

Article

Automatic Processing of Historical Japanese Mathematics (Wasan) Documents

Yago Diez ¹, Toya Suzuki ¹, Marius Vila ^{2,*} and Katsushi Waki ¹

¹ Faculty of Science, Yamagata University, Yamagata 990-8560, Japan; yago@sci.kj.yamagata-u.ac.jp (Y.D.); y.university.030@gmail.com (T.S.); waki@st.yamagata-u.ac.jp (K.W.)

² Department of Computer Science and Applied Mathematics, Girona University, 17003 Girona, Spain

* Correspondence: marius.vila@udg.edu

Featured Application: Automatic Method for the detection of kanji characters within historical Japanese “wasan” documents. Blob-detector-based kanji detection with deep learning-based kanji classification.

Abstract: “Wasan” is the collective name given to a set of mathematical texts written in Japan in the Edo period (1603–1867). These documents represent a unique type of mathematics and amalgamate the mathematical knowledge of a time and place where major advances were reached. Due to these facts, Wasan documents are considered to be of great historical and cultural significance. This paper presents a fully automatic algorithmic process to first detect the kanji characters in Wasan documents and subsequently classify them using deep learning networks. We pay special attention to the results concerning one particular kanji character, the “ima” kanji, as it is of special importance for the interpretation of Wasan documents. As our database is made up of manual scans of real historical documents, it presents scanning artifacts in the form of image noise and page misalignment. First, we use two preprocessing steps to ameliorate these artifacts. Then we use three different blob detector algorithms to determine what parts of each image belong to kanji characters. Finally, we use five deep learning networks to classify the detected kanji. All the steps of the pipeline are thoroughly evaluated, and several options are compared for the kanji detection and classification steps. As ancient kanji databases are rare and often include relatively few images, we explore the possibility of using modern kanji databases for kanji classification. Experiments are run on a dataset containing 100 Wasan book pages. We compare the performance of three blob detector algorithms for kanji detection obtaining 79.60% success rate with 7.88% false positive detections. Furthermore, we study the performance of five well-known deep learning networks and obtain 99.75% classification accuracy for modern kanji and 90.4% for classical kanji. Finally, our full pipeline obtains 95% correct detection and classification of the “ima” kanji with 3% False positives.

Keywords: Wasan; document processing; kanji detection; kanji recognition; deep learning



Citation: Diez, Y.; Suzuki, T.; Vila, M.; Waki, K. Automatic Processing of Historical Japanese Mathematics (Wasan) Documents. *Appl. Sci.* **2021**, *11*, 8050. <https://doi.org/10.3390/app11178050>

Academic Editor: Jose Santamaria Lopez

Received: 14 August 2021

Accepted: 26 August 2021

Published: 30 August 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

“Wasan” (和算) is a type of mathematics that was uniquely developed in Japan during the Edo period (1603–1898). The general public (samurai, merchants, farmers) learned a type of mathematics developed for people to enjoy, and Wasan documents have been used by Japanese citizens as a mathematics study tool or as a mental training hobby ever since [1,2]. These mathematics were studied from a completely different perspective from the mathematics learned in the West at the time and, in particular, did not have application goals [3]. The fact of Japan’s close commercial exchanges with western countries in this period resulted in these “Galapagos” mathematics that have made it to our days in a series of “Wasan books”. The most famous Wasan book is “Jinkokuki”, published in 1627 with a popularity [4] that resulted in many pirated versions of it being produced.

Three ranks exist in Wasan, depending on the difficulty of the concepts and calculations involved in them.

- The basic rank deals with basic calculations using the four arithmetic operations. Mathematics in this level correspond to those necessary for daily life.
- The intermediate rank considers mathematics as a hobby. Students of this rank learn from an established textbook (one of the aforementioned “Wasan books”) and can receive diplomas from a Wasan master to show their progress.
- The highest rank corresponds to the level of Wasan masters who were responsible for the writing of Wasan books. These masters studied academic content at a level comparable to Western mathematics of the same period.

By examining Wasan books, we can learn about this form of not only social, but also advanced mathematics developed uniquely in Japan. Geometric problems are common in Wasan books (see, for example [5]). It is said that the people who saw beautiful figures in Wasan books and understood the complicated calculations they conveyed felt a great sense of exhilaration. These feelings of joy produced by learning mathematics exemplify why, even today, people continue to learn Wasan and practice it as a hobby. We will focus in this subset of Wasan documents, including geometric problems.

This paper belongs to a line of research aiming to construct a document database of Wasan Documents. Specifically, we continue the work started in [6,7] to develop computer vision and deep learning (from now on CV and DL) tools for the automatic analysis of Wasan documents. The information produced by these tools will be used to add context information to the aforementioned database so users will be able to browse problems based on their geometric properties. A motivation for this work is to address the problem of the aging of researchers within the Wasan research area. Estimations indicate that, as current scholars retire, the whole research area may disappear within 15 years. Thus, we set out to make the knowledge of traditional Wasan scholars available for young researchers and educators in the form of an online-searchable context-rich database. We believe that such a database will result in new insights into traditional Japanese mathematics.

The documents considered in this study are composed of a diagram accompanied by a textual description using handwritten Japanese characters known as *Kanji* (see Figure 1 for an example). In this paper, we focus on two main aspects of their automatic processing:

- The description of the problem (problem statement) written in Edo period Kanji. Extracting this description would amount to (1) determining the regions of Wasan documents containing kanji, (2) recognizing each individual Kanji. Completing both steps would allow us to present documents along with their textual description and use automatic translation to make the database accessible to non-Japanese users.
- Determining the location of the particular kanji called “*ima*” 今. The description of all problems starts with the formulaic sentence “Now, as shown,...” 今有如. The “*ima*” (今, now) kanji marks the start of the problem description and is typically placed beside or underneath the problem diagram. Consequently, finding the “*ima*” kanji without human intervention is a convenient way to start the automatic processing of these documents.

To the best of our knowledge, this paper along with our previous work are the first attempts to use CV and DL tools to automatically analyze Wasan documents. This research relates to the optical character recognition (OCR) and document analysis research areas. Given the very large size of these two research areas, in this introduction we will only give a brief overview of the papers that are most closely related to the work being presented. Document processing is a field of research aimed at converting analog documents into digital format using algorithms from areas such as CV [8], document analysis and recognition [9], natural language processing [10], and machine learning [11]. Philips et al. [12] survey the major phases of standard algorithms, tools, and datasets in the field of historical document processing and suggests directions for further research.

An important issue in this field is object detection. Techniques to address it include Faster-RCNN [13], Single Shot Detection [14], YOLO [15] and YOLOv3 [16]. Two main types

of object detection algorithms exist: one and two-stage approaches. The one-stage approach aims at segmenting and recognizing objects simultaneously using neural networks [15], whereas the two-stage approach [13] first finds bounding boxes enclosing objects and then performs a recognition step. The techniques used are usually based on color and texture of the images. Historical Japanese documents, which are monochrome or contain little color and texture information, pose a challenging problem for this type of approach.

In our work, we classify handwritten Japanese kanji (see [17] for a summary of the challenges associated with this problem). While kanji recognition is a large area of research, fewer studies exist in DL approaches to the recognition of handwritten kanji characters. Most traditional Japanese and Chinese character recognition methods are based on segmenting the image document into text lines [18], but recently, segmentation-free methods based on deep learning techniques have appeared. Convolutional neural networks have been used to classify modern databases of handwritten Japanese kanji such as the ETL database [19], a large dataset where all the kanji classes have the same number of examples (200). For example, [20] used a CNN based on the VGG network [21] to obtain a 99.53% classification accuracy for 873 kanji classes. It shows how DL networks can be used to effectively discriminate modern kanji characters even if (as stated by the authors of the paper) the relatively low number of kanji classes considered limits practical usability. Other contributions focus on stroke order and highlight the importance of the dataset used (see [22] for an example).

As the kanji appearing in Wasan documents present both modern and classical characteristics, we also reviewed the work done in classical kanji character recognition using DL. All the studies that we encountered were based on the relatively recently published Kuzushiji dataset [23]. This publicly available dataset consists of 6151 pages of image data of 44 classical books held by National Institute of Japanese Literature and published by ROIS-DS Center for Open Data in the Humanities (CODH) [24]. We have focused on the *Kuzushiji-Kanji* dataset provided. This is a large and highly imbalanced dataset of 3832 Kanji characters, containing 140,426 64×64 images of both common and rare characters. Most DL research using this data focuses on initially recognizing individual characters (Kanji, but also characters belonging to the Japanese hiragana and katakana syllabaries) and subsequently classifying them. See [25,26] for possible strategies for character detection, [27] for object-detection based kanji segmentation and [24] for a state of the art on DL algorithms used with this database. This survey includes detailed analysis of the results of two recent character segmentation and recognition contests as well as a succinct overview of the database itself from the point of view of image analysis. Of most direct interest for our current goals is [28], where the classification of historical kanji is tackled. A LeNet [29] network was used to classify a subset of the Kuzushiji dataset that had been stripped of the classes with less than 20 examples and re-balanced using data augmentation. While a correct classification rate of 73.10% was achieved for 1506 kanji classes, this average was heavily biased by the classes with more characters. Even though some of the classes were augmented to include more characters, still the initially over-represented classes appear to have remained so. This is clearly shown by the reported average correct classification rate of 37.22% over all classes. This shows very clearly the unbalanced nature of the Kuzushiji dataset. Although DL networks are clearly also capable of classifying the characters it contains, rebalancing the dataset in a way that fits the final application is mandatory. In [28], the authors define some “difficult to classify” characters and obtain a final classification accuracy over 90% for the remaining classes.

In this work, we set out to use CV and DL techniques along with the existing modern and classical handwritten kanji databases to automatically process Wasan documents. Highlights:

- We have used a larger database of 100 Wasan pages to evaluate the algorithms developed. This includes manual annotations on a large number of documents and expands significantly on our previous work [6,7]. See Section 2.1 for details.

- We provide detailed evaluation of the performance of several blob detector algorithms with two goals: (1) Extracting full kanji texts from Wasan documents, and (2) locating the 今 kanji. Goal 1 is new to the current work and the study of goal 2 has been greatly expanded.
- We present a new, extensive evaluation of the DL part of our study, focusing on what networks works better for our goals. As a major novel contribution of this study we have compared in detail the performances of five well known DL networks.
- Additionally, we also investigate the effect that the kanji databases that we use have in the final results. We explore two strategies: (1) Using only modern kanji databases, which are easier to find and contain more characters, but have the disadvantage of not fully matching the type of characters present in historical Wasan documents, and (2) using classical kanji databases as these characters are closer to those appearing in Wasan documents, but also presenting increased variability and fewer examples. The modern kanji database used has been expanded in the current work and the ancient kanji database is used for the first time.
- Finally, the combined performances of the kanji detection and classification steps are studied in depth for the problem of determining the position of the “ima” kanji. For this application, the influence of the kanji database used to train the DL network has also been studied for the first time.

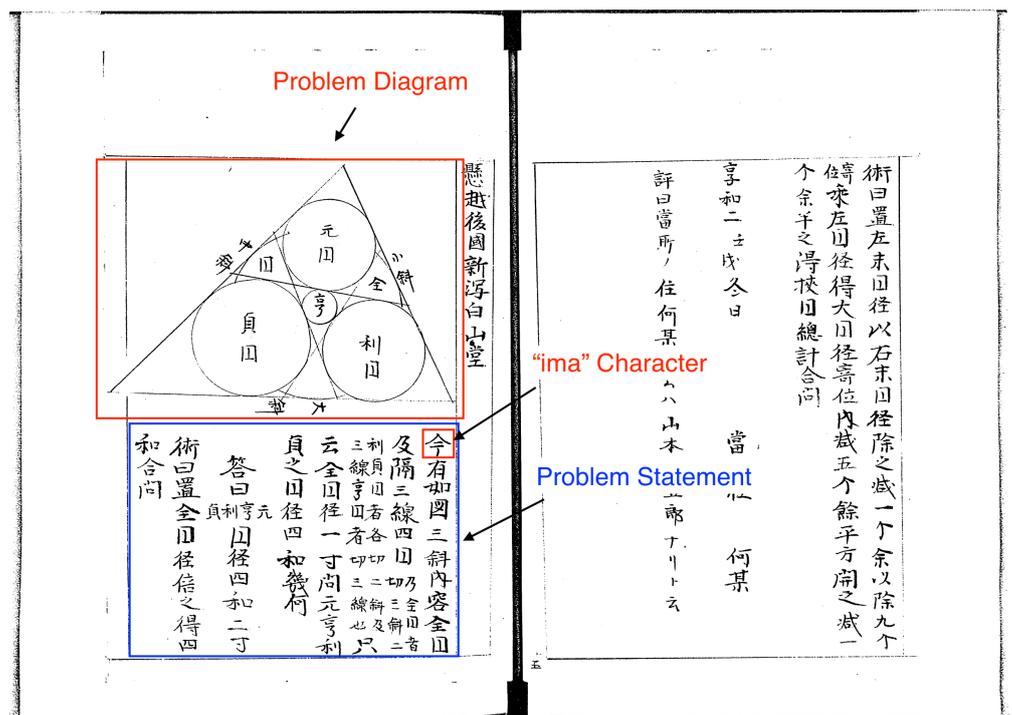


Figure 1. A Wasan page from a book in the Sakuma collection. (今) marks the start of the problem description. Typically this kanji is located either beside or under the geometric problem diagram description.

The rest of the paper is organized as follows. Section 2 contains a description of the algorithms that we have used to process Wasan documents. This includes the pre-processing steps used to build our working dataset as well as the algorithms developed for the automatic processing of Wasan documents. Section 3 presents experiments to evaluate each of the steps of our algorithmic process (separately and when used together). Section 4 discusses the implications of the results obtained. The paper ends with the conclusions and future research directions presented in Section 5.

2. Materials and Methods

In this section, we will describe the data used, provide details and reference of the techniques used and briefly explain how they were adapted to the particularities of Wasan documents.

2.1. Data

2.1.1. Wasan Images

Our Wasan document database was constructed by downloading parts of the public dataset a [30]. This data includes, among others, “The Sakuma collection”. This is a collection of the works of the famous Wasan scholar and mathematicians “Aida Yasuaki” and a disciple of his known to us only as “Sakuma”. We had access to images corresponding to 431 books, with each book containing between 15 and 40 pages with problem descriptions. All documents in the database had been manually scanned for digitization. The fact that the documents were already more than a hundred years old at the time of scanning meant that they presented small imperfections or regions that had accumulated small amounts of dirt that show in the scanned files as small black dots or regions. This particularity of the data made it necessary for us to implement a noise reduction step (Section 2.4). Additionally, the fact that the scanning was done by hand resulted in most of the documents presenting a certain tilt in their orientation. As the vertical and horizontal directions are important in the analysis of these documents (for example, the text is read from top-right to bottom left, with the kanji aligned vertically), it was necessary to correct this orientation tilt (Section 2.3).

We manually chose 100 images containing one instance of the “ima” character marking the start of the description of a problem. Subsequently, we manually annotated the position of all the kanji in 50 images. In order to do this, we manually marked rectangular regions encompassing each of the kanji characters present in each image. This annotation information was stored as binary images, see Figure 2. Finally, we also manually annotated the position of the “ima” kanji in all of the 100 images producing binary layers with only the rectangular regions corresponding to the position of this particular kanji.

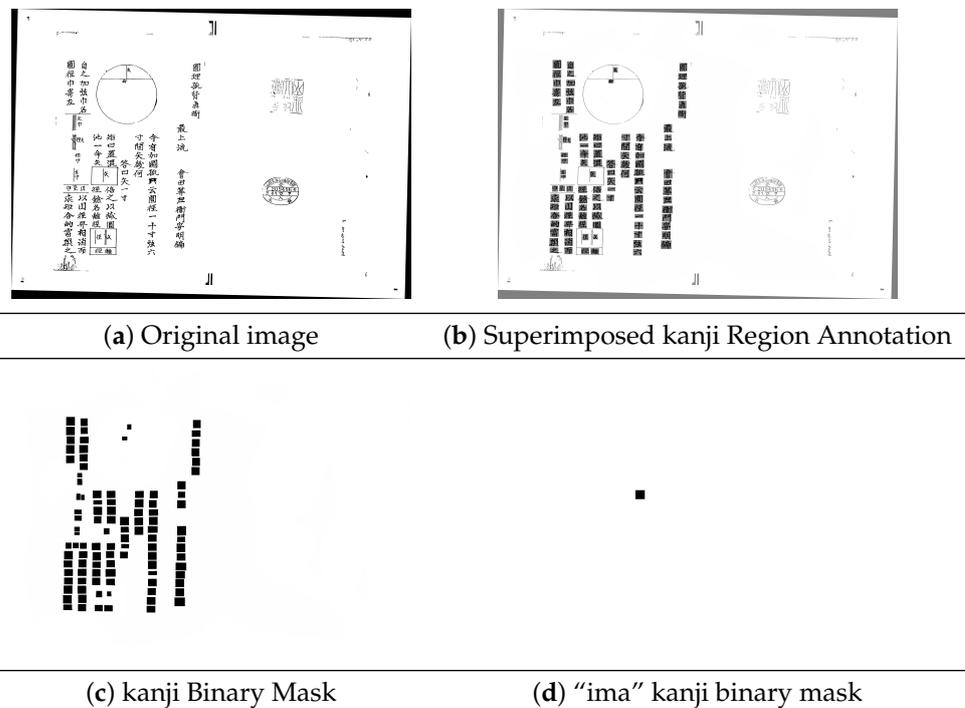


Figure 2. Data Annotation Process.

2.1.2. Kanji Databases

The use of kanji in Wasan documents is unique and itself of historical importance. The script in which most Wasan documents were handwritten, called 漢文 (“kanbun” or “chinese writing”), was initially only used in Japan by highly educated scholars and only became popular during the Edo period. The characters in Wasan documents have many similarities with modern Japanese kanji, but also present important differences. For example, the 弦 symbol, that today is generally thought to mean “string”, is used in Wasan documents to indicate the hypotenuse of a triangle. Conversely, the 円 kanji, used to indicate the technical term for “circle” in Wasan documents, has acquired other meanings since then and indicates, for example, the “yen” Japanese currency. Furthermore, some kanji have also changed their writing since the Edo period. See Figure 3 for some examples. This is exemplified by the “ima” kanji. Not only is this kanji of special importance in the analysis of Wasan documents, but it is also a kanji that has seen the way it is written from the Edo period to modern times. Additionally, given its relative simplicity, it is kanji that can be confused with other similar characters easily.

Unicode	Modern	Wasan Documents
U+4ECA	今	今
U+5408	合	合
U+5F97	得	得
U+6570	数	數
U+7B54	答	答

Figure 3. Some kanji characters have changed the way they are written between the Edo period and modern times. The “ima” 今 kanji character (top) is of particular interest for our study.

Thus, we needed datasets containing the largest possible number of example of handwritten characters adequately classified and matching those appearing in Wasan documents. Given the large number of existing kanji characters (currently the Japanese government recommends that all adults should be able to recognize all the kanji in a list containing 2136 characters for the purpose of basic literacy) it is impractical to produce these databases for a research project such as the current one so it is necessary to rely on publicly available datasets. In this study, we used two of them: the ETL dataset [19] and the Kuzushiji dataset [23]. The first is a large, balanced dataset made up of over 3036 modern kanji, each with exactly 200 examples. The second contains examples of 3832 different kanji from different texts of different historical periods and is heavily unbalanced. See Section 2.6 for details on how these two datasets were used for DL classification.

2.2. Algorithm Overview

In this section, we introduce the techniques used in our image processing pipeline (see Figure 4 for a visual representation), we also mention the goal they achieved and how they relate to each other. The following sections provide further details on each step.

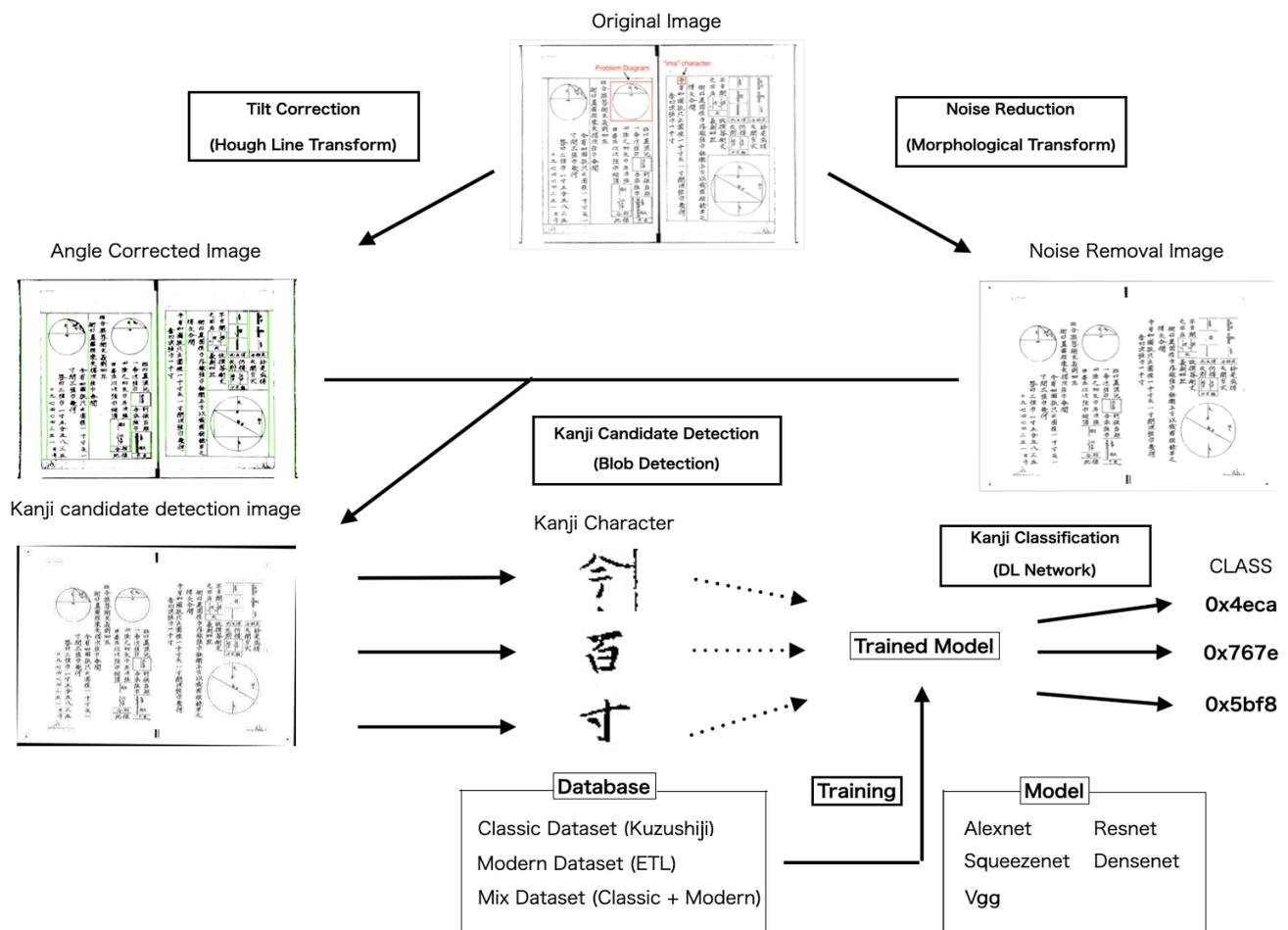


Figure 4. Image Processing pipeline.

- Preprocessing steps to improve image quality:
 - Hough line detector [31] to find (almost) vertical lines in the documents. These lines are then compared to the vertical direction. By rotating the pages so that the lines coincide with the vertical direction in the image we can correct orientation misalignments introduced during the document scanning process. See Section 2.3 for details.
 - Noise removal [32]. This step is performed to eliminate small regions produced by dirt in the original documents or minor scanning problems. See Section 2.4 for details.
- Blob detectors to determine regions candidate to containing kanji. See Section 2.5 for details.
 - Preprocessed images are used as the input for blob detector algorithms. These algorithms output parts of each page that are likely to correspond to kanji characters and store them as separate images. Three blob detector algorithms were studied (see Section 3.1 for the comparison).
- DL-Based Classification. See Section 2.6 for details.
 - In the last step of our pipeline, DL Networks are used to classify each possible kanji image into one of the kanji classes considered. Five different DL networks were studied for this purpose. See Section 3.2 for detailed results.
- The output of our algorithm is composed of the coordinates of each detected kanji along with its kanji category. We pay special attention to those regions classified as to

belong to the “ima” kanji. See Section 3.3 for the details focusing on the results related to the “ima” kanji as well as the combined performance of the blob detector and kanji classification algorithms for this particular case.

The techniques used are general-purpose, that can be used in other automatic Wasan document processing tools. These include the other uses of geometric primitives such as the vertical or horizontal lines detected for angle correction or the use of the same deep learning networks to classify salient features of Wasan documents such as equations (marked with distinctive rectangular frames) or the geometric shapes included in the geometric description of the problems.

2.3. Orientation Correction

Wasan documents frequently include “pattern lines” that helped writers maintain a coherent structure in their documents. For us these lines clearly show the tilt that the each page underwent during the scanning process. This tilt can be expressed as the angle between the vertical direction on the scanned image and any of these “almost vertical” pattern lines. We focused on finding vertical lines by (a) Detecting all lines, (b) Removing lines with slopes smaller than a set threshold (non-vertical) (c) Grouping neighboring line segments to find longer vertical lines. See Figure 5 for an example.

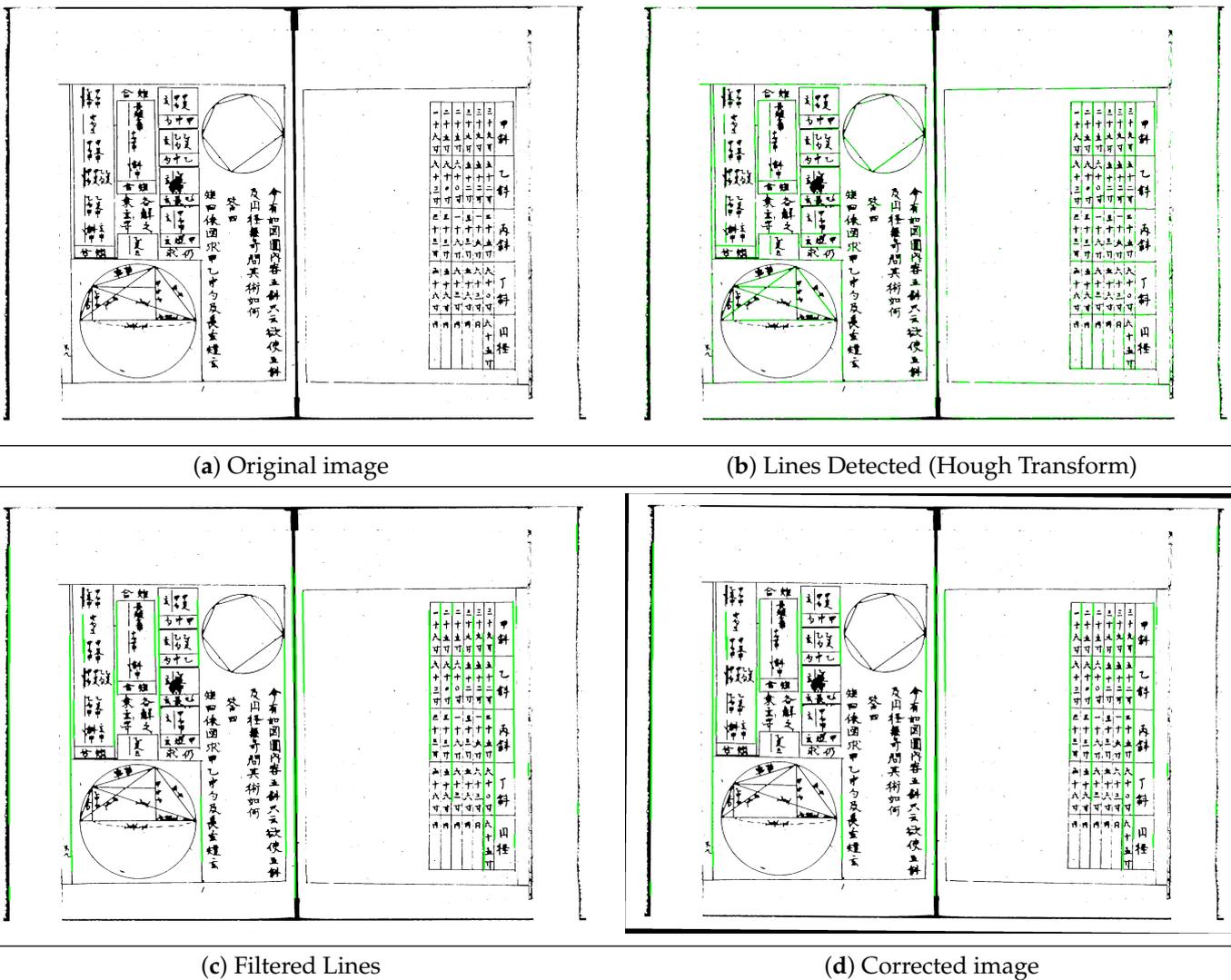


Figure 5. Orientation correction.

We detected lines using the Hough Line transform [33]. Due to the ink density of the original documents, written with calligraphy brushes, most of the long lines present in the documents (most importantly for us, the pattern lines) appear discontinuous in large zoom magnifications. This was a problem for deterministic versions of the Hough transform so we used the The probabilistic version [34] instead. Although we could detect lines, postprocessing was necessary in order to focus on lines pertinent for our orientation correction goal. Specifically lines under a set length where discarded as well as those with slopes that were not close enough to the vertical direction. Subsequently, we observed that many of the remaining lines could be grouped into longer lines. This happened because some parts of the original lines had sections that were faint or not visible. We grouped neighboring line endpoints and converted each group of lines into a “summary line”.

2.4. Noise Correction

We consider “noise” to be any spurious information appearing in an image. Usually, most of the noise present in images is a byproduct of image acquisition. However, in our data, the conservation state of the documents likely contributed to the image noise as small paper imperfections or small dirt particles may also appear as noisy regions in the digital images. It is impossible to remove all the noise from an image without producing distortions to it. However, in order for subsequent image analysis algorithms to produce satisfactory results, noise reduction is mandatory. Noise reduction in ancient documents is a particularly challenging research subject [35]. Generally, it is not possible to re-do the binarization process of the documents and it is necessary to work with images that have a low information-to-noise ratio [32].

In this case, we used the noise correction procedure described in [6] and illustrated in Figure 6. We use morphological operators [36–38] for reducing and eliminating noise regions in the images. Our procedure uses Dilation and Erosion operations to remove pattern lines and other document-shaping structures as well as to eliminate small noisy regions. In order to avoid deleting small parts of Kanji characters, the document starts with a Erosion operation to get all the strokes in each Kanji to be connected. See the aforementioned reference for a detailed description.

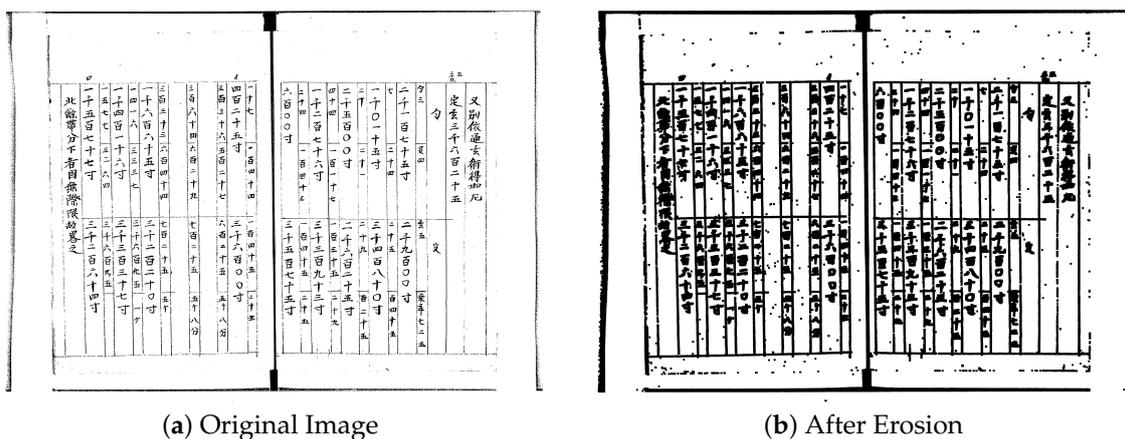


Figure 6. Cont.

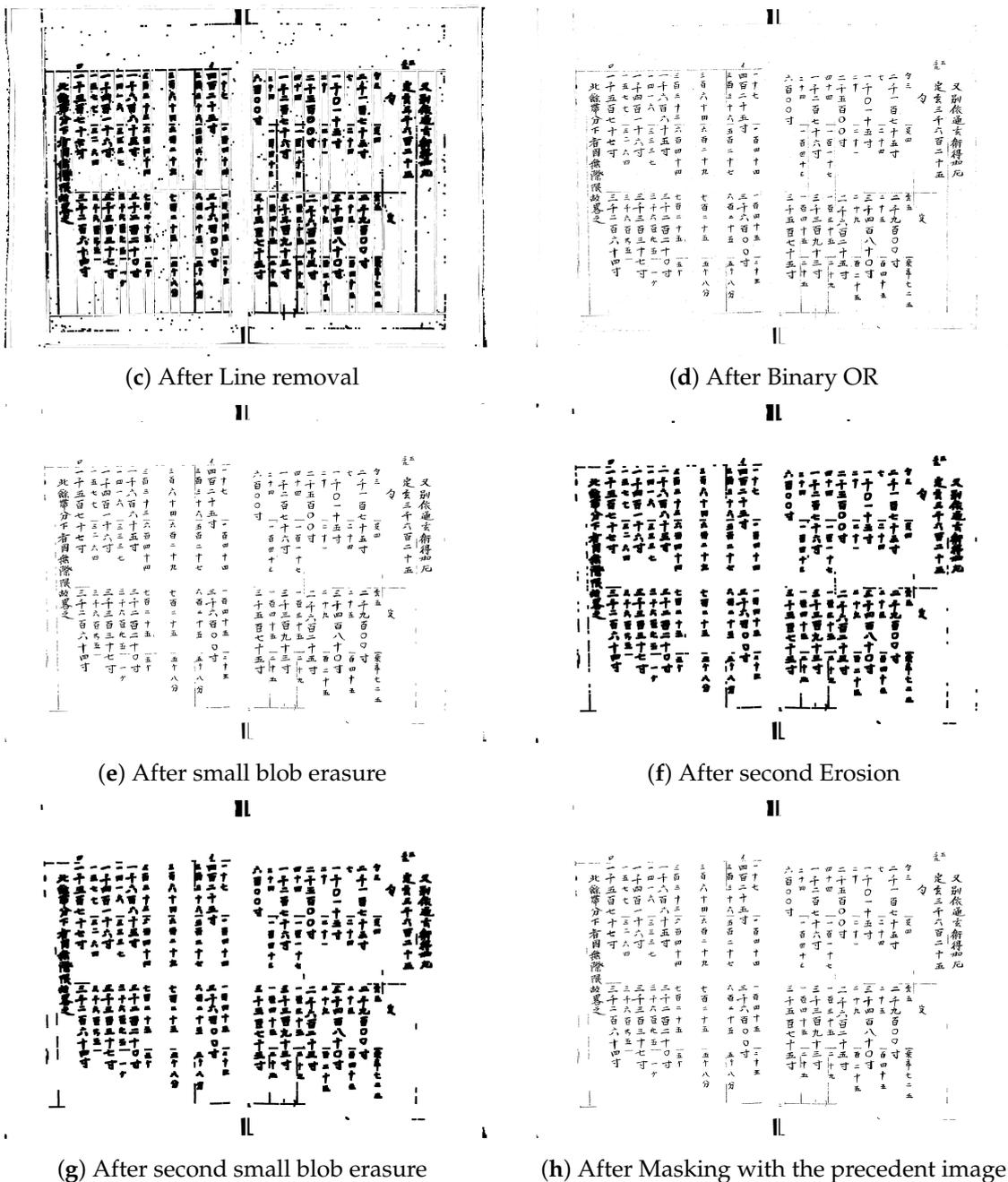


Figure 6. Example of the noise removal process.

2.5. Individual Kanji Segmentation: Blob Detectors

The characters appearing in Wasan documents were handwritten using calligraphy brushes. Each of these Kanji contains between 1 and 20 strokes. Consequently, each Kanji can be seen as small part of the image composed of smaller, disconnected parts. Furthermore, handwritten kanji (such as the ones in Wasan documents) often present irregular or incomplete shapes (see [17] for a succinct description of the particular challenges of handwritten kanji segmentation and recognition). We *blob detectors* to find the regions that are likely to contain Kanji in our images. We tested three different blob detectors implemented using the *SciKit* library [39]. Figure 7 presents an example of their results.

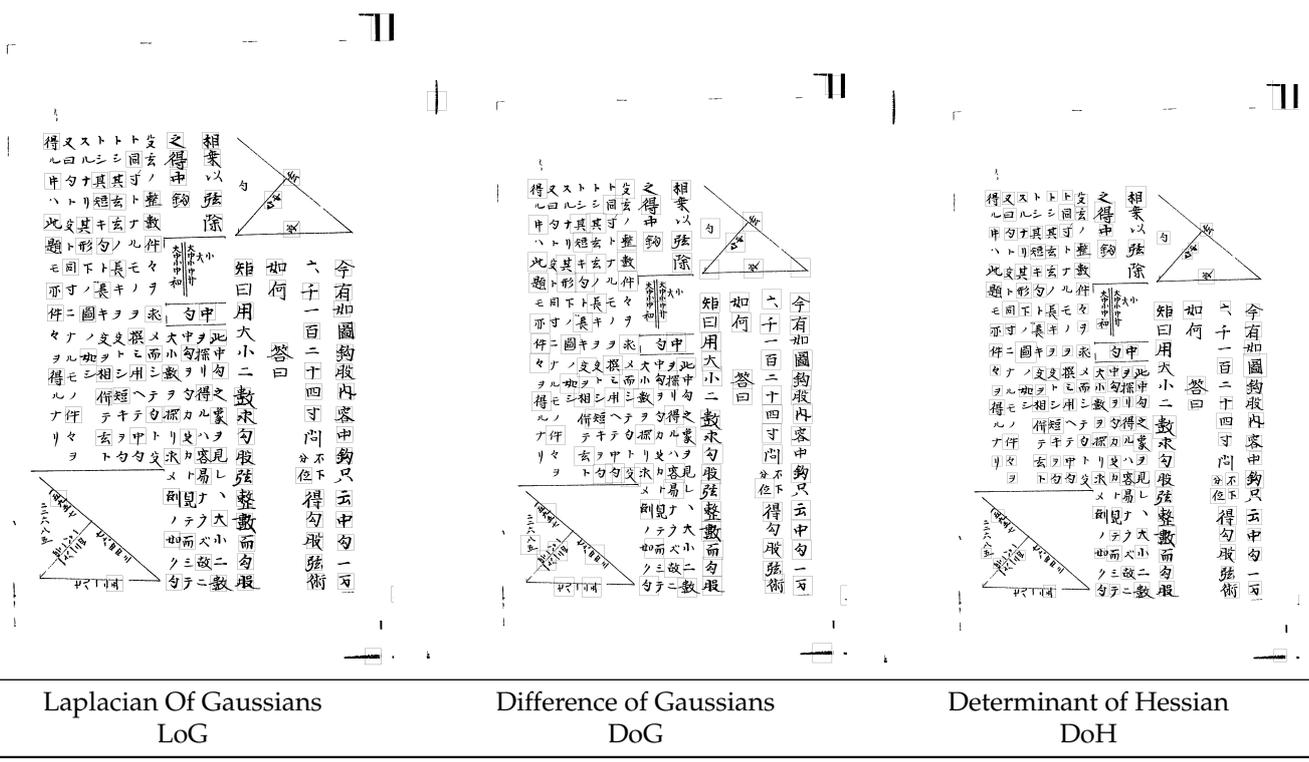


Figure 7. Example results of the three blob detector methods used for kanji detection.

2.5.1. Laplacian of Gaussian LoG

A Gaussian kernel is used on the input,

$$g(x, y, t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}}$$

t represents the scale of the blobs being detected. The Scale-space Laplacian operator is considered:

$$\nabla^2 L = L_{xx} + L_{yy}$$

In order to highlight extreme values with radius $r = \sqrt{2t}$ [40].

2.5.2. Difference of Gaussians DoG

Finite differences are used to approximate the Laplacian operator:

$$\nabla_{norm}^2 L(x, y; t) \approx \frac{1}{\Delta t} (L(x, y; t + \Delta t) - L(x, y; t))$$

This approach is known as the difference of Gaussians (DoG) approach [41].

2.5.3. Determinant of the Hessian DoH

The determinant of the Hessian a scale-space representation L is considered.

$$\det H_{norm} L = t^2 (L_{xx} L_{yy} - L_{xy}^2)$$

The blob detector is then defined as by detecting maxima of the aforementioned determinant.

2.6. Classification of Kanji Images Using DL Networks

In the final step of our algorithm, we focused on the classification of the obtained kanji images using DL networks. While these networks have achieved success in multiple classification tasks, their performance is conditioned by the problem being solved. This problem is most clearly expressed via the dataset from which the training and testing subsets are built. The data should faithfully reflect the practical problem that is being solved and be divided into training/validation and testing sets. The testing set should have absolutely no overlap with the training/validation set and, if at all possible, they should be independent (see [42,43] for discussion on good practices for data collection and subdivision).

Furthermore, the training/validation dataset should be built in accordance with the practical application goals. For example, in the case of heavily unbalanced datasets, the DL networks will tend to acquire biases towards the most frequent classes. If all the classes have the same importance for the application problems, this is not a problem, but in some cases the less frequent classes may actually be those of most interest for the application, and biased training sets may be considered (see [44] for an example).

2.6.1. Kanji Datasets Considered

Consequently, when considering the training datasets to be used to fine-tune DL networks, it is necessary to assess their degree of balance, but also their fitness of purpose for the application problem. In our case, we considered two datasets that reflect the peculiar characteristics of the kanji present in Wasan documents:

- The ETL dataset [19] is made up of 3036 kanji handwritten in modern style. The classes in this dataset are totally balanced and each kanji class is represented by 200 examples. As many of the kanji in the dataset represent relatively infrequent kanji, and in order to reduce the time needed to run the experiments, we chose a subset of the ETL database made up of 2136 classes of “regular use kanji” as listed by the Japanese Ministry of Education. For each of these kanji, we considered the 200 examples in the ETL database.
- The Kuzushiji-Kanji dataset is made up of 3832 kanji characters extracted from historical documents. Some of the characters have more than 1500 examples, while half the classes have fewer than 10. The dataset is, thus, very heavily imbalanced. Furthermore, many of the classes are also very infrequent. In order to achieve a slightly less imbalanced dataset made up of more frequent characters, we discarded the classes with less than 10 examples (obtaining a dataset with 1636 kanji classes) and downsampled the classes with more than 200 examples to contain exactly 200 randomly chosen examples.

We have used these two datasets with two main goals. In Section 3.2, we tackle the issue of to what extent DL networks can be used to discriminate the characters in each of the datasets. Subsequently, in Section 3.3, we focus on the adequacy of each of the two datasets (including data re-balancing) to the practical issue of detecting the “ima” kanji in Wasan documents.

2.6.2. DL Networks Tested

In order to study the performance of DL networks when used with these two datasets, we have considered five different deep learning architectures as defined on the FastAI Library [45].

1. Alexnet [46] is one of the first widely used convolutional neural networks, composed of eight layers (five convolutional layers sometimes followed by max-pooling layers and three fully connected layers). This network was the one that started the current DL trend after outperforming the current state-of-the-art method on the ImageNet data set by a large margin.

2. Squeezenet [47] uses so-called squeeze filters, including a point-wise filter to reduce the number of necessary parameters. A similar accuracy to Alexnet was claimed with fewer parameters.
3. VGG [21] represents an evolution of the Alexnet network that allowed for an increased number of layers (16 in the version considered in our work) by using smaller convolutional filters.
4. Resnet [48] is one of the first DL architectures to allow a higher number of layers (and, thus, “deeper” networks) by including blocks composed of convolution, batch normalization and ReLU. In the current work, a version with 50 layers was used.
5. Densenet [49] is another evolution of the resnet network that uses a larger number of connections between layers to claim increased parameter efficiency and better feature propagation that allows them to work with even more layers (121 in this work).

Taking into account the fact that we had a very large number of classes and a limited number of example for each of them, we decided to use the transfer learning property of deep learning networks to improve our kanji classification results. Taking into account previous practical experiences with these networks [50], we initialized all these DL classifiers using Imagenet weights [46]. Their final layers were then substituted by a linear layer with our number of classes. This final layer was followed by a softmax activation function. The networks thus constructed were trained using the manually annotated tree points and information of the tree health classes annotated in the orthomosaic. Initially, all the data were considered together and randomly divided into two subsets of 80% for training and 20% for testing.

3. Results

All tests were performed on a 2.6 GHz computer with an NVIDIA GTX 1080 GPU. The SciKit [39] and OpenCV [51] libraries were used for the implementation of the CV algorithms. The CNN presented where implemented using the Fastai Library [45] running over pytorch [52].

3.1. Experiment 1: Blob Detectors for Kanji Segmentation

This first experiment studies the performances of blob detectors to find Kanji characters. In order to do this, we considered several parameter combinations for the three methods (See https://scikit-image.org/docs/dev/auto_examples/features_detection/plot_blob.html for detailed documentation on the Scikit-learn methods used, accessed on 1 September 2020).

The main parameters considered were (min_{σ} , max_{σ} , num_{σ} , $threshold$, $overlap$). As it was impractical to run all parameter combinations with the full database, five Wasan images were used for the fine-tuning process, several parameter combinations were considered and the results for each of them was visually and automatically checked. min_{σ} and max_{σ} are related to the size of the blobs detected, and we obtained the best results with values $min_{\sigma} \in (35, 45)$, $max_{\sigma} \in (45, 50)$, the num_{σ} parameter related to the step size when exploring the parameter space and was set to 5. The threshold parameter relates to the variation allowed in scale space and was tested in the interval (0.01, 0.05). Finally, the overlap parameter relates to the degree of overlap allowed between different blobs. Values between 0 and 0.5 were tested.

For each blob detector and combination of parameters, we considered the blobs detected as candidates for containing kanji. Then, by comparing these blobs with our manual annotation of the position of the kanji we were able to determine the percentage of kanji that had been correctly detected (“True Positives” TP) as well as the regions that did not actually contain kanji (“False Positives” FP). By comparing the number of TP with the actual number of kanji regions in the manual annotation we could also determine the percentage of kanji detected/missed. Figure 8 shows the boxplots for the correctly detected kanji percentage (left) and FP detections (right) for the three blob detectors considered.

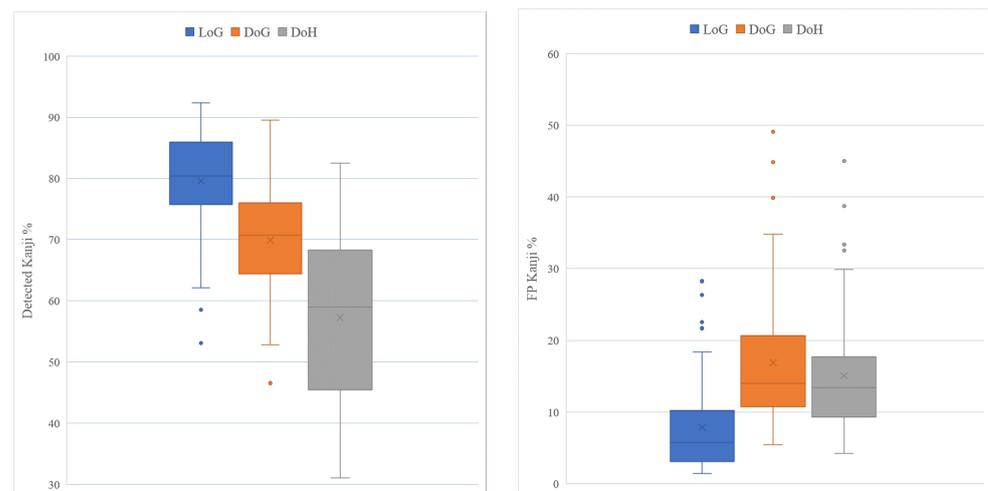


Figure 8. Comparison of the performance of blob detector algorithms for kanji detection.(left, correctly detected kanji %, right, false positive detection %).

The figure shows how the best performance was obtained by the LoG blob detector. The three blob detectors used obtained high success rates with average kanji detection percentages of 79.60, 69.89, 57.27 for the LoG, DoG and DoH detectors and standard deviations of 8.47, 8.10, 13.26. The LoG obtained the best average as well as the most stable performance (smaller standard deviation) with the DoG method obtaining a comparable performance. In terms of false positive detections, the three methods obtained averages of 7.88, 16.86, and 15.07, and standard deviations 6.54, 9.35, and 8.71. Once again, the LoG blob detector performed best and in this criterion the result was clearly superior to that of the DoG method, which proved to be confused more frequently by not properly cleaned up noise regions in the image producing a higher number of false positives.

3.2. Experiment 2: Classification of the Modern and Ancient Kanji Using Deep Learning Networks

In this section, we used several DL networks to classify Japanese kanji. We considered two databases. The first was a balanced database made up of 2136 modern, frequently used kanji characters, and the second was an unbalanced dataset (some kanji have 200 examples, others a little over 10) of kanji characters extracted from classical Japanese documents (see Section 2.6 for details).

For both databases, training and validation sets were considered. The data available in each case was randomly divided into 80% training and 20% validation. In order to limit the influence of hyperparameters, we tested 10 different learning rates ($(1 \times 10e^{-2}, 3 \times 10e^{-2} \dots 9 \times 10e^{-2}, 1 \times 10e^{-1}, 3 \times 10e^{-1} \dots 9 \times 10e^{-1})$). Figure 9 shows the compared performances for the ETL dataset for the five DL networks studied with detailed results for the 3 best performing networks (Densenet, Resnet and VGG) while Figure 10 shows results corresponding to the Kuzushiji dataset. The results are expressed in terms of the Error Rate ER, the ratio of images in the testing set that were misclassified.

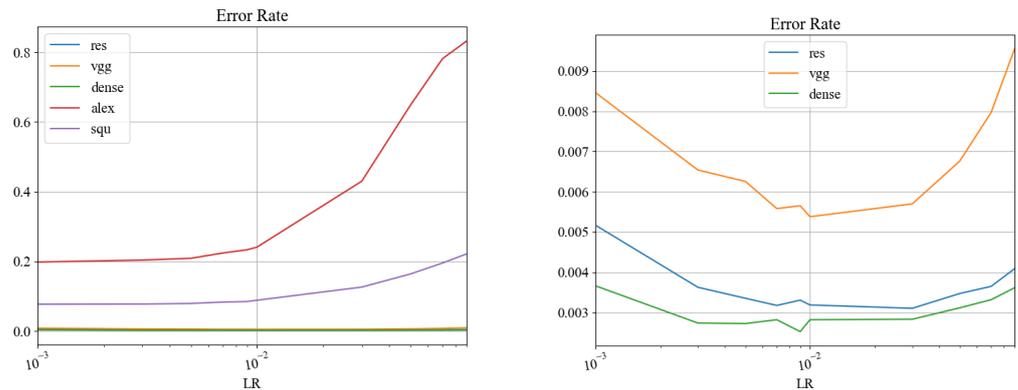


Figure 9. Compared performance of five DL networks for modern kanji classification (ETL dataset subset containing 2136 regular use kanji). **Left**, error rate (ER) results for all the DL networks considered. **Right**, details for the three best-performing networks.

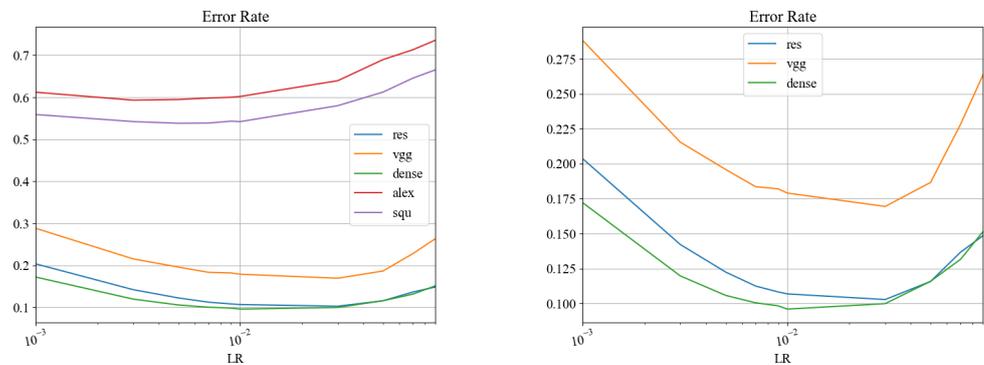


Figure 10. Compared performance of five DL networks for Classical kanji classification (Kuzushiji dataset rebalanced to ignore classes with fewer than 10 examples and cap the number of examples in a class to 200). **Left**, error rate (ER) results for all the DL networks considered. **Right**, details for the three best-performing networks.

Both Figures 9 and 10 show how best results were obtained by the bigger networks (Desenet, Resnet, VGG), with Squeezenet and Alexnet obtaining worse results. In both cases, the best results were obtained by the Densenet network with a similar performance obtained by the ResNet50 network. Best results were obtained at slightly different learning rates for each network, but all were close to $LR = 0.01$. For the ETL dataset, the best results in terms of ER for each network were: 0.0025 (Densenet), 0.0031 (ResNet), 0.0054 (VGG), 0.0773 (Squeezenet), and 0.1983 (Alexnet). For the Kuzujishi dataset, the corresponding results were: 0.096 (Densenet), 0.103 (ResNet), 0.169 (VGG), 0.538 (Squeezenet), and 0.593 (Alexnet).

3.3. Experiment 3: Classification of the “ima” kanji

In this final experiment, we studied the performance of our whole pipeline for the problem of detecting and classifying the “ima” kanji. Given the results obtained in Section 3.1, we used the LoG blob detector for kanji segmentation. Using the manual annotations of the positions of the “ima” kanji, we were able to ascertain that all present “ima” kanji had been detected by the blob detector. Consequently, the problem that remained was to correctly classify all kanji detected so the instances of the “ima” kanji present where correctly identified as such and no false positive detections (i.e., other kanji being wrongly classified as “ima”) were produced. To achieve this, we considered the Densenet network that obtained best results in Section 3.2. We considered this network trained using three different datasets: The ETL and Kuzushiji datasets introduced

in Section 2.6.1, but also a dataset (from now on “Mixed” dataset) built by adding a small number of images from the Kuzushiji dataset to the ETL dataset. Specifically, we identified 20 kanji that have changed their writing style from the Edo period until today and substituted all the example images that we had for them in the Kuzushiji dataset into the ETL dataset for regular use kanji. We repeated the experiment in Section 3.2 with the mixed dataset. These results followed the same tendencies of those presented in Figure 9 so, for the sake of brevity, we only report the best result obtained. This result was a ER of 0.0038 obtained by the Densenet network with a LR of 0.01. Table 1 contains TPR, the ratio of correctly predicted instances of the “ima” kanji over all existing “ima” kanji as well as FPR, the ratio over the total number of “ima” kanji of false positive detections (Kanji predicted as being “ima”, but really belonging to some other class).

Table 1. “Ima” kanji detection and classification.

Classif. DataSet	今 Detection %	今 Classification	
	(LoG Blob Detector)	TPR	FPR
Classic (Kuzushiji)	100	0.95	0.03
Modern (ETL)	100	0.10	0.04
Mixed	100	0.85	0.21

Table 1 shows how the best results in classification were obtained by the Kuzushiji dataset with 0.95 TRP and 0.03 FPR. This shows how the Densenet network trained with our re-balanced version of the Kuzushiji dataset can be effectively used to locate the “ima” kanji in historical Wasan documents. The results obtained by the ETL dataset were very low with only 10% of the instances of the kanji found (TPR of 0.10). This result was unexpected to us due to the superior classification accuracy, obtained in experiment Section 3.2. After reviewing the classes the “ima” kanji had been classified into, we realized that the examples in the modern dataset differed enough from the Edo period way of writing the kanji (see Figure 3 for it to be confused by the classifier with similar kanji. Specifically, 今 was often classified as 令, 命 or 合. In order to quantify this issue, we re-trained the Densenet network using a much smaller subset of the ETL dataset including 648 kanji and excluding 令, 命, 合 and we obtained a TPR of 64 and FPR of 6 for the classification of the “ima” kanji. Finally, the “Mixed” dataset obtained by adding examples of selected kanji to the ETL dataset obtained a TRP of 0.85 and a FPR of 0.21.

4. Discussion

We have presented what is, to the best of our knowledge, the first detailed analysis of computer vision tools and deep learning for the analysis of Wasan documents.

4.1. Kanji Detection

In Section 3.1, we studied the performance of three blob detector algorithms to segment individual kanji characters in Wasan documents. The best results were obtained by the LoG blob detector with the DoG method obtaining similar detection results with a large number of false positives. Visual analysis of the results revealed that the LoG has problems properly detecting the kanji in pages where they are written in two or more different sizes. Specifically, smaller kanji tend to be overlooked. Regarding false positives, they mostly belong to regions of the image corresponding to figures or noisy regions that had not been properly cleaned up. Nevertheless, the 79.60, 7.88 average results for TP and FP detection percentage represent a very promising result for the detected kanji to be used for automatic document structure analysis as well as a starting point for the automatic recognition of the extracted text.

4.2. Kanji Classification

In Section 3.2, we compared the performance of five DL network when classifying images corresponding to kanji characters for two databases, one balanced and depicting modern kanji (ETL) and another heavily unbalanced and depicting classical kanji (Kuzushiji). The fact that all networks behaved similarly for the two databases shows how we are dealing with similar problems. However, the error rates obtained for them were clearly different. The best result obtained using Densenet for the ETL dataset indicates that it can classify the images in the randomly-chosen validation dataset with around 0.25% error (ER = 0.0025). This is a very high accuracy ($\simeq 99.75\%$), which exceeds the 99.53% classification accuracy for 873 kanji classes [21] obtained with a modified VGG network. Our own results with the VGG network reproduce those of [21]. Given that our database included 2136 kanji (more than double that of [21]) shows that large DL networks can classify large balanced kanji datasets without loss in accuracy.

The error rate obtained when classifying the Kuzushiji classical kanji dataset was much higher, with a best obtained result of ER = 0.096. This is equivalent to a classification accuracy of $\simeq 90.4\%$ over all classes. These results show how the problem of classifying the images in this dataset is much more challenging for DL networks. Possible causes for this are the inferior image quality of the images depicting kanji in the historical database (due to document conservation issues) and, most importantly, to the lack of balance in it. This latest point becomes apparent when we check the average classification accuracy for the classes present in the validation set (an average of 81.72% was obtained). The eight-point difference between the average classification accuracy and the overall accuracy shows how the more frequent classes are classified more accurately and bias the overall result. These results improve those previously presented in [28] using a Lenet network. Our subset of the Kuzushiji dataset is comparable to the one where [28] obtained 73.10% accuracy (considering classes with at least 20 examples but without capping the number of examples for larger classes). Our results, however, reach the values of those reported in [28] by filtering “difficult to classify characters”. The effects of re-balancing the dataset as well as the use of networks with a larger number of layers become apparent in the 81.72% average classification accuracy for all classes obtained by our densenet network. This result clearly improves the 37.22% result reported in [28] for Lenet.

4.3. “ima” Kanji Detection

In Section 3.3, we presented a detailed study of how our algorithmic process can be used to locate an important structural element of Wasan Documents, the “ima” kanji that marks the location of the geometric problem diagram and is also the first character in the textual description. This tool uses two steps. In the first, blob detectors are used to select the regions containing kanji characters and in the second the characters contained in them are classified and those matching “ima” are identified. The “ima” kanji was detected in 100% of cases by the best-performing blob detector. This was probably due to this kanji having a very clearly delimited shape and it always being present at the beginning of the text description of the mathematical problems. This avoided some of the detection problems where nearby kanji might get mixed together or where parts of smaller kanji might get added to nearby larger ones.

Regarding the classification of the resulting kanji candidate regions, we used a Densenet network, as it had obtained the best results in the kanji classification experiment. In order to train it, we considered the ETL and Kuzushiji datasets, but also an enriched version of the ETL dataset using some kanji from the Kuzushiji. The reason to build this third dataset is that we observed that sometimes, the “ima” kanji 今 was confused with similar kanji (such as, for example the “rei” kanji 令) when classified with a network trained on the ETL dataset. We believe this happened due to the differences in writing style between Edo period and modern kanji, and it was our hope that by enriching the ETL dataset with examples closer to the Edo period scripture would ameliorate the problem. As a caveat, the mixed dataset does not account for all differences in style or kanji that

are infrequent in present times, but appear more often in Wasan documents. However, we believe that the adaptation that we present allows us to make the case that a tailored dataset using modern and classical kanji can be of interest when processing Edo-period Wasan documents. The best result obtained for the classification of the mixed dataset was a ER of 0.0038 obtained by the Densenet network with a LR of 0.01. This represents a slight drop in accuracy from the 99.75% obtained with the ETL database to 99.62%, still clearly better than that of the Kuzushiji dataset (accuracy $\approx 90.4\%$). The balance of the dataset still allowed us to maintain a consistent classification accuracy among all classes (average class-wise accuracy was 99.62%, the same as the overall accuracy).

Regarding the final results when classifying the “ima” kanji candidates extracted automatically from Wasan documents, the 0.95 TPR/0.03 FPR obtained by the network trained on the Kuzushiji dataset show how our approach can be used to effectively determine the position of the “ima” kanji for subsequent analysis of the structure of Wasan documents. With minimal correction from human users, this tool will be used to build a Wasan document database enriched with automatically extracted information. The very low results obtained by the network trained on the ETL dataset exemplifies the need to have a training dataset for classification that faithfully represents the practical problem that is being solved. While the Densenet network was able to classify the kanji in the ETL dataset with high accuracy, that did not help it properly locate the “ima” kanji as it appears in Wasan documents. As the kanji in Wasan documents present both classical and modern characteristics, we also tried a “Mixed” dataset that added to the ETL network some of the more frequent among those that have changed their writing style from the Edo period to current days. The results obtained by the Densenet network trained with this dataset in the detection of the “ima” kanji was a TPR of 0.85 and a FPR of 0.21. These results, while not as good as those obtained when training with the Kuzushiji dataset are still high and the mixed dataset is likely to produce better classification accuracy for the kanji that are not “ima”, so this version of our pipeline represents a promising starting point for a tool that can perform full text extraction and interpretation of Wasan documents.

5. Conclusions and Future Work

The process described in this study is the first step in the construction of a database of Wasan documents. The goal is to make this database searchable according to the geometric characteristics of each problem. Our results have shown how our algorithms can overcome the image quality issues of the data (mainly noise, low resolution and orientation tilt) and locate the occurrences of the “ima” Kanji reliably. We have also shown how deep learning networks can be used to effectively classify kanji characters of different styles and with datasets of varying degree of balance between their classes. We have not yet achieved full and satisfactory text interpretation of Wasan documents, but our methods can already detect about 80% of the kanji present on average with FPR of about 8%. We have shown how to construct mixed datasets that can confidently detect the “ima” kanji while at the same time maintaining the class balance and the modern kanji information contained in the ETL network. In future work, we will increase the size of our testing dataset in order to better measure the accuracy and consistency of the methods presented. We will also advance in the use of computer vision tools to detect other geometric primitives in the images such as the circles and triangles present in problem diagrams, but also rectangles used to enclose equations. We will improve the kanji detection step by using the pattern lines present in many documents as well as the structure of the text itself (with kanji characters always presenting lined up and evenly spaced) and we will continue to develop a kanji database that is fully suitable for the classification of Wasan kanji by combining existing datasets.

Author Contributions: Conceptualization, methodology, formal analysis, Y.D., K.W., M.V.; software, validation, Y.D., T.S.; resources, data curation K.W., T.S.; writing—original draft preparation, Y.D.; writing—review and editing, Y.D., K.W., M.V.; visualization, Y.D., T.S., M.V.; supervision, Y.D., K.W.; project administration, Y.D.; funding acquisition, K.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No Data is made available from the authors. Several Publicly available data sources are mentioned in the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Matsuoka, M. Wasan, and Its Cultural Background. In *Katachi and Symmetry*; Ogawa, T., Miura, K., Masunari, T., Nagy, D., Eds.; Springer: Tokyo, Japan, 1996; pp. 341–345.
2. Martzloff, J.C. A survey of Japanese publications on the history of Japanese traditional mathematics (Wasan) from the last 30 years. *Hist. Math.* **1990**, *17*, 366–373. [[CrossRef](#)]
3. Smith, D.E.; Mikami, Y. *A History of Japanese Mathematics*; Felix Meiner: Leipzig, Germany, 1914; p. 288.
4. Mitsuyoshi, Y. *Jinkouki*; Wasan Institute: Tokyo, Japan, 2000; p. 215.
5. Fukagawa, H.; Rothman, T. *Sacred Mathematics: Japanese Temple Geometry*; Princeton Publishing: Princeton, NJ, USA, 2008; p. 348.
6. Diez, Y.; Suzuki, T.; Vila, M.; Waki, K. Computer vision and deep learning tools for the automatic processing of WASAN documents. In Proceedings of the ICPRAM 2019—Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, Prague, Czech Republic, 19–21 February 2019; pp. 757–765.
7. Suzuki, T.; Diez, Y.; Vila, M.; Waki, K. Computer Vision and Deep learning algorithms for the automatic processing of Wasan documents. In Proceedings of the 34th Annual Conference of JSAI, Online. 9–12 June 2020; The Japanese Society for Artificial Intelligence: Kumamoto, Japan, 2020; pp. 4Rin1–10.
8. Pomplun, M. *Hands-On Computer Vision*; World Scientific Publishing: Singapore, 2022. [[CrossRef](#)]
9. Liu, C.; Dengel, A.; Lins, R.D. Editorial for special issue on “Advanced Topics in Document Analysis and Recognition”. *Int. J. Doc. Anal. Recognit.* **2019**, *22*, 189–191. [[CrossRef](#)]
10. Otter, D.W.; Medina, J.R.; Kalita, J.K. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 604–624. [[CrossRef](#)] [[PubMed](#)]
11. Dahl, C.M.; Johansen, T.S.D.; Sørensen, E.N.; Westermann, C.E.; Wittrock, S.F. Applications of Machine Learning in Document Digitisation. *CoRR* **2021**. Available online: <http://xxx.lanl.gov/abs/2102.03239> (accessed on 1 January 2021).
12. Philips, J.; Tabrizi, N. Historical Document Processing: A Survey of Techniques, Tools, and Trends. In Proceedings of the 12th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2020, Volume 1: KDIR, Budapest, Hungary, 2–4 November 2020; Fred, A.L.N., Filipe, J., Eds.; SCITEPRESS: Setúbal, Portugal, 2020; pp. 341–349. [[CrossRef](#)]
13. Cao, C.; Wang, B.; Zhang, W.; Zeng, X.; Yan, X.; Feng, Z.; Liu, Y.; Wu, Z. An Improved Faster R-CNN for Small Object Detection. *IEEE Access* **2019**, *7*, 106838–106846. [[CrossRef](#)]
14. Guo, L.; Wang, D.; Li, L.; Feng, J. Accurate and fast single shot multibox detector. *IET Comput. Vis.* **2020**, *14*, 391–398. [[CrossRef](#)]
15. Redmon, J.; Divvala, S.K.; Girshick, R.B.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016; IEEE Computer Society: Washington, DC, USA, 2016; pp. 779–788. [[CrossRef](#)]
16. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *CoRR* **2018**. Available online: <http://xxx.lanl.gov/abs/1804.02767> (accessed on 1 January 2021).
17. Tomás Pérez, J.V. Recognition of Japanese Handwritten Characters with Machine Learning Techniques. Bachelor’s Thesis, University of Alicante, Alicante, Spain, 2020.
18. Wang, Q.; Yin, F.; Liu, C. Handwritten Chinese Text Recognition by Integrating Multiple Contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1469–1481. [[CrossRef](#)] [[PubMed](#)]
19. ETL. ETL Character Database. 2018. Available online: <http://etlcnb.db.aist.go.jp/> (accessed on 20 November 2018).
20. Tsai, C. *Recognizing Handwritten Japanese Characters Using Deep Convolutional Neural Networks*; Technical Report; Stanford University: Stanford, CA, USA, 2016.
21. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations ICLR 2015, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings.
22. Grębowiec, M.; Protasiewicz, J. A Neural Framework for Online Recognition of Handwritten Kanji Characters. In Proceedings of the 2018 Federated Conference on Computer Science and Information Systems (FedCSIS), Poznań, Poland, 9–12 September 2018; pp. 479–483.
23. Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; Ha, D. Deep Learning for Classical Japanese Literature. *CoRR* **2018**. Available online: <http://xxx.lanl.gov/abs/cs.CV/1812.01718> (accessed on 1 January 2021).

24. Ueki, K.; Kojima, T. Survey on Deep Learning-Based Kuzushiji Recognition. In *Pattern Recognition. ICPR International Workshops and Challenges*; Del Bimbo, A., Cucchiara, R., Sclaroff, S., Farinella, G.M., Mei, T., Bertini, M., Escalante, H.J., Vezzani, R., Eds.; Springer International Publishing: Cham, Switzerland, 2021; pp. 97–111.
25. Saini, S.; Verma, V. International Journal of Recent Technology and Engineering IJ. *CoRR* **2019**, *8*, 3510–3515.
26. Ahmed Ali, A.A.; Suresha, M.; Mohsin Ahmed, H.A. Different Handwritten Character Recognition Methods: A Review. In *Proceedings of the 2019 Global Conference for Advancement in Technology (GCAT)*, Bangalore, India, 18–20 October 2019; pp. 1–8. [[CrossRef](#)]
27. Tang, Y.; Hatano, K.; Takimoto, E. Recognition of Japanese Historical Hand-Written Characters Based on Object Detection Methods. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing, HIP@ICDAR 2019*, Sydney, NSW, Australia, 20–21 September 2019; pp. 72–77. [[CrossRef](#)]
28. Ueki, K.; Kojima, T. Japanese Cursive Character Recognition for Efficient Transcription. In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods—Volume 1: ICPRAM, INSTICC, Valletta, Malta, 22–24 February 2020*; pp. 402–406.
29. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [[CrossRef](#)]
30. University, Y. Yamagata University Wasan Sakuma Collection (Japanese). 2018. Available online: <https://www.occonvert.com/japanese-ocr> (accessed on 20 November 2018).
31. Fernandes, L.A.; Oliveira, M.M. Real-time line detection through an improved Hough transform voting scheme. *Pattern Recognit.* **2008**, *41*, 299–314. [[CrossRef](#)]
32. Agrawal, M.; Doermann, D.S. Clutter noise removal in binary document images. In *Proceedings of the 2009 10th International Conference on Document Analysis and Recognition, Barcelona, Spain, 26–29 July 2009*; Volume 16, pp. 351–369. [[CrossRef](#)]
33. Illingworth, J.; Kittler, J. A survey of the hough transform. *Comput. Vis. Graph. Image Process.* **1988**, *44*, 87–116. [[CrossRef](#)]
34. Matas, J.; Galambos, C.; Kittler, J. Progressive Probabilistic Hough Transform. In *Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, Fort Collins, CO, USA, 23–25 June 1999; Volume 1, pp. 554–560. [[CrossRef](#)]
35. Arnia, F.; Muchallil, S.; Munadi, K. Noise characterization in ancient document images based on DCT coefficient distribution. In *Proceedings of the 13th International Conference on Document Analysis and Recognition, ICDAR 2015, Nancy, France, 23–26 August 2015*; pp. 971–975. [[CrossRef](#)]
36. Barna, N.H.; Erana, T.I.; Ahmed, S.; Heickal, H. Segmentation of Heterogeneous Documents into Homogeneous Components using Morphological Operations. In *Proceedings of the 17th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2018, Singapore, 6–8 June 2018*; pp. 513–518. [[CrossRef](#)]
37. Goyal, B.; Dogra, A.; Agrawal, S.; Sohi, B.S. Two-dimensional gray scale image denoising via morphological operations in NSST domain & bitonic filtering. *Future Gener. Comp. Syst.* **2018**, *82*, 158–175. [[CrossRef](#)]
38. Tekleyohannes, M.K.; Weis, C.; Wehn, N.; Klein, M.; Siegrist, M. A Reconfigurable Accelerator for Morphological Operations. In *Proceedings of the 2018 IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPS Workshops 2018, Vancouver, BC, Canada, 21–25 May 2018*; pp. 186–193. [[CrossRef](#)]
39. van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T.; The Scikit-Image Contributors. scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [[CrossRef](#)] [[PubMed](#)]
40. Lindeberg, T. Image Matching Using Generalized Scale-Space Interest Points. *J. Math. Imaging Vis.* **2015**, *52*, 3–36. [[CrossRef](#)]
41. Marr, D.; Hildreth, E. Theory of Edge Detection. *Proc. R. Soc. Lond. Ser. B* **1980**, *207*, 187–217. [[CrossRef](#)]
42. Diez, Y.; Kentsch, S.; Fukuda, M.; Caceres, M.L.L.; Moritake, K.; Cabezas, M. Deep Learning in Forestry Using UAV-Acquired RGB Data: A Practical Review. *Remote Sens.* **2021**, *13*, 2837. [[CrossRef](#)]
43. Wen, J.; Thibeau-Sutre, E.; Diaz-Melo, M.; Samper-González, J.; Routier, A.; Bottani, S.; Dormont, D.; Durrleman, S.; Burgos, N.; Colliot, O. Convolutional neural networks for classification of Alzheimer’s disease: Overview and reproducible evaluation. *Med. Image Anal.* **2020**, *63*, 101694. [[CrossRef](#)] [[PubMed](#)]
44. Cabezas, M.; Kentsch, S.; Tomhave, L.; Gross, J.; Caceres, M.L.L.; Diez, Y. Detection of Invasive Species in Wetlands: Practical DL with Heavily Imbalanced Data. *Remote Sens.* **2020**, *12*, 3431. [[CrossRef](#)]
45. Howard, J.; Thomas, R.; Gugger, S. Fastai. Available online: <https://github.com/fastai/fastai> (accessed on 1 January 2021).
46. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1, Red Hook, NY, USA, 3–6 December 2012*; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.
47. Iandola, F.N.; Moskewicz, M.W.; Ashraf, K.; Han, S.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <1 MB model size. *arXiv* **2016**, arXiv:1602.07360.
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
49. Huang, G.; Liu, Z.; Weinberger, K.Q. Densely Connected Convolutional Networks. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21–26 July 2016; pp. 2261–2269.
50. Kentsch, S.; Caceres, M.L.L.; Serrano, D.; Roure, F.; Diez, Y. Computer Vision and Deep Learning Techniques for the Analysis of Drone-Acquired Forest Images, a Transfer Learning Study. *Remote Sens.* **2020**, *12*, 1287. [[CrossRef](#)]

-
51. Bradski, G. The OpenCV Library. *Dr. Dobbs J. Softw. Tools* **2000**, *120*, 122–125.
 52. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. *Automatic Differentiation in PyTorch*; NIPS Autodiff Workshop: Long Beach, CA, USA, 2017.