Layout analysis using semantic segmentation for Imperial Meeting Minutes
Sayaka Iida¹, Yuki Takemoto¹, Yu Ishikawa², Masami Takata¹, Kazuki Joe¹
¹Nara Women's University, Japan
²Shiga University

Abstract - In this paper, we propose a layout analysis method for text extraction of Imperial Meeting Minutes. Character recognition accuracy depends on character extraction accuracy. In order to improve the accuracy of character segmentation, it is necessary to analyze the layout of the document image. Histogram analysis is usually used for layout analysis. However, it is difficult to perform general-purpose layout analysis using only histogram patterns, and it is necessary to visually consider the document configuration for accurate document area extraction. Therefore, we propose a layout analysis method using semantic segmentation. We apply the proposed method to Imperial Meeting Minutes and compare the case of layout analysis with histogram and extracted characters to confirm the usefulness of the proposed method.

Keywords: Layout Analysis, Imperial Meeting Minutes, Semantic Segmentation

1 Introduction

The National Diet Library [1] provides a web service called Imperial Meeting Minutes Search System [2]. Imperial Meeting Minutes Search System digitally publishes the fast recording of all sessions of Imperial Meeting during 1890-1947. The meeting minutes were published before font standardization, and therefore, with different fonts and formats depending on years. Their images are searchable by the tables of contents, index or speaker. The meeting minutes of last two years have been converted to text to be searchable by text while the meeting minutes prior to 1944 do not have any text data not to be searchable by text. Therefore, it is necessary to convert the non-textified Imperial Meeting Minutes into text.

Since the number of meeting minutes available for the Imperial Meeting Minutes Search System is huge (about 200,000), manual text conversion is practically impossible in cost. Although we proposed the character recognition method for early-modern printed books [3], a huge amount of training data is required to improve the recognition rate. Also, in order to automatically convert the meeting minute images into text, it is necessary to analyze the layout of the meeting minute images in detail. The character recognition depends largely on correct character clipping. It depends on the accuracy of document area clipping from document images. The meeting minute images are organized in columns with containing straight lines in addition to the main document parts. Layout analysis is required to clip the document area from the meeting minute images.

In general, histogram analysis is used for the layout analysis of document images. In this method, a pattern is detected from the shape and the change of a pixel projection histogram, and the document area is clipped from the document image according to the pattern. Since boundary area between document parts and other parts is very difficult to be clearly detected just using the histogram pattern, the performance of the general layout analysis with histogram is not always enough. In order to appropriately clip the document area, it is necessary to visually confirm the structure of the document. However, the number of undocumented meeting minutes currently available in the Imperial Meeting Minutes Search System is too huge to visually identify the layout and clip each area by hand. In this paper, we propose a layout analysis method using semantic segmentation [4] for the Imperial Meeting Minutes. We apply the method to some of the Imperial Meeting Minutes so that we compare the case of layout analysis with the accuracy of histogram and character extraction, and show the usefulness of the proposed method.

The structure of this paper is shown below. Section 2 introduces existing research on semantic segmentation, and Section 3 describes the histogram method, which is a classical layout analysis method. Section 4 proposes a layout analysis method using semantic segmentation, and Section 5 describes experimental methods, experimental results, and discussions. Section 6 gives conclusions.

2 Existing Research on Semantic Segmentation

Semantic segmentation is a method to cluster pixels belonging to the same object in an image. The most semantic segmentation methods mainly use CNN (Convolution Neural Network). There are two problems in semantic segmentation using CNN before FCN (Fully Convolution Network). The first problem is that the size of the input image has to be fixed in order to use the output from the fully-connected layer. The second problem is that the pool layer discards the location information. In the first full connectivity layer issue, FCN promotes the CNN architecture without the full connectivity layer. Because FCNs do not use the fully connected layer, segmentation maps are created using images of any size. Most approaches to semantic segmentation after FCN are CNN architectures that do not have the fully connected layer. For the second pooling layer problem, the encoder-decoder architecture is one of the main approaches for the position information retention. The encoder path is useful for high-
speed computations to recover object boundaries in the decoder path. This section describes existing CNNs that incorporate the encoder-decoder architecture.

SegNet [6] is one of the early encoder-decoder architectures. It mainly targets RGB landscape images and extracts roads, people, cars, etc. from the images. In SegNet, to increase the resolution of segmentation map, the index information of maximum pooling layers is sent to the decoder, and feature map upsampling is performed. Also, it performs batch normalization after the convolution layer to prevent gradient loss and explosion.

U-net [7] is a CNN proposed for segmentation of biomedical images. It mainly targets gray scale images such as radiographs and MRI images, and detects the position of internal organs and lesions from the images. In U-Net, the connection from the encoder to the decoder is skipped so that the sum for the channel is taken. U-net retains the detail features by the connection skip. Besides U-net, there are FCN and ResNet as architectures to make the connection skip. The difference with U-net is that FCN takes the channel sum while ResNet takes the residual out of the input.

The image size of the target data trained by CNN for the above mentioned existing semantic segmentation is up to 500 by 500 pixels. The image size of Imperial Meeting Minutes targeted in this paper is about 3,200 pixel width and 4,500 pixel height. It is desirable that the CNN architecture performing semantic segmentation for Imperial Meeting Minutes should be lightweight. Furthermore, the conference recording image is binary, and the boundary of the document area is blank. The document area is surrounded by blanks, so it is not necessary to keep the segmentation map boundary details. The boundaries between the character area and the frame area are clearly separated in black and white. It is considered easier to distinguish details than landscapes or biomedical images. Therefore, in this paper, we use SegNet Basic, which is a lightweight model to reduce the number of layers rather than SegNet.

3 Existing layout analysis method

Conventional layout analysis for document images typically creates rectangles using several histograms. In the histogram based method, first, a paper image is converted into a black and white binary image to obtain a projection histogram of black pixels constituting the document. A pattern is found from the shape of the histogram and the amount of change to be divided linearly. The extracted pattern is enclosed in a rectangle, and the document is extracted based on the area and the aspect ratio of the rectangle. However, it is difficult to distinguish the difference between the document and other elements contained in the document image using only the histogram pattern. To get accurate rectangles, we need to visually consider the composition of the document.

There are two ways to create rectangles: top-down and bottom-up. The top-down method analyzes the rough layout structure and then gradually creates small rectangles. For example, an image is clipped vertically or horizontally along a space expected to be a boundary of character strings or a boundary of a paragraph to create a rectangle of the document area. The top-down method cannot take account of cross-column headings or non-rectangular areas in the document image. The bottom-up method gradually integrates a rectangle detected from an image. Small parts in the document image are classified to be merged as rectangles taking into account the distance, shape, and area between the parts.

Figure 1 shows images of imperial meeting minutes targeted in this paper. The images of the meeting minutes are divided into two to five columns. Some have titles and other do not. Document columns and titles are framed. That is, in order to perform layout analysis on the meeting minute image, it is necessary to detect the line boundary and the pattern of the straight line from the histogram. However, the frame lines included in the meeting minute image are not clear straight lines, but are distorted due to the distortion of the original meeting record, or the deviation at the scanning time. As in the case of frame lines, the characters may be misaligned. In addition, meeting minutes contain noise such as ink stains. That is, in the meeting minute image, layout analysis using only the black pixel projection histogram along the horizontal vertical directions is difficult, and it is necessary to adjust the size and position of the rectangle visually to some extent. The proposed method uses semantic segmentation that does not require visual adjustment at the time of region extraction. Semantic segmentation is a method of automatically classifying pixels that belong to the same object, and suitable for extracting regions from images. A character area, a document area, and a frame area are extracted from the meeting minute image, and layout analysis is performed in the bottom-up manner.
We propose a layout analysis method using semantic segmentation for meeting minute image. We use the SegNet Basic architecture for the semantic segmentation. In the proposed method, layout analysis processing is performed using the segmentation map output by semantic segmentation. In addition, when training with meeting minute images, Gaussian filters with different filter sizes are applied to the meeting minute images for comparison.

The model is trained with a pair of meeting minute images and target labels to perform semantic segmentation. The labels used in this paper are the following six types.

- **Label i** Character area
- **Label ii** Character area and Frame area
- **Label A** Document area and Frame area
- **Label B** Document area and Character area, Frame area
- **Label C** Character area
- **Label D** Document area and Character area

The six label types are shown in Figure 2. In the labels, the red, yellow and green area represents character, frame and document area, respectively. Label i and ii do not include the document area while Label A to D include the document area. In the layout analysis process using Label i and ii, small areas are integrated to get a rectangle as in the bottom-up method.

In the layout analysis process using Label A to D, as is similar to the top-down method, the structure is roughly extracted and then the detail rectangle is created. Because the purpose of this layout analysis is to extract document and character areas, labels containing only frame areas are not generated. Meeting minute images are binary images so the color of binary images is expressed as a 256 gray scale. It is difficult to segment an image whose label boundary is blank as in document area, too. Therefore, the space between characters of meeting minute images is smoothed with a Gaussian filter shown in equation (1) to be used as an aid for border judgment of the document area. The kernel of the Gaussian filter used in this paper is square.

\[
Gauss(ksize) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{2ksize^2}{2\sigma^2}\right)
\]  

(1)

The width and the height of the filter mask are odd. The value \(\sigma\) is determined from the kernel size using equation (2).

\[
\sigma = 0.30 \left(\frac{ksize - 2}{2}\right) + 0.80
\]  

(2)

We compare the images applied with and without the Gaussian filter. The maximum Gaussian filter size is less than 60, assuming that the distance between two adjacent characters is approximately 60 pixels. The conditions of the size \(K(n)\) of the Gaussian filter are as follows.

- **Condition 1**: without Gaussian filter
- **Condition 2**: increase of 1 kernel size

\[
K(n) = 2n - 1 \quad \{n \in \mathbb{N} | 0 < n < 30 \}
\]  

(3)

- **Condition 3**: increase of 5 kernel size

\[
K(n) = 6n - 5 \quad \{n \in \mathbb{N} | 0 < n < 10 \}
\]  

(4)

- **Condition 4**: increase of 9 kernel size

\[
K(n) = 10n - 9 \quad \{n \in \mathbb{N} | 0 < n < 6 \}
\]  

(5)

Learning is performed with a total of 24 patterns for each of the six labels and each of the above four conditions. The number of meeting minute images to learn is five. In order to reduce the calculation, we divide 20 sheets into squares of 1,024px side and learn meeting minute images.

We explain the flow of the proposed method. First, we perform semantic segmentation to extract regions. Next, the preprocessing is performed to the output of the semantic segmentation. Finally, the character extraction processing is
performed. When outputting a character area in the character extraction processing, we use the character area label of the segmentation map.

The flow of the layout analysis processing using semantic segmentation is described as shown in Figure 4. First, using the trained model, we label the segments using semantic segmentation on the meeting minute images. Next, the original image is preprocessed using the segmentation map output. If the output segmentation map includes a frame area, we replace black pixels in the area determined as frame area with white pixels. If the output segmentation map includes document area, the image is clipped for each document area. Finally, the character extraction processing is performed as described above, but when the output segmentation map includes character area, the area extraction is performed using the character area.

5 Experiments

5.1 Experiments Method

In this paper, we compare the character extraction accuracy of Imperial Meeting Minute images. The target meeting minute images are used for training samples and test samples. We apply the histogram method and the proposed method to the meeting minute images to extract characters, and compare the proportions of correctly extracted characters. From the comparison results, the usefulness of layout analysis using semantic segmentation is confirmed. In the histogram method, the meeting minute images are divided into several stages using pixel projection histograms. In addition, we compare the results of the total 24 patterns of the six segmentation maps output by the proposed method and the four conditions of the Gaussian filter size extensions. The number of characters evaluated for learning and test images is 1,293 and 837, respectively. In the character extraction processing, rectangle integration fails when each part constructing a character are apart. Such characters were excluded from the evaluation. Since it is another open problem, we do not discuss in this paper.

5.2 Results and discussions

Layout analysis is applied to the learning and test images on the model and the extracted characters are compared. In the histogram method, 46.3% of characters in the learning images and 4.78% in the test images are extracted. The histogram approach cannot remove borders and brackets in documents. These noises are considered to hinder region extraction.

The results of semantic segmentation on test images using the proposed method are shown in Figure 5. When the frame is blurred or dirty, or when there is a blank area at the border of the frame, it is recognized as character area even if the frame area is incorrectly recognized. In some cases, frame joints or straight parts of characters may be recognized as frame area. In addition, ink stains may be recognized as a character or a frame. The misrecognized area is the noise of the subsequent processing. In the case of condition 4, the segmentation map is blurry. The reason is that the kernel size is too large.
Tables 1 and 2 show the proportion of characters extracted from each image by the proposed method. Table 1 shows the percentage of extracted characters for learning images. In the case of conditions 1, 2 and 3 of the proposed method, the proportion of extracted characters of the learning images is 84.9%, 78.8% and 82.1% in average, respectively. The proportion of extracted characters of the test images is 81.8%, 81.4% and 75.5% in average, respectively. Under conditions 1, 2 and 3, the proportion of characters extracted in average is higher than layout analysis using the histogram. This is because frame removal and document area extraction are performed from the document images. For condition 4 of the proposed method, the average proportion of extracted characters of the learning and test images is 21.5% and 14.7%, respectively. Condition 4 cannot extract most characters except label D. For label D with condition 4, 87.7% and 77.5% of the characters in the learning and test images are extracted, respectively. This is due to the fact that the segmentation map cannot be correctly generated except for label D.
In condition 4 with labels i and ii, the character area is used, but character extraction is failed because the output is blurred. In the case of condition 4 with labels A, B, and C, the vicinity of the frame line is divided into document areas. With regard to label D, the border line portion is not recognized as a document area as well as in the case of condition 4. When the proportion of extracted characters in Table 1 is the highest, label B is used and the Gaussian filter size extension is with condition 2. When the proportion of extracted characters in Table 2 is the highest, label B is used and the Gaussian filter is with condition 1. With labels i and ii, document area is not contained while with labels A to D, document area is contained. The average percentage of extracted characters with the labels that do not contain document area is 62.6% for learning images and 55.9% for test images. The average number of extracted characters with the labels that contain document areas is 69.0% for learning images and 66.9% for test images. Therefore, we find that the labels including document area is more effective. In addition, among the labels including document area, the average of the number of extracted characters with label B is the largest when condition 4 is excluded.

Table 3 and 4 shows the proportion of the average number of extracted characters of learning and test images with each (including/not including document area) label, respectively. We validate the effectiveness of blank parts interpolation by the Gaussian filter when performing semantic segmentation for binary images. We compare the conditions of the Gaussian filter when document area is not included and included. In the case of not included and condition 1, 92.5% of the text is extracted from the learning images while 79.6% from the test images. When text area is not included, the Gaussian filter with condition 1 achieves the highest percentage of character extraction. When character area is included with condition 3, 1.9% of the characters of learning images are extracted. In condition 2, 1.7% of characters are extracted from learning images and it is smaller than that in condition 1. From the test images, 1.0% of characters are extracted with condition 2. In the case of condition 3, 2.2% of characters are extracted and it is smaller than in condition 1. Therefore, when the text area is included, the blank part interpolation by the Gaussian filter may be effective but not always necessarily.

From the above, we find that the accuracy of character extraction is improved by performing layout analysis with the proposed method. The accuracy of character extraction depends on the accuracy of region extraction by semantic segmentation. In the future, we aim to improve the accuracy of semantic segmentation by increasing training data, changing conditions of Gaussian filter, and applying noise and blurring to input images. We confirm that it is not possible to correctly recognize the joint of the frame and the straight portion in characters as well as accurately extract character area and frame area. In addition, as applying semantic segmentation to binary images, white space is complemented with Gaussian filters, but the usefulness is not shown from the evaluation results.

### 6 Conclusions

In this paper, we propose a layout analysis method using semantic segmentation for the purpose of extracting the text of Imperial Meeting minute images. In the proposed method, the text area, the frame area, and the document area are extracted using semantic segmentation. The CNN architecture used for the semantic segmentation adopts SegNet Basic. There are six combinations of objects included in the label images to be trained. Furthermore, Gaussian filters of different kernel sizes are applied to the images to support segmentation of binary images.

In order to validate the proposed method, the ratio of the extracted characters by the layout analysis method using histogram and the proposed method was compared. Also, in the proposed method, the type of objects included in the label and the conditions of the Gaussian filter size are varied to compare the proportions of the character extraction. In the layout analysis method by histogram, meeting minute images are scanned for each column using a pixel projection histogram. In the proposed method, layout analysis is performed using the output of semantic segmentation, and the document area is extracted. Changing the object types contained in the labels and the Gaussian filter size, they are applied to the learning image to compare the proportions of
the extracted characters. The first to sixth labels are character area, character and frame area, document and frame area, document, character and frame area, character area, document and character area, respectively. There are four types of Gaussian filter size extension. Character extraction was performed on learning and test images using the histogram method and the proposed method. The number of characters extracted for each was compared. In the case of semantic segmentation, the label selection and Gaussian filter size were examined. The original images have 1,293 characters for learning and 837 characters for test. As a result, 46.3% and 4.78% of characters in the learning images and the test images were extracted by the histogram method, respectively. In the case of semantic segmentation, the label selection and Gaussian filter size were examined. The original images have 1,293 characters for learning and 837 characters for test. As a result, 46.3% and 4.78% of characters in the learning images and the test images were extracted by the histogram method, respectively. In the proposed method, when using the kernel sizes increasing of 9, the proportion of the extracted characters of the learning and the test images were 21.5% and 14.7% in average, respectively. Regardless to say, it showed poor performance. In the case of other increasing sizes, the proportion of the extracted characters of learning images with without filter, the kernel size of increasing of 1 and 5 is 84.9%, 82.1% and 78.8%, respectively. In the case of learning images, they are 81.8%, 81.4% and 75.0%. Therefore, compared with the histogram method, the proposed method is effective except in the case of using the kernel sizes increasing of 9. In addition, we showed the conditions where the number of extracted characters for learning and test images count the highest values. In the case of learning images using document area, text area and frame area as labels, 97.8% characters are extracted with kernel sizes increasing of 5. In the case of test images labeled with document area, text area and frame area, 87.2% characters are extracted without Gaussian filter.

As future work, it is necessary to improve the output accuracy of semantic segmentation. We aim to improve the accuracy by increasing the learning data, changing the conditions of the Gaussian filter, and applying de-noising and blurring to the input images.

7 References


Acknowledgment

This work is partially supported by Grant-in-Aid for scientific research from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT) No. 17H01829.