IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 20 June 2022, accepted 4 July 2022, date of publication 15 July 2022, date of current version 25 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3191429

RESEARCH ARTICLE

DKNet: Deep Kuzushiji Characters Recognition Network

DILBAG SINGH[1], (Senior Member, IEEE), C. V. ARAVINDA[2], MANJIT KAUR[1], (Senior Member, IEEE), MENG LIN[3], JYOTHI SHETTY[2], VIKRAM RAJU REDDICHERLA[2], AND HEUNG-NO LEE[1], (Senior Member, IEEE)

[1]School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
[2]N.M.A.M. Institute of Technology, Nitte, Karkala 574110, India
[3]Department of Electronic and Computer Engineering, The Graduate School of Science and Engineering, Ritsumeikan University, Kusatsu Shiga 525-8577, Japan

Corresponding author: Heung-No Lee (heungno@gist.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant by the Korean Government through MSIT (Artificial Intelligence Graduate School Program—GIST), under Grant 2019-0-01842; and in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2021-0-01835.

ABSTRACT Kuzushiji, a cursive writing style, had been extensively utilized in Japan for over a thousand years starting from the 8^{th} century. In 1900, Kuzushiji was not included in regular school curricula due to the change in the Japanese writing system. Nowadays Japanese natives are unable to read historical books that were written using Kuzushiji language. Therefore, libraries and museums have decided to build digital copies of the documents and books that were written in Kuzushiji language. Due to a limited number of trained experts, researchers have utilized machine and deep learning models to convert historical documents and books into a modern script that can be easily read by human beings. However, the existing deep learning techniques suffer from over-fitting and gradient vanishing problems. To overcome these problems, an efficient deep Kuzushiji characters recognition network (DKNet) is proposed. Initially, to remove noise from the training images, a trilateral joint filter is applied. Contrast limited adaptive histogram equalization (CLAHE) is then applied to enhance the visibility of filtered images. Thereafter, a pre-trained MobileNet is utilized to extract the features of Kuzushiji characters. MobileNet's final layers are removed, including the fully connected layer and softmax. The flatten layer is then applied to the input. A fully connected classification layer is then used with Rectified linear units (ReLUs) and dropouts. Dropouts are used to generalize the model, thus preventing the over-fitting problem. Finally, the softmax activation function is employed to provide the recognition results. To test the proposed model, actual documents are first segmented by using the proposed Maximally stable extremal regions (MSERs) and convexhull-based segmentation approach. Segmented characters are then recognized using the trained DKNet. Extensive comparative analyses reveal that DKNet achieves better performance than the competitive models in terms of various performance metrics. An efficient Application Programming Interface (API) is also designed for Japanese Kuzushiji ancient heritage character recognition to help the end-users.

INDEX TERMS Maximally stable extremal regions, deep learning models, Kuzushiji character recognition, MobileNet, InceptionNet, ResNet.

I. INTRODUCTION

Kuzushiji, a cursive writing style, had been extensively utilized in Japan for over a thousand years starting from the 8^{th} century. But, in 1900, Kuzushiji was not included in regular school curricula due to the change in Japanese writing system. Nowadays Japanese natives are unable to read historical books that were written using Kuzushiji language. Therefore, libraries and museums have decided to build digital copies of the documents and books that were written

The associate editor coordinating the review of this manuscript and approving it for publication was Donato Impedovo.

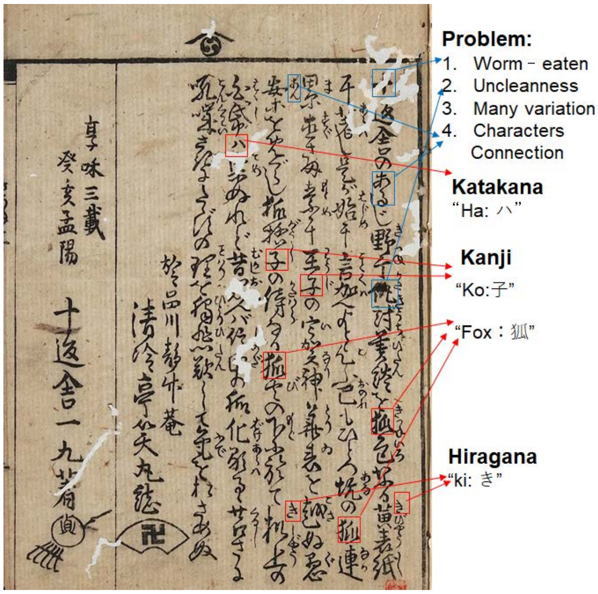


FIGURE 1. Typical page of a classical Japanese book.

in Kuzushiji language. But, due to a limited number of trained experts, researchers have utilized machine and deep learning models to convert historical documents and books into a modern script that can be easily read by human beings [1], [2].

To convert a single document or book page at a time, it is required to efficiently segment each character at a time [3]–[5]. In some cases, text detection becomes even more important than just text extraction until it can detect the shapes of regular or irregular shapes in the given images. Adding two Chinese syllabaries and a set of Chinese logograms into a complex logosyllabary system is the Japanese writing pattern, which is a fascinating study of tradition and innovation. The Japanese language uses incurved verbs and postpositions. It utilizes suffixes and particles to create words and clauses in sentences. Chinese characters (kanji) have been extensively utilized in a long history of use dating back centuries. The Japanese also used some Chinese characters with sound values for grammatical purposes. These characters were not instantly interpreted as phonetic or logographic representations of meaning [6], [7].

There were many variants of Kuzushiji characters and in some documents and books, these characters were connected with each other. Aging causes the documents to become unclean and some have been infected with worms [8]. Figure 1 shows a page of an early Japanese book that contains Kuzushiji characters.

- **Hiragana:** Manyoshu describes a system of writing native words in ancient Japan that uses manyogana. In Figure 1, Ki illustrates how these signs were reduced and simplified into sogana and finally to hiragana.
- **Katakana:** According to the Chinese Buddhist scriptures, the word kanji is an essential pronunciation aid. This character accumulated as grammatical suffixes, particles, and postpositions. But the literal meaning of kanji did not change. In Figure 1, Ha indicates that katakana is widely used to write non-Chinese words.
- **Allophones:** In Figure 1, Ko and Fox illustrate what may appear to be allophones in the Japanese syllabograms. People speaking the same language perceive these sounds as similar sounds.

A. CHALLENGES

Figure 1 demonstrates challenges and problems associated with Kuzushiji character recognition.

1) **Modern and Kuzushiji characters:** Some Kuzushiji characters are written in such a way that only context, particularly the identity of preceding characters can be used to recognize them. A simple line, for example, might signify a variety of characters based on the characters used before it. KuroNet can disambiguate by using the surrounding characters since it employs both local and global contexts. However, for efficient classification of characters, it is required to segment each character prior to the recognition process.
2) **Variation:** Although Kuzushiji has a huge number of characters (e.g., used dataset comprises 4328), their distribution is long-tailed. A significant portion of the characters only appears once or twice in the sample. This is owing to the Japanese language's peculiar structure that has two sorts of character sets, i.e., a phonetic alphabet (with simple and extremely frequent characters) and non-phonetic Kanji characters. Kanji is made up of both common and uncommon characters, some are very intricate and others are merely one or two straight lines.
3) **Hentaigana characters:** One feature of classical Hiragana or Hentaigana (the word literally translates to "Character Variations") that has a significant impact on recognition. Many characters that can be written exclusively in modern Japanese can be written in a variety of ways in pre-modern Japanese language [35].
4) **Aging processing:** It is hard to recognize the unclean characters due to infestation by worms.

Recently, researchers have utilized various deep learning models to improve the performance of Kuzushiji character recognition. Some of the popular models are as LeNet [16], AlexNet [17], GoogLeNet [18], VGG16, VGG19 [19], ResNet152V2 [21], InceptionNet [24], InceptionResNetV2 [20], XCeptionNet [25], and MobileNet [26]. But most of these models suffer from over-fitting and gradient vanishing problems. Additionally, an efficient approach is required to segment the characters prior to the recognition process.

B. CONTRIBUTIONS

To overcome the aforementioned problems, an efficient deep Kuzushiji characters recognition network (DKNet) is proposed. The main contributions of this paper are as follows:

- To overcome the over-fitting and gradient vanishing problems, an efficient deep Kuzushiji characters recognition network (DKNet) is proposed.
- To remove noise from the training images, a trilateral joint filter is employed. Contrast limited adaptive histogram equalization (CLAHE) is also applied to enhance the visibility of filtered images.
- In DKNet, a MobileNet's final layers are removed. The flatten layer is then applied to the input. Thereafter, a fully connected classification layer is implemented with Rectified linear units (ReLUs) and dropouts. Finally, the softmax activation function is employed to provide the recognition results.
- To test the proposed model, actual documents are first segmented by using the proposed Maximally stable extremal regions MSERs) and convexhull-based segmentation approach. Segmented characters are then recognized using the trained DKNet.
- An efficient Application Programming Interface (API) is also designed for Japanese Kuzushiji ancient heritage characters recognition to help the end-users.

The remaining paper is organized as follows. Section II discusses the literature review. Section III discusses the proposed DKNet. Section IV presents the proposed MSER and convexhull-based segmentation approach. Experimental results are presented in Section V. Section VI concludes the paper.

II. LITERATURE SURVEY

In [9], an adaptive neural network was utilized for character recognition by considering features of directional elements as feature vectors. In [11], line-by-line recognition was used to classify characters according to the attention mechanism. Lyu *et al.* [12] detected the text lines prior to the recognization process. Image processing operations and ARU-Net were used to identify text lines. It has achieved precision, recall, and F-score values as 95.2%,, −98.3%, and 96.6%, respectively. Le *et al.* [13] designed a Recognition system using an attention-based encoder-decoder (RAED). It re-process data by detecting the line's start and recognizing each character until the line has been completed.

Li *et al.* [10] utilized three and five-cross point non-directed graphs of Oracle-bone inscriptions (OBIs) such as blocks and holes. Another recognition method was also used (refer to [14]). Li *et al.* [15] developed a DNA-based model for detection of OBIs. However, Meng *et al.* [5] and Liu *et al.* [28] proved that all OBIs needed to be cut in advance which is not convenient for practical applications. Meng *et al.* [5] extended the Single shot multibox detector (SSD) for OBI detection and achieved precision, recall, and F values as 98%, 83%, and 88%, respectively. Clanuwat *et al.* [40] recognized the Kuzushiji characters using a deep learning model. Residual U-Net architecture (RU-Net) was applied to recognize the entire page with identification of all characters given on it. In [41], Japanese historical text was recognized by attention based encoder-decoder (AED) model. Multiscale features were extracted using a dense convolutional neural network. The target text was generated using Long short-term memory (LSTM) decoder.

In [45], handwritten Japanese text was recognized using Attention augmented convolutional recurrent network (AACRN). The performance of the model was evaluated on Kuzushiji and TUAT Kondate datasets. In [46], the Deep learning model(DLM) was applied to recognize the Kuzushiji characters. Ueki *et al.* [47] proposed a recognition system for the Kuzushiji characters by utilizing Multiple softmax outputs (MSOs). In [48], a 2-dimensional Context box proposal network (2DCbpn) was proposed to identify the Kuzushiji Characters. Features were extracted using VGG16 and then features were processed through a 2-D context. Thereafter, horizontal and vertical contexts were explored using Bidirectional LSTM (BLSTM).

Bing *et al.* [42] proposed an method to understand the ancient Japanese books using conventional image processing and ARU-Net. These methods were used for the detection of frames and segmentation od lines. Character recognition was done using by applying AlexNet. Nguyen *et al.* [43] used Gaussian mixture mode, LSTM, and CNN to recognize the Japanese kanji characters. Septiana *et al.* [44] proposed template matching algorithm to recognize the handwritten Japanese characters. It achieved the accuracy of 89.8%.

Horie *et al.* [6] recognized OBIs using deep learning models. While they achieved good accuracy, the experimental data were drawn from recent research, not the original rubbings. OBIs are a kind of hieroglyphs that were the original version of the modern Chinese characters that are widely used in China, Japan, and other Asian countries. One OBI character is one hieroglyph graph, with a meaning corresponding to one thing or one object. Lyu *et al.* [12] developed a Japanese historical character recognition system for the reading support system for historical documents by using the directional elements as feature vectors and designing a modular neural network as a pattern classification model.

Nguyen *et al.* [36] segmented the image into patches for specific characters and then classified each patch separately. This strategy is computationally better, but due to the contextual nature of many characters, it is unsuited for the overall Kuzushiji recognition job. Le *et al.* created datasets with separately segmented Kazushi characters but did not take into account the Kuzushiji recognition [13].

Unger [23] considered several characters that can be exclusively rewritten for current Japanese in different ways in pre-modern Japanese scripts. It seems to be one element of classical Hiragana or Hentaigana (meaning "Character Variations") that has a considerable impact on recognition. Japanese MNIST dataset [13] is significantly more difficult than the original MNIST dataset [16]. Because many pre-modern Japanese characters can be written in numerous ways. Therefore, there is a need to develop good models that must be able to reflect the multi-modal distribution of each class.

III. DEEP KUZUSHIJI CHARACTERS RECOGNITION NETWORK (DKNet)

Deep neural networks can be made lightweight using pre-trained models. Pre-trained models can reduce the number of parameters significantly when compared to networks with regular convolutions. To train the proposed model, MobileNet is used as a transfer learning model. MobileNet was designed by Google and it is an open-sourced model. This section initially briefly discusses various pre-trained models. Finally, the proposed deep Kuzushiji characters recognition network (DKNet) is discussed.

A. PRE-TRAINED LEARNING MODELS

This section discusses various pre-trained learning models which were trained on the ImageNet dataset.

1) RESIDUAL NEURAL NETWORK (ResNet)

It was presented at ILSVRC 2015. This architecture utilized the concept of skip connections. It comes up with 152 layers that allow us to train very deep neural networks without suffering from the gradient vanishing problem. Skipping connections are also called gated-recurrent units and are similar to recurrent neural networks based on recent successes [11], [24].

2) INCEPTION

The main aim of this architecture is to reduce the computational workload of deep neural networks. Originally proposed by GoogleNet, this model architecture was later applied to both Inception-V2 and Inception-V3. It can manipulate similar inputs in parallel to achieve better convergence [21], [25].

3) MobileNet

MobileNet is a well-known pre-trained model. It is an open-sourced model developed by Google as shown in the Table 1. To develop a lightweight model, it utilizes depth-wise separable convolutions. It minimizes the number of attributes than the network with regular convolutions with similar depth. Thus, it is a lightweight deep learning model [23], [26]. Two different hyper-parameters are used that can efficiently trade-off between accuracy and latency.

4) XCeptionNet

It is an extension of the inception model which can be viewed as an intermediate step between regular convolutions and depth-wise separable convolutions [22].

5) LeNet

It has five convolution layers. 32 images having 32×32 size for improving the classification performance. However, it has relatively few layers and simple architecture. LeNet cannot work properly for higher-resolution images.

TABLE 1. Mobile-net model architecture.

Type	Filter	Input
Conv/S2	3×3×3×32	224×224×3
Conv/dw/s1	3×3×32dw	112×112×32
Conv/s1	1×1×32×64dw	112×112×32
Conv/dw/s2	3×3×64dw	112×112×64
Conv/s1	1×1×128	56×56×64
Conv/s1	3×3×128dw	56×56×128
Conv/s1	1×1×128dw	56×56×128
Conv/dw/s2	3×3×128dw	56×56×128
Conv/s1	1×1×128×256	28×28×128
Conv/dw/s1	3×3×256dw	28×28×256
Conv/s1	1×1×256×512	14×14×256
5×Conv/s1/dw	3×3×512	14×14×512
Avgpool/s1	Pool×7×7	7×7×1024
FC/s1	1024×1000	1×1×1024
Softmax/s1	Classifier	1×1×1000

6) VGG

This was proposed by ILSVRC in 2014. It consists of 16 convolutional layers for uniform architecture. It is similar to AlexNet and has many filters and 138 million parameters. Hence, it has more computation power. Later, VGG-16 is extended to VGG-19 layers.

7) GoogLeNet

It achieved an error rate close to the human level in 2014. It consists of 22 layers with 224×224 sized images. The parameters are reduced from 60 million to 4 million [15].

8) InceptionNet

InceptionNet is created to reduce the computational burden of deep neural networks while maintaining performance. The idea is firstly proposed in GoogLeNet and then extended to Inception-V2, V3, and Inception-Res (Inception-V4). The key idea of InceptionNet is to compute multiple transformations over the same input map in parallel and provide the results into a single output. It has a compressed version of the spatial information that reduces the computation cost [16].

B. JOINT TRILATERAL FILTER

A joint trilateral filter is a well-known edge-preserving filter used to filer the Kuzushiji images. It considers a guided image G_{im} to filter the input image. Generally, the actual image I_m acts as a guided image. It can be implemented as:

$$J_{tf}(I_m) = \frac{\sum_{q\in\Omega} \rho^{pq}(G_{im}) \times I_q \times \sigma^2(I_q, G_{im})}{\sum_{q\in\Omega} \rho^{pq}(G_{im})} \tag{1}$$

Here, Ω is a kernel window at coordinate k dependent on the bilateral filter [37], [38].

A kernel weight, i.e., $\rho^{pq}(G_{im})$ can be computed as:

$$\rho^{pq}(G_{im}) = \frac{1}{|n|^2} \sum_{n:(p,q)\in\Omega} \left(1 + \frac{\left(G_{imp} - \mu_n\right)(G_{imq} - \mu_n)}{\sigma_n^2 + \epsilon}\right) \tag{2}$$

where $|n|$ shows pixels in window. μ_n and σ_n^2 are average and variance of G_{im} in local Ω. The weight associated to pixel q is maximum, if G_{imp} and G_{imq} are on similar sides of an edge, otherwise, a minimum weight is assigned to q.

C. CONTRAST LIMITED ADAPTIVE HISTOGRAM EQUALIZATION

The precision of Kuzushiji recognition is generally low if the images have poor visibility. The most of the enhancement methods are not effective for uneven lightening circumstances due to over-enhancement of images [39]. Therefore, CLAHE is used that can prevent excessive amplification of noise by limiting the contrast. It can be evaluated as:

$$f(D) = (1 - \omega y)((1 - \omega x)f_{ul}(D) + \omega x f_{bl}(D)) + \omega y((1 - \omega x)f_{ur}(D) + \omega x f_{br}(D)) \quad (3)$$

The mapping of four adjacent values of the histogram cumulative distribution function to pixel are desirable for every pixel. ωx and ωy indicate the distance among center of the left upper mask and pixels. $f()$ is cumulative distribution function. f_{br}, f_{bl}, f_{ul}, and f_{ur} represent below right, below left, upper left and upper right values in current patch, respectively. D represents the pixel coordinates.

D. PROPOSED DKNet

Deep neural networks can be made lightweight using MobileNet's depth-separable convolutions. It reduces the number of parameters significantly when compared with networks with regular convolutions. Figure 2 shows the proposed DKNet model. Initially, joint trilateral filter is used to remove the noise from training images. Thereafter, contrast limited adaptive histogram equalization (CLAHE) is used to enhance the visibility of filtered images. Thereafter, MobileNet is used to extract the potential filters. Final layers of MobileNet, i.e., fully connected layer and softmax are removed. Flatten layer is then used to flatten the input. Thereafter, a fully connected classification layer is used. It utilizes two fully connected layers with a Rectified linear unit (ReLU) and dropouts are used. Dropouts are used to generalized the model, thus can prevent the over-fitting problem. Finally, the softmax activation function is used to provide the recognition results.

E. VALIDATION ON SELECTED DOCUMENTS

The trained DKNet model is tested on Kuzushiji documents using the following steps.

1) Initially, MSER and convexhull are used to extract the characters.
2) Apply joint trilateral filter on each extracted character image.
3) Apply CLAHE on the filtered character image.
4) Use trained DKNet model to recognize the character.

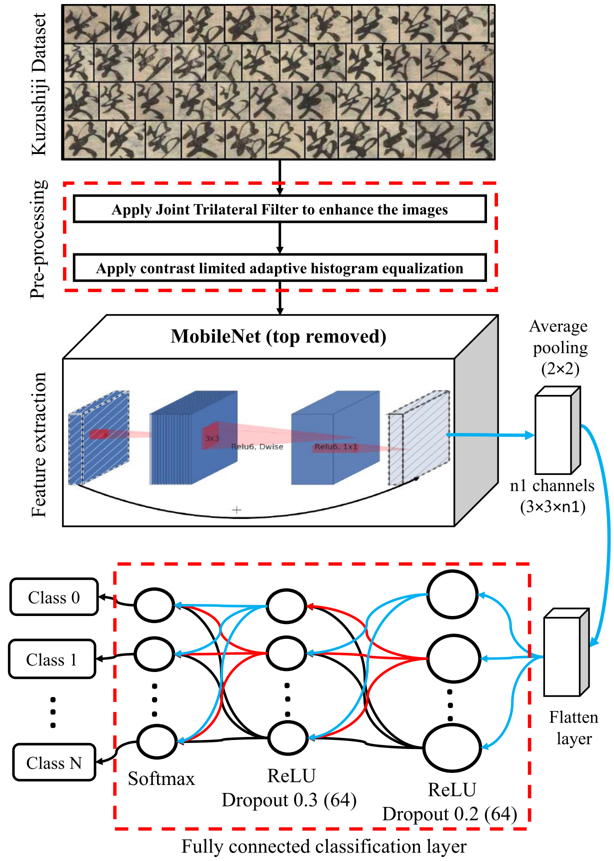


FIGURE 2. Diagrammatic flow of the proposed DKNet model.

IV. PROPOSED SEGMENTATION APPROACH FOR KUZUSHIJI CHARACTERS

This section discusses the proposed segmentation approach for Kuzushiji characters. Maximally stable extremal regions (MSER) and convexhull approaches are used to segment the characters from a given document.

Like a Scale-invariant feature detector (SIFT), MSER is a feature detector. It decomposes a given document to several co-variant regions, i.e., stable connected components or so-called MSERs. Figure 3 shows the diagrammatic flow of MSER technique.

To identify individual MSERs by characteristics of the Kuzushiji sample, MSER is applied to each document. To develop this pattern, binarization is achieved using OTSU threshold. The Pixel region of an area control is obtained by calculating the pixels region and constructing an ellipse using MinDiversity, MaxVariation, and Delta techniques. The contour is drawn around the arbitrary shape of the image using an elliptical frame that transposes the row index into a column index and vice versa. Based on a region seed, it returns a list of pixels within the region.

A. REGION OF GEOMETRIC PROPERTIES

The process of character detection and recognition can be achieved using following steps.

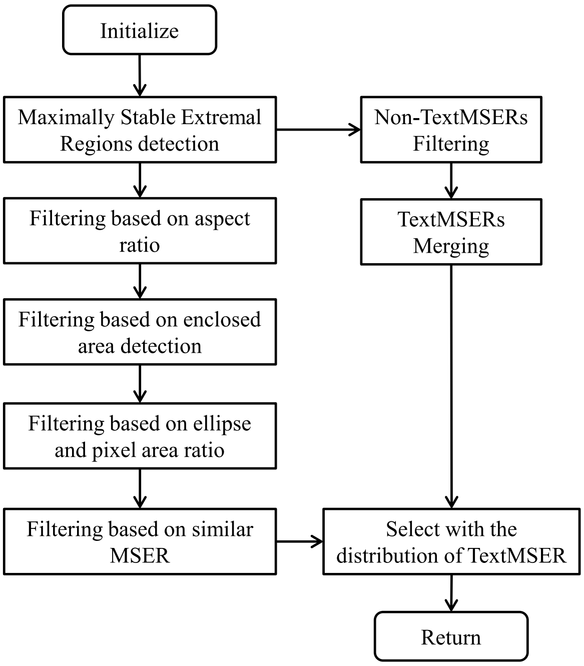


FIGURE 3. Diagrammatic flow of maximally stable extremal regions technique.

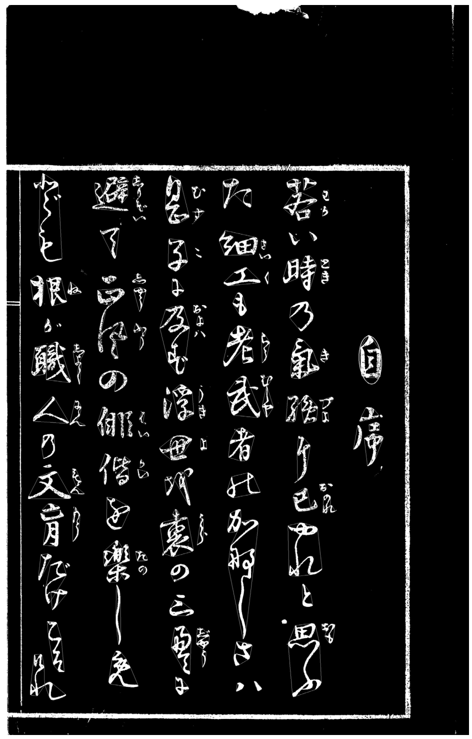


FIGURE 4. Region of image on which geometric properties are calculated.

- **Pre-processing:** Various conversion processes are used to convert RGB images to binary as shown in Figure 4.
- **Process:** An image was processed with each MSER algorithm as feature detector.
- **Properties:** Geometric attributes are employed to discriminate between non-text regions and text regions in the first pass for deleting the non-text regions from the obtained features (see Figure 5(a)).
- **Text regions:** To compute the remaining text area, the overlapping bounding boxes of these regions are combined to generate a bounded text region (see Figures 5(b) and 5(c)).
- **Detected Image:** The identified text bounding boxes in an image are examined using deep learning models to evaluate the actual and the predicted text.

- **Aspect ratio:** The bounding box ratio between width and height of the character is calculated using aspect ratio.
- **Eccentricity:** A given character's circularity could be found using Eccentricity.
- **Extent:** To identify a region for character encoding, the rectangle enclosing text is determined to be a given value using Extent.
- **Solidity:** The convex hull area ratio is calculated for a given region of a character based on the pixels in the convex hull area.

The aspect ratio, eccentricity, extent, and solidity values are set to be 0.48130841121495327, 0.8839539479029316, 0.8026494873423464, and 0.07339475469138594, respectively.

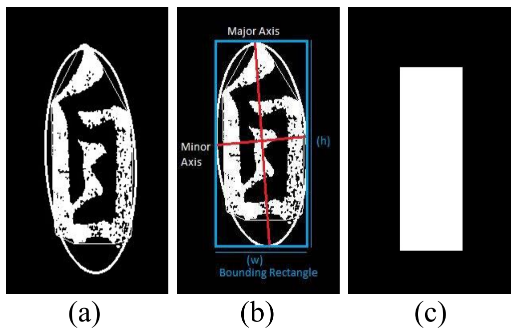


FIGURE 5. Geometric analysis using MSERs: (a) visualization of geometric properties, (b) rectangle image mask, and (c) sample image to calculate geometric properties using MSERs.

MSER calculates the mean square error of the pixels of an image over its background that have a constant intensity over its background. Text characters are assumed to have a comparable intensity and color contrast to their background. Some characters can be ignored if these characters have a small font size or represent the font's pronunciation.

B. PRE-PROCESSING OF MSER

- Initially, white to black intensity thresholds are swept by using a simple luminance thresholding algorithm.
- External regions can be calculated as

$$S_r = p|I(p) > I(q)\forall_p \in S_r, \forall_q \in B(S_r) \tag{4}$$

The gray-scale image I is indexed by p and q. $B(S_r)$ contains the boundaries of S_r as mentioned in Eq. (4).

An external region is maximally stable if it satisfies the following conditions, reject otherwise.

- The threshold δ, is the number of intensity threshold levels that must be reached for a region to be considered maximally stable during MSER computation.
- MSERs demonstrate a minimum level of variation which implies a high level of stability than the regions from previous threshold levels. It can be computed as follows:

$$V = \frac{area(R+\delta) - area(R)}{area(R)} \tag{5}$$

Here, R represents a threshold value for excising the region.

- A region beyond the area range or area size of the region in pixels should be excluded. For instance, an object whose area size is beyond 90% range of the whole image or less than 5% range would be rejected since it has no relevance for the actual test scenario as mentioned in Eq. (5).
- At different intensity thresholds, the maximum area variation exists between extremal regions. Therefore, the stability of results is limited. The variation in intensity thresholds leads to providing stable regions with tiny areas.

C. MSER REGIONS

The output of MSER algorithm is a set of pixels representing MSER regions. Ellipses can also be calculated to represent MSER regions. To represent a pixel list, it is intuitive to draw a minimum rectangle that encloses all of its coordinates. To represent ellipses, some mathematical operators are used to obtain the bounding box of each ellipse.

In the first method, a minimum enclosing rectangle for pixels is obtained within the region starting from different representations of MSER region. In the second method, a sampling set is used to find ROI coordinates. A bounding box of (x_{min}, y_{min}) and (x_max, y_{max}) can be easily derived for the entire list of pixels. These parameters identify the minimum and maximum coordinate values for the horizontal and vertical axes at any point. By geometrical transformation of MSER ellipses, ROI can be obtained in the form of bounding boxes as follows:

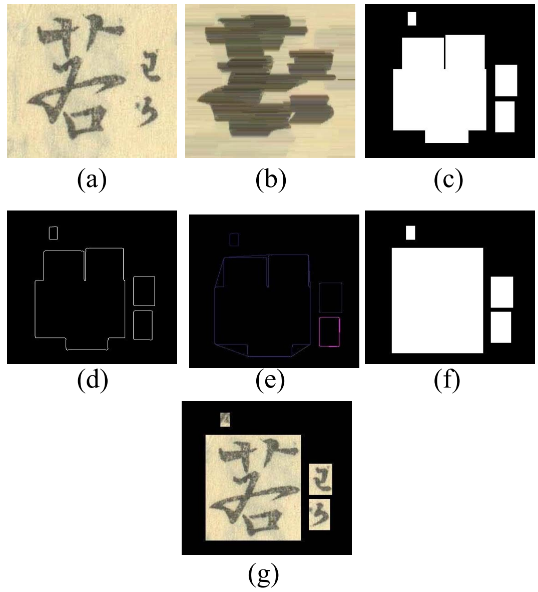
$$width = max(2a\cos\theta,\ 2b\sin\theta) \tag{6}$$
$$height = max(2a\sin\theta,\ 2b\cos\theta) \tag{7}$$

Here, a and b define the center values of extracted region. θ shows the rotation transformation.

To generate bounding boxes for MSER regions, ellipses are used. MSER regions often contain extra pixels near the text especially when text blends with its surroundings.

D. CHARACTER DETECTION APPROACHES

- **Approach 1: Morphological Image Operations** The characters in this approach are enlarged to a specified threshold limit. It means that all the single characters are grouped and the character's pronunciations are ignored.



(a) (b) (c) (d) (e) (f) (g)

FIGURE 6. Analysis of morphological operation: (a) Input image, (b) dilatation, (c) bounding box extension, (d) canny edge detection, (e) contour technique, (f) mask applied on new contours, and (g) character extracted after operations.

Figure 6(a) and 6(b) illustrate the failed attempt to find the dilated characters for the given image and a sample of such a failure is shown below.

- **Approach 2: Contour technique** It is found that the gap between a single disconnected character is approximately in the range of 7 to 10 pixels as opposed to the gap between the character and its pronunciation which is approximately 14 pixels. An extension of the bounding box is made for this image about $7px$ to match the other disconnected component of the same character. The boundary mask area is drawn outward by $7px$ and the sample is shown in Figures 6(c), 6(d), and 6(e), respectively. Approximately 90% of all disconnected characters within the boundary mask meet (see Figures 6(f) and 6(g)).

V. EXPERIMENTAL RESULTS

Experiments are performed on Xeon Ef 16200 v4 CPU and GPU GeForce GTX1080Ti $\times 4$ with memory as 64 GB and operating system is Ubuntu14.4. 3212 classes of Kuzushiji characters are used for building the DKNet model. The total number of training images is more than 600, 000. Kuzushiji dataset ID is obtained from the Center for Open Data in the Humanities (CODH) (refer to [29]). The parameter setting of DKNet and competitive models is presented in Table 2.

A. DATA AUGMENTATION

To augment the training data, three different augmentation techniques are also used. These techniques are random rotate, random blur, and random crop. The main objective is to augment the amount of training data to prevent the

TABLE 2. Parameters setting of DKNet and competitive models.

Parameter	Range
Padding pixels	0.1
Stride size	2
Filter size	7
Number of filters	6
Learning rate	0.01
epochs	100

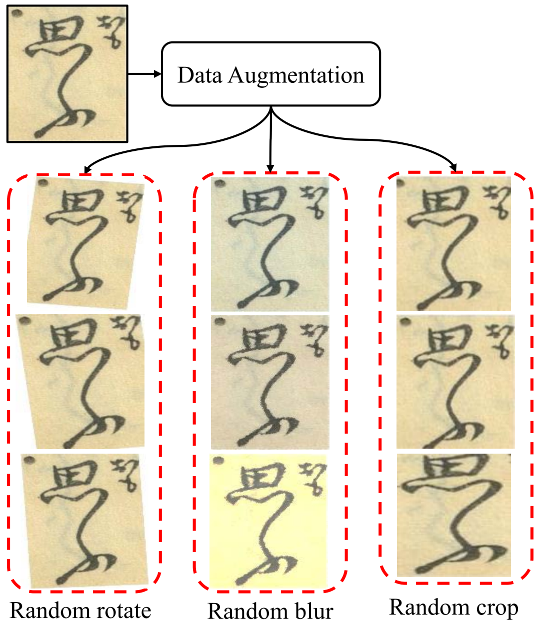


FIGURE 7. Impact of data augmentation techniques.

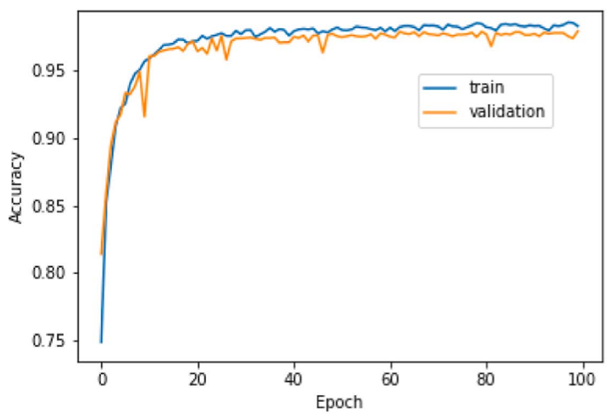


FIGURE 8. Training and validation accuracy of MobileNet.

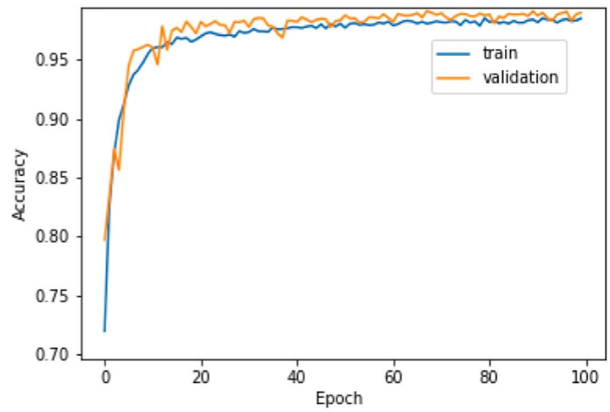


FIGURE 9. Training and validation accuracy of DKNet.

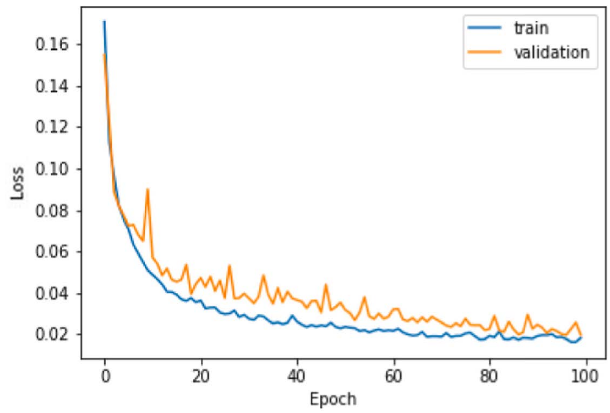


FIGURE 10. Training and validation loss of MobileNet.

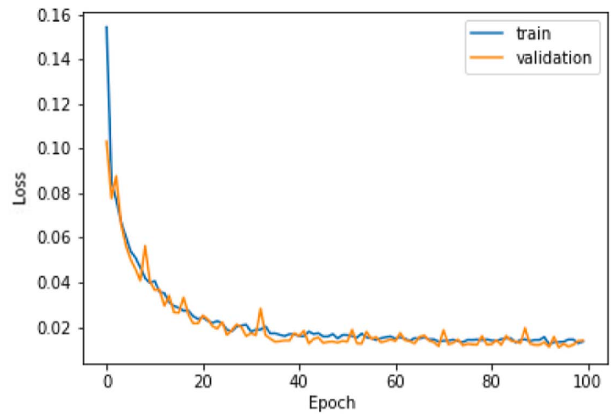


FIGURE 11. Training and validation loss of DKNet.

over-fitting and gradient vanishing problems. Figure 7 shows the impact of data augmentation techniques on a Kuzushiji character image.

B. TRAINING AND TESTING ANALYSIS

The training and validation accuracy analysis of MobileNet and DKNet are shown in Figures 8 and 9, respectively. The training and validation loss analysis of MobileNet and DKnet are shown in Figures 10 and 11, respectively. It is found that DKNet has achieved the better training and validation accuracy as compared to MobileNet. Hence, DKNet is selected and implemented in our online API.

C. PERFORMANCE ANALYSIS

The performance of DKNet is evaluated using confusion matrix-based metrics such as accuracy, F-measure, and area

TABLE 3. Training analysis of DKNet.

Model	AUC	CER	F-measure	SER	Accuracy
LeNet	94.15 ± 1.81	20.48 ± 1.58	94.87 ± 1.14	58.59 ± 1.43	94.91 ± 1.45
GoogLeNet	97.45 ± 1.23	20.14 ± 1.18	97.34 ± 1.52	57.18 ± 1.54	97.32 ± 1.34
VGG16	97.12 ± 1.34	19.29 ± 1.23	94.95 ± 1.11	56.08 ± 0.98	98.03 ± 0.99
VGG19	97.72 ± 0.44	18.04 ± 0.64	95.49 ± 0.44	55.72 ± 0.34	97.17 ± 0.32
ResNet152V2	97.59 ± 0.94	17.52 ± 0.93	97.21 ± 0.79	54.25 ± 0.74	97.24 ± 0.72
InceptionNet	97.41 ± 1.13	16.19 ± 1.53	97.25 ± 1.07	53.51 ± 0.95	97.53 ± 0.93
InceptionResNetV2	97.81 ± 1.03	15.84 ± 0.93	97.82 ± 0.74	52.94 ± 0.78	97.94 ± 0.84
XCeptionNet	97.81 ± 1.33	14.48 ± 1.01	97.83 ± 0.78	51.85 ± 1.04	97.88 ± 1.08
MobileNet	98.42 ± 0.68	13.41 ± 0.52	98.47 ± 0.63	50.74 ± 0.75	98.21 ± 0.85
Proposed DKNet model	**99.59 ± 0.39**	**11.36 ± 0.21**	**99.64 ± 0.35**	**48.61 ± 0.29**	**99.67 ± 0.31**

TABLE 4. Testing analysis of DKNet.

Model	AUC	CER	F-measure	SER	Accuracy
LeNet	94.75 ± 1.01	21.62 ± 1.72	94.74 ± 1.93	59.58 ± 1.92	94.71 ± 1.78
GoogLeNet	93.61 ± 1.31	21.54 ± 1.33	93.57 ± 1.64	58.51 ± 1.21	93.64 ± 1.23
VGG16	93.91 ± 1.24	20.77 ± 1.49	93.72 ± 1.21	57.74 ± 1.44	93.62 ± 1.44
VGG19	95.49 ± 1.41	19.19 ± 1.41	93.94 ± 1.41	56.21 ± 1.27	93.54 ± 1.22
ResNet152V2	94.31 ± 1.19	18.13 ± 1.14	94.15 ± 1.44	55.53 ± 1.43	93.51 ± 1.52
InceptionNet	95.21 ± 1.09	17.05 ± 1.14	93.44 ± 1.13	54.25 ± 1.09	94.83 ± 1.13
InceptionResNetV2	95.41 ± 1.17	16.81 ± 0.47	95.21 ± 0.89	53.29 ± 0.81	95.31 ± 0.22
XceptionNet	95.51 ± 1.25	15.12 ± 1.22	93.94 ± 1.22	52.02 ± 1.22	95.12 ± 1.46
MobileNet	96.63 ± 1.15	14.76 ± 1.51	95.32 ± 1.52	51.54 ± 0.42	96.34 ± 0.76
Proposed DKNet model	**98.42 ± 0.81**	**12.78 ± 0.17**	**98.75 ± 0.47**	**49.53 ± 0.83**	**98.57 ± 0.95**

under the curve (AUC). For evaluating handwriting recognition systems, we also use the Sequence Error Rate (SER) and Character Error Rate (CER) metrics [13]. DKNet's performance is evaluated through medians and uncertainty values (i.e., median $\pm IQR \times 1.5$).

The nine models such as LeNet, GoogLeNet, VGG16, VGG19, ResNet152V2, Inception, InceptionResNetV2, XCeptionNet, MobileNet, and DKNet are evaluated and the results are summarized in Tables 3 and 4.

Tables 3 and 4 illustrate the performance analysis of DKNet. DKNet achieves 99.67 and 98.57 accuracy values during training and testing. This means that the overfitting issue is not an issue. Further, DKNet was able to achieve an AUC value of 99.59 during training and 98.42 during testing. The proposed model is therefore least affected by the false positives and false negatives.

D. DISCUSSION

From the existing literature, it has been found that machine learning and deep learning techniques have been successfully used for Kuzushiji character recognition. Many models have been designed and implemented to recognize Kuzushiji characters from Japanese historical documents. Some well-known recognition models are RU-Net [40], AED [41], AACRN [45], 2DCbpn [48], MSOs [47], and RAED [13]. Although these models have shown significant results, the majority of these models have shown poor performance analysis. Therefore, an efficient deep Japanese recognition model using MSER is proposed. Table 5 demonstrates the performance analyses of DKNet and existing models.

RU-Net provides 86.83% accuracy and 85.53% F-measure. 2DCbpn provides better accuracy i.e., 93.32%, and F-measure i.e., 92.42 as compared to RU-Net. The accuracy of MSOs is 94.67% which is significantly better than RU-Net and 2DCbpn. DKNet provides 98.57% accuracy and 98.57% F-measure. Hence, it can be seen that DKNet outperforms the existing techniques in terms of accuracy and F-measure. The CER of AED is 32.34%. CER and SER of AACRN are 21.49% and 94.97%. CER of AACRN is better than AED. RAED provides 13.07% CER and 53.81% SER which is significantly better than AED and AACRN. DKNet provides 12.78% CER and 49.53% SER. It can be seen that DKNet outperforms AED, AACRN, and ADE in terms of CER and SER. Therefore, the proposed DKNet model is an efficient recognition system. In future, performnce of DKNet can be improved by tuning its hyper-parameters using metaheuristic techniques such as bat algorithm [50], particle swarm optimization [49], [51], etc.

TABLE 5. Performance comparison of DKNet with existing models.

Model	Database	Accuracy	F-measure	CER	SER
RU-Net [40]	CODH	86.83	85.48	-	-
AED [41]	-	-	-	32.34	-
AACRN [45]	CODH	-	-	21.48	94.97
2DCbpn [48]	CODH	93.32	92.42	-	-
MSOs [47]	CODH	94.67	-	-	-
RAED [13]	CODH	-	-	13.07	53.81
DKNet	CODH	**98.57**	**98.75**	**12.78**	**49.53**

E. EXPERIMENTAL ANALYSIS OF API

The recognition system is equipped on sever for online access. The CPU is Intel(R) Xeon(R) CPU E5-1410 v2 @ 2.80 GHz, RAM is 8G, and OS is Ubuntu 18.04.3 LTS. Apache HTTP Server was used for designing the API website.

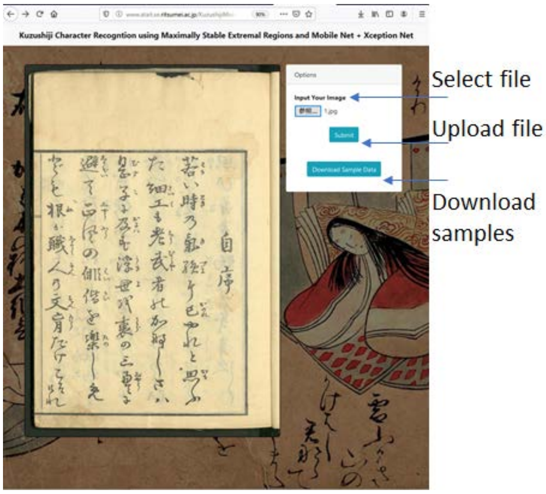


FIGURE 12. Interface of API.

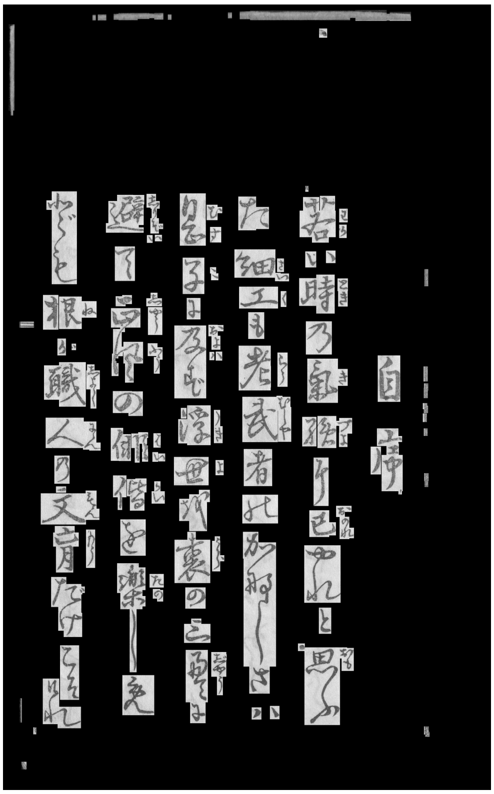


FIGURE 13. Masking MSER Technique.

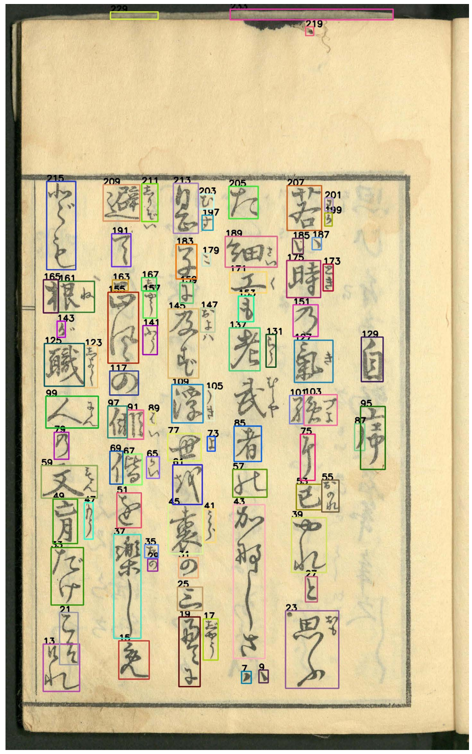


FIGURE 14. Segmentation of characters.

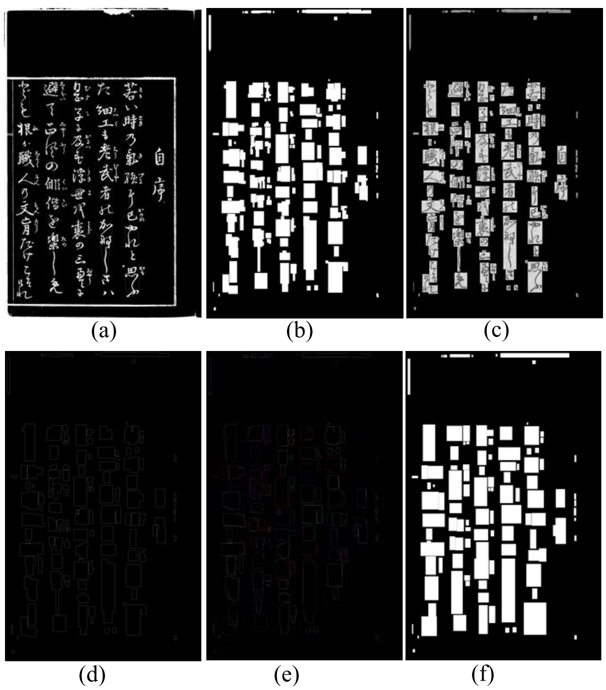


FIGURE 15. Extraction of characters: (a) MSER resultant image, (b) bounding mask drawn around possible characters, (c) text extracted with mask on the input image, (d) canny edge extracted from image mask, (e) convex hulls drawn around the canny edge extracted image, and (f) extraction of bounding mask from the convex hulls that contains characters.

Figure 12 shows the designed API, the target file can be selected from the local computer and submitted to our server. The user may download the sample files for testing the API.

Figures 13 and 14 show the obtained results of the proposed API. It is found that lots of characters are segmented correctly. The segmented characters are shown in the bottom of interface (see Figures 15 and 16). It is found that MSER with a bounding mask can achieve better results when there are significant gaps between characters. The proposed MSER and convex hull-based segmentation approach can segment characters when there is no or lesser gap between the characters (refer Fig. 16(a)). Whenever the user clicks the

(a) (b)

FIGURE 16. Extraction of single character: (a) extraction of text using Fig. 15(b) and (b) extraction of text using Fig.15(f).

characters, they will be recognized by the proposed model (refer Fig. 16(b)).

VI. CONCLUSION

A deep Kuzushiji characters recognition network (DKNet) was proposed to overcome over-fitting and gradient vanishing problems. First, the training images were filtered with a joint trilateral filter to remove noise. The filtered images were then enhanced with a contrast limited adaptive histogram equalization (CLAHE). A pre-trained MobileNet was then used to extract features from Kuzushiji characters. MobileNet's final layers, such as the fully connected layer and softmax, were removed. The flatten layer was then applied to the input. It was followed by a fully connected classification layer with rectified linear units (ReLU) and dropouts. Dropouts were used to generalize the model, thus preventing the over-fitting problem. Finally, the recognition results were obtained by applying the softmax activation function. To test the proposed model, actual documents were segmented using the proposed MSERs and a convexhull-based segmentation approach. Segmented characters were then recognized by the trained DKNet. Analyses of the competitive models revealed that DKNet surpassed the competitive models in terms of several performance metrics such as accuracy, F-measure, AUC, CER, and SER by 2.1675%, 1.9842%, 2.1075%, 1.6914%, and 1.7265%, respectively. An efficient Application Programming Interface (API) was also developed for Japanese Kuzushiji ancient heritage character recognition to help the end-users.

REFERENCES

[1] W. H. Bangyal and J. Q. A. Abbas, "Recognition of off-line isolated handwritten character using counter propagation network," *Int. J. Eng. Technol.*, vol. 5, no. 2, p. 227, 2013.

[2] Q. Abbas, W. H. Bangyal, and J. Ahmad, "Analysis of learning rate using BP algorithm for hand written digit recognition application," in *Proc. Int. Conf. Inf. Emerg. Technol.*, Jun. 2010, pp. 1–5.

[3] N. Fujisaki, Y. Ishikawa, M. Takata, and K. Joe, "Crawling low appearance frequency character images for early-modern Japanese printed character recognition," in *Advances in Parallel & Distributed Processing, and Applications*. Cham, Switzerland: Springer, 2021, pp. 683–695.

[4] C. V. Aravinda, M. Lin, A. Masahiko, and P. G. Amar, "A complete methodology for Kuzushiji historical character recognition using multiple features approach and deep learning Model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 8, pp. 1–7, 2020.

[5] L. Meng, B. Lyu, Z. Zhang, C. V. Aravinda, N. Kamitoku, and K. Yamazaki, "Oracle bone inscription detector based on SSD," in *Proc. Int. Conf. Image Anal. Process.* Cham, Switzerland: Springer, 2019, pp. 126–136.

[6] F. Horie, H. Goto, and T. Suganuma, "Synthetic scene character generator and ensemble scheme with the random image feature method for Japanese and Chinese scene character recognition," *IEICE Trans. Inf. Syst.*, vol. 104, no. 11, pp. 2002–2010, Nov. 2021.

[7] B. Lyu, Z. Wang, H. Li, A. Tanaka, K. Funumoto, and L. Meng, "Deep leaning based medicine packaging information recognition for medication use in the elderly," *Proc. Comput. Sci.*, vol. 187, pp. 194–199, Jan. 2021.

[8] A. Sayeed, J. Shin, M. A. M. Hasan, A. Y. Srizon, and M. M. Hasan, "BengaliNet: A low-cost novel convolutional neural network for Bengali handwritten characters recognition," *Appl. Sci.*, vol. 11, no. 15, p. 6845, Jul. 2021.

[9] T. Horiuchi and S. Kato, "A study on Japanese historical character recognition using modular neural networks," *Int. J. Innov. Comput., Inf. Control*, vol. 7, no. 8, pp. 5003–5014, 2005.

[10] F. Li and P.-Y. Woo, "The coding principle and method for automatic recognition of Jia Gu Wen characters," *Int. J. Hum. Comput. Stud.*, vol. 53, no. 2, pp. 289–299, Aug. 2000.

[11] X. Hu, M. Inamoto, and A. Konagaya, "Recognition of Kuzushi-Ji with deep learning method a case study of Kiritsubo chapter in the Tale of Genji," in *Proc. 33rd Annu. Conf. Jpn. Soc. Artif. Intell.*, 2019, pp. 1–2.

[12] B. Lyu, R. Akama, H. Tomiyama, and L. Meng, "The early Japanese books text line segmentation base on image processing and deep learning," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Aug. 2019, pp. 299–304.

[13] A. D. Le, T. Clanuwat, and A. Kitamoto, "A human-inspired recognition system for pre-modern Japanese historical documents," *IEEE Access*, vol. 7, pp. 84163–84169, 2019.

[14] Q. Li, Y. Yang, and A. Wang, "Recognition of inscriptions on bones or tortoise shells based on graph isomorphism," *Comput. Eng. Appl.*, vol. 47, no. 8, pp. 112–114, 2011.

[15] Q. S. Li and Y. X. Yang, "Sticker DNA algorithm of Oracle-bone inscriptions retrieving," *Comput. Eng. Appl.*, vol. 44, no. 28, pp. 140–142, 2008.

[16] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-V4, inception-ResNet and the impact of residual connections on learning," 2016, *arXiv:1602.07261*.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[22] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2016, pp. 4700–4708.

[23] K. Brown, *Encyclopedia of Language & Linguistics*, J. M. Unger, Ed., 2nd ed. Amsterdam, The Netherlands: Elsevier, 2006, pp. 95–102, doi: 10.1016/B0-08-044854-2/04566-1.

[24] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[25] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[26] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottleneck," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4510–4520.

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," 2015, *arXiv:1512.02325*.

[29] Center for Open Data in the Humanities (CODH). *Japanese Classic Cursive Script Data Set Book Title List [Data Set]*. Accessed: Feb. 26, 2020. [Online]. Available: http://codh.rois.ac.jp/char-shape/book/

[30] C. V. Aravinda, M. Lin, and P. G. Amar. *Kuzashi Recognition API*. Accessed: Feb. 26, 2020. [Online]. Available: http://www.atait.se.ritsumei.ac.jp/KuzushijiMser/

[31] L. Meng, C. V. Aravinda, K. R. U. K. Reddy, T. Izumi, and K. Yamazaki, "Ancient Asian character recognition for literature preservation and understanding," in *Proc. Euro-Mediterranean Conf.* Cham, Switzerland: Springer, 2018, pp. 741–751.

[32] C. V. Aravinda and H. N. Prakash, "A review on various handwritten cursive characters segmentation techniques," in *Proc. 2nd Int. Conf. Appl. Theor. Comput. Commun. Technol. (iCATccT)*, 2016, pp. 647–651.

[33] C. V. Aravinda and H. N. Prakash, "Template matching method for Kannada handwritten recognition based on correlation analysis," in *Proc. Int. Conf. Contemp. Comput. Informat. (ICI)*, Nov. 2014, pp. 857–861.

[34] L. Meng, "Recognition of Oracle bone inscriptions by extracting line features on image processing," in *Proc. 6th Int. Conf. Pattern Recognit. Appl. Methods (ICPRAM)*, 2017, pp. 606–611, doi: 10.5220/0006225706060611.

[35] L. Meng, "Two-stage recognition for Oracle bone inscriptions," in *Proc. ICIAP*, 2017, pp. 672–682, doi: 10.1007/978-3-319-68548-9.

[36] H. T. Nguyen, N. T. Ly, K. C. Nguyen, C. T. Nguyen, and M. Nakagawa, "Attempts to recognize anomalously deformed Kana in Japanese historical documents," in *Proc. 4th Int. Workshop Historical Document Imag. Process.*, 2017, pp. 31–36.

[37] D. Singh and V. Kumar, "Dehazing of remote sensing images using improved restoration model based dark channel prior," *Imag. Sci. J.*, vol. 65, no. 5, pp. 282–292, 2017.

[38] M. Kaur, D. Singh, V. Kumar, and K. Sun, "Color image dehazing using gradient channel prior and guided L_0 filter," *Inf. Sci.*, vol. 521, pp. 326–342, Jun. 2020.

[39] A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *J. VLSI Signal Process. Syst. Signal, Image Video Technol.*, vol. 38, no. 1, pp. 35–44, 2004.

[40] T. Clanuwat, A. Lamb, and A. Kitamoto, "KuroNet: Pre-modern Japanese kuzushiji character recognition with deep learning," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 607–614.

[41] A. D. Le, D. Mochihashi, K. Masuda, and H. Mima. (2018). *An Attention-Based Encoder–Decoder for Recognizing Japanese Historical Documents*. Pattern Recognition and Machine Understanding (PRMU). [Online]. Available: https://www.ieice.org/ken/paper/20181213H1IM/eng

[42] L. Bing, H. Tomiyama, and L. Meng, "Frame detection and text line segmentation for early Japanese books understanding," in *Proc. 9th Int. Conf. Pattern Recognit. Appl. Methods*, 2020, pp. 600–606, doi: 10.5220/0009179306000606.

[43] H. T. Nguyen, T. Nakamura, C. T. Nguyen, and M. Nakawaga, "Online trajectory recovery from offline handwritten Japanese kanji characters of multiple strokes," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 8320–8327.

[44] Y. Septiana, A. Mulyani, D. Kurniadi, and H. Hasanudin, "Handwritten recognition of Hiragana and Katakana characters based on template matching algorithm," in *Proc. IOP Conf. Mater. Sci. Eng.*, vol. 1098, no. 3, Mar. 2021, Art. no. 032093.

[45] N. T. Ly, C. T. Nguyen, and M. Nakagawa, "Attention augmented convolutional recurrent network for handwritten Japanese text recognition," in *Proc. 17th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Sep. 2020, pp. 163–168.

[46] L. Sichao and H. Miwa, "Algorithm using deep learning for recognition of Japanese historical characters in photo image of historical book," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst.* Cham, Switzerland: Springer, 2019, pp. 181–189.

[47] K. Ueki, T. Kojima, R. Mutou, R. S. Nezhad, and Y. Hagiwara, "Recognition of Japanese connected cursive characters using multiple softmax outputs," in *Proc. IEEE Conf. Multimedia Inf. Process. Retr. (MIPR)*, Aug. 2020, pp. 127–130.

[48] A. D. Le, "Detecting Kuzushiji characters from historical documents by two-dimensional context box proposal network," in *Proc. Int. Conf. Future Data Secur. Eng.* Cham, Switzerland: Springer, 2019, pp. 731–738.

[49] W. H. Bangyal, A. Hameed, J. Ahmad, K. Nisar, M. R. Haque, A. A. A. Ibrahim, J. J. P. C. Rodrigues, M. A. Khan, D. B. Rawat, and R. Etengu, "New modified controlled bat algorithm for numerical optimization problem," *Comput., Mater. Continua*, vol. 70, no. 2, pp. 2241–2259, 2022.

[50] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, "A systematic literature review on particle swarm optimization techniques for medical diseases detection," *Comput. Math. Methods Med.*, vol. 2021, pp. 1–10, Sep. 2021.

[51] W. H. Bangyal, A. Hameed, W. Alosaimi, and H. Alyami, "A new initialization approach in particle swarm optimization for global optimization problems," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–17, May 2021.