End-to-End Pre-Modern Japanese Character (Kuzushiji) Spotting with Deep Learning

Tarin Clanuwat (Research Organization of Information and Systems, Joint Support-Center for Data Science Research, Center for Open Data in the Humanities, National Institute of Informatics)

Alex Lamb (MILA, Université de Montréal)

Asanobu Kitamoto (Research Organization of Information and Systems, Joint Support-Center for Data Science Research, Center for Open Data in the Humanities, National Institute of Informatics)

Abstract - Kuzushiji has been used as a cursive writing style in Japan for over a thousand years. However, following the reform of Japanese language textbooks in the year 1900, Kuzushiji was no longer taught in schools. Thus at present, nearly all Japanese natives cannot read books written or published before 150 years ago. As a result, many important pieces of Japanese history are only accessible to a handful of specially trained scholars. This has motivated the use of machine learning to read Kuzushiji. To accomplish this, we propose a new way of using the U-Net architecture from Deep Learning to spot characters in manuscripts. For other researchers to use our system to search for characters or words in manuscripts written with Kuzushiji, only an image of the page of text needs to be given to our system as input. Other Kuzushiji word spotting method have limitations in characters or words written in different Jibos (字母) or root characters in Hentaigana (or variant Kana). Our proposed U-Net character spotting system not only gives results with state-of-the-art accuracy, our system also resolves the Jibo problem, allowing our system to search for input characters in the text. While we present results for character spotting, we discuss keyword search as an exciting area for future work.

1. Introduction

According to the General Catalog of National Books [1] there are over 1.7 million books written or published in Japan before 1867. We estimate that in total there are over 3 million books preserved nationwide. Despite ongoing efforts to create digital copies of these documents—a safeguard against fires, earthquakes, and tsunamis—most of the knowledge, history, and culture contained within these texts remains inaccessible to researchers and the general public. We have a lot of digitized images of the manuscripts and books, but we still don't have appropriate way to search for information or make manuscripts readable to non-expert users. The main problem is because the text in all materials was written in Kuzushiji style.

Kuzushiji (cursive style Japanese characters) had been used in Japanese writing in printing system for over a thousand years. However, the standardization of Japanese language textbooks as in "the 16th article of the enforcement regulations" and "the No. 1 table" of "Elementary School Order" in Meiji 33th (1900) [2], not only unified the writing type of hiragana and, but also made most Japanese cursive writing style obsolete for modern writing and printing systems. Therefore, most Japanese natives cannot read books written or published prior to the reforms.

This challenge has motivated the use of automated systems for reading and transcribing Kuzushiji texts. Such systems have been successfully built to automatically recognize texts in other languages, but Kuzushiji presents a unique challenge as (1) many characters are hard to recognize in isolation without context from the surrounding text because many of them look similar. On the other hand, the opposite problem also exists: (2) a single character may be written in different ways. This is a major problem especially for Hiragana characters because each character in Hentaigana or variant Kana has many root characters or Jibo (字母) mapped to them. Additionally, (3) the vocabulary size is large due to the use of Kanji. Moreover, some Kanji only appear rarely in the dataset, making the character shape dataset highly label-imbalanced.

We explore a new system for recognizing and transcribing Kuzushiji text using deep neural networks. Our system addresses (1) by considering the entire page as a whole, and using the U-Net architecture to predict the marginal character probabilities across positions. Additionally, deep learning based classifiers naturally address (2) as many characters written in different ways can map to the same representation.
Additionally, these intermediate outputs (the probabilities of characters across the image) could be useful by themselves for tasks other than transcription. For example, researchers could search for all sequences which potentially match a given input string (key word spotting). While our system is designed specifically to address challenges with Kuzushiji, the method itself is domain-agnostic and should be able to work on other languages with relatively little modification.

2. Kuzushiji Problem

Despite the promise of deep learning technology, Pre-modern Japanese OCR remains a challenging problem. First, Kuzushiji characters often overlap with each other. Such connectedness makes character segmentation difficult, even for trained researchers. Also, many manuscripts have irregular layouts with, for example, characters wrapped around images, in multiple blocks, or of different sizes. Additionally, the size of a character relative to those around it may be its only distinguishing feature. For example, in pre-modern hiragana "Ri" (利)

and “Wa” (和) share almost the same shape, but different in the aspect ratio. “Ri” is written slightly smaller than other characters.

Figure 1: The similarity between “Wa” (和) and “Ri” (利) [3]

Several examples of this phenomenon shown below illustrate the importance of context in correctly recognizing Kuzushiji.

Another example of Kuzushiji character similarity problem is to distinguish between hiragana Ku (く), an iteration mark (く or くの字点) and Te (て) by looking at just one character at a time.

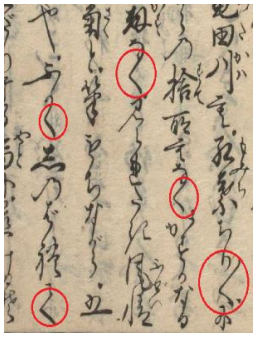

Figure 2: shows the similarity of Ku (く), iteration mark (く) and Te (て).
From right to left:
-Iteration mark (the word read as Chika ‘chika’).
-Ku (the word read as Na ‘ku’).
-Ku (the word read as Na ‘ku’).
-Ku (the word read as Fuka ‘ku’).
-Te (the words read as Shinobare ‘te’)

The K

Thus the most straightforward solution of segmenting the text into characters and training a recognition system separately over the characters does not perform well for Kuzushiji. This motivates the main contribution of our work: recognizing Kuzushiji by allowing the model to use the entire page as context.

3. Related Work

There has been a great deal of research on applying machine learning to transcribe pre-modern Japanese manuscripts. All of them struggled with the challenges in the Kuzushiji problem which make it a hard research problem to solve. Moreover, typical character recognition methods require many steps of preprocessing such as layout analysis, binarization, or image cropping which require a lot of work before reaching the character recognition step. However, this breaks with the way that a person would naturally read kuzushiji, in which the whole page was available as context.

As in standard keyword spotting research, many methods have generated cropped images for each word or each column to be processed separately [4]. This method has limitation if the word was written in hiragana since it could be written with different style using different Jibo. This method has challenge if the searching word was not written with the same Jibo for hiragana characters.

Another Kuzushiji recognition research written by the winning team of PRMU algorithm contest [5] which used deep learning for Kuzushiji recognition gave good accuracy. However, since the data given in the contest was cropped into short sequence of characters, it would take a lot of effort to preprocess digital images in the archive to match the dataset.

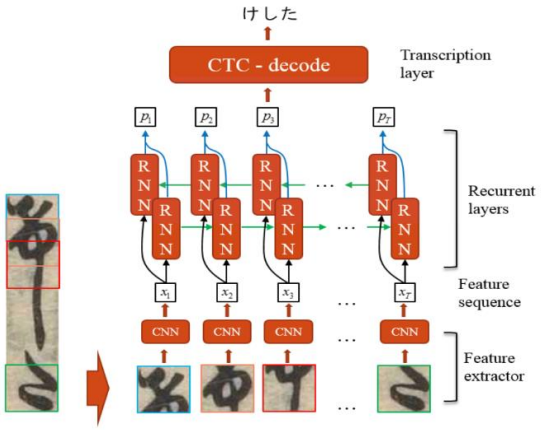

Figure 3: Method in Kana recognition using deep learning in [5]

As mentioned earlier, we aimed to avoid excessive preprocessing steps so that our system could be used directly with images from libraries, museum or archives, and also so that our system would not give up important pieces of contextual information during preprocessing.

4. Kuzushiji Dataset

Kuzushiji dataset was created by the National Institute of Japanese Literature (NIJL), and is curated by the Center for Open Data in the Humanities (CODH). Since 2014, NIJL and other institutes have started a national project for digitizing about 300,000 Japanese old books, transcribing some of them, and sharing them as open data for promoting international collaboration.

During the transcription process, a bounding box was created for each character, but literature scholars did not think they were worth sharing. From a machine learning perspective, CODH suggested to make a separate dataset for bounding boxes on a page, because that can be used as the basis for many machine learning challenges. As a result, the Kuzushiji dataset was released in November 2016, and now the dataset contains 3,999 character types and 403,242 characters.

We expect that number to increase to one-million-character shape images by the end of 2018. This sizable dataset opens up the possibility to use deep learning, which has recently enabled impressive advances in computer vision, pattern recognition, document image analysis and many other fields.

The Kuzushiji dataset from CODH website is structured in downloadable link of each books, one book has one zip file which contains 2 folders and 2 csv files. The first folder is “characters” which contains images of all characters appeared in the book organized in folders

labeled by Unicode. The filename structure of character image is Unicode, Book ID, image number, X coordinate, Y coordinate respectively. For example:

U+4E0D_200014740-00003_1_X1623_Y1845.jpg
U+4E0D is ふ.
200014740 is book id of *Ugetsu Monogatari*.
00003_1 is image number 3. _1 means right page and _2 is left page.
X1623 is x pixel coordinate of the character.
Y1845 is y pixel coordinate of the character.

The Coordinate csv file which is one of the main information we used in our method contains not only pixel coordinate of each characters, but also the width and height of the character. With this information, we can create the bounding box as in figure 4 and 5.

Unicode	Image	X	Y	Block ID	Char ID	Width	Height
U+96E8	200014740-00002_2	1616	405	B0001	C0001	200	160
U+6708	200014740-00002_2	1662	610	B0001	C0002	129	184
U+7269	200014740-00002_2	1644	847	B0001	C0003	165	160
U+8A9E	200014740-00002_2	1640	1067	B0001	C0004	180	206
U+5E8F	200014740-00002_2	1670	1281	B0001	C0005	136	334

Figure 4: The coordinate csv file in Kuzushiji dataset.

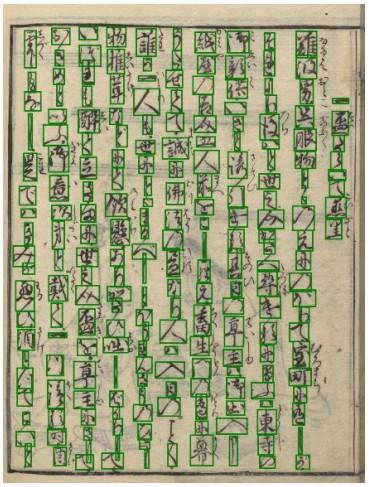

Figure 5: Bounding box created from pixel coordinates in the coordinate csv file [6].

As mentioned earlier, the Kuzushiji dataset contains digital images taken from 15 pre-modern Japanese manuscripts and woodblock printed books dated as early as mid-18th century. Many of them are cookbooks from Edo period. There are two books in the dataset which are popular fictions from that time, *Kōshoku Ichidai Otoko* (好色一代男) [6] and *Ugetsu Monogatari* (雨月物語) [7]. The genre of the books was the main reason why we chose them for the model. This is because the layout of story books is more diverse than those of cookbooks and characters appeared are not biased to one genre (for example cookbooks focus on food).

However, in our method, we didn't use character images. We used information from csv file and give the location of each character to the model using csv file.

At this time, we only use 2 books to train the model, but we plan to expand the training data using the whole Kuzushiji dataset in the future.

The Kuzushiji dataset will not only serve as a dataset for advanced classification algorithms, but also contribute to more creative areas such as generative modelling, adversarial robustness, few-shot learning, transfer learning and domain adaptation.

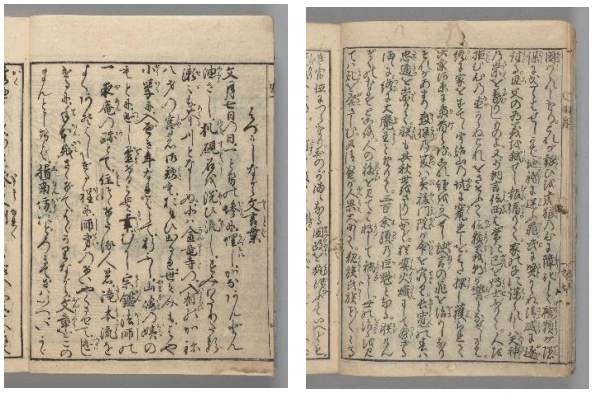

Figure 6: Left, image from *Kōshoku Ichidai Otoko*. Right, image from *Ugetsu Monogatari*.

5. Method

5.1 The Kuzushiji Recognition Task

The task of Kuzushiji spotting consists of mapping from a book written in Kuzushiji to an estimation of where characters are located on each page. We can think of this in the most general terms as trying to estimate a conditional distribution p(Y|X), where Y is the set of character locations and X is the book. We can first begin to describe our solution by discussing the simplifying assumptions in this probabilistic framework. First, we assume that a page's characters are conditionally independent given the image of the page - that is to say that we make predictions only using a single page at a time and without multi-page context. This is justified by the amount of contextual information that is already contained within a single page.

Secondly, for a given page, we reframe the variable Y (the set of characters on the page) as a character assignment at each pixel in the image. These two representations are not strictly equivalent. For example, if the same character appears twice in a row with the same-sized bounding box, in the pixel representation this is the same as the character appearing once but at twice the size. Another difference in the representations is that it assumes that each pixel only has a single character - in general this is true for Kuzushiji except at pixels very close to the boundaries between character.

Given that we assign a character to each pixel position in the book image, we make a further simplifying assumption that each pixel's character assignment is conditionally independent given the entire page's image. This assumption is not well justified, but we introduce a mechanism later (based on taking the mode of the character assignment distribution) for compensating for this flaw in the model.

Additionally, we estimate a second distribution for whether a point is within 5 pixels of the center of the character's position. Note that this distribution is agnostic to the actual identity of the character. This serves two purposes. First, it allows our system to identify that a character is present when it is easy to determine that it is a character, even if it is unable to determine exactly which character it is. Second, it disambiguates between the cases discussed earlier where two small characters which are adjacent and the same and one large character could be ambiguous.

Up until now, we have motivated the framing of the problem in terms of estimating the character identity for each pixel position given the entire image of the page as context. We can think of a 512x512 page's image as a (512,512,3) tensor. And we can think of the character identity per position as a (512,512,C) tensor where C is the number of possible characters.

We elected to use Deep Learning to learn these conditional distributions. Because this task involves learning both local and global context, we employ the U-Net architecture which was originally developed for semantic segmentation of biological cells [9].

Given our conditional independence assumptions, we elected to use maximum likelihood training with a multinomial distribution for each position, which has parameters estimated by the neural network. This is equivalent to using the cross-entropy loss for each pixel in the output.

This method is computationally expensive when the vocabulary size is large, so we opt to only model the most common characters. This still allows us to spot many characters, but being able to spot all characters is an important area for future work.

Finally, as noted earlier, there is an issue that each pixel's character assignment was treated as independent distributions. This means that if the character's identity is ambiguous, multiple conflicting character assignments could be made for a given character. We address this by first considering all the pixels in the 5x5 bounding box around the center of a character (using the estimate from our model of the center of character's positions, as explained earlier). Then if the probability of a given character in each position exceeds a threshold (in our case 90%), we add that position into a clustering algorithm which runs across the entire page. In our case we used DBSCAN. The result of this DBSCAN is a single assignment for each character position that we detect.

Figu

5.2 U-Net

The U-Net is a convolutional neural network architecture that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg, Germany [9]. The U-Net network consists of two parts, first, the contracting path and then the expansive path, hence the u-shaped architecture. The contracting path is a typical convolutional network, each followed by a rectified linear unit (ReLU) and a max pooling operation. During the contraction, the spatial information is reduced while the number of filters is increased. The expansive pathway combines both local information and global context, allowing for spatially precise prediction that doesn't discard global information.

Due to the size of the images, we elected to use stochastic gradient descent without minibatches (i.e. a single image per update). The original U-Net architecture used batch normalization, which is not well suited to small batch sizes. More concretely, we found that when using batch normalization in our case, the model would produce many spurious detections on unusual pages. To address this problem, we used the Group Normalization technique [10] (which normalizes across all positions and a subset of the filters) and found that it significantly improved results.

6. Experiment Setup and Results

We trained on data from two books: *Ugetsu Monogatari* and *Koushoku Ichidai Otoko*, but in order to test the model with the same layout and similar characters, we divided the images into train and test set on 80/20 ratio. From total 526 images, we chose 420 images randomly to train and 106 to test the accuracy. Below we show a ground-truth image with bounding boxes for three characters (ground truth) shown for clarity:

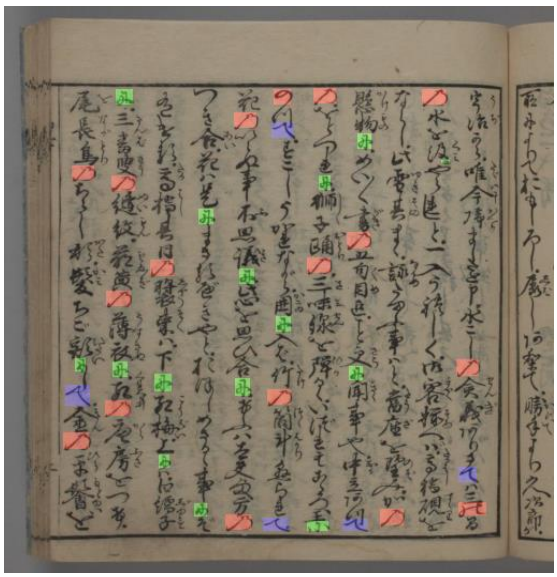

Figure 7: Ground-truth image with bounding boxes for tree characters. Red: No (の), green: Ni (に), blue: Te (て)

We trained using the Adam optimizer [11] and with a single example in the batch, due to memory constraints and the size of the image. While the original U-Net used batch normalization, we found that this led to spurious detections on pages not containing significant amount of hiragana. We corrected this by replacing the batch normalization with group normalization, which has been shown to produce superior results when training on small batch sizes. After a single epoch if training, we achieve the following prediction in Figure 8.

Finally, in order to test and calculate the accuracy of the model, we particularly chose 10 characters which are in similar shape and hard to recognize for even for human as shown in Figure 9. In our hypothesis, we supposed the

model would do poorly on these characters.

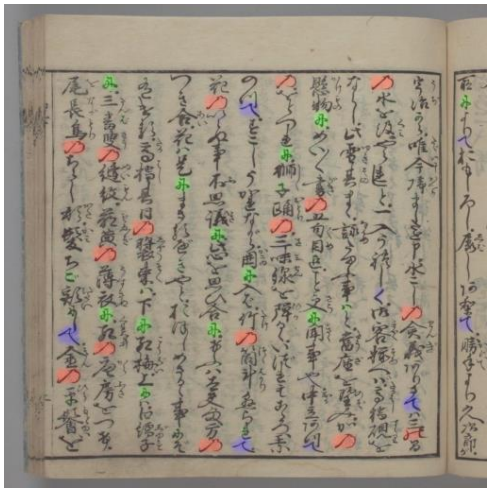

Figure 8: Prediction by the model after the first epoch. The model missed one No (の) on top of 6th column and some light green on other characters like Ha (は) and Re (れ) on 3rd and 6th columns which considered very few. The model also gave prediction of characters on previous page on the right side even though the ground truth doesn't contain the data on this image.

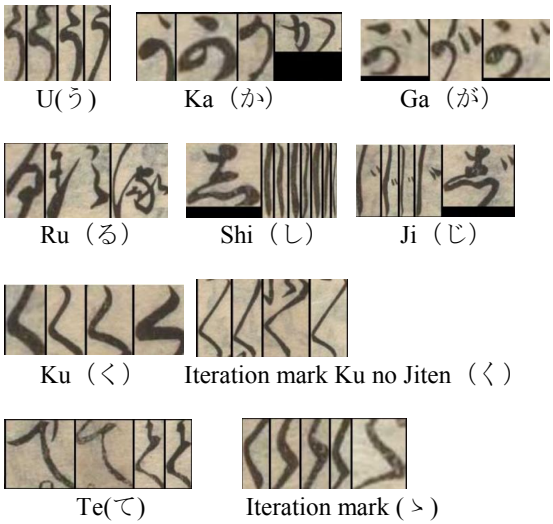

Figure 9: 10 characters chosen to test the model.

We trained for 30 epochs since train and test losses don't get lower afterward. It took average 7 minutes per epochs with two NVIDIA GeForce GTX 1080Ti.

Overall the model performed well considering we only trained the model on 420 pages. However, the accuracy for Ji (じ) is quite low because the model missed the Dakuten (゛) and not detected anything or detected as Shi (し). Even though the accuracy is low, when the model gives high probability that the character is Ji, it rarely made mistake.

One interesting result from this experiment is the iteration marks. Ku iteration mark or くの字点 is considered to be hard to distinguish from hiragana Ku (く) for human. However, the model made very few mistakes when detected it. This is probably because in Kuzushiji, most of the time the iteration mark was written on far right of the column or probably they are slightly bigger than other characters. We don't know for sure what feature the model consider a character as Ku iteration mark instead of hiragana Ku, but the accuracy is high for this character.

For the small iteration mark (ゝ) for hiragana is considered to be very hard to recognize for human because it is very small and always connected with above character. However, the model performed quite good. This is because the whole page data keeps the aspect ratio of the character unlike the cropped character image.

Finally, the model does well with overlapped characters as well. Some of interesting results are shown in figure 11.

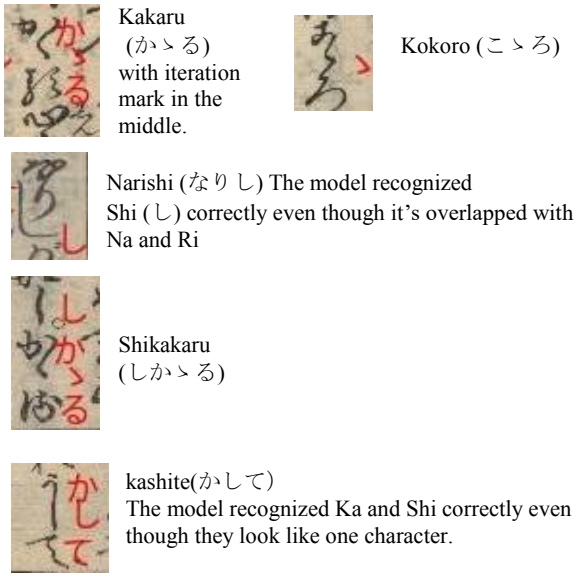

Figure 11: Interesting results from the model.

Figure 12: The whole page result of *Kōshoku Ichidai Otoko*. The transcribed characters are shown on the right side of characters for convenience.

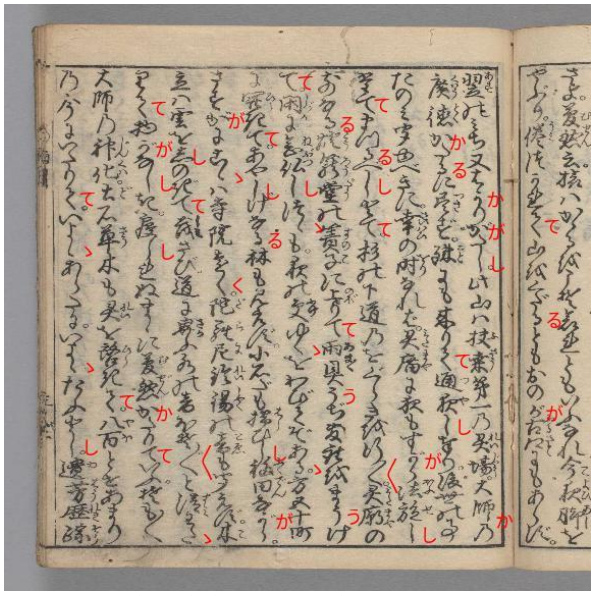

Figure 13: The whole page result of Ugetsu Monogatari. Interesting part is the model detected character on part of right page cropped in the image even though the ground-truth csv doesn't contain the data of this part in for the same page image.

7. Future work

A natural step forward would be to address the complete task of Kuzushiji recognition. There are significant differences between our current system and what would be required for Kuzushiji recognition. One is that our current system does not scale very well in the total number of characters which will take a lot of GPU memory in training.

Small characters next to the main text or "Rubi" (ルビ) or Furigana (ふりがな) is something that the model doesn't perform well or not detected at all. Even though manuscripts before 17th century rarely used Rubi, it is very common in Edo period and we hope to deal with the problem in future.

Another challenge for the model is the dataset we used was created mostly from woodblock printed books which are different from handwritten books or Shahon (写本). We still don't know how well the model will perform on handwriting characters.

At the moment, the model may perform well on books from Edo period because of the data the model was trained with. However, manuscripts from earlier eras used a lot more types of Jibo in hiragana. We hope we can expand the dataset with older manuscripts so the model will be more useful in future.

8. Conclusion

A significant amount of Japanese writing prior to the reforms in 1900 is unintelligible and remains inaccessible to all but specially trained scholars. Most of the existing text remains unconverted to the contemporary Japanese writing system. This has motivated a significant amount of work on using machine learning to perform this conversion. Traditionally these systems have worked by segmenting the text into distinct characters and converting them separately. However, this approach fails for several reasons, the most significant one being the highly contextual nature of the Kuzushiji writing system. We have proposed a new system for recognizing keywords in Kuzushiji text which aims to get around these limitations by considering context throughout the image instead of classifying each character individually.

References

[1] Iwanami Shoten 1963 *The General Catalog of National Books* (『国書総目録』), Iwanami Shoten, Tokyo.

[2] Koichi Takahiro 2013 Notation of the Japanese Syllabary seen in the Textbook of the Meiji first Year. The bulletin of Jissen Women's Junior College 34, pp. 109-119 [https://ci.nii.ac.jp/els/contents110009587135.pdf?id=ART0010042265]

[3] Nakano Kōichi 1978 *Hentaigana No Tebiki* （『変体仮名の手引』） ,Musashino Shoin, Tokyo

[4] Many research papers such as Kengo Terasawa et al, 2005 Word Spotting for Historical Document Images, the Meeting on Image Recognition and Understanding or Kengo Terasawa 2009 Slit Style HOG Feature for Document Image Word Spotting (10th International Conference on Document Analysis and Recognition)

[5] Nguyen et al 2017 Attemps to Recognize Anomalously Deformed Kana in Japanese Historical Documents, IEICE PRMU 2017 Algorithm Contest.

[6] Neal Digre, Carpedm python library for downloading, viewing and manipulating image data, originally developed for the Kuzushiji dataset. [https://github.com/SimulatedANeal/carpedm]

[7] Saikaku Ihara 1682 *Koshoku Ichidai Otoko* (『好色一代男』), Aratoya Magobei Kashin, Osaka. Data from the Kuzushiji dataset.

[8] Akinari Ueda 1776 *Ugetsu Monogatari* (『雨月物語』), Shinsaibashi kitakyuhojimachi, Osaka. Data from the Kuzushiji dataset.

[9] Olaf Ronneberger, Philipp Fischer and Thomas Brox 2015 U-Net: Convolutional Networks for Biomedical Image Segmentation [https://arxiv.org/abs/1505.04597] [http://codh.rois.ac.jp/char-shape/book/200014740/]

[10] Yuxin Wu and Kaiming He 2018. Group Normalization. ECCV 2018. [http://openaccess.thecvf.com/content_ECCV_2018/papers/Yuxin_Wu_Group_Normalization_ECCV_2018_paper.pdf]

[11] Diederik P. Kingma and Jimmy Ba 2014 Adam: A Method for Stochastic Optimization. [http://arxiv.org/abs/1412.6980]