SN Computer Science (2020) 1:177
https://doi.org/10.1007/s42979-020-00186-z

ORIGINAL RESEARCH

KuroNet: Regularized Residual U-Nets for End-to-End Kuzushiji Character Recognition

Alex Lamb[1] · **Tarin Clanuwat**[2] · **Asanobu Kitamoto**[2]

Received: 30 January 2020 / Accepted: 3 May 2020 / Published online: 21 May 2020

Abstract

Kuzushiji, a cursive writing style, had been used in Japan for over a thousand years starting from the eighth century. Over 3 million books on a diverse array of topics, such as literature, science, mathematics and even cooking are preserved. However, following a change to the Japanese writing system in 1900, Kuzushiji has not been included in regular school curricula. Therefore, most Japanese natives nowadays cannot read books written or printed just 150 years ago. Museums and libraries have invested a great deal of effort into creating digital copies of these historical documents as a safeguard against fires, earthquakes and tsunamis. The result has been datasets with hundreds of millions of photographs of historical documents which can only be read by a small number of specially trained experts. Thus there has been a great deal of interest in using machine learning to automatically recognize these historical texts and transcribe them into modern Japanese characters. Our proposed model KuroNet (which builds on Clanuwat et al. in International conference on document analysis and recognition (ICDAR), 2019) outperforms other model for Kuzushiji recognition. In this paper, KuroNet achieves higher accuracy while still recognizing entire pages of text using the residual U-Net architecture from adding more regularization. We also explore areas where our system is limited and suggests directions for future work.

Keywords Kuzushiji · Character recognition · Machine learning · Japan · Historical document

Introduction

Kuzushiji or cursive style Japanese characters were used in the Japanese writing and printing system for over a thousand years from the eighth century to the beginning of the nineteenth century. However, the standardization of Japanese language textbooks (known as the *Elementary School Order*) in 1900 [21] unified the writing type of Hiragana and also made the Kuzushiji writing style obsolete and incompatible with modern printing systems. Therefore, most Japanese natives cannot read books written just 150 years ago. However, according to the General Catalog of National Books [11] there are over 1.7 million books written or published in Japan before 1867. Overall it has been estimated that there are over 3 million books preserved nationwide [4]. The total number of documents is even larger when one considers non-book historical records, such as personal diaries. Despite ongoing efforts to create digital copies of these documents, most of the knowledge, history and culture contained within these texts remain inaccessible to the general public. One book can take years to transcribe into modern Japanese characters. Even for researchers who are educated in reading Kuzushiji, the need to look up information (such as rare words) while transcribing as well as variations in writing styles can make the process of reading texts time-consuming. Additionally entering the text into a standardized format after transcribing it requires effort. For these reasons, the vast majority of these books and documents have not yet been transcribed into modern Japanese characters. This has led to a great deal of interest in automatically

This article is part of the topical collection "Document Analysis and Recognition" guest edited by Michael Blumenstein, Seiichi Uchida and Cheng-Lin Liu.

✉ Tarin Clanuwat
tarin@nii.ac.jp

Alex Lamb
lambalex@iro.umontreal.ca

Asanobu Kitamoto
kitamoto@nii.ac.jp

[1] Mila, Université de Montréal, Quebec, Canada

[2] ROIS-DS Center for Open Data in the Humanities, National Institute of Informatics, Tokyo, Japan

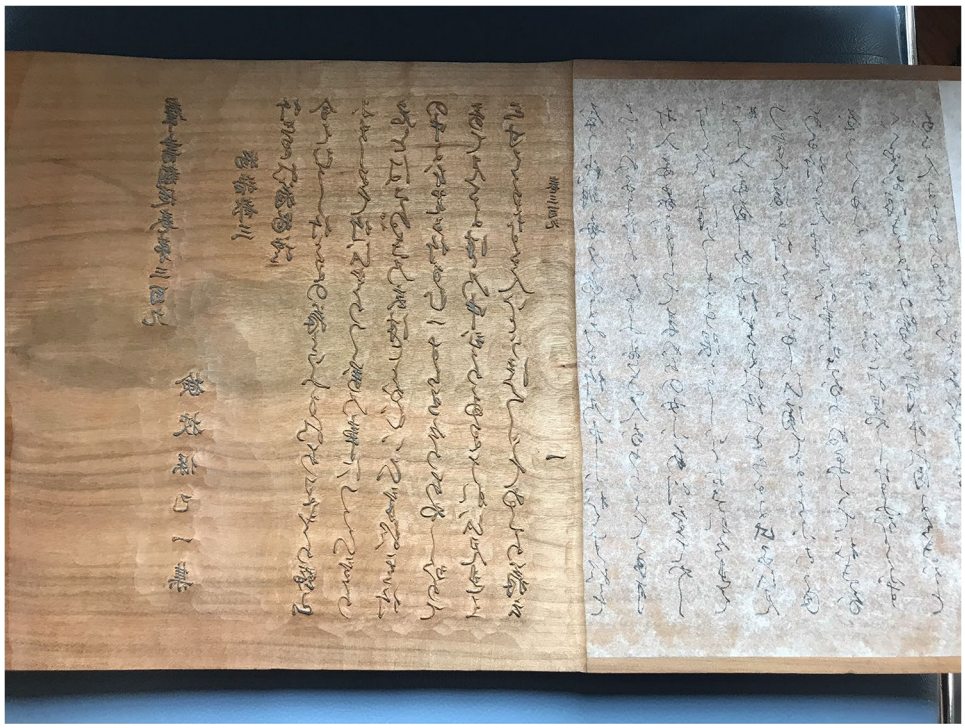

Fig. 1 An example of a Woodblock for book printing (replica) carved using handwritten text on a thin paper attached to a woodplank on opposite side. (Hanawa Hokiichi Museum, Tokyo)

converting these documents into the modern Japanese writing system.

A Brief Primer on the History of the Japanese Language

We introduce a small amount of background information on the Japanese language and writing system to make the recognition task more understandable. Since Chinese characters entered Japan prior to the eight century, the Japanese wrote their language using Kanji (Chinese characters in the Japanese language) in official records. However, from the late ninth century, the Japanese began to add their own character sets: Hiragana and Katakana, which derive from different ways of simplifying Kanji. Individual Hiragana and Katakana characters don't contain independent semantic meaning, but instead carry phonetic information (like letters in the English alphabet). The usage of Kanji and Hiragana in the Heian period (794–1185 A.D.) depended upon the gender of the writer: Kanji was used by men and Hiragana was used by women. Only in rare cases such as that of Lady Murasaki Shikibu, the author of *the Tale of Genji*, did women have any Kanji education. This is why stories like *the Tale of Ise* or collections of poems such as *Kokinwakashū* were written in mostly Hiragana and official records were written in Kambun (a form of Classical Chinese used in Japan written in all Kanji without Hiragana). However, even in Kambun official record like *Midōkampakuki* which was written mostly in Kambun recorded conversation and poems in Hiragana. This is the main reason why the number of character classes in Japanese books can greatly fluctuate. Some books have mostly Kanji while other books have mostly Hiragana. Additionally, Katakana was used for annotations in order to differentiate them from the main text.

In pre-modern Japanese documents, two printing systems were used: movable type printing and woodblock printing. Woodblock printing dominated the printing industry in Japan until the nineteenth century (see Fig. 1). Even though these books were printed, characters carved in the block were copied from handwritten documents. Therefore, woodblock printed characters and handwritten characters share similar visual features. Since woodblocks were created from a whole piece of wood, it was easy to integrate illustrations with texts. Thus, many books have pages where text is wrapped around illustrations. Another note is that even though a lot of pre-modern Japanese documents were written in clean and separated columns, this layout is far from universal. Text in pre-modern Japanese was read from top to bottom (vertically) and then left to right (the same reading order as modern Japanese). However, the reading style can differ from document to document due to the layout. Many times the irregular layout of text makes reading order extremely hard to determine.

One of the reasons for this is the "Chirashigaki" writing style. The Chirashigaki style was a very common writing style for poems and Hiragana letters (see Fig. 2). It was popular because of the beauty of the layout. A writer

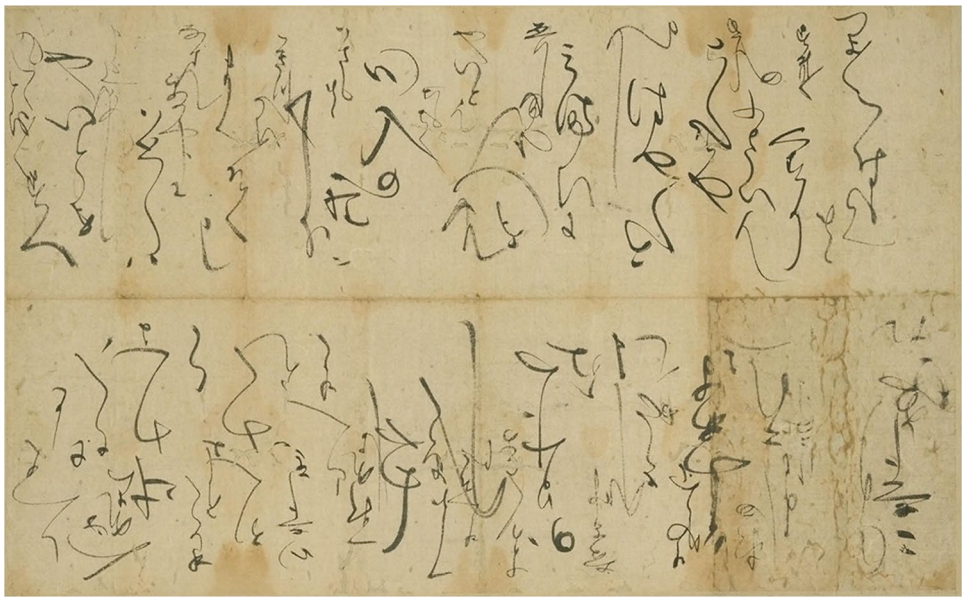

Fig. 2 A Kana (Hiragana) Letter by Hosokawa Tadaoki (1563–1600) to his wife (The National Diet Library) is an example of Chirashigaki writing style in letter which was a common way to write; however, it is extremely hard for a human, let alone an OCR system, to decipher the reading order

would be considered skillful if they could use this writing style elegantly. Instead of writing character in straight columns, the text was written in curve and ascending lines. The space between columns will also vary. Because of the popularity of Chirashigaki, when books were printed by carving text on a plank of wood in the Edo period (seventeenth–nineteenth century), publishers tried to imitate the Chirashigaki writing style.

Chirashigaki and irregular layouts in pre-modern Japanese documents are the reason why transcribing Kuzushiji using text sequence is very hard and not practical in the real world outside of carefully selected and prepared datasets. It is also a reason why Kuzushiji recognition models like KuroNet, which don't assume that the text is aligned as a sequence achieves higher accuracy and is able to work with a greater variety of documents.

Our method offers the following contributions:

- An improved general algorithm for pre-modern Japanese document recognition which uses no pre-processing and is trained using character locations instead of character sequences.
- Demonstration of our novel algorithm on recognizing pre-modern Japanese texts. Our results dramatically exceed the previous state-of-the-art. The train data and test data are completely separated. We trained the model on 29 books and tested the model on 15 books. Using KuroNet we found that:
  - 6 books had an F1-score between 90 and 100%
  - 9 books had an F1-score between 80 and 90%
- A solution to recognize Kuzushiji text even when it includes illustrations.
- An exploration of why our approach is well-suited to the challenges associated with the historical Japanese character recognition task.

KuroNet

The KuroNet method is motivated by the idea of processing an entire page of text together, with the goal of capturing both long-range and local dependencies. KuroNet first receives an image (from the random cropping process) of size 976×976. Then the image is passed through a U-Net architecture to obtain a feature representation of size $C \times 976 \times 976$, where C is the number of channels in the final hidden layer (in our experiments we used $C = 64$).

We refer to the input image as $x \sim p(x)$ and refer to the correct character at each position (i, j) in the input image as $y_{ij} \sim P(y_{ij}|x)$.

We can model each $P(y_{ij}|x)$ as a multinomial distribution at each spatial position. This requires the assumption that characters are independent between positions once we've conditioned on the image of the page. Another issue is that the total number of characters in our dataset is relatively large (over 4000), so storing the distribution parameters of the multinomial at each position is computationally expensive. For example at a 976×976 resolution this requires 15 gigabytes of memory just to store these values.

To get around this, we introduce an approximation where we first estimate if a spatial position contains a character or

if it is a background position. We can write this Bernoulli distribution as $P(c_{ij}|x)$, where $c_{ij} = 1$ indicates a position with a character and $c_{ij} = 0$ indicates a background position. Then we model $P(y_{ij}|c_{ij} = 1, x)$, which is the character distribution at every position which contains a character.

This allows us to only compute the relatively expensive character classifier at spatial positions which contain characters, dramatically lowering memory usage and computation. This approximation is called Teacher Forcing and it has been widely studied in the machine learning literature [9, 13]. The Teacher Forcing algorithm is statistically consistent in the sense that it achieves a correct model if each conditional distribution is correctly estimated, but if the conditional distributions have errors, these errors may compound. To give a concrete example in our task, if our estimate of $p(c_{ij}|x)$ has false positives, then during prediction we will evaluate $p(y_{ij}|c_{ij} = 1, x)$ at positions where no characters are present and which the character classifier was not exposed to during training. Besides computation, one advantage to using teacher forcing is that simply estimating if a position has a character or not is a much easier task than classifying the exact character, which may make learning easier.

The result of our training process is two probability distributions:

$P(y_{ij}|c_{ij} = 1, x)$ and $P(c_{ij} = 1|x)$. As we produce estimates that may be inconsistent across different spatial positions, we use clustering as a post-processing step to ensure that each character image in the text is only assigned a single class. To do this we used the DBSCAN clustering algorithm [8], which does not require the number of classes to be tuned as a hyperparameter, and we report the hard cluster centers.

Training and Architecture Details

We trained for 156 epochs with a batch size of one. We found that using higher-resolution images improved results qualitatively, so we elected to use larger resolutions even though this required us to use a batch size of 1 to stay within GPU memory. We used the SGD with momentum optimizer, based on the experimental result found elsewhere that it improves generalization [3, 25] over adaptive optimizers. We used a learning rate of 0.001 which we dropped to 0.0001 at 100 epochs, and we used a fixed momentum of 0.90.

We used the residual FusionNet variant [17] of the U-Net architecture [18] to compute the features. We used four downsampling layers followed by four upsampling layers, with skip connections passing local information from the downsampling layers to the upsampling layers. We used 64 channels in the first downsampling layer and doubled the number of channels with each downsampling layer, following the procedure used by [18]. On each upsampling layer, we halved the number of channels. Our final hidden representation had 64 channels at each position.

The most significant change we made to the [17, 18] architecture was to replace all instances of batch normalization [10] with group normalization [26]. Batch normalization makes the assumption that every batch has the same batch statistics (that each feature will have the same mean and variance). This approximation is often poor when the batch size is small, which is especially true in our case as we used a batch size of one. While our results with batch normalization were okay, we noticed that the model often struggled with unusual pages—for example in pages containing large illustrations it would struggle to not predict characters over those parts of the image. We found that these issues were resolved by using group normalization. When using group normalization, we always used 16 groups.

Regularization

We also added a few simple regularizers to improve generalization performance. First we used mixup to slightly modify the input images [29]. We used a variant where the labels are unchanged but the mixing distribution is Beta$(\alpha, \alpha + 1)$. We set $\alpha = 0.4$ and to be conservative we clamped the resulting α to be between 0.0 and 0.3. Our goal with this setup was to encourage the model to only mix in a small amount from different examples while retaining the original label. Thus we picked relatively conservative hyperparameters that led to images which did not make it too hard to read the original image. [23] explored the possibility of underfitting when doing Mixup, and we wanted to avoid this possibility. We provide an example of mixup with $\lambda = 0.3$ in Fig. 3.

Many books are written on a relatively thin paper, so the content of the adjacent page is often faintly visible through the paper. This can be seen in the center image in Fig. 3. While it is somewhat subjective, the images produced by mixup appear similar to images where the adjacent page's content are faintly visible. Thus Mixup may have an added benefit of helping to encourage the model to ignore the adjacent page, as it is not useful for correct character recognition.

As an additional regularizer, we slightly perturbed the brightness on each update between 90 and 110%, to simulate variations in lighting when the books were photographed as well as differences in the darkness of the paper (which are not relevant for the recognition task).

Additionally when evaluating the KuroNet model, we use a model with an exponential moving average of the parameters seen during training (with $\beta = 0.9999$). This improved results, especially in early epochs, and is essentially computationally free. This technique has been used successful in semi-supervised learning [22] and has also become commonplace in improving the sample quality from generative models [27].

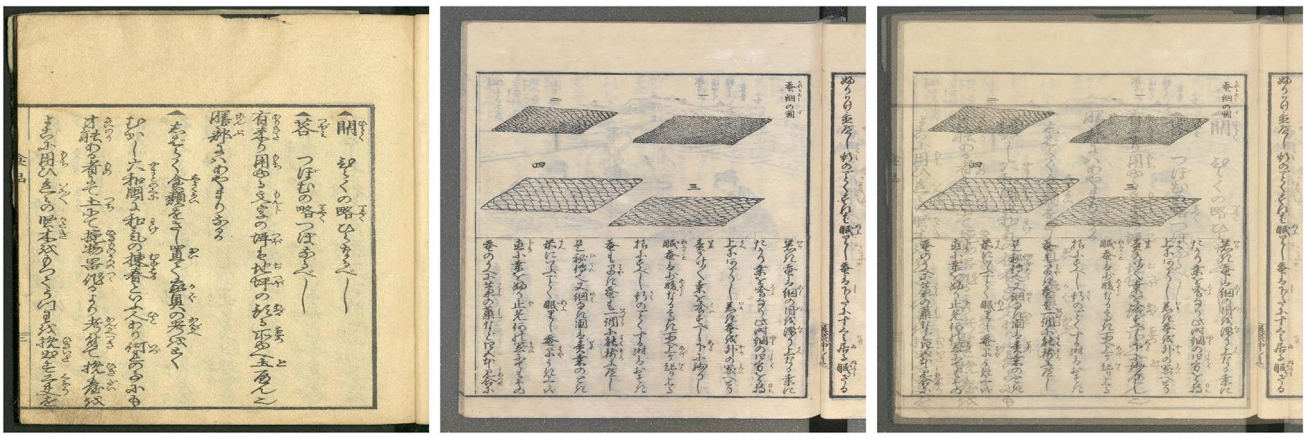

Fig. 3 An example of images used during training with the mixup regularizer [29]. We mix the left image (30%) with the middle image (70%) to produce the right image (mixed). 30% is the strongest mixup rate that we could have sampled during training

Related Work

KuroNet

The KuroNet technique is also described in a conference proceeding [6].

In this work, we introduce several changes to the model, which we show together lead to substantial improvements:

- We added a random cropping regularization in which we train on aspect-ratio preserved crops from the original image.
- We evaluate on a higher resolution at test time than what was used during training, which is enabled by the use of a fully convolutional architecture.
- The conference paper [6] used the Adam optimizer [12] but here we have switched to SGD, as there is research suggesting that it generalizes better.
- We use an exponential moving average of the parameters [28] for prediction. We found that this made convergence faster and have less variance between epochs.

Work Prior to KuroNet

Previous approaches have considered aspects of the kuzushiji recognition task, although no complete end-to-end system has been proposed prior to this work (excluding the original KuroNet [6]). An approach specifically for character spotting using U-Nets was proposed in [5]. However, this considered detecting only 10 chosen Hiragana characters and thus was not an attempt to make the document fully readable, but rather to pick out a few specific common characters.

Segment and Classify Characters Individually One approach that has been explored is to first segment the image into patches for individual characters and then classify each patch separately. This approach is very computationally appealing, but is inappropriate for the general Kuzushiji recognition task due to the contextual nature of many characters. This was explored by [16], where they used a dataset for the paper in which the problem was simplified with input given as a single column of already extracted kuzushiji characters.

The Kuzushiji-MNIST (KMNIST [4]) produced datasets consisting of individually segmented kuzushiji characters. The datasets were created to be used as an alternative benchmark for machine learning algorithms, but not to be directly applicable for end-to-end Kuzushiji recognition.

Sequence Models A widely studied approach to handwriting recognition involves learning a sequence model over the characters which is conditioned on the image of the text [2]. This was explored for Japanese historical documents specifically by [14], but using documents between 1870 and 1945 which characters in the their dataset, while old style prints, are not Kuzushiji, but closer to modern Japanese characters.

The sequence modeling approach has a major limitation for Kuzushiji because the layout of the text is not necessarily sequential, and in many cases trying to produce a sequential ordering for the text would lead to substantial ambiguity. The dataset that we used did not have an explicit sequential structure (see "Data" section). Another disadvantage is that generating characters sequentially usually requires sampling the characters one-by-one and in-order. This prediction may be slower than prediction with our model, which processes the entire page in parallel. Although the correct kuzushiji character sequence is difficult to define, we think this problem is important for the future of kuzushiji recognition research which we discuss in "Future Work" section.

Challenges in Kuzushiji Recognition Task

We identify several challenges in Kuzushiji which make it challenging for standard handwriting recognition systems and explain how KuroNet addresses these challenges.

Context Kuzushiji characters are written in such a way that some characters can only be recognized by using context, especially the identity of the preceding characters. For example, a simple line can indicate a few different characters, depending on the preceding characters. Because KuroNet uses both local and global context, it is able to disambiguate by using the surrounding characters. This is a challenge for models which segment the characters before classifying them individually.

Large Number of Characters The total number of characters in Kuzushiji is very large (our dataset contains 4328 distinct characters), but their distribution is long-tailed and a substantial fraction of the characters only appears once or twice in the dataset. This is due to the unique structure of the Japanese language—which consisted at the time of two type of character sets, a phonetic alphabet (with simple and extremely common characters) and non-phonetic Kanji characters. Kanji consists of both common and rare characters: some of them are highly detailed, while others are just one or two straight lines.

The large number of characters presents challenges for both computation and generalization. KuroNet addresses this challenge computationally by using Teacher Forcing to only evaluate the character classifier at positions where characters are present. However, it may still present a challenge for generalization, because many Kanji characters only appear a few times in the training data.

Hentaigana One characteristic of Classical Hiragana or *Hentaigana* (the term literally translates to "Character Variations") which has a huge effect on the recognition task is that many characters which can be written a single way in modern Japanese could be written in multiple different ways in pre-modern Japanese. This is one reason why the pre-modern Japanese variant of the MNIST dataset [4] is much more challenging than the original MNIST dataset [15]. Many of the characters in pre-modern Japanese have multiple ways of being written, so successful models need to be able to capture the multi-modal distribution of each class.

Sayre's Paradox A well-studied problem in document recognition [20] is that with cursive writing systems the segmentation and recognition tasks are entangled. This is successfully handled by sequence models which use the entire image as context (for example, using attention) or convolutional models which have access to larger context from the page. This provides further motivation for KuroNet using the same hidden representations to predict $p(y_{ij}|c_{ij}, x)$ and $p(c_{ij}|x)$.

Annotations Versus Main Text Pre-modern Japanese texts, especially in printed books from the Edo period (seventeenth–nineteenth century), are written such that annotations are placed between the columns of the main text (usually in a smaller font). These annotations mostly act as a guide for readers on how to pronounce specific Kanji and were written in either Hiragana or Katakana (in our dataset from the Edo Period, Hiragana was more commonly used). For our task, we did not consider annotations to be part of the text to be recognized since our dataset doesn't contain labels for annotations (and if it did, we would still want to discriminate between the main text and annotations). In this setting, our model needs to use context and content to discriminate between the main text and the annotations so that the annotations can be ignored. Another problem that makes annotation challenging is difficulty in determining whether the small characters are annotations or the main text. This type of small text is one reason why automatically extracting character sequences is hard in kuzushiji books. A particular challenging example is shown in Fig. 4.

Layout The layout of Kuzushiji characters does not follow a single simple rule, so it is not always trivial (or even possible) to express the characters as a sequence. Some examples of this include characters being written to wrap around or even integrate into illustrations. Another example is a writing style used for personal communications where the ordering of the characters is based on the thickness of their brush strokes in addition to their spatial arrangement. An example of this is shown in Fig. 5. Still another practice involves the use of coded symbols to indicate breaks and continuation in text. This is a major challenge for systems that assume the data are in a sequence. As KuroNet does not make this assumption, it does not have any general difficulty with this, and our model performs well on several books which have non-sequential layouts. Additionally our data ("Data" section) did not have any sequential information in its labels and in practice we found that using heuristics to estimate the sequential ordering was quite challenging, especially due to the presence of annotations.

Experiments

The primary goal of our experiments is to validate the extent to which KuroNet is able to successfully recognize real historical Japanese documents. We evaluate both models on realistic setting which is books never seen by the model. As secondary goals, we wish to understand the cases where KuroNet doesn't perform well, to motivate future applied research on Kuzushiji, and also to study where it does perform well, to suggest which ideas from KuroNet could be more broadly useful in other areas of document recognition.

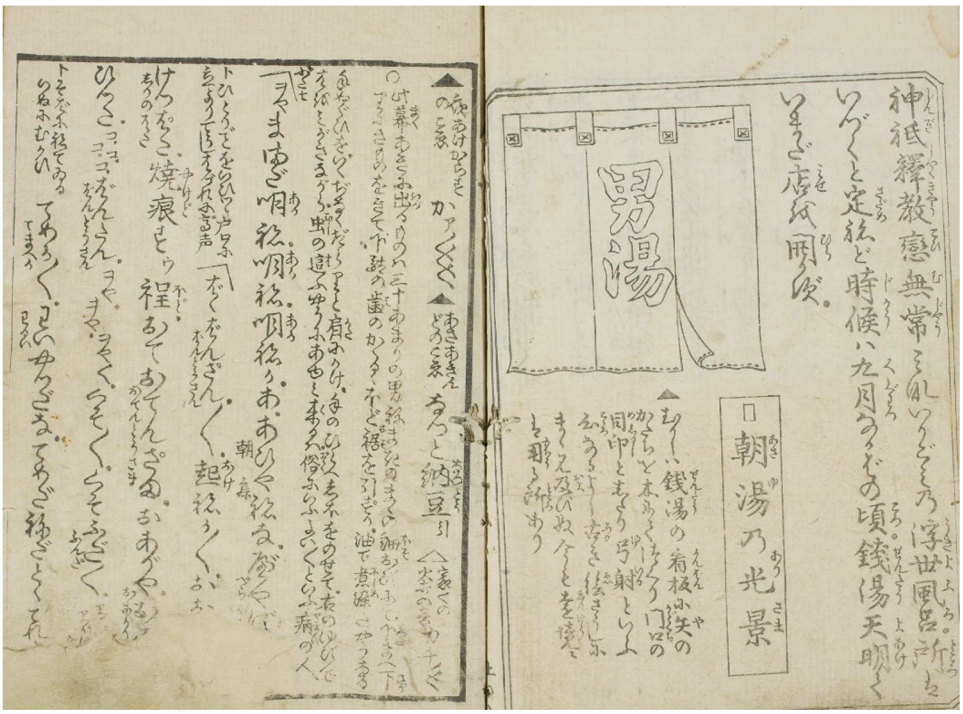

Fig. 4 Ukiyoburo, a literature from Kuzushiji dataset (Book ID 200015779 from the data in "Data" section). The page shows that there are many double column texts. Although the second column seems like annotation, it's actually explanations about the scene which are considered as main text and not annotations. This type of main text alignment was very common in Kokkeibon or comical book genre in the Edo period

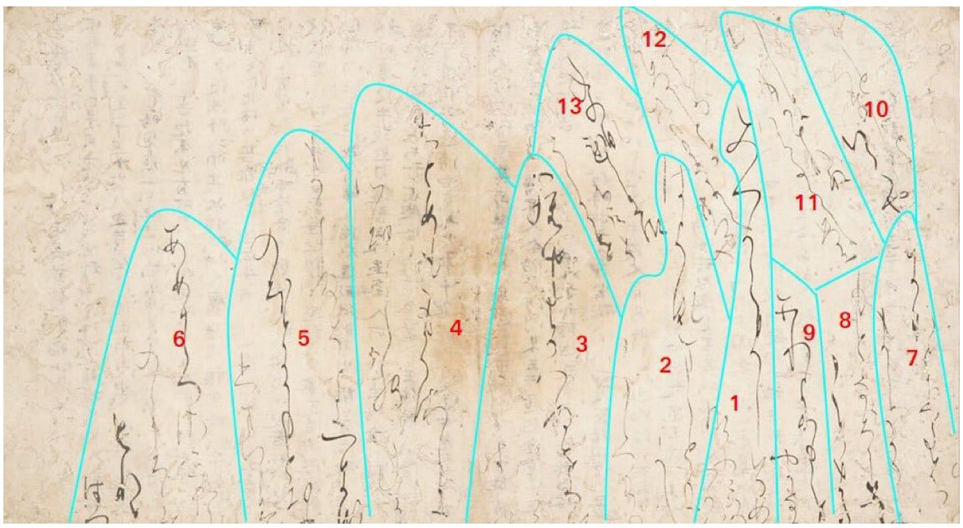

Fig. 5 An example showing how the text is not necessarily laid out in a clear sequential order. The red numbers indicate an ordering for how the text should be correctly read. The reading order of this type of text is determined by the size of characters and the darkness of the ink. The correct reading order begins with the biggest and darkest characters on the page

Data

The Kuzushiji dataset.[1] that we use to train and evaluate KuroNet was created by the National Institute of Japanese Literature (NIJL) and is curated by the Center for Open Data in the Humanities (CODH). When it was first released in 2016, the dataset had only 80,000 character images. The dataset has been updated since and as of the end of 2019, the dataset 1,086,326 character images with 4328 character classes. The detail of Kuzushiji dataset is as shown in (Table 1).

The dataset was created mostly from woodblock printed book from the Edo period (1603–1868). Only a few of them were handwritten. As mentioned earlier in the history of the Japanese language, even though the text was printed, it was carved from handwritten manuscripts. Therefore, the characters are very close to handwritten ones. One big difference is in printed books, the layout is normally easier to understand than the layout in handwritten documents.

The Kuzushiji dataset consists of 3 parts.

[1] http://codh.rois.ac.jp/char-shape/book/.

Table 1 List of books in Kuzushiji Dataset

Book ID	Title	Genre	Year	Total classes	Total chars
100241706	Usonarubeshi	Literature	1834	786	8527
100249371	Teisahiroku	Food	1852	726	9580
100249376	Gozenkashi Hidenshou	Food	1718	401	11822
100249416	Mochigashi Sokuseki Teseishu	Food	1805	468	7950
100249476	Meshihyakuchinden	Food	n/a	639	7838
100249537	Ryourichinmishu	Food	1764	817	12358
200003076	Koushoku Ichidaiotoko	Literature	1682	1668	63959
200003803	Genji Monogatari	Literature	1654	237	11132
200003967	Oragaharu	Literature	1878	1112	11197
200004107	Ninin Bikuni	Literature	1660	711	8636
200004148	Chinsetsu Yumiharizuki	Literature	1807	1971	38572
200005598	Keiseikai Shijuhatte	Literature	1790	658	16133
200005798	Seken Munesanyou	Literature	1692	1211	37887
200006663	Chiguchi	Literature	n/a	78	121
200006665	Kirishitan Monogatari	Literature	1639	420	16883
200008003	Kagakuteiyou	Literature	1850	841	12791
200008316	Bukegiri Monogatari	Literature	1682	1176	37707
200010454	Genji Monogatari	Literature	n/a	193	11566
200014685	Nansou Satomi Hakkenden	Literature	1842	1731	15864
200014740	Ugetsu Monogatari	Literature	1765	1922	44832
200015779	Ukiyoburo	Literature	1813	1743	60381
200015843	Nippon Eitaigura	Litearture	1688	1669	50087
200017458	Soga Monogatari	Literature	1671	166	29584
200018243	Tama Kushige	Literature	1789	676	13623
200019865	Wominaheshi	Literature	n/a	760	39183
200020019	Chikusai Monogatari	Literature	1727	312	33163
200021063	Usuyuki Monogatari	Literature	1691	314	17593
200021071	Isoho Monogatari	Literature	n/a	610	45358
200021086	Isoho Monogatari Chukan	Literature	1659	718	15284
200021637	Touseiryouri	Food	n/a	417	4871
200021644	Kashiwa Funabashi	Food	1841	778	12313
200021660	Yousanhiroku	Agriculture	1803	1711	32525
200021712	Manbouryouri Himitsubako	Food	1786	821	24480
200021763	Zenburyourishou	Food	n/a	700	11397
200021802	Ryouri Monogatari	Food	1643	555	19607
200021851	Katemono	Medical	1802	427	5599
200021853	Nichiyou Souzaiso Fujino	Food	1836	594	9046
–	Chinkyaku Sokuseki Houchou	–	–	–	–
200021869	Ryourikata Kokoroe no Koto	Food	n/a	330	3003
200021925	Shinpen Ikoku Ryouri	Food	1861	687	4259
200022050	Ryouri Hidenshou	Food	1684	255	9545
200025191	Nise Monogatari	Literature		468	21702
brsk00000	Butsurui Shouko	Dictionary	1775	2171	75462
hnsd00000	Hiyokurenri hananoshimadai	Literature	1838	1907	83492
umgy00000	Shunshoku Umegoyomi	Literature	1832	1660	79415

- Whole page of Kuzushiji document images (Total 5879 images).
- Each character images labeled by character types.
- CSV files for each book giving pixel coordinates and character size of every characters in the dataset.

How we determined what to use in training and testing is purely according to Kaggle Kuzushiji Recognition competition[2] we hosted in July 2019. Since it was a competition so the label of test data couldn't be publicly available during competition period (3 months from July–October 2019) so the test data were the private dataset CODH had at that time. Our test data consist of 15 books:

200004107, 200005798, 200006665,
200008003, 200008316, 200010454,
200015843, 200017458, 200018243,
200019865, 200020019, 200021063,
200021071, 200021086, 200025191.

The dataset used for [6] is a precursor to the dataset used for the experiments in this paper. The previous dataset consisted of only 684,165 character images which is only 68% of the dataset used in this paper.

Setup

We included images without text labels in training and treated the entire page as $c_{ij} = 0$ for training $p(c_{ij}|x)$. Generally, pages without labels consisted primarily of an illustration or a cover and a very small amount of text.

In KuroNet, we performed random cropping during training to vary the exact position and size of the crop used for training. Then at test time, we evaluated by running the U-Net on the entire uncropped page but at a higher resolution than was used for training (which is allowed because the U-Net is a fully convolutional architecture). Suppose the desired final size of the training image is s (in practice we used $s = 976$). We set a cropping ratio of $r = 50\%$, and thus we load the original image and re-scale to a resolution of $s/r \times s/r$. We then select a random size for the crop s_c uniformly from s to s/r. We then randomly sample an offset in the x dimension uniformly between s_c and s/r and randomly sample an offset in the y dimension between s_c and s/r. We then select a square crop of size $s_c \times s_c$ with the sampled x and y dimension offsets, and then re-scale this selected crop to a final desired size of $s \times s$.

Effectively, this technique preserves the aspect ratio of the original image, but trains on a variety of different re-scalings during training, and also provides a variable amount of boundary surrounding the page. Intuitively we would expect that this would make the model more robust to variability in character size as well as the quality of the scanned image. In practice, one issue is that this approach leads the model to see a variety of characters, but with the size of characters kept the same or increased. To make the test setting more similar to this, we evaluated the same model but at a higher resolution. In practice, we found using resolutions between 1024×1024 and 1280×1280 at test time slightly improved results. However, when using higher resolution images, KuroNet often recognizes annotations which were written in smaller characters. These annotations don't have ground truth in the dataset because they were omitted in data creating process. Hence, KuroNet sometimes has more false positive due to higher resolution images.

Quantitative Results

Even though KuroNet achieves better results through improved regularization (random cropping at training), several types of errors still occur in recognition. One is that a character is present but our model makes no prediction (false negative). Another is that the model predicts a character at a position where no character is present or predicts the wrong character when a character is present (false positive). Refer to Fig. 6 for more details.

We use precision and recall which are two basic quantitative metrics to describe performance. The number of correctly predicted characters divided by the total number of predicted characters is defined as precision. We define recall as the number of correctly predicted characters divided by the total number of characters present in the ground truth. To score a true positive, the model must provide center point coordinates that are within the ground truth bounding box and a matching Unicode label. We also report the F1-score, which is simply the harmonic mean of the precision and the recall. Even though the metric gives idea how the model performs overall, there are problems with this metric which discuss in "Problem of the Evaluation Metric: F1 Score" section.

We present our quantitative results in four different tables. In Table 2, we show overall results of KuroNet and ablations where regularization is removed. For comparison, we also reduced the model's capacity by using a standard U-Net instead of a residual U-Net [18] and only predicting the 400 most common characters (instead of the 4000 predicted by KuroNet). We refer to this second ablation as UNet-Small. This result is average of precision, recall and F1 from total 15 books in the test data. KuroNet with additional regularization leads to superior results. Random cropping in the training process helps KuroNet deal with different character sizes better. In Table 3, we report F1-score of each books in the test data. As would be expected, KuroNet with added regularization gives higher accuracy in most books in the test data. However, there are three books (200008003, 200008316, 200010454) where the added regularizations slightly lowers F1. In Tables 4 and 5, we show the precision and recall across different books.

[2] https://www.kaggle.com/c/kuzushiji-recognition.

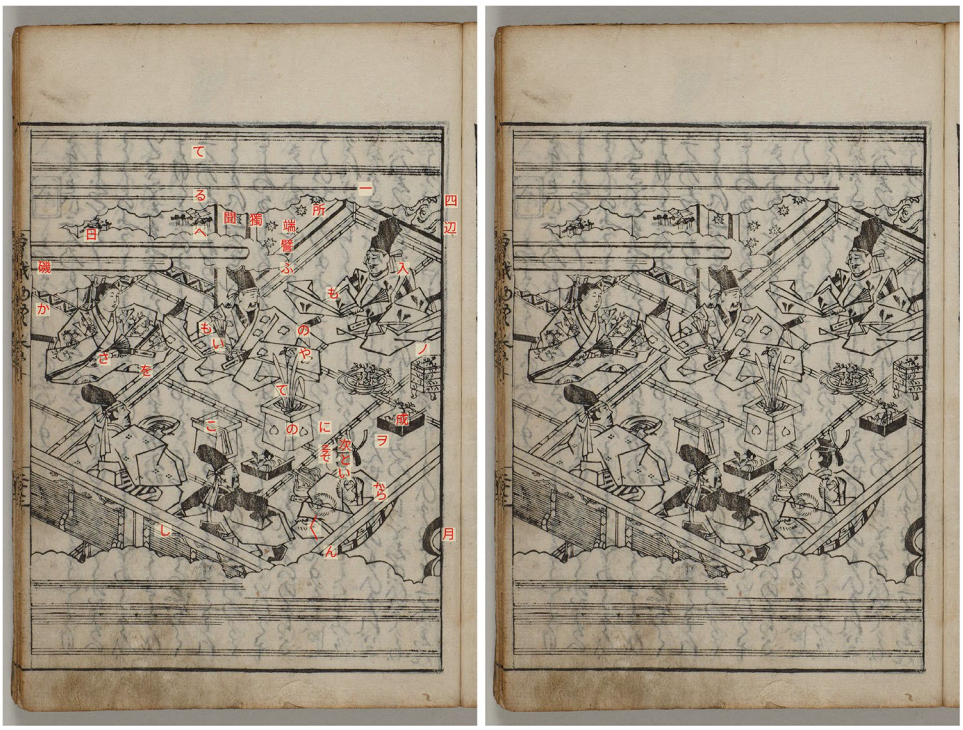

Fig. 6 Through better regularization, Kuronet greatly reduces the number of predictions which are false positives (right) compared to the baseline KuroNet with less regularization (left)

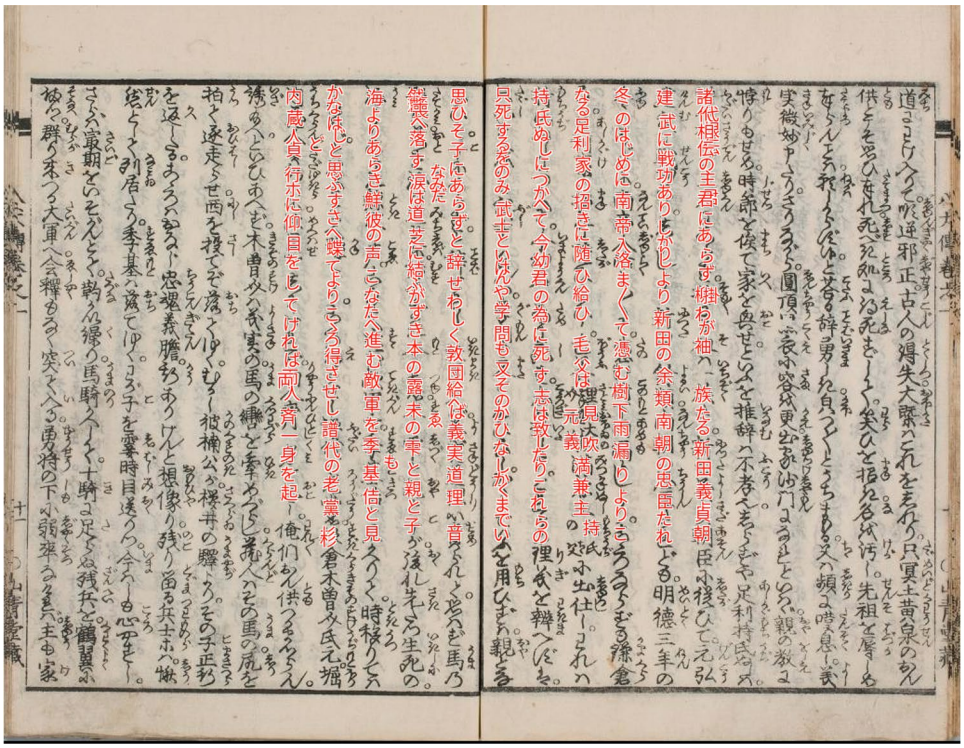

Fig. 7 KuroNet is better able to transcribe only partial of a page due to the random cropping introduced in training process. This is an important feature because some images are just mainly illustration and we only need to transcribe short caption in the image instead of feeding the whole picture into the model

We also found that adding random cropping to image in training process helps KuroNet handle image in different size better. While KuroNet only trained on full-page image without cropping, when we give partial of a page image into KuroNet, the model struggles to recognize the character due to image resizing when perform prediction task. However, for KuroNet with the random cropping regularization, we can select only part of a page and give to the model and it still performs relatively well. (See Fig. 7)

Although random cropping helps KuroNet to be able to deal with variable text sizes better, the model still struggles with very small or very big text. This also includes annotation. In Kuzushiji dataset, annotation is not included because it's very small text next to main text. Therefore, labeling annotation is hard and very time-consuming. However, since KuroNet with random cropping regularization can deal with font size better, many times the model tries to predict annotations which are not in the ground truth csv. Even though the model predicts them correctly, these annotation will be counted at false positive when we calculate the overall accuracy.

Analysis of Results

We have identified areas where our model still needs improvement:

- Difficulty in correctly recognizing characters which are very large in size. When the area of one character is large, the model gives duplicate prediction results even though the character predicted is correct as seen in Fig. 8. Sometime, the model also tries to separate one big Kanji into smaller radicals so it gives prediction in multiple smaller characters. We also see poor performance on small characters (see Figs. 9, 10).
- Although the overall F1 score is high for KuroNet, the model still struggles when predicting rare Kanji. This is probably not the model problem, but rather dataset problem. In order to make the model be able to generalize for most kind of documents, we need more data (preferably handwritten document data) to train the model.
- Since the model doesn't assume text sequence, therefore the model cannot give text output at the moment. KuroNet is extremely useful to aid human readers when they read Kuzushiji documents. However, in order to give output as text, we need to adjust the model so that it can also predict the text sequence. One thing to keep in mind is that text sequence in Kuzushiji documents is a very hard problem. Text in straight columns is easy to define. However, there are a substantial number of documents in irregular layouts such as Chirashigaki (see "A Brief Primer on the History of the Japanese Language" section).

Problem of the Evaluation Metric: F1 Score

Although F1 score is a simple way to give us an idea about the model's overall performance, there is one downside of F1 score that ought to be considered. Since F1 is calculated from the harmonic mean between precision and recall, the score can often be higher if the model doesn't try to predict rare characters. It's often better to leave a blank space to become a false negative rather than predicting something and getting a false positive. For KuroNet, we found that if we don't let the model try to predict all character classes (roughly 2400 classes appeared in the test data), but decrease the number of class in half such that it only predicts the most commonly appearing 1240 classes, the model will get slightly higher F1 (0.8984 rather than 0.8957).

Fig. 8 An example of large font sizes in the title page of a book, which causes our model to struggle (predictions in red). Even though the KuroNet with improved regularization can predict some characters correctly which is a lot of improvement over the less regularized KuroNet result, it gives duplicate predictions because the characters are very large in size. Another problem is Kanji consists of radicals which sometime similar to one smaller characters. The model struggles to see if the radical is one character or a part of big character. Finally, these title pages appear at most once per book and supervised learning often struggles when the amount of labeled data is small

Since the purpose of Kuzushiji recognition task is to transcribe Kuzushiji document, the model should be able

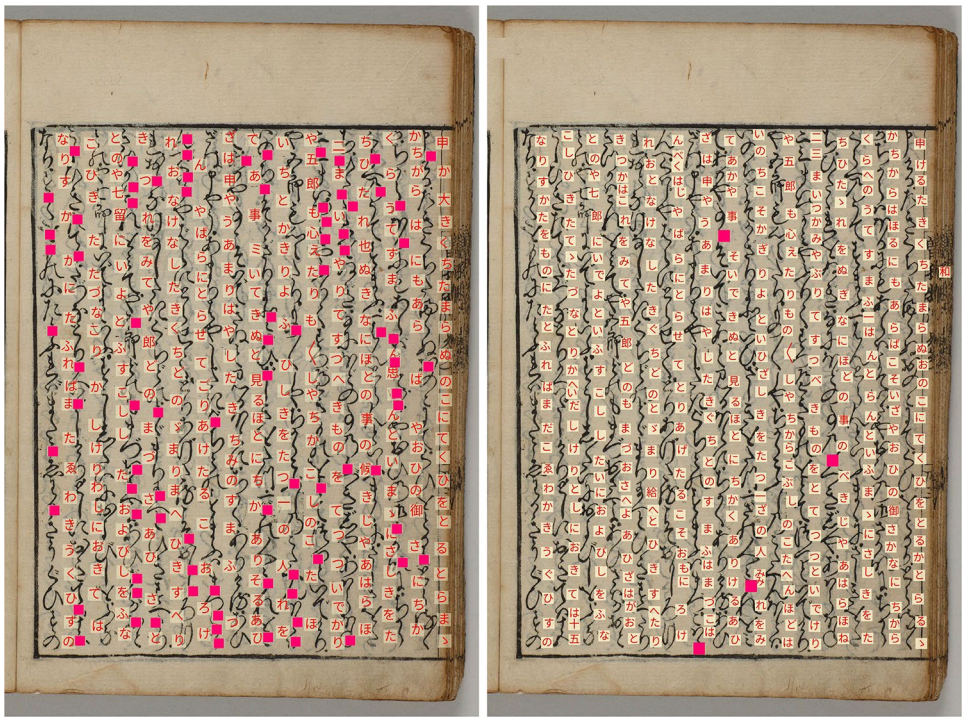

Fig. 9 Result of the same page in the book 200021063 which KuroNet has its poorest result, because the book has an unusually high number of character per page. Hence, the space between each character is very small. The pink squares indicate where false negative is present. While KuroNet without limited regularization (left image) missed many characters, KuroNet with added regularization (right image) only missed a few of them

Fig. 10 Because KuroNet doesn't assume character sequence when perform prediction task, the method can be used with document where text is wrapped around illustration which was very common layout in pictorial story books. However, the prediction accuracy relies greatly on the quality of the image and the size of characters. If characters are too small, the model will give false negative as seen in some part of this figure

to transcribe overall document correctly and also able to transcribe less frequent characters. Improving model performance by cutting number of predictable classes is not suitable in real situation. However, in Kaggle Kuzushiji Recognition competition, many participants cut down the number of classes so their models get higher F1 score.

Whether leaving a rare character blank is desirable is a nuanced question. For a reader, is it better to read an incorrect character or a blank position? From the view of F1-score, the blank position leads to a substantially better score. However from a reader's perspective, we'd argue that it's often better for the model to predict an incorrect

Table 2 Overall performance of KuroNet and the effect of improved regularization across 15 books which not seen at all during training

Total 1996 pages	UNet-Small	KuroNet	Kuronet+ImprovedReg
Precision	0.5221	0.7964	0.8889
Recall	0.4731	0.7509	0.9025
F1 Score	0.4943	0.7730	0.8957

Table 3 UNet-Small, KuroNet and KuroNet+ImprovedReg F1 score by book

Book ID	UNet-Small	KuroNet	KuroNet+Reg
200004107	0.344023	0.828114	**0.883649**
200005798	0.47206	0.722134	**0.884032**
200006665	0.583233	0.811294	**0.899238**
200008003	0.722185	**0.933349**	0.922256
200008316	0.563525	**0.854751**	0.852443
200010454	0.344856	**0.855555**	0.844926
200015843	0.468846	0.744535	**0.910278**
200017458	0.546979	0.731627	**0.949569**
200018243	0.529266	0.887782	**0.900186**
200019865	0.424169	0.782885	**0.895329**
200020019	0.429851	0.745377	**0.901095**
200021063	0.403956	0.352337	**0.880328**
200021071	0.665483	0.838477	**0.905842**
200021086	0.671698	0.753097	**0.887031**
200025191	0.244377	0.714276	**0.884105**

Best F1 scores are in bold

Table 4 UNet-Small, KuroNet and KuroNet+ImprovedReg Precision score by book

Book ID	UNet-Small	KuroNet	KuroNet+Reg
200004107	0.369393	0.778708	0.872809
200005798	0.491396	0.775391	0.878944
200006665	0.603044	0.807131	0.883257
200008003	0.755867	0.931166	0.917325
200008316	0.596003	0.839333	0.842049
200010454	0.389114	0.852287	0.843034
200015843	0.499367	0.742832	0.901105
200017458	0.546855	0.806243	0.946381
200018243	0.54152	0.885414	0.87668
200019865	0.444678	0.789542	0.887304
200020019	0.471764	0.793021	0.892866
200021063	0.456546	0.549717	0.895414
200021071	0.694383	0.841072	0.902501
200021086	0.720118	0.796496	0.881725
200025191	0.251042	0.714985	0.87776

Table 5 UNet-Small, KuroNet and KuroNet+ImprovedReg Recall score by book

Book ID	UNet-Small	KuroNet	KuroNet+Reg
200004107	0.322528	0.884214	0.894761
200005798	0.455374	0.675723	0.889179
200006665	0.565221	0.815499	0.915808
200008003	0.692647	0.935542	0.927240
200008316	0.536254	0.870747	0.863097
200010454	0.311062	0.858848	0.846826
200015843	0.443089	0.746247	0.91964
200017458	0.548013	0.669653	0.952779
200018243	0.518649	0.890162	0.924987
200019865	0.406153	0.77634	0.903500
200020019	0.397357	0.703133	0.909477
200021063	0.376026	0.259251	0.865742
200021071	0.639358	0.835898	0.909207
200021086	0.644957	0.714183	0.892401
200025191	0.239268	0.713568	0.890543

character. One reason is that many characters share common visual features, so if the model's predicted character is visually similar to the correct character, it may still be a useful aid in reading the document. For example, many kanji characters contain a symbol on the left and a symbol on the right, so the model's prediction may successfully aid reading if it is correct on the left but incorrect on the right. At the same time, it is difficult to quantify the trade-off between the cost of incorrect predictions and the value of making certain incorrect predictions.

Future Work

Since KuroNet model can recognize Kuzushiji in the Kuzushiji dataset relatively well with high accuracy already, the next problem we need to consider is how to make the model able to generalize across broader type of documents. Although we have over one million characters in the Kuzushiji dataset already, this number is still much smaller than the number of characters in real-world situations. Moreover, the dataset was created from printed books especially in the Edo period, it will take more work to make the model be able to recognize handwritten documents especially messy ones in irregular layout.

CODH has released the KuroNet as a publicly available service on its website, and thus it will be possible to get feedback from human users in order to get correct labels for

images outside the Kuzushiji dataset. The Kuzushiji dataset consumes both time and budget to create. However, if we can use KuroNet to reduce the cost in creating the dataset, we should be able to get labels for a greater variety of data.

One thing that we can consider is that the amount of unlabeled data which is available vastly exceeds the amount of labeled data, especially when one only considers the labeled data which are public and easily accessible (the dataset we used is public and has 44 labeled books). The Pre-Modern Japanese Text dataset, also from the same institution, contains 3126 unlabeled books, so this task could be an excellent setting for using semi-supervised learning. This has been successfully demonstrated in a variety of settings. Dumoulin et al. [7] demonstrated successful semi-supervised learning using latent variables inferred from generative models. Strong results on semi-supervised learning have been demonstrated [1, 24] using variants of the mixup algorithm, which we already use as a regularizer for supervised learning ("Regularization" section).

An additional area for future work would be to create better methods to improve generalization on Kanji characters, which is challenging due to the Kanji alphabet's large vocabulary size. In KuroNet, we model $p(y_{ij}|c_{ij}, x)$ as a multinomial distribution at each position—thus the final weight matrix in the character classification output layer has separate parameters for each character. If we could somehow group or identify related kanji characters, then we could share more parameters between them and perhaps generalize much better. There is extensive work from the Machine Learning literature on "Few Shot" learning, where few examples are available for a particular class [19].

Our current dataset and model does not contain labels for the annotations placed between text columns ("Challenges in Kuzushiji Recognition Task" section). Thus our model is trained to ignore these annotations. In the future, it would be useful to produce a model which can read them.

One of the biggest challenges in future work is to get text output from KuroNet. We need to explore the task of automatically converting these character coordinate lists into a single text sequence for a page in reading order. These text sequences are necessary for machine translation, cataloguing and search, which are critical for Kuzushiji recognition research. This problem sounds simple but creating algorithms that can handle most documents is a very challenging task due to irregular layout in each page which is common in pre-modern Japanese documents.

Conclusion

We have proposed and experimentally evaluated a model for recognizing pre-modern Japanese text. We have shown that several aspects of these documents make this task challenging, including the large vocabulary size, long-range interactions and complicated layouts. We have proposed and experimentally validated KuroNet, a new updated approach which addresses these challenges by jointly reading and recognizing complete pages of text. We have also identified several open challenges in this area to motivate future work. In particular, our model only trains on labeled data and does not take advantage of the plentiful unlabeled data which is available for this task. KuroNet gives high recognition accuracy and uses no pre-processing when making predictions, making it easy to apply to real-world data. As a result, we hope that this will be an initial step toward making pre-modern Japanese books more accessible to the general public and preserving the cultural heritage of the Japanese people.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Beckham C, Honari S, Verma V, Lamb AM, Ghadiri F, Hjelm RD, Bengio Y, Pal C. On adversarial mixup resynthesis. In: Advances in neural information processing systems. 2019. p. 4348–4359.
2. Bezerra BLD, Zanchettin C, de Andrade VB. A hybrid RNN model for cursive offline handwriting recognition. In: Brazilian symposium on neural networks. 2012. p. 113–118. https://doi.org/10.1109/SBRN.2012.41.
3. Chen J, Gu, Q. Closing the generalization gap of adaptive gradient methods in training deep neural networks. arXiv preprint arXiv:1806.06763 (2018).
4. Clanuwat T, Bober-Irizar M, Kitamoto A, Lamb A, Yamamoto, K, Ha D. Deep learning for classical Japanese literature. Neural information processing systems (Neurips) creativity workshop. http://arxiv.org/abs/1812.01718 (2018).
5. Clanuwat T, Lamb A, Kitamoto A. End-to-end pre-modern Japanese character (kuzushiji) spotting with deep learning. In: Proceeding of information processing society of Japan, Jinmoncom. 2018.
6. Clanuwat T, Lamb A, Kitamoto A. Kuronet: Pre-modern Japanese kuzushiji character recognition with deep learning. In: International conference on document analysis and recognition (ICDAR). 2019.

7. Dumoulin V, Belghazi I, Poole B, Mastropietro O, Lamb A, Arjovsky M, Courville A. Adversarially learned inference. 2016.
8. Ester M, Kriegel HP, Sander J, Xu X, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceeding of Association for the advancement of artificial intelligence (AAAI). 1996.
9. Goyal A, Ke NR, Lamb A, Hjelm RD, Pal C, Pineau J, Bengio Y. Actual: Actor-critic under adversarial learning. arXiv preprint arXiv:1711.04755 (2017).
10. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015).
11. Iwanamishoten: the general catalog of national books. Iwanamishoten. 1963.
12. Kingma DP, Ba J. Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
13. Lamb AM, GOYAL AGAP, Zhang Y, Zhang S, Courville AC, Bengio Y. Professor forcing: a new algorithm for training recurrent networks. In: Advances in neural information processing systems. 2016. p. 4601–4609.
14. Le Duc A, Mochihashi D, Masuda K, Mima H. An attention-based encoder-decoder for recognizing Japanese historical documents. In: Pattern recognition and machine understanding (PRMU). https://www.ieice.org/ken/paper/20181213H1IM/eng/ (2018).
15. LeCun Y. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/ (1998).
16. Nguyen HT, Ly NT, Nguyen KC, Nguyen CT, Nakagawa M. Attempts to recognize anomalously deformed kana in japanese historical documents. In: Proceedings of the 4th international workshop on historical document imaging and processing. ACM; 2017. p. 31–36.
17. Quan TM, Hildebrand DGC, Jeong W. Fusionnet: a deep fully residual convolutional neural network for image segmentation in connectomics. CoRR abs/1612.05360. http://arxiv.org/abs/1612.05360 (2016).
18. Ronneberger O, Fischer P, Brox T. U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer; 2015. p. 234–241.
19. Santoro A, Bartunov S, Botvinick M, Wierstra D, Lillicrap T. One-shot learning with memory-augmented neural networks. arXiv preprint arXiv:1605.06065 (2016).
20. Sayre KM. Machine recognition of handwritten words: a project report. Pattern Recognit. 1973;5(3):213–28. https://doi.org/10.1016/0031-3203(73)90044-7.
21. Takashiro K. Syllabary seen in the textbook of the meiji first year. The bulletin of Jissen Women's Junior College. Jissen Joshi Tankidaigaku Kiyou. 2013. p. 109–119.
22. Tarvainen A, Valpola H. Weight-averaged consistency targets improve semi-supervised deep learning results. CoRR abs/1703.01780. http://arxiv.org/abs/1703.01780 (2017).
23. Verma V, Lamb A, Beckham C, Najafi A, Courville A, Mitliagkas I, Bengio Y. Manifold mixup: learning better representations by interpolating hidden states. 2019.
24. Verma V, Lamb A, Kannala J, Bengio Y, Lopez-Paz D. Interpolation consistency training for semi-supervised learning. In: International joint conference on artificial intelligence (IJCAI). arXiv:1903.03825 (2019).
25. Wilson AC, Roelofs R, Stern M, Srebro N, Recht B. The marginal value of adaptive gradient methods in machine learning. In: Advances in neural information processing systems. 2017. p. 4148–4158.
26. Wu Y, He K. Group normalization. arXiv preprint arXiv:1803.08494 (2018).
27. Yazıcı Y, Foo CS, Winkler S, Yap KH, Piliouras G, Chandrasekhar V. The unusual effectiveness of averaging in gan training. arXiv preprint arXiv:1806.04498 (2018).
28. Yazıcı Y, Foo CS, Winkler S, Yap KH, Piliouras G, Chandrasekhar V. The unusual effectiveness of averaging in GAN training. In: International conference on learning representations. 2019.
29. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017).