

Constructing an enriched digital wasan (和歌) database with general-purpose deep learning models

Yago Diez

yago@sci.kj.yamagata-u.ac.jp

Yamagata University

Mariano Cabezas

University of Girona

M`arius Vila

University of Girona

James Harry Morris

National Institutes for the Humanities

Katsushi Waki

Yamagata University

Research Article

Keywords: Document Analysis, Image Processing, Deep Learning, Object Detection, Wasan

Posted Date: October 1st, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11084430/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Constructing an enriched digital wasan (和算) database with general-purpose deep learning models

Yago Diez^{1*}, Mariano Cabezas^{2,3}, Màrius Vila⁴, James Harry Morris^{5,6},
Katsushi Waki⁷

^{1*}Faculty Of Science, Yamagata University, Yamagata, Japan.

²Research Institute of Computer Vision and Robotics (VICOROB), University of Girona, Girona, Spain.

³Macquarie University Hearing Research Centre, Faculty of Medicine, School of Health Science and Nursing, Macquarie University, Sydney, Australia.

⁴Institute of Informatics and Applications, University of Girona, Girona, Spain.

⁵Center for Innovative Research, National Institutes for the Humanities, Tokyo, Japan.

⁶Research Department, National Museum of Japanese History, Sakura, Japan.

⁷School of Collaborative Regional Innovation and Data Science, Yamagata University, Yamagata, Japan.

*Corresponding author(s). E-mail(s): yago@sci.kj.yamagata-u.ac.jp;

Contributing authors: mariano.cabezas@udg.edu; marius.vila@udg.edu;
morrisjamesharry@gmail.com; waki@sci.kj.yamagata-u.ac.jp;

Abstract

Purpose. Wasan (和算) documents represent a unique Japanese cultural mathematical heritage but they are difficult to read and process automatically, placing them out of reach for most readers. This work examines whether general-purpose computer vision and deep learning tools, without specially designed methods, can support the construction of an enriched digital database of such documents.

Methods. Working with the Sakuma Bunko collection held at Yamagata University, we address two tasks. For noise reduction, three observers scored global and local thresholding methods against explicit criteria. For character detection, we compared two classical blob detectors with seven deep learning object detection architectures, trained on 19,937 annotated characters. For context, two established systems, the National Diet Library’s Kotenseki OCR and KuroNet, were also run on the same test pages.

Results. All seven architectures exceeded a 0.90 F1 score, with the best model reaching 0.947 precision and 0.964 recall. From the denoising methods, Sauvola’s threshold was preferred over Otsu’s because it preserved the text better. Finally, annotation time per scan fell from around two hours to three minutes when starting from deep learning predictions when compared to annotating from scratch.

Conclusion. Historical document analysis problems of this difficulty can now be addressed with general-purpose tools. What determines the outcome is not the choice of architecture but the annotated data and an understanding of how these tools behave on the material at hand.

1 Introduction

The formal study of mathematics was brought to Japan from China during the Asuka Period (592–710) and became an important part of the Japanese bureaucracy and the upper echelons of Japanese society [1, 2]. However, it is not until the Edo period (1603–1867) that the development of Japanese mathematics (wasan 和算) really began to emerge as a discipline distinct from the mathematics of the continent [3]. During this period, Japanese mathematics developed in relative isolation. Although there was still influence from overseas with mathematical treatise imported from China and Korea helping to spur greater interest in mathematics during the 17th century, Japan’s restrictive trade policy limited the number of trade partners and with it the outside influence in science [3, 4]. In this context, mathematics changed from an intellectual pursuit for erudites into a wider cultural phenomenon, taken up voluntarily and in public, for pleasure, regardless of wealth or gender [5]. This culture left two main types of material trace. Sangaku (算額)–votive tablets inscribed with geometric problems and dedicated at shrines and temples, where a result could be displayed publicly and a problem left open as a challenge to later visitors–of which around 900 are thought to survive nationwide¹. And mathematical books, which combined advanced mathematical content with a strong recreational character and circulated as a pastime, were studied throughout Japan.

Formally, wasan texts often contain geometric problems, including determining the length and area of figures using logic and deductive reasoning. A textual description accompanies the graphical depiction of the problem, written in a literary standard called kanbun (漢文, Sinitic writing). There were also materials aimed at people who had acquired only basic levels of literacy such as the *Jinkōki* 塵劫記 (first published in 1627). This paper focuses only on texts written in kanbun. Kanbun uses Chinese characters and syntax

to represent Japanese content, sometimes alongside a system of diacritical and syntactic markers that tell the reader how to reorder and augment the characters so as to conform to Japanese grammar [6]. Wasan texts are highly structured: for example, all problem descriptions start with the sequence of characters 今有如圖, meaning “Now, as shown in the figure”. In addition to the figures and the text, wasan documents also include equations and, in handwritten documents, lines to aid writing. Figure 1 shows a typical page, with the opening formula, the problem description, the accompanying diagram and the answer highlighted.

This historical mathematical culture can be an important asset in mathematical education in the present day. Currently, however, it is largely out of reach. Due to the use of kanbun, which is no longer widely read, historical character forms, and highly specific technical vocabulary, these pages are difficult to read for the general public. Furthermore, their irregular layout makes them challenging to process automatically. Making this vast trove of material usable requires the preservation of the documents as well as finding ways to process them efficiently and present them in a language that modern audiences can understand.

This paper is part of a larger project aimed at building a database of wasan documents, working with the Sakuma Bunko (佐久間文庫), a historical archive held in Yamagata University Library² that currently comprises 1,280 volumes, all of which have been digitized. Scanned images alone provide limited access to the contents of a collection of this size. Our aim is to establish a framework that uses computer vision (CV) and deep learning (DL) tools to extract information from the documents automatically and provide it alongside them to “enrich” the database.

The framework consists of three stages that can be generalized to different types of information extraction. The first stage is the preparation of the material to ease the extraction of information and limit the effect of conservation and

¹<https://sangaku-archive.org/>

²<https://wasan-database.yum-archives.net/manuscripts>

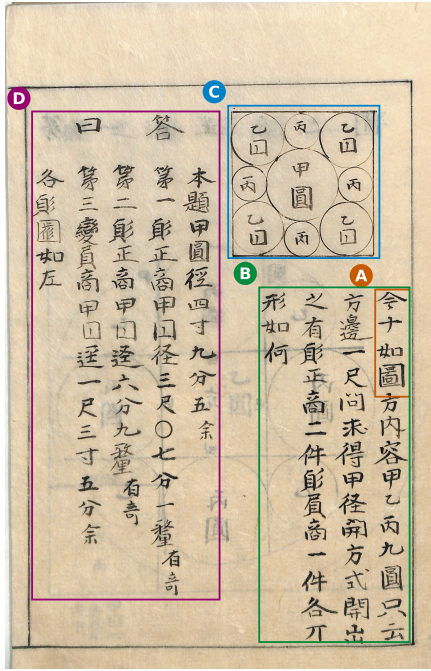


Fig. 1: A page from the Sakuma Bunko showing the structure of a wasan problem. A: the opening formula 今有如图 (“Now, as shown in the figure”), which begins every problem description. B: the problem description, read from top to bottom and right to left. C: the diagram depicting the problem. D: the answer, introduced by 答.

scanning artefacts. The second stage is the definition and annotation of the information of interest. This stage requires expert knowledge of the documents and accounts for the greater part of the effort involved. The final stage is the training and evaluation of DL models. This evaluation should focus on the final practical goal. Given suitable annotations, the same procedure yields a tool for targets such as the positions of the equations, salient kanji such as 今, or the graphical depictions of the problems. This paper follows the framework through two tasks in detail, noise reduction and character detection, reporting the cost and the outcome of each. Tools developed following this philosophy can aid document interpretation and allow aggregating and searching by content. Consequently, this can lead to a deeper study and understanding of these documents by a wider public and large-scale linguistic processing of pre-modern Japanese materials.

2 State of the art

While our database is composed of Japanese documents from the Edo period, the methodology required to process them is related to other problems in CV and optical character recognition (OCR). Two research areas are particularly relevant to our work: OCR for premodern Chinese sources, where considerable progress has been made [7, 8], and the recognition of *kuzushiji* (くずし字), cursive writing in which the great majority of premodern Japanese books were written. Neither transfers directly to kanbun, for which no dedicated OCR system exists despite its use from the classical period to the modern era [6, 9, 10].

Kuzushiji recognition, largely built on the publicly available dataset [11] of the same name, has proven potentially useful for application to wasan documents. In particular, KuroNet [12] is a popular browser-based software tool to automatically transcribe premodern Japanese materials written in cursive script. KuroNet was introduced over a series of papers [12–14] and subsequently deployed for practical use in the mobile application *miwo* [15]. Rather than segmenting pages into lines, as most OCR engines do, KuroNet takes a semantic segmentation approach, using a U-Net that predicts the presence of a character at each image position (pixel) and which specific character. These locations are also classified to determine which specific character was found. This design was motivated by premodern Japanese layouts in which text may wrap around illustrations usually with irregular reading order. However, network outputs are given in a pixel format and to identify characters these neighbouring predictions are merged with DBSCAN (a classical clustering algorithm). The pipeline is thus a hybrid approach that requires manual fine-tuning for the classical components. The authors report 0.889 precision, 0.903 recall and 0.896 F1 over 1,996 pages from 15 held-out books [12].

Recognition of kuzushiji was also the subject of a Kaggle competition in 2019³. From all the submitted solutions, object detection approaches were well represented among the leading solutions with cascade Mask R-CNN [16] as the competition winner. To clarify, object detection is the

³<https://www.kaggle.com/competitions/kuzushiji-recognition>

task of predicting, for every object of interest in an image, a bounding box enclosing it together with a class label (check subsection 3.3 for further details). While such competitions demonstrate that general-purpose object detection DL networks perform well with this material they should not be considered as a systematic comparison of methods due to their focus on a specific dataset. On the commercial side, systems for kuzushiji recognition also exist. The most prominent commercial solution is TOPPAN’s *Fuminoha*. Development of this tool began in 2015 and it was initially offered as a transcription viewer with a collaborative reading environment and AI-OCR assistance. While reported accuracies are above 90%, technical details have never been published, hindering the assessment and comparison of the tool with other research approaches⁴.

Independently of these developments, the National Diet Library developed an OCR system for historical Japanese material [17]. The system is composed of three parts: a layout recognition model based on the above-mentioned cascade Mask R-CNN [16], a character-string recognition model derived from TrOCR [18], and a reading-order sorting model. Its training data includes transcriptions produced by Minna de Honkoku⁵, a crowdsourced platform for reading historical Japanese documents that now benefits from the developed OCR system. The division of labour is dictated by the parts of the system: TrOCR operates on cropped text-line images and performs no localization of its own, therefore a detection network is needed before it and an ordering step after. The system has been applied to around 80,000 public-domain items amounting to 4.3 million images with a reported median F1 score of approximately 0.92 over 3,028 evaluation images. This evaluation was performed on a transcription problem, comparing the characters in the output against those in a reference transcription without any regard to their location in the page.

Regarding our particular problem, Morris and Karaisl [10] tested KuroNet on printed kanbun, the register used in a large portion of wasan texts, and compared it with the NDL Kotenseki OCR system Lite Web version. Working from manually counted transcription errors rather than from

localization, they reported an overall character error rate of 9.77% for KuroNet across the texts they tested (which included several wasan publications), with four of twelve texts above 20%. As Sangiacomo and colleagues reported [19], a corpus of transcriptions with 85–90% word accuracy can be used in text-mining projects, and a word accuracy rate of 90% is “more than sufficient for performative semantic analysis”. Output of this quality is therefore already of analytical value, even where it would not satisfy an editor. More interesting for our purposes is their comparison: the NDL system achieved consistently lower character error rates, yet its reliance on line segmentation produced reading-order errors, particularly on pages containing *warigaki* (double columns inserted within a single line), alongside other issues, to the point that the more accurate transcription was the more laborious one to edit. Since *warigaki* is common in mathematical texts, character-based detection appears more suited to the layout in wasan documents.

Work on kanbun is also advancing on the language side. Wang and colleagues [6] constructed the first Classical-Chinese-to-Kanbun parallel dataset and demonstrated that fine-tuned language models can perform character reordering and translation at levels exceeding both a rule-based baseline and non-expert human annotators, while explicitly placing OCR of ancient books outside their scope. Their scoping decision is telling for our purposes: reliable detection remains the prerequisite for any subsequent linguistic processing.

In terms of our previous work, we previously focused on auxiliary OCR processing tools such as angle correction and noise reduction [20, 21]. Furthermore, we also addressed character classification from cropped images, obtaining 99.75% classification accuracy on a balanced set of 2,136 modern kanji classes and 90.4% on a balanced subset of 1,636 classes of the kuzushiji classical kanji dataset [21]. Two distinct problems arise in this setting of unequal difficulty. Character *classification* focuses on which character appears in an image patch. DL networks have demonstrated in various fields that they are outstanding for this task, and the real difficulty lies with the assembly and balance of a training set: the Kuzushiji data is highly imbalanced, with a long tail of rare characters, and reported accuracies vary enormously

⁴<https://www.toppan.com/ja/joho/fuminoha/>

⁵ndl-lab/ndl-minhon-ocrdataset

depending on how many classes are retained and how the remainder are re-balanced. Instead, character *detection* focuses on the location of the characters on the page, and in wasan documents this is the harder problem. Characters vary greatly in size, pages can present dirt and insect damage, ink has a wide range of intensities, ruled lines and equations compete with the text as possible character traces, and the variability of the brushwork adds to the ambiguity of which marks of ink are part of a character.

When compared to documents that consist mostly of text, wasan documents differ in one key aspect for evaluation. Meaningful figures (geometric diagrams that are not just decorative illustrations) are central and almost always present, frequently in poor condition. KuroNet is trained to ignore characters inside illustrations, and Morris and Karaisl accordingly exclude them from their analysis [10]. For our purpose, there is a need for the opposite: characters appearing within the graphical description of a problem carry mathematical content and can support the automatic construction of problem taxonomies. Therefore, it is critical to keep them as part of the annotations and evaluation.

3 Materials and methods

The methodology used in this work relies heavily on DL models, a set of models within the artificial intelligence (AI) umbrella that has been used effectively in many research areas focusing on image analysis. In supervised approaches expert data is used to tune the parameters of the models. We begin by sketching how a problem like ours is posed and solved using this technology. Typically a four-step strategy is followed: Formalization, Annotation, Training and Testing. In the formalization step, we express our practical problem in mathematical terms. In our case, this means finding the bounding boxes that contain each character. Bounding boxes are the smallest rectangles that fully contains a shape and we define them through the x,y coordinates of their corners. Once the problem is formalized, annotations provided by human experts are created. These annotations should represent the correct answer on a set of examples. In our particular case, this is expressed with the positions of each character marked by hand on 75 scans (see Fig. 2 for an example).

With this dataset, supervised training focuses on finding the model parameters that best reproduce the annotations. These models take the raw data as input, and should predict the annotations as output. In our particular case, we use eight pretrained object detection models that are re-trained on the annotated data. The input in that case are images, and the output are the coordinates of the boxes. Finally, to guarantee that the model was well trained, an evaluation step (known as testing) should be performed with a new set of annotated data that was not used for training and that is relevant to the problem being solved. In our case, we used held-out scans, under the criteria of Section 3.4.2. This design makes supervised DL approaches easier to implement than classical CV algorithms requiring encoding expert knowledge in the model or algorithm. By fitting a practical problem into the existing mathematical template and providing annotations, researchers can tackle a large variety of practical problems. What determines their success is having enough annotated data to capture the characteristics of the practical problem. While the annotation effort remains significant, models and the field have evolved and it is now possible to get useful results with less annotated data than earlier generations of AI methods. This is, in large part, why DL has become so ubiquitous in several research fields.

The remainder of this section focuses on the description of the data and its annotations (Section 3.1), the noise removal methods studied (Section 3.2), the classical and DL detectors compared (Section 3.3), and how the results of both tasks were evaluated (Section 3.4).

3.1 Data

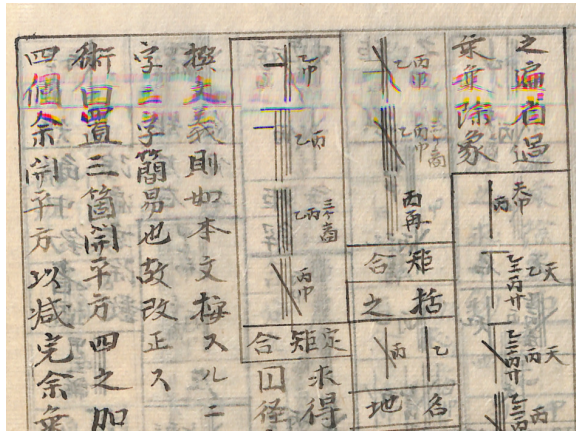
The Sakuma Bunko is a collection of wasan materials assembled by Watanabe Ichi (渡辺一, 1767–1839), a samurai of the Nihonmatsu Domain who transcribed numerous Edo-period texts. In particular, this collection covers the works of Aida Yasuaki (会田安明, 1747–1817), founder of the Saijō-ryū (最上流) school, and Watanabe’s disciples Sakuma Shitsu (佐久間質, 1786–1854) and Sakuma Tsuzuki (佐久間續, 1819–1896), who ran a private school in Miharu and taught wasan to thousands of students in the Tohoku region. The collection was donated by Sakuma Tsuzuki’s



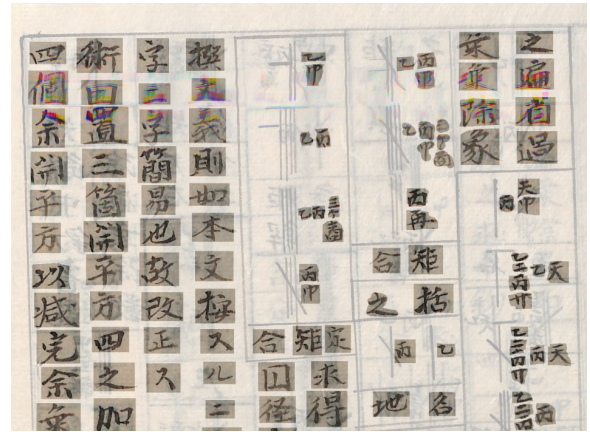
(a) Image resulting from scanning two book pages



(b) Overlay of character bounding boxes



(c) Detail of upper part of left page



(d) Character overlay on detail subimage

Fig. 2: Problem visualization. Our goal is to process digitized documents such as the one illustrated in (a) to obtain the position of all characters, overlaid in (b). To better highlight characteristics of these documents we focus on two closeups of the documents. (c) shows some of the challenges present in the data cause by scanning artefacts and ink from the adjacent page (ink bleedthrough). (d) showcases the complex distribution of characters in the pages as well as their differences in size.

grandson and transferred to Yamagata University (located in Aida's hometown) for the 150th anniversary of Aida's death in 1966. It comprises 1,280 volumes and documents including not only the development of Saijō-ryū wasan but also how wasan was received and taught locally⁶.

The 431 works by Aida that form the core of the collection were made available as PDFs through the university repository in 2008, covering roughly 30% of the holdings at a resolution that did not reproduce the different inks faithfully. The remaining 849 volumes were digitized in a project

that began in July 2024 and finished in July 2026, producing 29,496 scans. The volumes were digitized at Yamagata University Library using a PFU ScanSnap SV600 overhead scanner, at a scanning density of 600 dots per inch. The resulting images are $9,922 \times 7,012$ pixels, stored as baseline JPEG conforming to the JFIF 1.00 standard with 8-bit precision across three colour components with file weights in the range of 60 and 65 MB. At this resolution minute characters, wasan-specific symbols and red-ink annotations can be examined as if working from the physical volumes. A selection

⁶<https://klibs1.kj.yamagata-u.ac.jp/drupal/ja/node/168>

of 164 volumes is currently served through an IIIF endpoint as a prototype database⁷.

Wasan books are read vertically, from top to bottom and from right to left. The books were scanned so each image contains a double-page spread containing the verso of one leaf and the recto of the next leaf. This corresponds to what modern readers would consider the right-hand and left-hand pages (Figure 2, (a)). For simplicity we will continue to use the general purpose notation. The first image of a scanned volume contains its front cover and the last its back cover.

Several artifacts due to scanning and conservation are present in the resulting images. Such artifacts not only make it harder to read them, but also to process automatically. Due to the books' age and their frequent use, their state of conservation is sub-optimal: dirt, staining, insect damage and paper imperfections can be observed as small noisy regions. Furthermore, the paper thinness makes it partially transparent and ink from the reverse of a sheet bleeds through making it visible. This is observed as faint ghost characters that compete with the text. This issue of *bleedthrough* can be prevented at scanning time by placing a sheet of paper in the pocket between the recto and verso of each leaf. However, in our case, the collection is so large that inserting and removing a sheet for every scan would have added considerably to the time required for each volume. The books are also fragile, and much of the digitization was carried out by people who are not specialists in historical document handling, for whom the additional manipulation of each leaf carries a risk to the material. Having established through preliminary tests that bleedthrough and the other sources of noise could be removed afterwards by image processing, we chose to scan faster and clean the images later. The noise reduction described in Section 3.2 is therefore not incidental to this work but the counterpart of a deliberate decision regarding digitization.

Data annotation

Three image processing specialists, with experiences ranging from 4 to over 25 years, annotated the positions of the characters in 75 double-page scans drawn from 8 books, yielding 19,937 annotated characters. The set was divided in a 90/10 split between a training set and a test set.

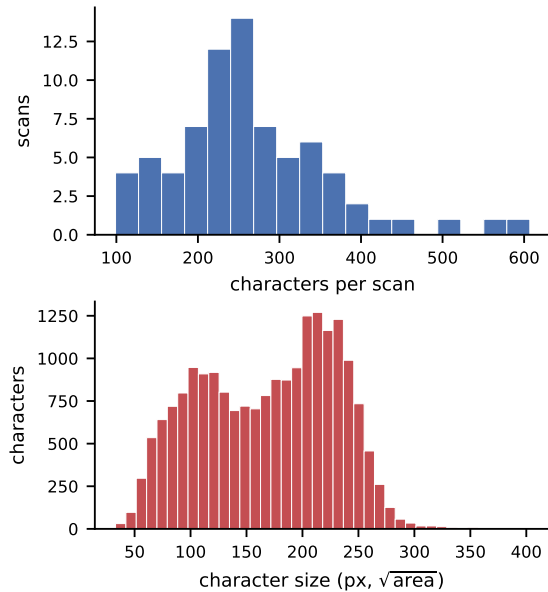


Fig. 3: Distribution of the characters in the annotated set. Top: number of characters per double-page scan. Bottom: character size, given as the square root of the annotated region area. The peaks of the bimodal distribution correspond to the main text and to the smaller characters found in equations and annotations.

Figure 3 summarizes the character distribution of the annotated set. Scans contain 248 characters on average, but the specific count per scan ranges from 99 to 607: some pages contain a dense amount of text while others are little more than a diagram. Character size also varies widely: annotated regions range from 71 pixels (5th percentile) to 255 (95th percentile). Character size distribution is bimodal, reflecting the difference between the main text and the smaller characters used for equations and pronunciation annotations. As Figure 2 shows, all of these characters were annotated as independent characters. In this respect, our problem is more demanding than several of the approaches reviewed in Section 2. Annotation comprises the principal cost of a project of this kind, and in our case, the initial annotations of the 75 double-page scans (made completely manually) took two hours for a double-page scan but, as we will detail in Section 4, this time can be greatly reduced by starting from results provided by the automatic methods and correcting them.

⁷<https://wasan-database.yum-archives.net/manuscripts>

3.2 Noise removal

The characteristics described above, along with the beige appearance of the aged paper, hamper reading and automatic processing. Before applying any threshold to binarize the images, each scanned page was converted to grayscale to obtain a single intensity value per pixel. To ease binarization, we considered noise reduction techniques to enhance the appearance of foreground characters against a uniform background. We first considered the classical noise removal approach used in our previous work [21] based on masking and morphological operators. However, better results were obtained by only applying a threshold, a much simpler and faster approach. Since ink and paper differ in intensity, a threshold assigns each pixel to foreground (ink) or background (paper). We evaluated several global and local methods, including Li’s, Yen’s, Niblack’s, and Wolf-Jolion’s approaches. Among the methods evaluated, Otsu’s method [22] and Sauvola’s method [23] provided the best results for global and local thresholding, respectively.

Global methods compute a single intensity threshold for each image. Specifically, Otsu’s method selects the threshold minimizing the intra-class variance of the two resulting groups, or, equivalently, maximizing the variance between them. The method is fast and requires no user-defined parameters, but it rests on the assumption that one value separates ink from paper across the entire page. That assumption is problematic with our material, where paper tone varies with staining and water damage, and where bleedthrough occupies intensities between those of the ink and the clean paper. Otsu’s method was applied independently to each scanned page, using a single global threshold for the entire image estimated from its complete intensity distribution.

Local methods compute a different threshold for each pixel from the intensity statistics of a window centered on it, which allows the threshold to follow variations in paper tone and illumination across a single page. We present results of Sauvola’s method, for which the threshold at a pixel is computed from the local mean and

standard deviation within a neighborhood. Specifically, the threshold is computed as

$$T(x, y) = m(x, y) \left[1 + k \left(\frac{s(x, y)}{R} - 1 \right) \right], \quad (1)$$

where $m(x, y)$ and $s(x, y)$ denote the local mean and standard deviation, respectively, within the neighborhood centered at (x, y) , k controls the influence of the local standard deviation, and R is the dynamic range of the standard deviation. In our experiments, Sauvola’s method was applied independently to each scanned page using a 271×271 -pixel neighborhood and $k = 0.3$, which were selected empirically by comparing the quality of the resulting binarization. The implementation used $R = 128$. Where the local contrast is low, as in a region containing only stained paper or faint bleedthrough, the threshold adapts to the local background intensity, reducing the likelihood that these variations are classified as foreground. Where a character is present, the higher local contrast allows the character strokes to be retained. Thus, Otsu’s and Sauvola’s methods provide complementary approaches to the non-uniform backgrounds present in the scanned pages, using global and local thresholding, respectively.

3.3 Object detection

In order to perform automatic document processing it is necessary to identify the main constituent parts of those documents. In particular, character detection is a necessary step towards tasks like layout determination, linguistic processing or location of salient characters. Rather than opting for a heavily tailored strategy to our problem and data, we used general-purpose DL object detection networks to test the level of readiness of this technology for processing our data. We will return to this point in Section 4.

As mentioned in Section 2, object detection is the task of predicting bounding boxes containing objects of interest in an image. DL detectors are conventionally grouped into families according to the strategy they use to predict bounding boxes. Two-stage detectors work in two passes: the first proposes regions of the image that are likely to contain an object, and the second decides what object is contained in those regions and adapts its box to fit it. Faster R-CNN [24] was a pioneer for

these types of approaches, and Mask R-CNN [25] extended it by also predicting which pixels inside each box belong to the object. These models usually rely on what is called a “backbone” classification network to extract meaningful features (with ResNet being the most common option). One-stage detectors, as the name implies, perform everything with a single pass, predicting boxes and their contents directly. This improvement came with an increase in speed at the cost of accuracy. From this family of models, YOLO [26] and SSD [27] are the most commonly used approaches, while RetinaNet [28] and FCOS [29] are later refinements of the same idea. Recently, due to the emergence of transformer models, new versions of each family not using convolutional neural networks have been presented, with DETR [30] being the main representative. All of these architectures were developed for natural images, in which colour is important and objects often large and comparatively few. However, historical Japanese documents are usually modelled as binary images and contain several hundred small objects of highly variable size on each page. While these architectures have been applied successfully across a wide variety of domains, whether they transfer to material like ours is still an open question. Rather than committing to a single architecture, we evaluated seven spanning the three families, listed in Table 1.

In order to provide a baseline for comparison and to put DL detectors in the context of similar work, we also provide results for classical CV detectors. Within those, blob detectors locate regions that differ from their surroundings in intensity. For instance, the Determinant of Hessian (DoH) detector [31] examines the matrix of second-order derivatives of the image at a range of scales and responds where that matrix indicates a blob-like structure, returning the position and scale at which the response is extremal. This was the best performing detector in our previous work with wasan data [21]. Another CV detector is maximally stable extremal regions (MSER) [32] by Matas and colleagues. MSER considers images binarized at every intensity level, and those connected components whose area changes the least with threshold variations are retained. Because it depends on relative rather than absolute intensity, MSER is in principle robust to the kind of intensity variation these pages exhibit.

Regarding the implementation of DL object detector networks, the scanned pages are far larger than the input sizes they can easily handle with limited memory. Consequently, each page was divided into square tiles (also known as patches) using a sliding window with 20% overlap. The goal is to avoid missing characters due to the tiling: a character cut by a tile boundary should be captured whole in the adjacent tile. Ground truth boxes were derived from the annotation masks and expressed relative to each tile. Tiles of 1024 and 2048 pixels were used, as shown in Table 1. However, YOLO was only run at 1024 pixels and the remaining architectures at 2048, and the consequences of that difference are discussed in Section 4.2.

Tiling was applied for the training/validation and test sets. All models were initialized from weights pretrained on natural images and then retrained in full on the annotated data; no layers were frozen and no architectural modification was made for this material. In terms of hyperparameters for the optimisation, the models were trained using default configurations: the torchvision models were trained with SGD (momentum 0.9, weight decay 0.0005) and a step learning rate schedule, for up to 200 epochs with early stopping after 10 epochs without improvement, while DETR was trained with AdamW. Learning rates, and the score and non-maximum suppression thresholds applied at prediction time, were selected by grid search for each architecture. The results reported in Section 4.2 correspond to the best configuration found for each architecture. For the sake of brevity we do not reproduce the full grids here, but a figure summarizing the spread of performance within each of them is provided as supplementary material ⁸.

3.4 Evaluation of the results

The two tasks addressed in this paper call for two different types of evaluation. Character detection has a ground truth that can be annotated as sets of bounding boxes. As a consequence, the results of the different methods can be quantitatively compared with the ground truth. In the case of noise removal there is no correct binarization of a page

⁸<https://github.com/nicill/wasanDBResources/tree/main/supplementary%20material>

Table 1: The object detection architectures evaluated in this work. Mask R-CNN was tested with two different backbones. All models were initialized from weights pretrained on natural images and then retrained in full on the annotated wasan data.

Method	Family	Backbone	Tile size	Implementation
DETR	Transformer	ResNet-50	2048	HuggingFace Transformers
YOLOv8n	One-stage	CSPDarknet	1024	Ultralytics
FCOS	One-stage	ResNet-50 FPN	2048	torchvision
RetinaNet	One-stage	ResNet-50 FPN v2	2048	torchvision
SSD300	One-stage	VGG-16	2048	torchvision
Faster R-CNN	Two-stage	MobileNetV3-L FPN	2048	torchvision
Mask R-CNN	Two-stage	ResNet-50 FPN v2	2048	torchvision
		ConvNeXt-B FPN	2048	torchvision

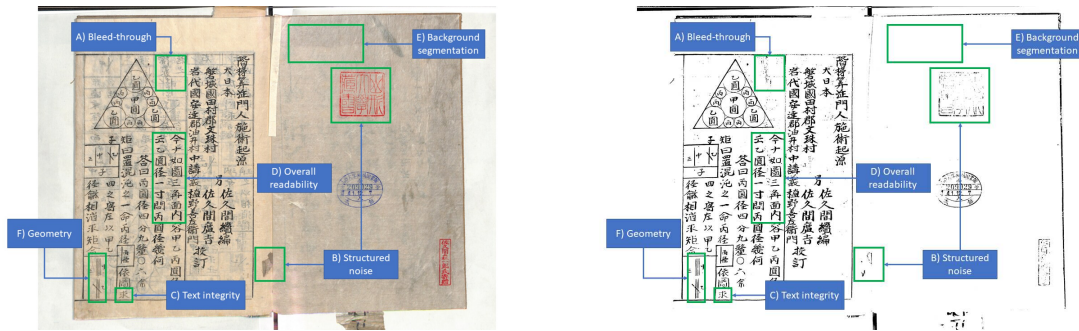


Fig. 4: An image from our database before (left) and after (right) Sauvola noise reduction. The subjective criteria used to evaluate the quality of noise reduction algorithms are exemplified by the blue text boxes

to compare against, so it has to be assessed by human observers against explicit criteria.

3.4.1 Noise removal evaluation

While human experts can assess whether one binarization is preferable to another, it is not possible to define pixel level ground-truth regarding noise reduction. Therefore, we evaluated this criterion subjectively. Three observers independently scored the output of both methods on ten pages selected to span the range of degradation present in the collection, giving 60 assessments in total. Figure 4 shows a page before and after processing, with the criteria marked on it.

Four criteria were used: the integrity of the text (A), the removal of ink bleedthrough (B), the preservation of the lines that make up equations and figures (C), and the removal of structured noise such as stamps and library marks (D). Each

is scored from 0 to 100 along two axes whose product gives the score for that criterion. The first axis is the severity of the worst case present on the page. For text integrity this runs from no problem (100) through fragmented strokes (75) and whole strokes lost (50) to the character disappearing altogether (25); for the other three criteria it measures how much of the unwanted content survives, from nothing (100) to clearly recognizable characters (0). The second axis captures how much that worst case matters. For criteria A and C it is the frequency of the problem across the page, from 1.0 for an isolated case to 0.1 for a generalized one. For criteria B and D it is instead the proximity of the residual noise to the text, from 1.0 when none is present to 0.25 when it sits within one character width of a character, on the grounds that noise far from any text does not impede reading.

The criteria are combined into a single score as

$$S = 0.40 A + 0.30 B + 0.15 C + 0.15 D \quad (2)$$

weighting damage to the text above the rest, since a page whose characters have been thinned or broken is of no use however clean its background. Criterion C does not apply to pages containing neither equations nor figures; in that case its weight is redistributed over the other three, giving $S = 0.45 A + 0.35 B + 0.20 D$. Two of the ten pages fell into this category.

3.4.2 Character detection evaluation

Detection performance is measured by comparing the boxes predicted by an algorithm against the ground truth (GT) boxes produced during annotation. A prediction is considered a *true positive* (TP) when it overlaps a GT box by at least 50% in terms of the intersection over union (IoU) metric, and a *false positive* (FP) when it does not reach that threshold. A GT box with no sufficiently overlapping prediction is considered a *false negative* (FN), (a missed character). The 50% threshold is common practice in object detection research.

However, other definitions are possible. A common alternative considers one box matching another when its centre falls inside it. In that case, a TP is defined as a prediction whose centre lies within a GT box, and so on. This is a coarser criterion and less widely used, but it is the one adopted by several of the systems reviewed in Section 2, and we report it occasionally to refine the description of the performance of some algorithms. From these counts we compute:

Precision, the proportion of correct detections among all detections:

$$precision = \frac{TP}{TP + FP} \quad (3)$$

Recall, the proportion of detected objects among all real objects:

$$recall = \frac{TP}{TP + FN} \quad (4)$$

F1-score, the harmonic mean of the two:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (5)$$

Intuitively, precision answers how many of the characters predicted by an algorithm were correct, recall how many of the real characters were found, and F1 summarizes how well an algorithm finds characters while avoiding mistakes. The two are not equally important here. Because the results are used to build a database in which detections are corrected by hand before publication, we consider recall the more important of the two: errors that lower precision, such as strokes of a character being detected as separate character, are easier to fix than missed characters.

4 Results and discussion

The experimental setup was aimed at assessing the performance of general-purpose CV and DL algorithms when used to solve practical tasks in wasan document processing. Specifically, section 4.1 covers noise reduction, where the goal is to binarize a page to clearly differentiate characters against a uniform background, to ease the access to the textual information for readers or subsequent automatic processing. Section 4.2 covers character detection, where the goal is to precisely determine the position of each character. Character detection is a necessary step towards transliteration as well as layout and content analysis via detection of salient characters.

The data and the code used in this study are publicly available for download in a GitHub repository: <https://github.com/nicill/wasanDBResources>. The repository contains links to the annotated data as well as all the code used in the experiments. As the code is rather complex, the repository also contains links to tutorials explaining the working principles and details of the code. These tutorials are based on Google Colaboratory notebooks as well as YouTube videos and are free to use. These contents are part of ongoing comprehensive data science courses aimed at students and researchers whose main field is not programming or data science but are interested in acquiring these skills. All experiments were run on a Linux computer with an NVIDIA GeForce RTX 4090 24 GB GPU and an Intel Core i9-14900K processor, using PyTorch 2.6.0 and torchvision 0.21.0 with CUDA 12.4, Ultralytics 8.3.3 for YOLO and HuggingFace Transformers 4.57.1 for DETR.

4.1 Experiment 1, Noise reduction

In addition to reporting the numerical results in our experiments, we interpret their performance regarding the practical tasks the algorithms were designed for. Most importantly, and as Figure 4 (right) exemplifies, the two noise reduction methods were generally successful. Both algorithms managed to highlight the characters in the pages, presenting the information in binary form and reducing the presence of the effects of dirt, changes in the paper color and ink bleedthrough. However, in a single case, Otsu’s automatically selected global threshold obscured a significant portion of the page, making some of the characters impossible to read.

For the rest of the section, we focus on finer aspects of the noise reduction and assess its quality in terms of the extent to which the algorithms erased part of the ink in some of the characters, making them more difficult to read, and how each algorithm dealt with the non-character elements in the pages. Table 2 includes the averages for three observers over 10 pages for the criteria introduced in Section 3.4.1.

Table 2: Noise removal score averages for all observers. Higher is better. Criterion (weight in combined score)

Criterion	Otsu	Sauvola
A. Text integrity (0.40)	43.0	53.9
B. Bleedthrough (0.30)	77.3	72.0
C. Geometry (0.15)	55.5	60.2
D. Structured noise (0.15)	53.8	57.8
Combined score	57.2	61.2

Sauvola obtained a higher score on seven of the ten pages, while Otsu performed better on only two (with a tie on the last one). While both methods dealt with bleedthrough satisfactorily, Otsu obtained the best score for that criterion. However, Sauvola obtained better results for all the remaining criteria: text integrity, geometry, and structured noise, with the largest difference occurring in text integrity (53.9 vs. 43.0). This difference is particularly relevant because text integrity carried the largest weight in the combined score (0.40). Even though the general meaning of the text in the pages is still understandable, Otsu

damaged the text considerably more, losing whole strokes or entire characters where the paper is dark, which is consistent with the limitations of the approach anticipated in Section 3.2. Both methods dealt with geometry components and structured noise without interfering with character readability, and the diagrams and equations remained understandable after noise reduction.

The margin between the two methods, 4.0 points on a 100-point scale, is small relative to the variation between pages, which have a standard deviation of around 12 points for both methods. More importantly, the observers disagree with one another considerably more than the methods differ: the mean range between the highest and lowest score given to the same page and method is 35.3 points, and one observer scored consistently higher than the other two across the whole set. Two of the three observers rated Sauvola higher on average, while the third rated the two methods as essentially similar. This is an inherent difficulty of subjective evaluation rather than a defect of this protocol, but it means these numbers indicate preference rather than absolute improvement. Overall, these results suggest that Sauvola is the preferable method for noise reduction in our collection, mainly because it better preserves text while maintaining satisfactory removal of bleedthrough and structured noise.

4.2 Experiment 2, Character detection

We evaluated the results of seven Object DL networks used to detect characters in our documents. For the sake of completeness, we also included the performance of classical CV detectors. Best precision/recall values obtained were 0.59/0.24 for MSER and 0.44/0.10 for DoH. Digging a little deeper, we also studied the centroid-based precision and recall values, and here the same detections gave F1-scores of 0.78 for DoH and 0.73 for MSER. Both approaches are implemented and compared in their corresponding notebook of the online course⁹. The noticeable gap between the values for the two criteria highlights the differences in what they actually measure. While

⁹<https://colab.research.google.com/drive/1NpXRJdOMfVxKzpYsDgW-edHFSBlfZ0n?usp=sharing>

the centroid figures suggest a successful performance, there are matched boxes that no human observer would accept: a box several times the size of the character it should enclose, or one mostly displaced to the location of its neighboring character. In both cases, matching satisfies the criterion of one center falling inside the other box. However, those boxes have no practical use and would require major fixes. Due to this low performance, we will exclude them from the rest of the comparison. Nonetheless, these algorithms do not require annotations to work properly and they can be useful in simpler document processing tasks.

For the rest of the section we focus on DL detectors, presenting numerical data to provide a high-level overview of the characters that were detected/missed as well as qualitative results to exemplify their performance and illustrate their limitations:

DL detectors, Quantitative evaluation:

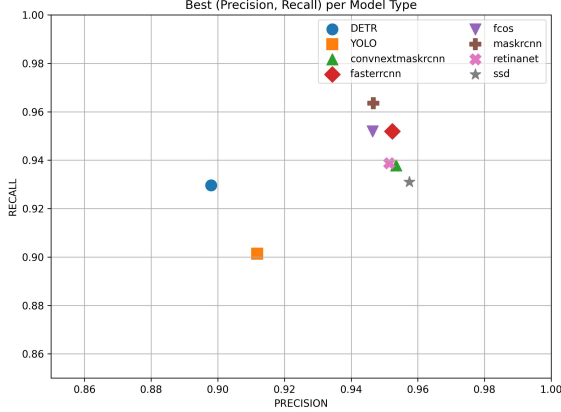
Taking into account the intuitive definitions of precision (percentage of times a predicted box truly contains a character) and recall (percentage of existing characters detected), we consider recall to be more important for the problem at hand. This is motivated by our use of these detectors as a semi-automatic tool to annotate new pages. Wrong predictions tend to correspond to one single character being broken down into more than one detection or, in a small number of cases, dirt regions or parts of diagrams being falsely predicted as characters. Both these errors are easy to spot and fix with our annotation protocol whereas annotating missed characters (which happens more with lower recall) is slightly more time consuming. For the precise definitions of precision and recall used, see Section 3.4.2. Figure 5 shows the results for the DL Object Detector methods studied.

All the architectures perform well, with high F1 scores between 0.907 and 0.955. Mask R-CNN gives the best overall balance and, for our purposes, the best recall at 0.964. These results are not only high but they also led a practical impact on the annotation process of our database. Without any support from these models, annotating a scan from scratch initially took around 120 minutes: the annotator defined a black region over every character, and those regions were then shrunk automatically until they reached the boundary black pixels belonging to each character.

Once the classical CV detectors provided a first automatic annotation, we corrected those annotations by shrinking boxes automatically, adding regions for missed characters, separating wrongly joined characters, and deleting regions that did not correspond to characters. This second pipeline brought the time down to approximately 30 minutes per scan. Finally, using the DL detectors reported here to generate a first annotation, correcting a scan took only 3 minutes. As it stands, the semi automatic annotations created by DL networks can be provided to users with a warning about possible minimal mistakes. Creating human-corrected annotations for the full database in a timely manner is now possible and it is likely that any version of the networks trained with more data or algorithms slightly tailored to the data could result in even faster annotation time.

Detection is also unlikely to be the limiting factor in a full transcription pipeline. Combining the 0.964 recall obtained here with our previous experience in classification suggests that the 85–90% accuracy range that Sangiacomo and colleagues report as usable in text-mining projects is well within reach. In particular, the 90.4% classification accuracy we previously reported for kanji [21] would result in roughly 87% of the characters on a page both found and correctly read. With the mixed modern-and-classical training set of that same work, the final accuracy would rise to approximately 96%. A pipeline built from these components would therefore sit at or above that threshold, which is a marked change from our earlier work, where blob-based detection recovered 79.60% of the characters [21] and detection rather than classification was the bottleneck.

Regarding the method comparison, there is a key aspect that highlights some of the larger differences. The YOLO results were obtained with 1024-pixel tiles while the other architectures used 2048, a choice forced by memory constraints. When all architectures were run at 1024 pixels, YOLO was the best of them and the others performed worse than the number presented. Mask R-CNN, for instance, reached 0.88 F1 rather than 0.955. The likely explanation is that these networks were originally trained to find objects in natural images, where each object occupies a larger fraction of the image than a kanji does in a densely written page, and that the larger tiles bring the characters closer to the sizes the



Type	Method	Prec.	Rec.	F1
T	DETR	0.898	0.930	0.914
1S	YOLO	0.912	0.901	0.907
	FCOS	0.946	0.952	0.949
	RetinaNet	0.951	0.939	0.945
	SSD	0.957	0.931	0.944
2S	Faster R-CNN	0.952	0.952	0.952
	Mask R-CNN	0.947	0.964	0.955
	ConvNeXt M-RCNN	0.954	0.938	0.946

Fig. 5: Best detection performance across architectures. Precision/recall plot (left). Detailed Precision/Recall values (right). T: transformer-based, 1S: one-stage detectors, and 2S; two-stage detectors.

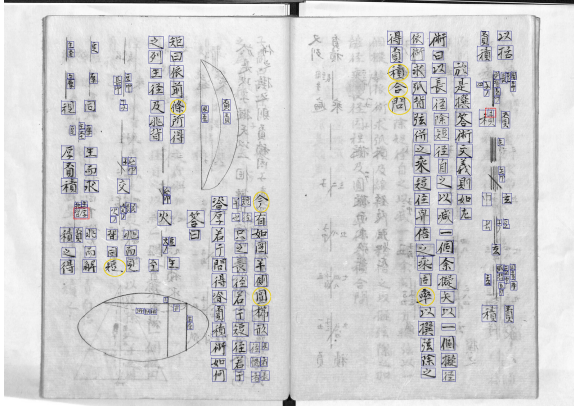
networks expect. The same architecture therefore moved from roughly 0.90 to roughly 0.95 F1 through a better-informed choice of a single parameter, with no changes in the model. What determines the outcome is not just the architecture chosen but the combination of good annotated data with an understanding of both the tools and the material, as argued in Section 2. At the same time, the fact that around 0.90 was achieved without that understanding shows the level of performance of the technology even when it is not used optimally.

These results, however, do not imply that out-of-the-box object detection is the best possible approach to character detection in wasan documents. It is likely that a method designed around the specific characteristics of these documents would show an even better performance. But at values close to 0.95 for precision and recall, what they show is that it is a sufficient approach to solve the problem. Every architecture in Figure 5 was taken as published, initialized from weights trained on natural images, and retrained on our data with minor code adaptations. None was designed for binarized pages, for objects this small, or for several hundred of them at a time. That they still reach this level of performance means the barrier to attempting new document processing tasks is no longer the design of a dedicated method but understanding modern computer vision processing technologies and being able to produce expert annotations that express the problem. For example, detecting the

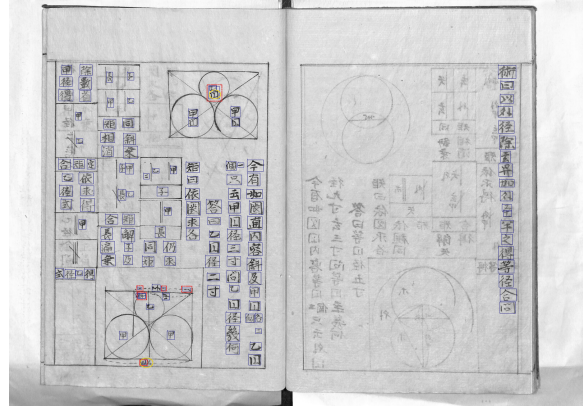
equations, the graphical descriptions of problems, or a particular salient character would only require a new annotated sets for each case. We believe this represents an opportunity for dedicated document processing tasks to be developed efficiently even by small multidisciplinary teams and with limited DL knowledge.

DL detectors, Qualitative evaluation:

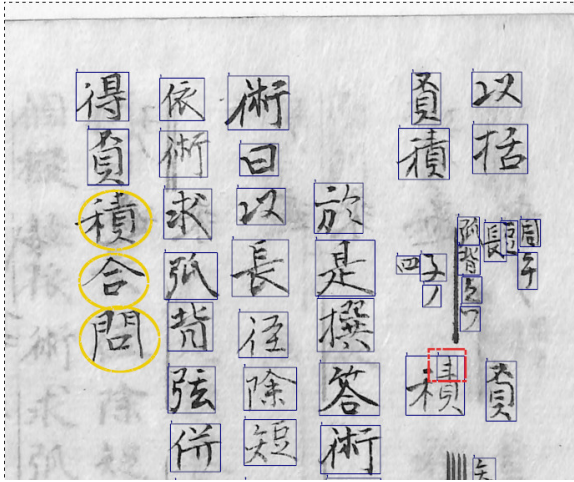
Overall, all methods were better at identifying large characters, especially those lying in regular rows. On the one hand, these characters often appear as regular regions of intersecting strokes and can also be easier to identify by human observers. On the other hand, these larger and relatively isolated characters represent the vast majority of examples in terms of the area they occupy. In this respect, any DL network trained with a database that presents similar characteristics would be biased towards detecting them. Characters within figures or pronunciation annotations had usually a smaller size, were located very close to other characters and can appear rotated. Properly delimiting these characters is often difficult even for human annotators. These characters are sometimes difficult to read and in some cases the exact same stroke may be a part of a larger character or a character of its own depending on the context and position. All of these issues explain the dominant role in detection errors of smaller characters. Nevertheless, the detection networks studied managed to produce an excellent overall output as illustrated in Figure 6. The two example pages are shown in



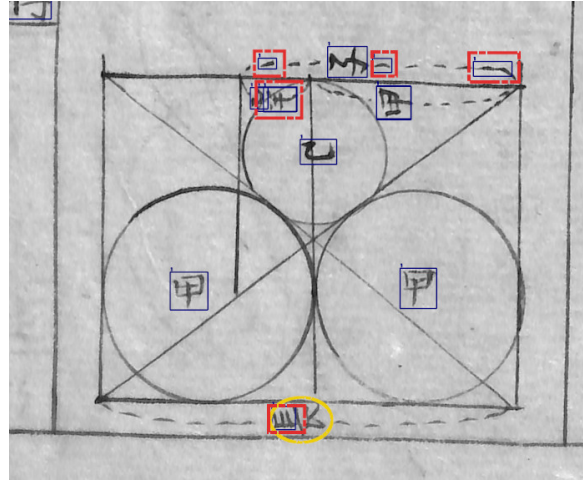
(a)



(b)



(c)



(d)

Fig. 6: Examples of the results obtained by the best performing method (Mask R-CNN). Blue boxes indicate the predictions of the model. Red boxes represent false positive detections. Yellow ellipses indicate missed detections (false negatives). (a,b) show two full pages and show (c, d) detailed views of smaller sections of the pages above. False negatives can occur even when a part of the character is detected (lower part of d). False positives can happen when a small part of a correctly detected character is detected a second time (c, lower part) or when parts of the figure that do not contain characters (dirt, parts of the graphical description of a problem) are mistakenly identified as characters (d, upper part of the diagram).

grayscale to help visualize the prediction boxes and highlighted mistakes.

The matching criterion used is strict. As an example, the two overlapping errors in the lower part of subfigure (d) show how even in cases where a character is partially found, unless the overlap between ground truth and automatically detected boxes is large enough, the character is not considered correctly detected. The figure taken as a whole shows the high level of performance

achieved by the studied methods. Even though bleedthrough is clearly present in the original images, the detectors ignore it while detecting the vast majority of the characters present. The few illustrated errors correspond to characters of atypical sizes and orientations. Furthermore, incorrect predictions usually occur within characters. Either by joining two adjacent characters or by detecting only part of one.

4.3 Comparison with commonly used OCR systems.

To provide further context for the performance of our methods, we run the test images through two popular software tools: the NDL Kotenseki OCR tool [17] and KuroNet [12] available online¹⁰. This comparison is meant to illustrate how other methodologies can be used with similar data and how changes in focus affect the final result. This should not be read as a comparison of method performance with the DL detectors, as the outputs given by the OCR softwares did not provide us with detailed enough information to quantitatively evaluate detection. In particular, both methods are unsuited to process characters inside figures. The NDL OCR expects relatively regular document layout and extracts rows of characters. Additionally, the output of KuroNet was a superimposition of the characters predicted over the original image with a constant font size and did not report any coordinates of boxes or characters. Consequently, incorrect character region detections following our definitions of matching were difficult to ascertain. Figure 7 shows the result of one page with both software tools. For the sake of brevity we omit detailed results here but the interested reader can find detailed tables and image results in the supplementary material repository¹¹.

The NDL OCR was very sensitive to bleedthrough and other types of noise. We tested it both the original images and with a noise reduced version and decided to present denoised results because they were vastly better. As long boxes with sequences of characters were predicted, we considered any character falling inside one of the predicted boxes to have been correctly detected. When any character was found to belong to more than one box it was counted as a false detection for every time it appeared in any box after the first. Characters not covered at least 50% by any box were considered missed and boxes not including any character were counted as one wrong prediction each. Precision values for the 10 pages studied ranged from 0.896 to 0.985 and recall values from 0.785 to 1. Overall, aggregating

all predictions and considering the 2,269 characters in the 10 pages the pooled values were a 0.939 precision, 0.889 recall and a F1 score of 0.913.

We also scored KuroNet against our annotations to provide a quantitative evaluation. Unlike the NDL system, it was not observably affected by bleedthrough and other noise artefacts when run on the original images. Scoring also required a different procedure: KuroNet returns an image in which each recognized character is drawn in place with a fixed font size, so the detection has only a position without its extent (box). Therefore, we could only evaluate its results by whether it fell on the correct character. Detections not corresponding to any character, including characters marked twice, were counted as wrong predictions. Precision ranged from 0.897 to 0.993 and recall from 0.623 to 0.935, with pooled values over the same 2,269 characters of 0.948 precision and 0.822 recall, for an F1 score of 0.880.

Overall, both methods found a large percentage of characters with better precision than recall. Even though they missed some characters, they made relatively few wrong predictions. Both methods also struggled with similar characters and NDL had issues with irregular text layouts, such as double columns. The evaluation is slightly flattering to NDL compared to KuroNet due to the chosen matching criteria. However, the fact that a character is inside a large box does not necessarily mean that it was actually detected as a unique character in a sequence. The main source of missed characters for both methods was characters inside of figures. These characters are often small and sometimes appear rotated. That often means that they do not conform to the linear groups that NDL expects and they would not survive the DBSCAN clustering step in KuroNet because their classification score would be very low due to the rotation. These are limitations acknowledged by the authors of both software tools so these numbers show more the challenging characteristics of our data than any shortcomings of the software tools.

To reiterate, none of this is meant as a direct comparison of detection performance. Even though the best values obtained by a model in our experiments (0.947 precision, 0.964 recall, 0.955 F1) are overall higher than those of both NDL and KuroNet, this fact stems from the focus of our experiments and the training data used. These issues appear in subtle ways that require

¹⁰<https://mp.ex.nii.ac.jp/kuronet/>

¹¹<https://github.com/nicill/wasanDBResources/tree/main/supplementary%20material>

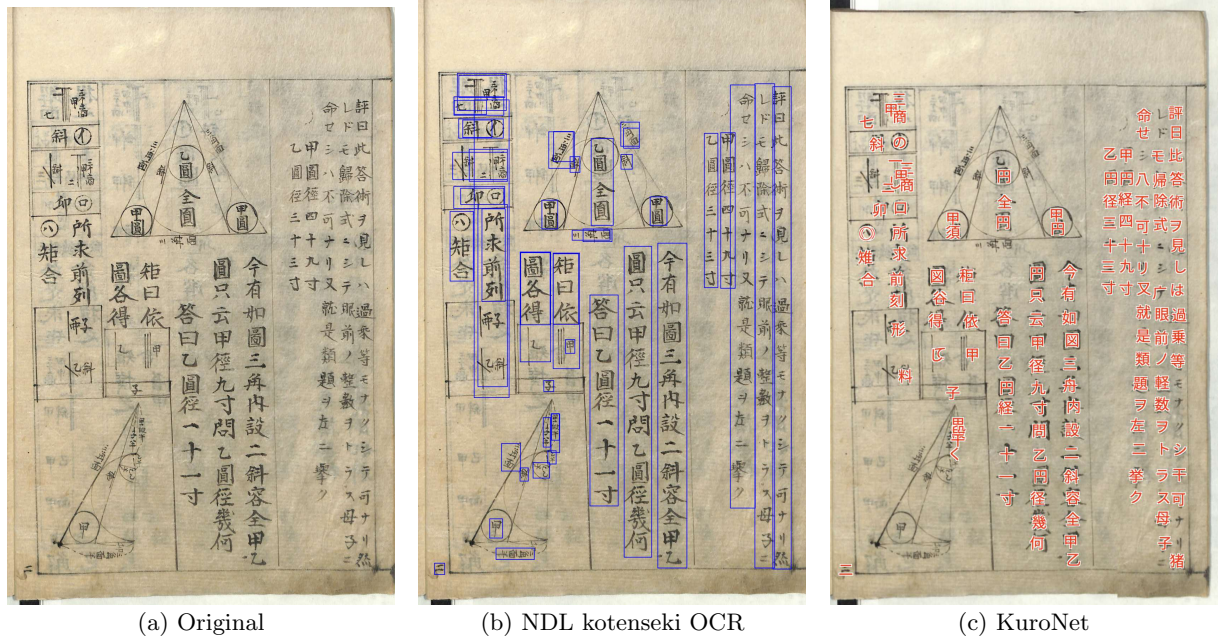


Fig. 7: Output of the two external systems on one example page. NDL segments the page into text lines, while KuroNet identifies individual characters and renders their transcription in place.

deep understanding of the data and the technology used. For example, the results of the NDL Kotenseki OCR were low when the original images were used due to the bleedthrough effect. This is likely explained by the use of data that had no bleedthrough when training the NDL OCR (so a noise reduction step like those presented in Section 4.1 was presumably assumed). The problem is not that DL models cannot handle noise (as shown by our own results and those of KuroNet), but that the system was not designed with that in mind.

Taken together, these results suggest that the appropriateness of a tool depends as much on the collection and its intended use as on its capabilities. The NDL pipeline serves several million images of largely regular, single-column printed material, and its objective is a readable transcription in correct reading order. A staged pipeline with a purpose-built recognition model is a reasonable answer to that problem. KuroNet targets kuzushiji books in which the layout is irregular and text can wrap around or be incorporated within illustrations. Our collection is smaller, handwritten and irregular in layout, and what we need from it is the position and extent of each

character. Under those conditions the requirement is localization rather than transcription, and, as the results above show, general-purpose object detection networks meet it without any component built specifically for the task. The broader claim we make is therefore not that any one architecture is best, but that CV and DL have matured to the point where a problem of this difficulty can be addressed with general-purpose tools: no new architecture was designed for this work, and no classical component required tuning.

5 Conclusion

In this paper we explored the use of general-purpose CV and DL tools for solving practical document processing problems arising from the construction of a digitized database of the Sakuma Bunko collection deposited at Yamagata University. We addressed two key tasks in detail: noise reduction and character detection.

Noise reduction is a necessary step for many image processing algorithms. For example, we observed how it improved the results of the NDL OCR. In our case it also allowed us to reduce the scanning time of our document database. In terms

of numerical results, thresholding alone proved sufficient while also being simpler than the masking and morphological pipeline presented in our previous work. Of the methods tested, Sauvola’s scored 61.2 against Otsu’s 57.2 on our subjective criteria and rated higher on seven of the ten pages evaluated by the 3 human observers. Overall, the two methods present different but comparable behaviors: Sauvola preserves the text considerably better, 53.9 against 43.0 on text integrity, while both methods remove bleedthrough and structured noise well enough that neither interferes with reading the characters or the diagrams.

Regarding character detection, the seven object detection architectures evaluated had high F1 scores with values between 0.907 and 0.955. The best of them, Mask R-CNN, reached 0.947 precision and 0.964 recall. This was achieved on handwritten pages carrying bleedthrough, dirt and insect damage, on which the 19,937 annotated characters ranged from 71 to 255 pixels across and were laid out in diagrams and equations rather than in regular columns. None of these networks was designed for this material, or for any data resembling it. The classical blob detectors we used as a baseline reached only 0.59/0.24 precision and recall for MSER and 0.44/0.10 for DoH, highlighting the gap between classical CV and DL approaches.

For additional context, we also ran two established OCR systems on the same ten test pages, containing 2,269 annotated characters. The NDL OCR system reached an F1 of 0.913 and KuroNet 0.880. As Section 4.3 explains, these figures are not directly comparable with ours, since each system reports its output in a different format and each system was scored under a different criterion. However, these results highlight that both systems, built for adjacent materials and given no data or information from this collection, locate the vast majority of the characters, and that their errors are what was expected: characters inside figures, small and rotated characters, and, for the NDL system, pages with irregular layouts.

We evaluated both tasks with practical application in mind rather than as academic benchmarks, our discussion is focused in on practical implications of the results. The reduction in annotation time reported in Section 4.2, from around two hours per scan to three minutes, makes annotation of the whole collection a realistic goal, and

provides a cheap path for extension to other document processing tasks. The balance between the two stages of a transcription pipeline has also shifted: blob-based detection recovered 79.60% of the characters in our previous work [21], where detection was clearly the limiting step, against the 0.964 recall obtained here. Two things determined our results, and neither was the choice of architecture. The first was the annotated data: 75 double-page scans from 8 books were enough. The second is understanding the tools to obtain optimal results: the same Mask R-CNN model moved from 0.88 to 0.955 F1 through just a small change of tile size alone, from 1024 to 2048 pixels. This decision is independent of the choice of model and invisible to anyone unfamiliar with how these networks were originally trained. This leads to two recommendations for projects of this kind. First, clearly define your goal when analysing these documents because this goal will dictate the annotation process and the annotation will dictate everything else. Second, make sure that the team combines experts that understand the material and experts that understand the tools. Neither alone is sufficient: the domain expert can define and annotate what knowledge is worth finding, while the technological expert turns that information into a solvable problem and can recognize when an off-the-shelf method is enough for the task at hand.

Determining character positions for the collection opens several research directions. Since detection is no longer the limiting step, transcription becomes an ambitious but realistic goal that can lead to searching by content. The same pipeline can be redirected at other targets with no new research and just new annotations: the equations, the graphical descriptions of the problems, and salient sequences such as the opening sequence 今有如圖 for problem statements. Currently, our team is pursuing these directions as next stages of the project.

Finally, we want to remark that large projects of this kind are only possible with multidisciplinary teams. Furthermore, it is key to train rare future hybrid experts that can bridge both sides: researchers with enough understanding of document studies to know what is worth extracting, and enough CV experience to know how to extract it. As mentioned, the tile size result is a

small example of this conclusion. Neither specialist alone would have found it. To help build that profile, the procedures used in this work and the evaluation of their performance are documented in a series of open online courses in CV-based document analysis¹², aimed at researchers whose main field is not programming or DL.

Funding. This work was supported by JSPS KAKENHI Grants-in-Aid for Scientific Research (C), Grant Numbers 21K11899 and 24K15650.

References

- [1] Smith, D.E., Mikami, Y.: A History of Japanese Mathematics. The Open Court Publishing Company, Chicago (1914)
- [2] Ueno, K.: Japanese mathematics from the seventh to the sixteenth century. In: Morgan, D.P., Quan, T. (eds.) Mathematics and Astronomy in the Ancient World, pp. 163–173. Springer, Cham (2025)
- [3] Murata, T.: Indigenous japanese mathematics, wasan. Journal of Japanese Trade and Industry **20**(2), 50–55 (2001)
- [4] Cao, S.: A comparative analysis of japanese mathematics in the Edo period in terms of cultural dimensions with other global mathematical traditions. In: Proc. of the 3rd Inte. Conf. Applied Physics and Mathematical Modeling, pp. 124–129 (2025). <https://doi.org/10.54254/2753-8818/2025.DL26845>
- [5] Ogawa, T.: A review of the history of Japanese mathematics. *Revue d’histoire des mathématiques* **7**, 137–155 (2001)
- [6] Wang, H., Shimizu, H., Kawahara, D.: Kanbun-LM: Reading and translating classical chinese in japanese methods by language models. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 8589–8601 (2023). <https://doi.org/10.18653/v1/2023.findings-acl.545>
- [7] Sturgeon, D.: Large-scale optical character recognition of pre-modern chinese texts. *International Journal of Buddhist Thought & Culture* **28**(2), 11–44 (2018) <https://doi.org/10.16893/IJBTC.2018.12.28.2.11>
- [8] Yang, H., Jin, L., Sun, J.: Recognition of chinese text in historical documents with page-level annotations. In: 2018 16th Int. Conf. on Frontiers in Handwriting Recognition (ICFHR), pp. 199–204 (2018). <https://doi.org/10.1109/ICFHR-2018.2018.00043>
- [9] Horvath, A.: Digital brush talk: Challenges and potential connections in east asian digital research. In: Fiormonte, D., Chaudhuri, S., Ricaurte, P. (eds.) Global Debates in the Digital Humanities, pp. 127–140. University of Minnesota Press, Minneapolis (2022)
- [10] Morris, J.H., Karaisl, A.: Accuracy isn’t all: Testing kuronet for kanbun ocr. *Digital Studies/Le champ numérique* **15**(1) (2025) <https://doi.org/10.16995/dscn.17257>
- [11] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep Learning for Classical Japanese Literature. arXiv:1812.01718. <https://arxiv.org/abs/1812.01718> (2018)
- [12] Lamb, A., Clanuwat, T., Kitamoto, A.: Kuronet: Regularized residual u-nets for end-to-end kuzushiji character recognition. *SN Computer Science* **1**(3), 177 (2020) <https://doi.org/10.1007/s42979-020-00186-z>
- [13] Clanuwat, T., Lamb, A., Kitamoto, A.: End-to-end pre-modern japanese character (kuzushiji) spotting with deep learning. In: Proc. of IPSJ SIG Computers and the Humanities Symposium, pp. 15–20 (2018)
- [14] Clanuwat, T., Lamb, A., Kitamoto, A.: Kuronet: Pre-modern japanese kuzushiji character recognition with deep learning. In: 2019 Int. Conf. on Document Analysis and Recognition (ICDAR), pp. 607–614 (2019). <https://doi.org/10.1109/ICDAR.2019.00103>
- [15] Clanuwat, T., Kitamoto, A.: “miwo” ai kuzushiji recognition application for document examination. In: Proc. of IPSJ Humanities and Computer Symposium (2021)

¹²<https://colab.research.google.com/drive/1mjBugZmwa6OSxNaqkrCu4OCsIKf2Wy7P?usp=sharing>

- [16] Cai, Z., Vasconcelos, N.: Cascade R-CNN: High quality object detection and instance segmentation. *IEEE TPAMI* **43**(5), 1483–1498 (2021) <https://doi.org/10.1109/TPAMI.2019.2956516>
- [17] National Diet Library: Experimental OCR Conversion of Rare Books and Old Materials (since 2022). https://lab.ndl.go.jp/data_set/r4.kotenocr_en/. Access: 2026-09-08 (2024)
- [18] Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: TrOCR: Transformer-based optical character recognition with pre-trained models. In: *Proc. of the Thirty-Seventh AAAI Conf. on Artificial Intelligence*, pp. 13094–13102 (2023). <https://doi.org/10.1609/aaai.v37i11.26538>
- [19] Sangiacomo, A., Hogenbirk, H., Tanasescu, R., Karaisl, A., White, N.: Reading in the mist: High-quality optical character recognition based on freely available early modern digitized books. *Digital Scholarship in the Humanities* **37**(4), 1197–1209 (2022) <https://doi.org/10.1093/llc/fqac014>
- [20] Diez, Y., Suzuki, T., Vila, M., Waki, K.: Computer vision and deep learning tools for the automatic processing of wasan documents. In: De Marsico, M., Baja, G., Fred, A.L.N. (eds.) *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019)*, Prague, Czech Republic, pp. 757–765 (2019). <https://doi.org/10.5220/0007555607570765>
- [21] Diez, Y., Suzuki, T., Vila, M., Waki, K.: Automatic processing of historical japanese mathematics (wasan) documents. *Applied Sciences* **11**(17), 8050 (2021) <https://doi.org/10.3390/app11178050>
- [22] Otsu, N.: A threshold selection method from gray-level histograms. *IEEE Trans. Systems, Man, Cybernetics* **9**(1), 62–66 (1979) <https://doi.org/10.1109/TSMC.1979.4310076>
- [23] Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. *Pattern Recognition* **33**(2), 225–236 (2000) [https://doi.org/10.1016/S0031-3203\(99\)00055-2](https://doi.org/10.1016/S0031-3203(99)00055-2)
- [24] Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*, vol. 28 (2015)
- [25] He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (2017)
- [26] Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788 (2016). <https://doi.org/10.1109/CVPR.2016.91>
- [27] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *Computer Vision – ECCV 2016*, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
- [28] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proc. IEEE International Conference on Computer Vision (ICCV)* (2017)
- [29] Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully convolutional one-stage object detection. In: *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)* (2019)
- [30] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: *ECCV 2020*, pp. 213–229 (2020)
- [31] Lindeberg, T.: Image matching using generalized scale-space interest points. *Journal of Mathematical Imaging and Vision* **52**(1), 3–36 (2015) <https://doi.org/10.1007/s10851-014-0541-0>
- [32] Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing* **22**(10), 761–767 (2004) <https://doi.org/10.1016/j.imavis.2004.02.006>