical Doc. Imaging and Processing, 2011, pp. 133–139.

[11] S. S. Bukhari, T. M. Breuel, A. Asi, and J. El-Sana, “Layout Analysis for Arabic Historical Document Images Using Machine Learning,” in Proc. of the 13th Int'l Conf. on Frontiers in Handwriting Recognition, 2012, pp. 639–644.

[12] C. Panichkriangkrai, L. Li, and K. Hachimura, “Character segmentation and retrieval for learning support system of Japanese historical books,” in Proc. of the 2nd Int'l Workshop on Historical Doc. Imaging and Processing, 2013, pp. 118–122.

[13] A. Antonacopoulos, C. Clausner, C. Papadopoulos, and S. Pletschacher, “Historical Document Layout Analysis Competition,” in Proc. of the 11th Int'l Conf. on Document Analysis and Recognition, 2011, pp. 1516–1520.

[14] K. Chen, M. Seuret, M. Liwicki, J. Hennebert, and R. Ingold, “Page segmentation of historical document images with convolutional autoencoders,” in Proc. of the 13th Int'l Conf. on Document Analysis and Recognition, 2015, pp. 1011–1015.

[15] G. Renton, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, “Handwritten Text Line Segmentation Using Fully Convolutional Network,” in Proc. of the 14th Int'l Conf. on Document Analysis and Recognition, 2017, pp. 5–9.

[16] Y. Xu, F. Yin, Z. Zhang, and C.-L. Liu, “Multi-task Layout Analysis for Historical Handwritten Documents Using Fully Convolutional Networks,” in Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence, 2018, pp. 1057–1063.

[17] E. Shelhamer, J. Long, and T. Darrell, “Fully Convolutional Networks for Semantic Segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 4, pp. 640–651, 2017.

[18] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. of the 3rd Int'l Conf. on Learning Representations, 2015.

[19] F. Yu and V. Koltun, “Multi-Scale Context Aggregation by Dilated Convolutions,” in Proc. of the 4th Int'l Conf. on Learning Representations, 2016.

[20] N. Otsu, “A Threshold Selection Method from Gray-Level Histograms,” IEEE Trans. Syst. Man. Cybern., vol. 9, no. 1, pp. 62–66, Jan. 1979.

[21] D. P. Kingma and J. L. Ba, “Adam: a Method for Stochastic Optimization,” in Proc. of the 3rd Int'l Conf. on Learning Representations, 2015.

