

Seal-Robust KCR: A Robust Kuzushiji Character Recognition Framework under Seal Interference*

Rui-Yang Ju, Kohei Yamashita, Hirotaka Kameko, Shinsuke Mori
Kyoto University, Kyoto, Japan

<https://ruiyangju.github.io/Seal-Robust-KCR>

Abstract

Kuzushiji was one of the most widely used cursive writing systems in pre-modern Japan. Due to its highly cursive forms and extensive glyph variations, most modern Japanese readers are unable to read Kuzushiji characters. Consequently, recent studies have focused on developing automated Kuzushiji character recognition (KCR) methods, which have achieved strong performance on relatively clean Japanese historical document images. Although seals frequently appear in Japanese historical documents, existing methods often fail to maintain recognition accuracy under seal interference, particularly when seals overlap with characters. To address this challenge, we propose a seal-robust KCR framework. Based on character detection, classification, and ordering, the proposed framework additionally incorporates document restoration to mitigate seal interference, thereby improving overall recognition performance. In addition, we introduce a novel synthetic data augmentation strategy to enhance the performance of character detection models. We further correct annotation errors, reconstruct the dataset, and create a synthetic test set to simulate severe seal interference. Experimental results demonstrate the effectiveness of the proposed framework in mitigating the impact of seal interference on KCR. Compared with a conventional baseline and NDLkotenOCR, it achieves relative character error rate (CER) reductions of 39.7% and 5.9%, respectively, on the real test set, and 50.1% and 41.7%, respectively, on the synthetic test set.

1. Introduction

Kuzushiji (くずし字), a traditional cursive writing system widely used in pre-modern Japan, is found extensively in Japanese historical documents. Although it shares the same underlying character system as modern Japanese, including Kanji (漢字, Chinese-origin logographic characters) and Kana (仮名, Japanese syllabaries), its character

forms differ substantially from their modern counterparts. Furthermore, Kuzushiji encompasses multiple cursive writing conventions and numerous variant forms of both Kanji and Kana [23]. Following the Meiji Restoration, the adoption of modern printing technologies and the establishment of modern educational systems led to a decline in the use of Kuzushiji [9]. As a result, most modern Japanese readers are unable to read historical documents written in Kuzushiji, and only a limited number of trained specialists can accurately interpret them.

Recent advances in deep learning have enabled models trained on expert-annotated datasets to automatically recognize Kuzushiji characters [5]. To date, several Kuzushiji character recognition (KCR) methods have been developed to assist the general public in interpreting Japanese historical documents, including Komonjo Camera [27], developed by TOPPAN Inc.; miwo [4] and KuroNet [7, 16], developed by the Center for Open Data in the Humanities (CODH); NDLkotenOCR [21] and NDLkotenOCR-Lite [1], developed by the National Diet Library (NDL).

Japanese historical documents frequently contain seals that serve as marks of ownership, identity, and provenance. These seals were often stamped not only by the original creators of the documents but also by subsequent owners and collectors. Typically rendered in red ink, they may contain stylized characters representing the owners' names, social status, or personal aspirations. As a result, a single Japanese historical document may contain multiple seals accumulated over time, many of which partially overlap with the original handwritten Kuzushiji characters.

From the perspective of KCR, seals pose a significant challenge and can lead to recognition errors [12]. However, none of the above KCR methods explicitly addresses seal interference. To address this limitation, we propose Seal-Robust KCR, a Kuzushiji character recognition framework robust to seal interference in Japanese historical documents.

The main contributions of this work are as follows:

- (a) We correct annotation errors in 1,000 Japanese historical document images, reconstruct the dataset, and create a synthetic test set to simulate severe seal interference.

*This work was supported by JSPS KAKENHI Grant Number 25H01242, and JST SPRING Grant Number JPMJSP2110.

- (b) We propose Seal-Robust KCR, which incorporates a training-free document restoration algorithm to mitigate seal interference and improve recognition accuracy.
- (c) We introduce a synthetic data augmentation strategy for training character detection models, improving detection performance under severe seal interference without modifying the detector architecture.
- (d) We develop a training-free character ordering algorithm that achieves higher accuracy and faster inference speed than the ordering method used in NDLkotenOCR.

2. Related Work

KCR methods can be broadly categorized into two types: character-level and text line-level methods. In character-level KCR methods, each single character is first detected in Japanese historical document images and cropped into separate patches. Each patch is then classified into a Unicode code point and transcribed into its corresponding modern Japanese character. The transcribed characters are subsequently ordered to produce the final recognition result. Representative methods include miwo [4], a mobile application, and KuroNet [7, 16], a web-based application, both developed by the CODH. Specifically, both employ U-Net-based [24] semantic segmentation [6] for pixel-wise character segmentation, followed by Unicode classification of the segmented characters to produce the recognition result.

Another type of KCR is the text line-level method. These methods first perform layout recognition to detect all Kuzushiji text lines in Japanese historical document images. The detected text lines are then recognized as modern Japanese text, and the corresponding text lines are subsequently ordered to produce the final recognition result. Representative methods include NDLkotenOCR [21], a public optical character recognition (OCR) service for Japanese historical documents introduced by the NDL in 2022, with updated versions released in 2023 and 2024. Since NDLkotenOCR requires GPU acceleration during inference, the NDL further introduced a lightweight variant, NDLkotenOCR-Lite [1], to enable efficient processing in CPU-only environments. Specifically, NDLkotenOCR employs Cascade Mask R-CNN [3] for layout recognition, TrOCR [17] with RoBERTa-small [19] as the decoder for text line recognition, and LightGBM [13] for text line ordering. In contrast, NDLkotenOCR-Lite uses RTMDet [20] for layout recognition and PARSeq [2] for text line recognition while retaining the same text line ordering method.

In addition to the methods developed by research institutes, commercial KCR applications have also been introduced by industry. For example, TOPPAN Inc. has been developing KCR technologies since 2015. In 2023, its Fuminoha project released a mobile application named Komonjo Camera [27], which integrates two recognition engines: Komonjo AI and Kotenseki AI. Komonjo AI is

designed for documents containing Kanji characters with variable stroke widths, while Kotenseki AI is optimized for documents primarily composed of Kana with relatively uniform stroke widths. Since the application is not open source and no related technical reports or academic publications have been released, the specific methods remain unclear.

Furthermore, several datasets and competitions for KCR, as well as studies focusing on specific KCR sub-tasks, have emerged in recent years. For example, in 2019, the CODH released a Japanese historical document dataset and organized the Kuzushiji Recognition competition on Kaggle [14], which attracted 338 participants. In addition, Sakana AI introduced Metom [10] in 2025, a Vision Transformer (ViT)-based [8] classifier designed for recognizing individual Kuzushiji characters. The pre-trained model was trained on 649,932 character samples, validated on 216,644 samples, and evaluated on 216,645 samples, achieving a micro accuracy of 0.9722 and a macro accuracy of 0.8354.

Although the above methods achieve strong recognition performance on relatively clean Japanese historical document images, their recognition performance is often significantly affected by seal interference [12].

3. Proposed Method

3.1. Overall Framework

This work builds upon a character-level KCR pipeline. As described in Section 2, the conventional character-level KCR pipeline obtains the final recognition results through Kuzushiji character detection, character classification, and character ordering, as shown by the blue flow in Figure 1.

To mitigate the impact of seal interference on KCR, we extend the conventional pipeline by incorporating document restoration. Specifically, document restoration is performed in parallel with Kuzushiji character detection using a color-based thresholding algorithm to remove seal interference. Since the restoration process is computationally lightweight and requires less processing time than character detection, it does not introduce additional latency to the overall pipeline. The restored document image is then used in place of the original image to crop character patches, after which character classification and character ordering are performed as in the conventional pipeline. The overall pipeline of the proposed framework is shown by the red flow in Figure 1.

3.2. Synthetic Data Augmentation

To enhance the robustness of detection models against seal interference, we propose a synthetic data augmentation (SDA) strategy that constructs additional training samples with severe seal interference by randomly overlaying a large number of red seals onto original historical document images. Specifically, the document images in the original training set typically contain only 0 to 2 real seals, mak-

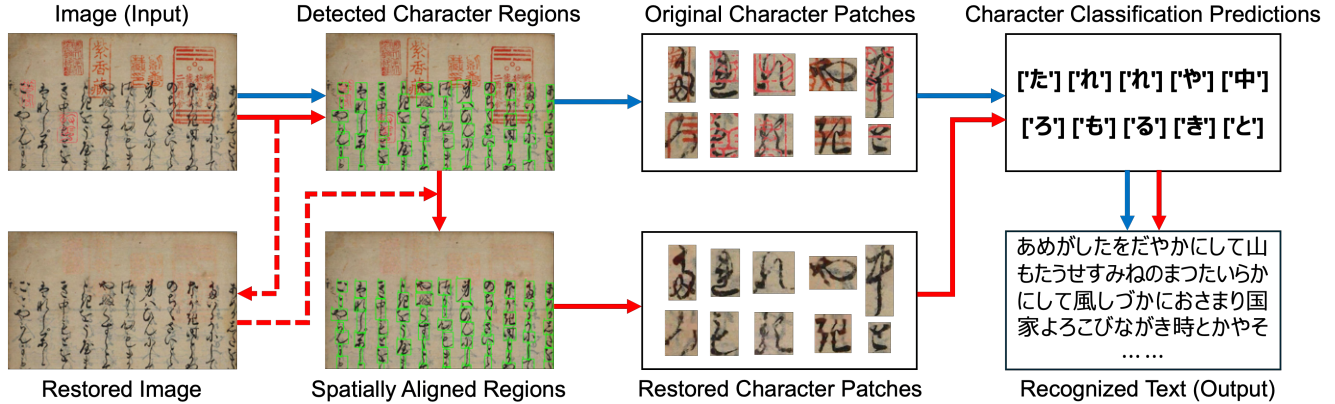


Figure 1. Conventional pipeline (blue flow) and the proposed pipeline (red flow) for seal-interfered Japanese historical document images. Dashed arrows indicate additional processes performed in parallel with character detection without affecting the detection results.

ing it difficult to adequately simulate complex overlap scenarios between characters and seals. Therefore, additional synthetic training samples with severe seal interference are obtained using collected high-resolution seal images.

These seal images are collected from pre-modern East Asian historical artifacts and primarily contain Kanji characters and traditional decorative patterns. These synthetic samples provide additional cases of character and seal overlap during training, enabling the detection model to learn more robust character detection under seal interference.

For the detailed synthesis process, the seal images are first processed to remove their backgrounds and then randomly scaled, with the longest side constrained to between 100 and 300 pixels to simulate seals of varying sizes in real historical documents. Since historical document layouts vary across images, the valid document region is determined individually for each document based on its layout. To prevent seals from extending beyond this region, their placement is restricted to a rectangular area corresponding to the valid document region. For each newly synthesized seal position, the overlap between seals is constrained to ensure that no more than two seals overlap at any location. Finally, the scaled transparent seals are overlaid onto the original document images using alpha compositing. As shown in Table 1, after applying the proposed SDA strategy, the training and validation sets increased from 800 to 1,600 images and from 100 to 200 images, respectively. Each synthetic image contains approximately 10 seal instances on average.

3.3. Kuzushiji Character Detection

This work employs real-time object detection models to detect individual Kuzushiji character regions in historical document images. Dense character arrangements and complex handwriting styles in Japanese historical documents pose several challenges for character detection, including missed detections, duplicate detections of partial character regions, and false positives caused by document stains. Under seal

Table 1. Statistics of the dataset splits and the proposed SDA strategy, where Avg. S/P denotes the average number of seals per page.

Set	Real		Synthetic		Total Pages
	#Pages	Avg. S/P	#Pages	Avg. S/P	
Train	800	0–2	800	≈10	1,600
Validation	100	0–2	100	≈10	200
Real Test	100	0–2	–	–	100
Synth. Test	–	–	100	≈10	100

interference, the issue becomes even more challenging, as seals contain engraved Kanji characters that closely resemble Kuzushiji characters, causing the models to mistakenly detect them as the valid characters.

During inference, the detection model outputs bounding boxes corresponding to Kuzushiji characters along with their confidence scores. In documents containing degraded backgrounds or noise interference, stains may be mistakenly detected as Kuzushiji characters with extremely low confidence scores (e.g., 0.001), as shown in Figure 2. Furthermore, characters with elongated strokes or highly cursive handwriting may be detected multiple times as separate instances, as shown by the two rightmost examples in Figure 2. The confidence scores of these duplicate detections, which typically correspond to partial character regions, are generally below 0.1. Therefore, to improve the reliability of the detection results, only bounding boxes with confidence scores higher than 0.1 are retained, improving the likelihood that each bounding box corresponds to a character instance.

3.4. Kuzushiji Document Restoration

To mitigate interference caused by red seals, we propose a training-free, color-based thresholding algorithm for document restoration. Given an input document image $I \in \mathbb{R}^{H \times W \times 3}$, let R , G , and B denote the RGB (red, green, and blue) channel intensities at each pixel. Since seal regions

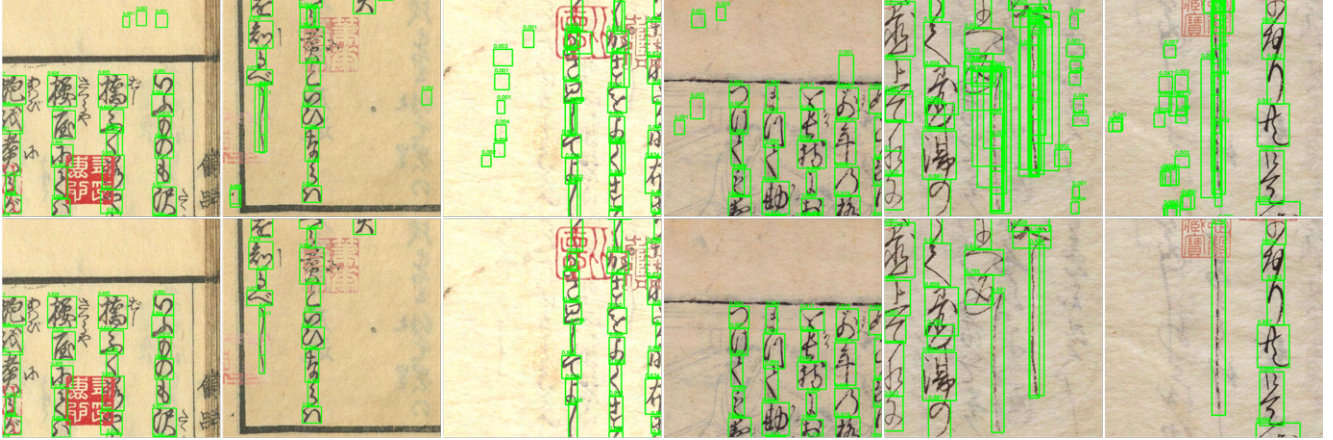


Figure 2. The top row shows low-confidence bounding boxes produced by the detection model. Stains in Japanese historical documents may cause false positives, resulting in background noise being mistakenly detected as Kuzushiji characters with confidence scores as low as 0.001. The bottom row shows the detection results with a confidence threshold of 0.1, which effectively removes most false positives.

typically exhibit stronger responses in the red channel than in the green and blue channels, they are detected using a channel-ratio-based thresholding rule. Specifically, a pixel is classified as a red seal candidate if it satisfies:

$$(R \geq \tau_r) \wedge (R \geq \tau_{rg} \cdot G) \wedge (R \geq \tau_{rb} \cdot B), \quad (1)$$

where τ_r denotes the minimum red channel intensity threshold, while τ_{rg} and τ_{rb} control the dominance of the red channel over the green and blue channels, respectively. This rule helps suppress false positives caused by non-seal reddish regions while preserving most seal candidate pixels.

The resulting binary mask M is further refined using morphological dilation to slightly expand the detected seal boundaries and compensate for color bleeding around seal edges. In this work, a 3×3 square kernel with a single dilation iteration is used by default. The detected seal regions are subsequently removed through image inpainting:

$$I_{\text{restored}} = \text{Inpaint}(I, M, \rho), \quad (2)$$

where ρ denotes the inpainting radius. We adopt Telea’s fast marching-based image inpainting method [25], which efficiently propagates surrounding image structures into the masked regions and is well suited for lightweight document restoration without requiring additional training.

3.5. Kuzushiji Character Cropping

After Kuzushiji character detection and document restoration, character-level cropping is performed based on the predicted bounding boxes for subsequent single-character classification. Specifically, the top-left coordinates (x, y) , width w , and height h of each predicted bounding box are used to locate the corresponding spatially aligned region in the restored document image. The corresponding character region is then cropped from the restored document image and used as input for character classification.

3.6. Kuzushiji Character Classification

Character classification is performed for each cropped character patch. Given the complex handwriting styles, large number of character categories, and various challenges in historical documents, including degradation, blurring, and seal interference, we directly employ the pre-trained Metom model [10] as the character classifier. This model is capable of classifying 2,703 categories of Kuzushiji characters and provides strong feature representation capabilities.

However, the original implementation of Metom supports only single-image inference. To improve inference efficiency, we extend the implementation to support batch inference for character classification. In addition, the Scaled Dot-Product Attention (SDPA) [28] acceleration mechanism is enabled to increase the throughput of large-scale character classification. Finally, the Top-5 candidate characters are retained for each input patch, while the Top-1 prediction is used as the final classification result.

3.7. Kuzushiji Character Ordering

We propose an efficient training-free Kuzushiji character ordering algorithm, as detailed in Algorithm 1 and illustrated in Figure 7. Since the ordering process relies on the spatial positions of characters, the algorithm takes the predicted bounding box information of each character as input, including x_i , y_i , w_i , and h_i , while the predicted character label \hat{c}_i is used only to produce the final text output.

First, the column list \mathcal{C} is initialized, and the center coordinates (cx_i, cy_i) of each character are computed from its bounding box. The column-center x-coordinate cx_j^{col} is maintained to represent the horizontal position of the j -th column. The characters are then ordered in descending order of cx_i and processed sequentially from right to left.

To determine whether a character belongs to an existing

Algorithm 1 Proposed character ordering algorithm

Require: Predictions $\{x_i, y_i, w_i, h_i, \hat{c}_i\}_{i=1}^N$, $\lambda = 0.8$ **Ensure:** Ordered text T

```
1:  $\mathcal{C} \leftarrow []$ ,  $cx_i \leftarrow x_i + \frac{w_i}{2}$ ,  $cy_i \leftarrow y_i + \frac{h_i}{2}$ 
2:  $\mathcal{R} \leftarrow \text{sort}(\{1, \dots, N\}, cx_i, \downarrow)$ ,  $\tau \leftarrow \lambda \cdot \frac{1}{N} \sum_{i=1}^N w_i$ 
3: for  $i \in \mathcal{R}$  do
4:   if  $\mathcal{C} = \emptyset$  then
5:      $\mathcal{C} \leftarrow \mathcal{C} ++ [\{i\}]$ ,  $cx_{\mathcal{C}_1}^{col} \leftarrow cx_i$ 
6:   else
7:      $j^* \leftarrow \arg \min_{1 \leq j \leq |\mathcal{C}|} |cx_i - cx_{j^*}^{col}|$ 
8:     if  $|cx_i - cx_{j^*}^{col}| \leq \tau$  then
9:        $\mathcal{C}_{j^*} \leftarrow \mathcal{C}_{j^*} \cup \{i\}$ 
10:       $cx_{j^*}^{col} \leftarrow \text{median}(\{cx_k \mid k \in \mathcal{C}_{j^*}\})$ 
11:    else
12:       $\mathcal{C} \leftarrow \mathcal{C} ++ [\{i\}]$ ,  $cx_{\mathcal{C}_1}^{col} \leftarrow cx_i$ 
13:    end if
14:  end if
15: end for
16:  $T \leftarrow \text{Concat}_{\mathcal{C}_j \in \mathcal{C}} (\hat{c}_k \mid k \in \text{sort}(\mathcal{C}_j, cy_k, \uparrow))$ 
17: return  $T$ 
```

column, the threshold τ is computed as λ times the average predicted character width, where $\lambda = 0.8$. When processing each character, if no column has been established, a new column is created and its center x-coordinate is initialized to the character cx_i . Otherwise, the algorithm identifies the column whose center is closest to the current character along the x-axis and computes the corresponding horizontal distance. If this distance does not exceed τ , the character is assigned to that column, and the column-center $cx_{j^*}^{col}$ is updated as the median of all cx_i values within the column. Otherwise, a new column is appended to the column list, and its center x-coordinate is initialized to the character cx_i .

After all characters have been assigned to columns, the characters within each column are ordered in ascending order of cy_i , thereby arranging them from top to bottom. Since new columns are appended to \mathcal{C} while characters are processed in descending order of cx_i , the column list naturally preserves the right-to-left reading order, eliminating the need for additional column ordering. Finally, the predicted character labels \hat{c}_i are concatenated according to the column order and the character order within each column to produce the final text output T .

4. Experiments

4.1. Dataset Correction and Reconstruction

We collected 1,000 publicly available Japanese historical document images from 13 different Japanese historical books provided by the CODH [22], covering a wide range of writing styles, document layouts, and degradations. Additional details of the dataset are provided in Section A of



Figure 3. Red and green bounding boxes indicate the annotations newly added in this work and existing annotations, respectively.

the supplementary materials. To further improve the quality and reliability of the dataset, we systematically re-examined the annotations of all 1,000 images. The results revealed that 267 document images contained missing annotations. Therefore, we manually added the missing character bounding boxes with the assistance of a Kuzushiji expert.

As shown in Figure 3, the red bounding boxes indicate the additional annotations added in this work, while the green bounding boxes represent the annotations originally included in the dataset. After correcting the annotation errors, we reconstructed the dataset. Specifically, the 1,000 images were randomly divided into training, validation, and test sets with a ratio of 8:1:1. The statistics of the reconstructed dataset are presented in Table 1.

4.2. Synthetic Test Set Construction

To evaluate robustness under severe seal interference, we constructed a synthetic test set from the real test set using high-resolution seal images distinct from those used in the SDA strategy. Examples are shown in Figure 4. Following the synthesis process described in [12], we overlaid 10 seal instances onto each document image while restricting seal overlap to at most two seals, as summarized in Table 1.

4.3. Evaluation Metrics

This work conducts both stage-wise and end-to-end evaluations of KCR. For Kuzushiji character detection, we employ standard object detection metrics [18], including the number of model parameters (Params), floating point operations (FLOPs), precision (P), recall (R), average precision at an IoU threshold of 0.50 (AP₅₀), and mean average precision over IoU thresholds ranging from 0.50 to 0.95 (AP_{50:95}). For document restoration, we use image quality metrics, including peak signal-to-noise ratio (PSNR, in dB) and structural similarity index measure (SSIM) [31]. For character classification, we adopt standard image classification met-



Figure 4. Seals in the top row are present in real historical documents, while those in the bottom row are synthetically overlaid.

rics [15], including Top-1 and Top-5 accuracy. For character ordering and end-to-end KCR, we use the character error rate (CER), a widely adopted evaluation metric in OCR. In addition, the inference speed (Speed) is measured in frames per second (FPS) for all evaluations.

4.4. Implementation Details

All experiments were conducted on Ubuntu systems equipped with an Intel Core i5-11600K CPU and either an NVIDIA RTX PRO 6000 Blackwell Max-Q GPU (96 GB) or an NVIDIA RTX A5000 GPU (24 GB). Specifically, training was performed on RTX PRO 6000, stage-wise evaluation was conducted on both GPUs, and end-to-end KCR evaluation was performed on RTX A5000. During inference, character detection and classification were executed on the GPU, while document restoration, character cropping, and character ordering were performed on the CPU.

For character detection, all models were trained for 1,000 epochs with a batch size of 16 and an input resolution of 640×640 pixels. Stochastic Gradient Descent (SGD) was used as the optimizer with an initial learning rate of 0.01. For document restoration, parameter analyses are provided in Section B of the supplementary materials. The selected parameters were $\tau_r = 90$, $(\tau_{rg}, \tau_{rb}) = (1.3, 1.3)$, and the inpainting radius $\rho = 3$. In addition, document restoration was executed on the CPU with up to six parallel workers. For character classification, inference was performed with a batch size of 1,280. For character ordering, analyses of different threshold scales λ and column-center update strategies are provided in Section C of the supplementary materials. The selected configuration used $\lambda = 0.8$ and the median as the column-center update strategy.

4.5. Kuzushiji Character Detection

We trained several state-of-the-art real-time object detectors with and without the proposed SDA strategy, and the results

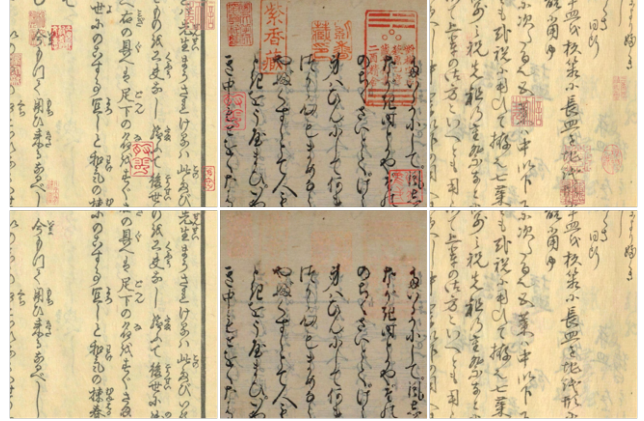


Figure 5. Qualitative comparison between the original historical document images (top row) and the restored results of synthetic historical document images (bottom row) by our proposed method.

are presented in Table 2. Since the SDA strategy only increases the amount of training data without modifying the model architecture, models trained with the SDA strategy retain the same Params and FLOPs as their counterparts trained without the SDA strategy, while maintaining comparable inference speeds in terms of FPS. The experimental results show that, for the same model architecture, the SDA strategy improves performance across all evaluation metrics, including P, R, AP₅₀, and AP_{50:95}. Among all evaluated models, YOLO11-L trained with the SDA strategy achieved the best overall performance and was therefore selected as the character detector in the proposed framework.

Notably, the SDA strategy doubles the amount of training data, resulting in a longer training time per epoch, as reflected by T/E in Table 2. However, models trained with the SDA strategy generally achieve their best performance at earlier epochs. Therefore, the additional training overhead introduced by the SDA strategy can be partially mitigated in practice through the use of an early stopping mechanism.

4.6. Kuzushiji Character Classification

To evaluate the effectiveness of document restoration for character classification, we conducted an ablation study. To ensure a fair comparison, all ablation experiments were based on bounding boxes detected by the YOLO11-L model trained with the SDA strategy. Specifically, the baseline method employed Metom [10] to classify character patches cropped from the original document images, while the proposed method first restored the document images and then classified character patches cropped from the restored images using the same classifier. A qualitative comparison between the original and restored document images is presented in Figure 5. Additional qualitative results are provided in Section D of the supplementary materials.

The quantitative results of the ablation study are reported

Table 2. Quantitative comparison of different detectors for Kuzushiji character detection. The average training time per epoch (T/E) was measured on an NVIDIA RTX PRO 6000 Blackwell GPU (96 GB), while Epoch indicates the epoch achieving the best performance.

Method	Efficiency					Real Test Set				Synthetic Test Set			
	Params	FLOPs	Epoch	T/E	Speed	P	R	AP ₅₀	AP _{50:95}	P	R	AP ₅₀	AP _{50:95}
RT-DETR-R50 [32]	41.94M	125.6G	988	30.06	11.76	93.6	90.3	94.3	70.1	91.1	85.0	90.8	65.2
+ SDA (Ours)	41.94M	125.6G	736	56.13	11.74	96.3	93.1	95.6	70.5	95.6	92.2	95.0	68.4
YOLOv9-C [30]	25.32M	102.3G	621	10.97	10.72	97.8	93.3	96.5	82.6	95.8	88.0	93.8	77.8
+ SDA (Ours)	25.32M	102.3G	406	19.35	10.45	97.8	93.7	96.5	82.9	97.7	93.1	96.4	81.7
YOLOv10-L [29]	25.77M	127.2G	781	12.60	11.03	98.0	92.0	96.3	82.5	95.7	87.2	93.6	77.5
+ SDA (Ours)	25.77M	127.2G	415	23.87	11.05	98.3	92.7	96.5	83.1	98.0	92.4	96.4	81.8
YOLO11-L [11]	25.28M	86.6G	549	10.89	10.52	97.9	92.9	96.4	83.0	95.8	87.7	93.7	78.1
+ SDA (Ours)	25.28M	86.6G	440	18.81	10.52	98.0	93.5	96.5	83.3	97.6	93.1	96.4	82.1
YOLOv12-L [26]	26.39M	82.1G	909	16.86	10.53	97.4	93.4	96.4	82.7	95.5	87.7	93.7	77.6
+ SDA (Ours)	26.39M	82.1G	408	31.69	10.38	97.5	93.6	96.5	83.2	97.5	92.8	96.3	81.9

Table 3. Ablation of document restoration for character classification on the synthetic test set. Speed@A and Speed@P represent the inference speeds on A5000 and PRO 6000 GPUs, respectively.

Method	Top-1 Acc.	Top-5 Acc.	Speed@A	Speed@P
Baseline	94.22	97.64	1.19	4.76
+ Rest.	95.66	98.62	1.19	4.76

in Table 3. Across 18,350 character patches detected on the synthetic test set, the baseline achieved Top-1 and Top-5 accuracy of 94.22% and 97.64%, respectively. After incorporating document restoration, the proposed method achieved superior classification performance, improving Top-1 accuracy to 95.66% and Top-5 accuracy to 98.62%.

Since document restoration and character detection are executed in parallel on the CPU and GPU, respectively, and the restoration process requires less processing time than detection, its execution is effectively hidden by detection. As shown in Table 3, incorporating document restoration does not introduce additional latency to classification.

4.7. Kuzushiji Character Ordering

Among existing methods, only NDLkotenOCR [21] explicitly reports the use of LightGBM [13] for character ordering in its Version 1 model, while the ordering strategies adopted by other methods are not described in detail. Therefore, we selected LightGBM as the baseline for quantitative comparison with the proposed method. To ensure a fair comparison, both methods performed character ordering using the same character classification results obtained by the proposed method in Table 3. Since LightGBM requires supervised training, it was trained using all character annotations provided by the CODH [22]. Both methods were evaluated on a CPU, and the results are presented in Table 4.

Table 4. Quantitative comparison between the baseline and the proposed methods for character ordering on the synthetic test set.

Method	Training	CER	Speed@CPU
LightGBM [13]	Yes	21.25	15.38
Ours	No	13.67	419.27

Compared with the baseline, the proposed method is training-free and achieves superior performance, reducing the CER from 21.25% to 13.67% while increasing the inference speed on the CPU from 15.38 FPS to 419.27 FPS.

4.8. Visualization

Figure 6 presents visualization results obtained by projecting the recognized modern Japanese characters onto the original document images. We set the bounding boxes to green and display the recognized characters at a font size of 64 pixels, facilitating intuitive interpretation of Japanese historical documents by the general public.

4.9. End-to-end KCR Evaluation

We compare the proposed method with the conventional baseline and existing methods for end-to-end KCR. Since miwo [4], KuroNet [7, 16], and Komonjo Camera [27] are not open source, fair comparisons could not be conducted. For NDLkotenOCR [21], we used the publicly released Version 3 pre-trained model, which was trained on 28,134 images and 456,746 text lines. For NDLkotenOCR-Lite [1], we also used the publicly released pre-trained model, which was trained on 502,065 text lines. The conventional baseline consists of the YOLO11-L model [11] trained without the SDA strategy for character detection, pre-trained Metom model [10] for character classification without document restoration, and LightGBM [13] for character ordering.



Figure 6. Visualization of the input images (top row) and the recognition results projected onto the corresponding images (bottom row).

Table 5. Quantitative comparison of different methods for end-to-end KCR on an NVIDIA RTX A5000 GPU and an Intel Core i5-11600K CPU. Mem. denotes peak GPU memory usage (GB).

Method	Real Test Set		Synth. Test Set			
	CER	Speed	Mem.	CER	Speed	Mem.
NDL-Lite [1]	21.76	0.40	–	47.82	0.35	–
NDL [21]	12.73	0.51	7.52	23.43	0.37	16.20
Baseline	19.86	0.37	2.81	27.41	0.35	2.81
Ours	11.98	0.41	2.81	13.67	0.39	2.81

The quantitative results are presented in Table 5. Note that the inference speed of NDLkotenOCR-Lite was measured on the CPU, while the other methods were evaluated on the GPU. The proposed method achieved the lowest CER on both the real and synthetic test sets, reaching 11.98% and 13.67%, respectively. In terms of inference speed, NDLkotenOCR [21] and the proposed method achieved the highest speeds on the real and synthetic test sets, with 0.51 FPS and 0.39 FPS, respectively. Regarding GPU memory usage, the baseline and the proposed method require only 2.81 GB of peak GPU memory, substantially less than NDLkotenOCR, while NDLkotenOCR-Lite [1] did not consume GPU memory. All methods exhibited lower inference speeds on the synthetic test set due to the increased computational cost caused by more severe seal interference.

5. Discussion

One major factor causing the CER of the proposed method to remain above 10% in end-to-end KCR is the presence of a limited number of documents with complex layouts, in which the content of a single page is divided into multiple independent text blocks. As shown by the right example in Figure 7, the proposed method (shown in green) performs character ordering under a regular layout assumption, while

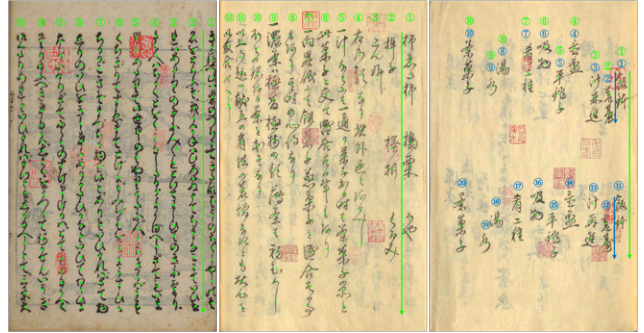


Figure 7. The left two examples show regular layouts where our method correctly orders the characters, while the right example presents a complex layout that leads to ordering errors.

the page contains two independent text blocks that should be ordered separately (shown in blue). Such layouts are relatively uncommon in Japanese historical documents and exhibit little structural consistency, making them difficult to process effectively using simple rule-based methods.

Nevertheless, the three rightmost examples in Figure 6 show that the visualization results can still assist users in interpreting documents with complex layouts.

6. Conclusion

This work proposes Seal-Robust KCR, a robust Kuzushiji character recognition framework for Japanese historical documents affected by seal interference. Experimental results demonstrate that the proposed framework improves recognition performance on the real and synthetic test sets.

Although this work was evaluated on a limited-scale dataset, the results demonstrate the effectiveness and robustness of the proposed framework. In future work, we plan to collect more high-quality data, further improve the processing of complex layouts, and deploy the proposed method through web and mobile applications for public use.

References

- [1] Toru Aoike. Development of NDLkotenOCR-Lite, a lightweight OCR that runs at high speed in a CPU environment. In *IPSJ SIG-CH*, pages 181–186, 2024. 1, 2, 7, 8
- [2] Darwin Bautista and Rowel Atienza. Scene Text Recognition with Permuted Autoregressive Sequence Models. In *ECCV*, pages 178–196, 2022. 2
- [3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE TPAMI*, 43(5):1483–1498, 2019. 2
- [4] Tarin Clanuwat and Asanobu Kitamoto. “miwo” AI Kuzushiji Recognition Application for Document Examination. In *IPSJ SIG-CH*, pages 302–309, 2021. 1, 2, 7
- [5] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep Learning for Classical Japanese Literature. In *NeurIPS Workshop*, 2018. 1
- [6] Tarin Clanuwat, Alex Lamb, and Asanobu Kitamoto. End-to-End Pre-Modern Japanese Character (Kuzushiji) Spotting with Deep Learning. In *IPSJ SIG-CH*, pages 15–20, 2018. 2
- [7] Tarin Clanuwat, Alex Lamb, and Asanobu Kitamoto. KuroNet: Pre-Modern Japanese Kuzushiji Character Recognition with Deep Learning. In *ICDAR*, pages 607–614, 2019. 1, 2, 7
- [8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021. 2
- [9] Yuta Hashimoto, Yoichi Iikura, Yukio Hisada, SungKook Kang, Tomoyo Arisawa, et al. The Kuzushiji Project: Developing a Mobile Learning Application for Reading Early Modern Japanese Texts. *DHQ*, 11(1), 2017. 1
- [10] Yuki Imajuku and Tarin Clanuwat. Metom. <https://huggingface.co/SakanaAI/Metom>, 2024. 2, 4, 6, 7
- [11] Glenn Jocher, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO11. <https://yolol11.com/>, 2024. 7
- [12] Rui-Yang Ju, Kohei Yamashita, Hirotaka Kameko, and Shinsuke Mori. DKDS: A Benchmark Dataset of Degraded Kuzushiji Documents with Seals for Detection and Binarization. *IJDAR*, 2026. 1, 2, 5
- [13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *NeurIPS*, 30, 2017. 2, 7
- [14] Asanobu Kitamoto, Tarin Clanuwat, and Mikel Bober-Irizar. Kaggle Kuzushiji Recognition Competition: Challenges of Hosting a World-Wide Competition in the Digital Humanities. *Journal of JSAI*, 35(3):366, 2020. 2
- [15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *NeurIPS*, 25, 2012. 6
- [16] Alex Lamb, Tarin Clanuwat, and Asanobu Kitamoto. KuroNet: Regularized Residual U-Nets for End-to-End Kuzushiji Character Recognition. *SN Computer Science*, 1(3):1–15, 2020. 1, 2, 7
- [17] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models. In *AAAI*, pages 13094–13102, 2023. 2
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, pages 740–755, 2014. 5
- [19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*, 2019. 2
- [20] Chengqi Lyu, Wenwei Zhang, Haian Huang, Yue Zhou, Yudong Wang, Yanyi Liu, Shilong Zhang, and Kai Chen. RTMDet: An Empirical Study of Designing Real-Time Object Detectors. *arXiv preprint arXiv:2212.07784*, 2022. 2
- [21] Kiyonori Nagasaki, Ikki Ohmukai, and Masahiro Shimoda. Enhancing Digital Scholarly Editions Through Advanced OCR Accuracy: A New Methodological Development. In *IPSJ SIG-CH*, pages 177–182, 2023. 1, 2, 7, 8
- [22] National Institute of Japanese Literature. Japanese Classical Book Kuzushiji Dataset (日本古典籍くずし字データセット), 2016. 5, 7
- [23] National Institute of Japanese Literature. What is Kuzushiji? (くずし字って何?). <https://www.nijl.ac.jp/koten/kuzushiji/post.html>, 2018. 1
- [24] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *MICCAI*, pages 234–241, 2015. 2
- [25] Alexandru Telea. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools*, 9(1): 23–34, 2004. 4
- [26] Yunjie Tian, Qixiang Ye, and David Doermann. YOLOv12: Attention-Centric Real-Time Object Detectors. *NeurIPS*, 38: 78433–78457, 2026. 7
- [27] TOPPAN Inc. Fuminoha: Komonjo Camera (古文書カメラ). <https://camera.fuminoha.jp>, 2023. 1, 2, 7
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *NeurIPS*, 30, 2017. 4
- [29] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. YOLOv10: Real-Time End-to-End Object Detection. In *NeurIPS*, pages 107984–108011, 2024. 7
- [30] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. In *ECCV*, pages 1–21, 2024. 7
- [31] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE TIP*, 13(4):600–612, 2004. 5
- [32] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs Beat YOLOs on Real-time Object Detection. In *CVPR*, pages 16965–16974, 2024. 7