

Experimental Application of a Japanese Historical Document Image Synthesis Method to Text Line Segmentation

Naoto Inuzuka and Tetsuya Suzuki ^a

Graduate School of Systems Engineering and Science, Shibaura Institute of Technology, Saitama, Japan

Keywords: Text Line Segmentation, Historical Document, Deep Learning, Data Synthesis.

Abstract: We plan to use a text line segmentation method based on machine learning in our transcription support system for handwritten Japanese historical document in Kana, and are searching for a data synthesis method of annotated document images because it is time consuming to manually annotate a large set of document images for training data for machine learning. In this paper, we report our synthesis method of annotated document images designed for a Japanese historical document. To compare manually annotated Japanese historical document images and annotated document images synthesized by the method as training data for an object detection algorithm YOLOv3, we conducted text line segmentation experiments using the object detection algorithm. The experimental results show that a model trained by the synthetic annotated document images are competitive with that trained by the manually annotated document images from the view point of a metric intersection-over-union.

1 INTRODUCTION

We plan to use a text line segmentation method based on machine learning in a transcription support system for handwritten Japanese historical document in Kana which we are developing. Because it is time consuming to manually annotate a large set of document images for training data for machine learning, we are searching for a data synthesis method of annotated document images.

In this paper, we report our synthesis method of annotated document images designed for a handwritten Japanese historical document. The method makes it easy to automatically generate a lot of annotated document images.

The organization of this paper is as follows. In section 2, we summarize related work. We then explain characteristics of a target handwritten Japanese historical document and our synthesis method of annotated document images designed for the document in section 3, and evaluate the synthesis method by text line segmentation experiments in section 4. In section 5 we state concluding remarks.

2 RELATED WORK


2.1 A Transcription Support System for Handwritten Japanese Historical Kana

We briefly explain our transcription support system for handwritten Japanese historical document in Kana (Sando et al., 2018; Yamazaki et al., 2018). Kana is a kind of Japanese characters, and it is difficult to read handwritten Kana used in historical documents because of the following reasons.

- It is difficult to segment characters because they are cursive.
- There exist similar shape characters with different syllables.

The characteristic point of the system is that the system outputs optimal combinations of multiple results of character segmentation, multiple reading order among segmented characters, and multiple results of character recognition.

The system consists of three parts: a document image analyzer, a constraint solver and a graphical user interface (GUI) which integrates them. The document image analyzer segments characters, decides

^a <https://orcid.org/0000-0002-9957-8229>

reading order among the segmented characters, and outputs multiple recognition results for each of the segmented characters. The constraint solver solves a constraint satisfaction problem (CSP) based on the output of the document image analyzer, and outputs solutions of the CSP as transcription results. The solver searches solutions which minimize the total cost of occurrence costs of words and connective costs between words with reference to an electrical dictionary for morphological analysis (Ogiso et al., 2010). The entire system can be used through the GUI.

The GUI and the constraint solver have been developed, but the document image analyzer have not yet. The first step of the document image analysis will be text line segmentation.

2.2 Training Data Generation for Document Image Analysis

A problem in text line segmentation based on machine learning is to construct training data. It is time consuming to manually annotate a large set of document images. In this section, we briefly present three works related to construction of training data for document image analysis. None of them targets on historical Japanese document image synthesis.

Capobianco et al. proposed a historical document image generation tool (Capobianco and Marinai, 2017). The generation process is as follows. First, a user extracts background images from some examples of document images using a tool. The user, then, describes structure of a document in XML with reference to the document images. The generator takes the extracted background images, document structure in XML, fonts, and a dictionary as inputs and generates document images. To construct pages with various characteristics, it is possible not only to randomly decide text line height and repetition times of items but also to specify a possibility to generate a text line. In addition, it is possible to rotate document images and add noise on document images.

Pondenkandath et al. proposed a synthesis method of historical document images using a deep neural network (Pondenkandath et al., 2019). The method transforms a given document generated by \LaTeX to a handwritten-like historical document image using a deep neural network. The authors experimentally compared CycleGAN and Neural Style Transfer for image transformation, and concluded that CycleGAN is more promising than Neural Style Transfer.

Aoike et al. constructed a layout data set of documents usable for machine learning and made it public (Aoike et al., 2019). The target documents are digi-

tal data included in digital collections of the National Diet Library of Japan. The layout data set was constructed by revising recognition results by a document layout recognizer based on machine learning. To improve accuracy, the recognizer was updated by machine learning using the revised layout data as training data every time 100 to 200 pages were manually processed.

3 A JAPANESE HISTORICAL DOCUMENT IMAGE SYNTHESIS METHOD

We explain characteristics of a target Japanese handwritten historical document from the view point of page layout, a model constructed for the document, and a method which generates handwritten-like document images from a given model.

3.1 Page Layout of a Japanese Handwritten Historical Document

We chose the Tales of Ise (Reizei, 1994) as a target Japanese handwritten historical document. It is relatively easy to generate document images similar to those of the document because it consists of vertically written text lines and its page layout is simple.

The characteristics of its page layout are as follows.

- C1.** It includes notes representing the sources of poems as shown in Fig.1 (1)
- C2.** Some text lines are folded at the bottom of pages as shown in Fig.1 (2)
- C3.** Some text lines have readings or supplementary explanations alongside them as shown in Fig. 1 (3)
- C4.** Some characters in adjoining text lines touch each other as shown in Fig.1 (4)
- C5.** Center lines of some text lines are leaning as shown in Fig.1 (5)
- C6.** There exist paragraphs with supplementary paragraphs alongside them as shown in Fig.1 (6)
- C7.** There exist characters written in a cursive style.

3.2 A Document Model for the Target Document

We designed a document model for documents with characteristics shown in section 3.1 except the characteristic C7 (characters in a cursive style). In this

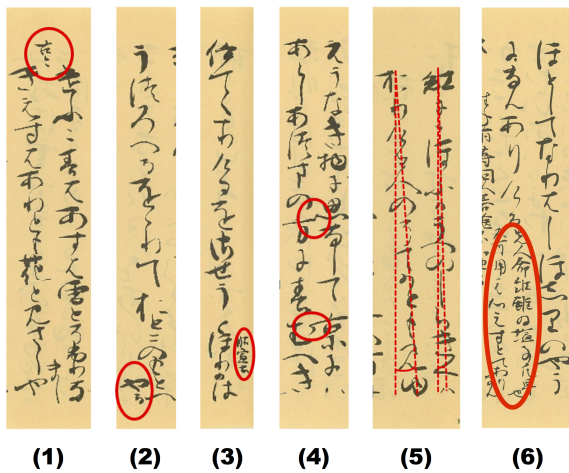


Figure 1: Characteristics of the target document’s layout adopted from (Inuzuka and Suzuki, 2020) (Circles and lines are added in document images scanned from (Reizei, 1994)).

section, we explain elements of the model: line representing a text line, paragraph representing a paragraph and format representing the entire format of a document.

A line element represents a text line and has the following attributes: the body width, the height of the text line, the center line, ruby characters, and the folding part of the text line shown in Fig.2. The center line of a text line is represented as a polyline on which characters of the text line are placed. It is for the characteristic C5 shown in section 3.1. The vertical space between characters is specified in *format*. A sequence of ruby characters of a text line is also a line placed alongside a character of the text line. It is for the characteristic C3 shown in section 3.1. A folding part of a text line is also a line whose top margin can be specified and the folding part is placed closely to the text line. It is for the characteristics C2 and C4 shown in section 3.1.

A paragraph element represents a paragraph and has the following attributes: four margins around the paragraph (top, bottom, left, and right), three annotations around the paragraph (top, left, and right), the minimum space between first characters of adjoining text lines, and the sequence of line elements as shown in Fig.3. Each of three annotations around a paragraph is also a paragraph. They are for the characteristics C1 and C6 shown in section 3.1. The left and right annotations of a paragraph are placed closely to the paragraph. Adjoining text lines of a paragraph are placed closely with the specified minimum space between their first characters. These are for the characteristic C4 shown in section 3.1.

A *format* represents the entire format of a document, and has the height and the width of each page,

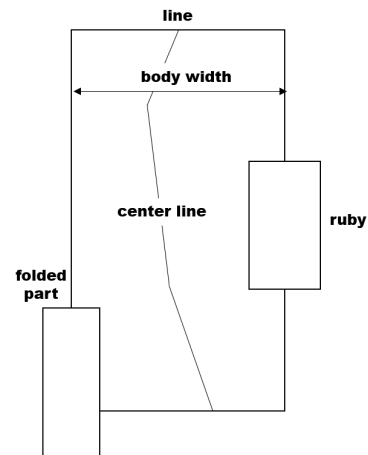


Figure 2: A line element of the document model adopted from (Inuzuka and Suzuki, 2020).

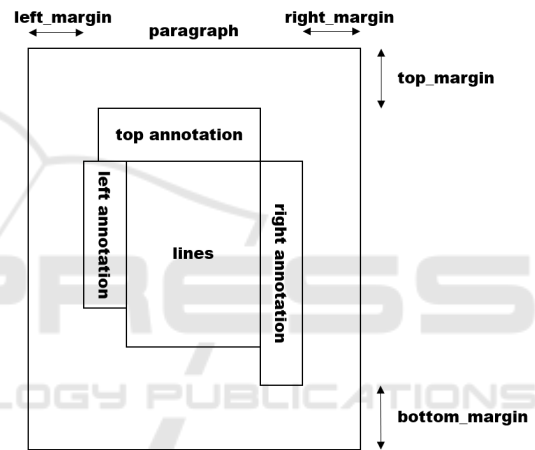


Figure 3: A paragraph element of the document model adopted from (Inuzuka and Suzuki, 2020).

four margins (top, bottom, left, and right) of each page and a sequence of paragraphs as shown in Fig.4.

3.3 A Document Image Synthesis System

We implemented a document image synthesis system in Python, which consists of four command line interface commands: *fonts*, *format*, *typeset*, and *print*. Fig.5 shows the document image synthesis process by them.

The *fonts* command extracts at most *n* fonts for each Japanese Kana from Kuzushiji-49(Clauwat et al., 2018) which is a data set of deformed Kana with white on black. The command reverses white and black in the extracted fonts, and removes white lines at the top and the bottom of each font with black on white. It finally outputs the resulting font data to a python object file called a font data file.

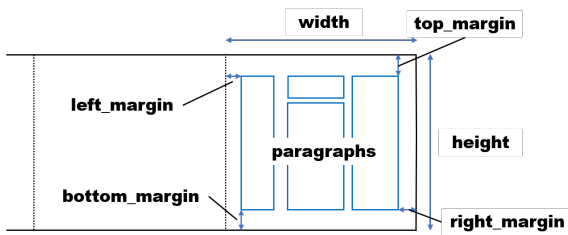


Figure 4: A format element of the document model adopted from (Inuzuka and Suzuki, 2020).

The `format` command generates a format based on the document model described in section 3.2 as a dictionary object in Python, and outputs it to a file called a format file. It randomly generate various paragraphs and text lines.

The `typeset` command takes both a font data file and a format file as its input, and outputs the result of typesetting to a file called a typesetting file. The command puts a randomly selected character in Kana with a randomly selected font of it in the font data file according to a format specified by the format file.

The `print` command reads a typesetting file and generates both document images and rectangles as annotations, each of which surrounds a text line.

4 EXPERIMENTS

4.1 Objective

To check the effectiveness of document images generated by the method described in section 3.3 as training data in text line segmentation, we conducted text line segmentation experiments using a machine-learning-based object detection algorithm. The details are described in the following.

4.2 Method

4.2.1 Image Data Sets

We prepared three sets of document images as follows.

- I1.** Document images binarized and extracted from the reference (Reizei, 1994)
- I2.** Binary document images generated by our method with roughly adjusted parameters
- I3.** Binary document images generated by our method with carefully adjusted parameters

Document images in the data set I1 were preprocessed as follows. They were binarized by Otsu's

method (Otsu, 1979) and resized to 512-pixel width and 512-pixel height. Annotations for them were done manually. An annotation is a bounding box of a text line as shown in Fig.6. In our experiments, each folding part of text lines and each of ruby character sentences were also treated as text lines.

Document images in the data set I2 are synthetic document images generated by our method with roughly adjusted parameters.

Document images in the data set I3 are synthetic document images generated by our method with carefully adjusted parameters as follows.

- At most 100 fonts were randomly selected for each character in Kana.
- The selected grayscale fonts were completely binarized.
- The selected font data were randomly shrunk or expanded with some probability.
- Margins in pages and width of text lines were adjusted with reference to the reference (Reizei, 1994).
- The vertical space between characters were randomly set 0 to 10 pixels.

Fig.7 shows a document image generated under the configuration.

We divided the data sets I1, I2, and I3 as follows. The data set I1 was divided into three sets I1-1, I1-2, and I1-3 for a machine-learning-based object detection algorithm YOLOv3 (Redmon and Farhadi, 2018). I1-1, I1-2 and I1-3 are a training set, a validation set, and a test set respectively. The training set and the the validation set of a data set were used in training process. The test set was used to evaluate a trained model in text line segmentation. The data sets I2 and I3 were also divided as I1. Table.1 shows the number of document images in I1, I2, I3, and their subsets. The training set, the validation set and the test set of each data set appear in a ratio 2:1:1.

4.2.2 Models

To create the following three models using YOLOv3, we used a learning rate 0.001, a batch size 16, and epoch 4000 in training process.

- M1.** a model trained by the data sets I1-1 and I1-2
- M2.** a model trained by the data sets I2-1 and I2-2
- M3.** a model trained by the data sets I3-1 and I3-2

4.2.3 Tests

We combined the trained models and the data sets for text line segmentation as follows.

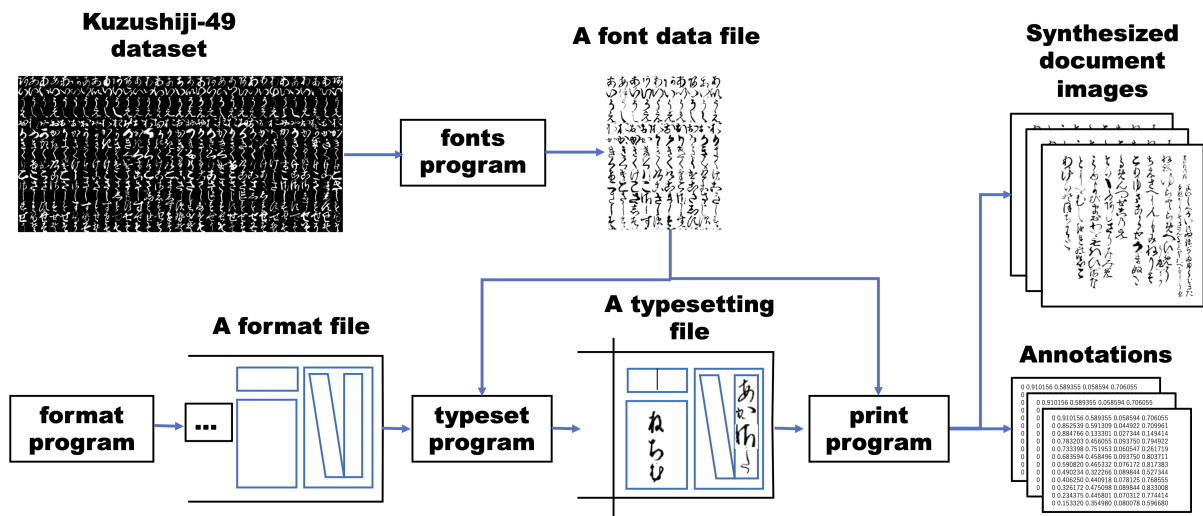


Figure 5: The document image synthesis process amended from (Inuzuka and Suzuki, 2020).

Table 1: The number of document images in data sets.

Data set	Subset for training (# of images)	Subset for validation (# of images)	Subset for test (# of images)
I1	I1-1 (83)	I1-2 (41)	I1-3 (42)
I2	I2-1 (243)	I2-2 (121)	I2-3 (122)
I3	I3-1 (243)	I3-2 (121)	I3-3 (122)

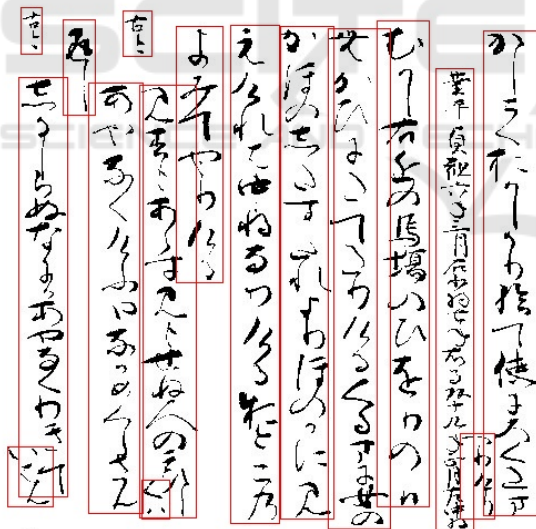


Figure 6: An annotated document image (Rectangles are added to a binarized document image scanned from (Reizei, 1994)).

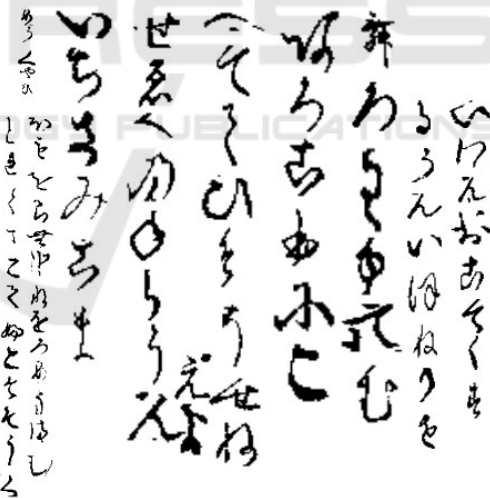


Figure 7: A synthetic document image adopted from (Inuzuka and Suzuki, 2020).

- Case 1. The model M1 and the data set I1-3
- Case 2. The model M2 and the data set I2-3
- Case 3. The model M2 and the data set I1-3
- Case 4. The model M3 and the data set I3-3
- Case 5. The model M3 and the data set I1-3

For example, we segmented text lines in document images in I1-3 using M1 in case 1.

We used intersection-over-union (IoU) as an evaluation metric. IoU between two regions A and B is defined as follows.

$$Intersection\ Over\ Union\ (IoU) = \frac{|A \cap B|}{|A \cup B|}$$

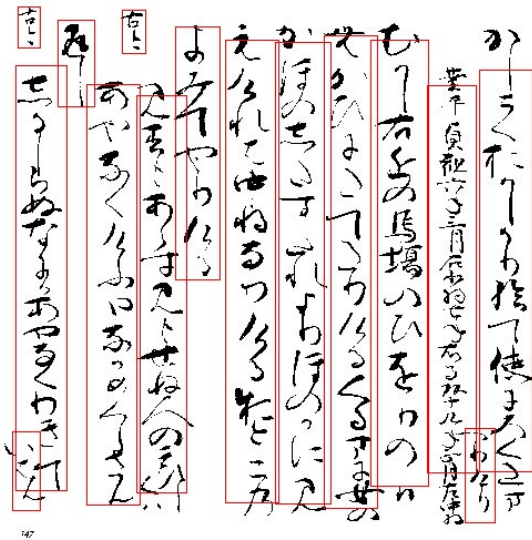


Figure 8: An example of text line segmentation in case 1.

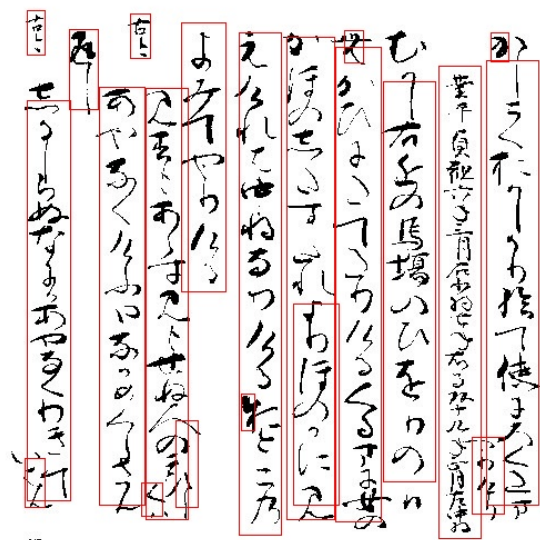


Figure 10: An example of text line segmentation in case 5.

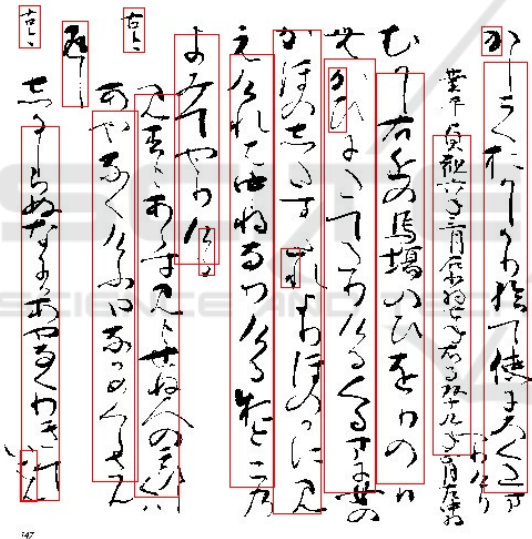


Figure 9: An example of text line segmentation in case 3.

We calculated IoU between the ground truth and a predicted region for each document image, and averaged them.

4.3 Results

Table 2 shows the resulting average IoUs.

- In case 1, case 2 and case 4 where data sets for training, validation and test are derived from a data set, average IoUs are at least 0.892.
- The average IoU in case 5 is 0.877 while that in case 3 is 0.806. Parameter adjustment in our document image synthesis method improved the av-

erage IoU by 0.71 points.

- The average IoU in case 1 is 0.897 while that in case 5 is 0.877.
- The number of detected rectangles in case 1 is 514 while that in case 5 is 682.

Fig.8, Fig.9 and Fig.10 show the resulting text line segmentation in case 1, 3, and 5 respectively. Fig.6 shows text lines to be detected.

- In Fig.8, Fig.9, and Fig.10, some detected text lines lack their heads and/or ends.
- In Fig.9 and Fig.10, some detected text lines are redundant. For example, a part of a text line is detected as a text line.

4.4 Evaluation

We evaluate the experimental results as follows.

- The results in case 1, case 2 and case 4 show that trained models works well.
- The results in case 3 and case 5 show that parameter adjustment in our method improved results in text line segmentation.
- The results in case 1 and case 5 show that a model trained by the synthetic annotated document images can be competitive with that trained by the manually annotated real document images from the view point of IoU.
- We need another metric which evaluates both the accuracy of an object detector and the number of detected objects because more text lines were detected in case 5 than in case 1 though the average IoU in case 5 is competitive with that in case 1.

Table 2: Average IoU and the number of detected text lines.

Case	Model	Data sets for training and validation	Data set for test	Avg. IoU	# of detected text lines
Case 1	M1	I1-1 and I1-2	I1-3	0.897	514
Case 2	M2	I2-1 and I2-2	I2-3	0.892	1,676
Case 3	M2	I2-1 and I2-2	I1-3	0.806	574
Case 4	M3	I3-1 and I3-2	I3-3	0.903	1,320
Case 5	M3	I3-1 and I3-2	I1-3	0.877	682

- If we use the document image synthesis method and the object detection algorithm for text line segmentation, we need post processes such as removal of redundant detected text lines and expansion of detected text lines.

5 CONCLUSION

We proposed an annotated Japanese historical document image synthesis method, and experimentally applied it to text line segmentation using a machine learning-based object detection algorithm YOLOv3 where synthetic document images were used as training data for YOLOv3. The experimental results show that a model trained by the synthetic annotated document images can be competitive with a model trained by the manually annotated real document images from the view point of intersection-over-union. Parameters in our method are, however, needed to manually adjust to generate such competitive document images.

Future work would be as follows. To confirm applicability of our method, we need to apply our method to same type of other historical documents because we applied it to only a historical document. Automatic parameter adjustment methods for document image synthesis and post-processing for segmented text lines will improve text line segmentation results. The post-processing, for example, would be to remove segmented text lines included in other segmented ones and to extend segmented text lines to include missing parts of the text lines. Our document image synthesis method will be also applicable to character segmentation.

REFERENCES

Aoike, T., Kinoshita, T., Satomi, W., and Kawashima, T. (2019). Construction and publication of book layout datasets for machine learning. In *Proceedings of the Computers and the Humanities Symposium*, volume 2019, pages 115–120.

Capobianco, S. and Marinai, S. (2017). Docemul: A toolkit to generate structured historical documents. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 01, pages 1186–1191.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. (2018). Deep learning for classical japanese literature. *CoRR*, abs/1812.01718.

Inuzuka, N. and Suzuki, T. (2020). Text line segmentation for japanese historical document images using deep learning and data synthesis. *SIG Technical Reports (CH)*, 2020-CH-122(4):1–6.

Ogiso, T., Ogura, H., Tanaka, M., Kondo, A., and Den, Y. (2010). Development of an electrical dictionary for morphological analysis of classical japanese. *SIG Technical Reports (CH)*, 2010-CH-85(4):1–8.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Pondenkandath, V., Alberti, M., Diatta, M., Ingold, R., and Liwicki, M. (2019). Historical document synthesis with generative adversarial networks. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 146–151.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv*.

Reizei, T. (1994). *Tales of Ise (photocopy)*. Kasama Shoin.

Sando, K., Suzuki, T., and Aiba, A. (2018). A constraint solving web service for a handwritten japanese historical kana reprint support system. In van den Herik, H. J. and Rocha, A. P., editors, *Agents and Artificial Intelligence - 10th International Conference, ICAART 2018, Funchal, Madeira, Portugal, January 16-18, 2018, Revised Selected Papers*, volume 11352 of *Lecture Notes in Computer Science*, pages 422–442. Springer.

Yamazaki, A., Sando, K., Suzuki, T., and Aiba, A. (2018). A handwritten japanese historical kana reprint support system: Development of a graphical user interface. In *Proceedings of the ACM Symposium on Document Engineering 2018*, New York, NY, USA. Association for Computing Machinery.