International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Recognition of Historical Kannada Manuscripts using Convolution Neural Network

Rajithkumar B. K[1], Dr.B.V. Uma[2], Dr. H.S. Mohana[3]

Submitted: 27/01/2024 **Revised**: 05/03/2024 **Accepted**: 13/03/2024

Abstract: Document image analysis has emerged as a field of study with growing importance over the last few decades. Historical documents add another challenge of physical degradation that needs to be tackled in the pre-processing. The main focus of the present work is the classification and identification of Kannada old stone inscription characters. The characters are segmented into lines, words, and characters for easier processing. The segmented images are then preprocessed in order to extract the essential features and remove the redundancies in the image. The preprocessed data is then augmented in order to compensate for the lack of datasets, and the existing dataset is trained in order to create data for the training phase. The machine learning model, Convolutional Neural Network (CNN), is selected. The classifiers based on each model are trained, and the performance of each model is evaluated. The model developed for recognizing Kannada characters achieved a validation accuracy of 95.9%. This outcome demonstrates a significant achievement in processing and digitizing ancient Kannada scripts, considering the complex nature of the language and the diverse characteristics of individual handwriting.

Keywords: *Historical Kannada documents, Image recognition, Document analysis, Cultural heritage preservation Convolutional Neural Network*

1. Introduction

The language used in Karnataka is Kannada, a Dravidian language. Thirteen vowels, thirty-four consonants, and two letters that are neither consonants nor vowels make up the forty-nine phonemic letters of Kannada, which are written in a script that is not Latin [1]. Since there are more characters and more character repetitions in Kannada, character identification is a more difficult challenge. Preservation of rich linguistic and cultural history is very much the need of the hour in present times, where rich legacies are gradually being forgotten. Most of this is handwritten or inscribed on stones whose remnants can still be found in archaeological places like Hampi and Halebeedu [2]. In this digital age where tools and means are available to identify and digitally preserve records for posterity, it is required to use the tools to preserve rich cultural legacies.The objective of the work is to devise a system to read and process Kannada script, digitize the information, and store it electronically. Datasets in the form of leaf manuscripts and stone inscriptions are collected from the Kuvempu Institute of Kannada Studies, Manasagangotri, Mysore, and Hampi (UNESCO World Heritage Site), Vijayanagara. Various preprocessing steps, namely gray scaling, image binarization, edge detection, cropping, etc., are applied to the collected datasets to remove redundancy in the information contained and make processing faster. The preprocessed images are fed to the optical character recognition system, which uses CNN model algorithms to recognize individual Kannada characters. Comparisons are drawn between images with different noise levels. Finally, images are analyzed, and a conclusion about the model is drawn [3].

[1] R.V College of Engineering Bengaluru, Visvesvaraya Technological University, Belagavi,India -590018,
ORCID ID : 0000-0002-1608-2085
[2] R.V College of Engineering Bengaluru, Visvesvaraya Technological University, Belagavi,India -590018
[3] Navkis College of Engineering Hassan, Visvesvaraya Technological University, Belagavi,India -590018
** Corresponding Author Email: Rajith.bkr@rvce.edu.in*

2. Related Works

In [1], recognition of handwritten characters was focused, and the captured characters are difficult to recognize due to the dissimilar script styles and shape of character.In order to solve this issue, initial training data was performed, which was later supplemented by validation and testing data. In [2], the database was created, and it contains 100,000 words from 600 users compiled. The characteristics of the scripts, as well as the number of symbols, effectively trained the data for recognition. All of the words are included in the database list. In [3], the author proposes a three-stage character segmentation method for separating Kannada characters. The three-line character segmentation includes separating each into three segments: the top zone, base and compound characters make up the middle zone, and consonant make up the bottom zone. In [4], author demonstrates the implementation of a feature extraction method based on zone metrics. The image is split up into n equal zones, and the character centroid is computed.

The survey paper [5] expresses the tasks associated with the recognition system, including binarization, enhancement, and layout analysis. It compares existing architectures like U-Nets and encoder-decoder networks and looks into the different historical datasets available, such as Bentham, Washington, and Ratskaporolle used to solve problems with imperfect backgrounds, stains, and uneven illumination. In [6], various binarization methods such as global fixed threshold, Otsu threshold and Markov models are implemented on a set of images with printed text. The results show that the Otsu threshold with little to no denoising performs the best. [7] presents page layout analysis consisting of classifying blocks and subsequently lines that can be performed using ANNs. Block segmentation extracts text regions in the reading order using U Net, while line segmentation involves segmenting text into smaller lines using tools like ARU Net and Kraken. Synthetic data is also made use of here, and test cases included completely randomly generated text, historical words in German, and modern German words with far lower accuracies of about 50%. In [8] presents a novel methodology for segmenting a document page into words. The proposed segmentation is based on evaluate blobs, i.e., the connection of individual letters or characters as a single entity based on a Laplacian on a Gaussian operator. The results show a promising detection of 99.12% of words with 87% accuracy.

The literature reveals substantial work on the preprocessing of ancient scripts, such as noise removal, thinning, binarization, and segmentation. The Devanagri, Tamil, and Kannada stone inscription characters are recognized using Support Vector Machines(SVM) and Neural networks algorithms. But recognition of ancient Kannada words in inscriptions is still an open challenge because some characters used in inscriptions have been partly or completely erased due to weather conditions or war, etc.

3. Design Methodology

The design methodology is shown in Fig. 1 and it consists of segmentation, pre-processing, Augmentation, Feature extraction and evaluation of model parameters.

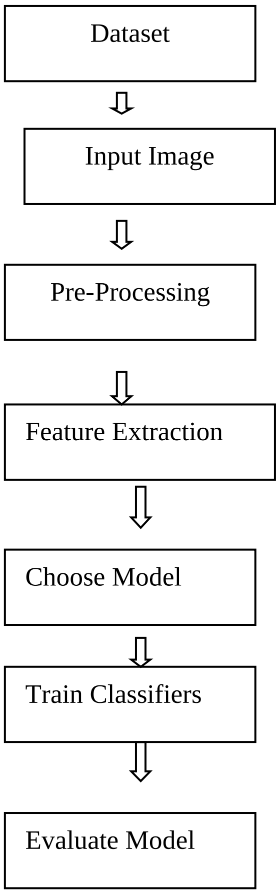

Fig 1: Block diagram for the proposed system

3.1 Dataset Creation: Datasets are created using sources from the Vijayanagar Empire and Mysore, Karnataka, India, using ordinary digital cameras. Around ten thousand images are captured in Hampi, Vijayanagar, India, and leaf manuscripts are collected from the Kuvempu Institute of Kannada Studies, University of Mysore, India. Hampi, the UNESCO world heritage site and the second largest city of that time, is the capital of the Vijayanagara Empire. Period: 14th–16th century The stone inscriptions that they carved are relevant to the culture of that time. Stone inscriptions are a source of important historic information, such as about kings, kingdoms, and governance during that period of time. The collected image datasets are then preprocessed using image preprocessing techniques such as gray scaling, binarization, edge detection, skew correction, etc. The output of this stage passes to the segmentation stage, where character segmentation is performed and bounding boxes are obtained for each character. The characters in the image are resized to a set uniform pixel area and Images are labeled and split for training and testing set, and given to the CNN recognition model.

3.2 Preprocessing

3.2.1 Gray Scaling

The RGB of an image will be converted to a grayscale image using average method or weighted method.

The weighted method, also called the luminosity method, the weighs of red, green, and blue according to their wavelengths, as shown in equation 1.

$$\text{Grayscale} = 0.299R + 0.587G + 0.114B \quad (1)$$

In the proposed method the standard NTSC conversion formula is used for grayscale conversion.

3.2.2 Image Binarization Word recognition tasks do not require gradients or color information. A number of binarization techniques were implemented and analyzed in order to select the ideal technique. The following methods are analyzed:

A. Isodata thresholding: randomly considers a pixel threshold “t” within the range of available intensity values in a grayscale image. It determines two means of either class higher or lower than the pixel threshold ”t” given by mL and mH. The threshold is then updated as the average of mL and mH. This is depicted in Equation 2.

$$t = (mL + mH)/2 \quad (2)$$

This process is iteratively continued until convergence upon a single threshold, which is then utilized as the image threshold.

B.Mean binarization

This thresholding method utilizes the average of all the grayscale pixel intensity values as its threshold and the image based on its pixel intensity values is likely to be skewed, as shown in equation 3.

$$t = \sum I(x, y))/n \quad (3)$$

where I(x, y) represents the pixel intensity value for any pixel at location (x, y), and n represents the number of pixels in the region.

C.Sauvola Thresholding

The Sauvola method is a local thresholding algorithm that takes into consideration the local mean and standard deviation. It is a good algorithm to use for slightly more complex images. A local threshold is calculated by using a sliding rectangular field of consideration across the image, as shown in equation 4.

$$t = m(x, y) + k * n(x, y) \quad (4)$$

Equation 4 depicts this. Here, m(x, y) represents the mean of the local window, n(x, y) represents the variance, and k is a constant set at -0.2.

D. Binarization: Otsu Thresholding

This algorithm minimizes the difference between classes by finding the threshold or weighted average of the difference between two classes (background and foreground). The grayscale range is 0-255 (0or 1 for floating point values). According to Equation 5, all pixels with a value less than 100 will be the background of the image, and all pixels with a value greater than or equal to 100 will be the foreground of the image. This happens if you set it to 100.

$$\sigma^2_W = W_b\sigma^2_b + W_f\sigma^2_f \quad (5)$$

Evaluation Metric: PSNR

Peak signal-to-noise ratio (PSNR) is an expression for the ratio between the signal and noise. There is a high positive correlation between PSNR and visual quality, but it cannot be a sole indicator. PSNR is most easily defined via the mean squared error (MSE), as shown in equation 6.

$$PSNR= (2N-1)^2/MSE \quad (6)$$

G. Evaluation Metric: SSIM

The measure of similarity between two images is called Structural Similarity Information Measure (SSIM). Unlike PSNR, which uses the concept of absolute error, SSIM includes the concept of proximity. Brightness time, contrast time, and sampling time are three terms calculated to determine the quality index of the SSIM Index. Equation 7 shows that all exponents are combinations of three terms.

$$SSIM(x, y) = [l(x, y)]\ \alpha\cdot[c(x, y)]\ \beta\cdot[s(x, y)]\ \gamma \quad (7)$$

Where x and y are the images,

The System Similarity Index (SSIM) is a method that calculates how similar two images are to each other. SSIM values range from 0 to 1; where 1 means the reconstructed image is exactly the original image. SSIM values for the well-developed method are generally 0.97, 0.98, and 0.99. Table 1 explains that the PSNR of the Otsu method is comparable to and better than the Sauvola method. The final processing was done using the Otsu method.

Table 1: Performance Metric Analysis

Metric/Method	Isodata	Otsu	Sauvola
PSNR	12.7876	12.4864	10.5448
SSIM	0.7955	0.788	0.7300

3.2.3 Edge Detection and Cropping

Edge detection is an image-processing technique that is used to identify the boundaries (edges) of objects or regions within an image. The resultant image after binarization is considered input, and the sum of all pixels in each row of the image is determined using equation 8. Compete the mean of the sums recorded using equation 9 and crop out the regions that fall below the mean.

$$avg = \frac{\sum_i^{rows}\sum_j^{cols} aij}{no.of\ rows} \tag{8}$$

$$textarea = \sum_j^{cols} aij - avg \tag{9}$$

3.2.4 Denoising

The captured images contain salt and pepper noise, and to remove such noise, various standard filtering methods are analyzed and their PSNR is calculated. Spatial filtering techniques such as mean, median, and Gaussian filters are analyzed with varied kernel sizes in order to accommodate different types of images.

A.Gaussian Smoothing: This filter focuses on giving more weight to the central pixel, which is unlike the mean and median filters. It uses a 2-D distribution called a point spread function. The Gaussian kernel outputs a weighted average at the center, which permits gentler smoothing and the preservation of edges, as shown in Fig. 2.

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \qquad \frac{1}{273} \times \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Fig 2: Gaussian filters of size 9 and 25 respectively

3.2.5 Skew Correction

Each word will be converted to slope correction and Hough transform for slope detection. The Equation 10 shows that if the character's angle is positive, the character's slope is clockwise, otherwise the character's slope is counterclockwise. Set it to show a visual tilt angle of -1.5 degrees.

$$(angle) = \sum\sum_0^{rows} pixel \tag{10}$$

The angle for which f (angle) is maximum over a range of angles is the skew angle, and the image is rotated by the skew angle to make it as horizontal as possible.

3.2.6 Line Segmentation

Segmentation is nothing but breaking the whole image into subparts to process them further as shown in Fig. 3

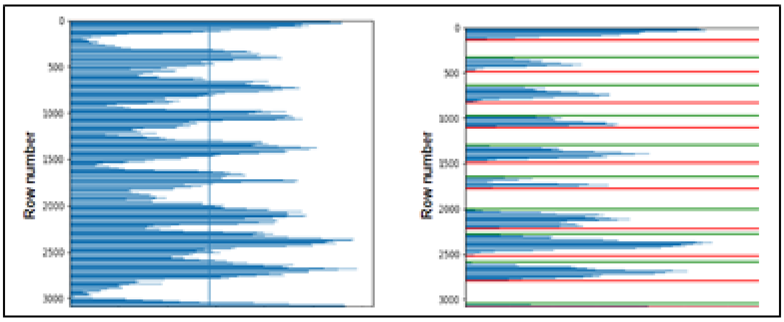

Fig. 3: Line Segmentation

The text zones are detected, and the rows of skew-corrected text are summed up after the image is negated. The negative images contained text zones showing peaks. Finally, the average is taken, and the sum is normalized to this average. Regions with high non-zero values are marked as text zones. The green line in the graph indicates the start of the text zone, and the red line indicates the end of the text zone.

3.2.7 Character Segmentation

The function of character segmentation is to divide the image of a string of characters into smaller images representing different characters. It is one of the determining factors of the Optical Character Recognition (OCR) system. The linked object collects an image and divides its pixels into objects based on the connectivity of the pixels (e.g., all pixels in a link have a ratio of pixels using multiples and connected to each other). Once each group is defined, each pixel is labeled according to its material, which can be colored or grayscale (color labeling).The 4-way and 8-way communication is possible. When using a 4-way connection, measure the pixel's connectivity by checking the top, bottom, left and right pixel position.

3.3 Convolutional Neural Network

It is a type of neural network that used to process data with a known grid-like topology. The Keras is a deep learning library that simplifies the creation of CNNs. It is composed of two stacks of convolutional and maximum pooling layers. Normalization, dropout, and flattening layers are then applied to convert the feature set to a 1-dimensional vector, which is then fed to fully connected dense layer.

$$\frac{\partial E}{\partial w_{ab}} = \sum_{i=0}^{N-m}\sum_{j=0}^{N-m}\frac{\partial E}{\partial x_{ij}^{l}}\frac{\partial x_{ij}^{l}}{\partial w_{ab}} = \sum_{i=0}^{N-m}\sum_{j=0}^{N-m}\frac{\partial E}{\partial x_{ij}^{l}}y_{(i+a)(j=b)}^{l-1}. \qquad (11)$$

In Equation 11 error function is E, and it compute for the preceding layer with constant a ,b for each neuron output $\frac{\partial E}{\partial x_{ij}^{l}}$.

3.3.1 Input Image

The Red, Green, and Blue colors planes of input stone inscription images are separated. The ConvNet's condense the image into a simpler format without loosing essential information using equation 12.

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, egg. RGB) (12)

3.3.2 Convolution Layer: The Kernel

In the first part of the convolution process, the elements that complete the convolution process are called kernels or filters. Select matrix K, 3x3x1. The kernel moves nine times with step length = 1 (no step) and does the matrix multiplication of K and the viewport P in which it is currently hovering

3.3.3 Padding

The function of the convolution process is to extract high-level features such as edges from the input image. The first ConvLayer usually saves low-level properties such as edges, colors, and gradient directions. Adding additional layers also allows the architecture to evolve to a higher level, providing us with a network that understands all the images in the dataset.

This process leads to two types of results: one is a situation in which the dimensionality of convolutional features is smaller than the dimensionality of the input, and the other is a situation in which the dimensionality of convolutional features increases or remains unchanged. This can be done with appropriate padding. After scaling a 5x5x1 image to a 6x6x1 image and applying a 3x3x1 kernel to it, a convolution matrix of size 5x5x1 is obtained. That's why Same Padding was created. On the other hand, if the same method is done without padding, it produces a matrix with the exact dimensions of the kernel (3x3x1).

3.3.4 Normalization Layer

If the pixel value is negative, make it zero. Otherwise, keep the same value. It should be applied after every convolution layer. Applying ReLU doesn't change the dimension.

3.3.5 Pooling Layer

The pooling layer, like the convolutional layer, is in charge of shrinking the convolutional feature's spatial size. Through dimensionality reduction, the amount of computing power needed to process the data will be reduced. Additionally, it helps to extract dominating characteristics that are rotational and positional invariant, retaining the effectiveness of the model training process. Max pooling and average pooling are the two different types of pooling. The maximum value from the area of the image that the kernel has covered is returned by Max on the basis of the increased complexity of capturing low-level details even further, but at the cost of more computational power.

3.3.6 Classification: Fully Connected Layer (FC Layer)

Implementing a fully connected layer is a generally inexpensive way to learn non-linear combinations of the high-level properties represented by the convolutional layer's output. The fully connected layer is now learning a function in that region that might not be linear. To be used with multi-level perception, the input image must now be flattened into a column vector. The flattened output is fed into a feed-forward neural network, and each training iteration uses back propagation. The model can detect dominant and particular low-level features to classify images using the Softmax Classification approach across multiple epochs.

4 Results & Discussions

4.1 Dataset Collection:

The database was created for Hoysala, Mysore, Belur, and Halibedu places in Karnataka, India as shown in Fig. 4. The 5000 images are collected, and for training, around 2500 images are stored in a database sample.

Fig. 4: Kannada Stone Inscriptions

4.2 Gray Scaling of Hoyasala Stone Inscriptions

The Fig. 5 and 6 are the results obtained from the gray-scaling of Hoyasala stone inscriptions. The gray scaling is done to reduce the amount of information contained in a 3D pixel value to a 1D pixel value, which resulted in an increase in the amount of information held by each pixel value and made computations simpler.

Fig.5: Image of Kannda Stone inscription

Fig. 6: Gray Scaling of Kannada Stone inscription

4.3 Image Binarization of Hoyasala Stone Inscriptions

The result of binarization of the Hoyasala stone Inscriptions image is shown in Fig. 7. A 256-level grayscale binary image is converted to a black and white image. A threshold must be selected for the grayscale image before starting the binarization process. The OTSU binarization method, which results in thresholding, is used in the planning process. If the pixel is higher

than the threshold, it is considered white (255); If it is lower than the threshold, the pixel is considered black (0).

Fig. 7: Image Binarization of Kannada stone inscription

4 . 4 Smoothening of Hoyasala Stone Inscriptions

The smoothening process is done using median filter that averages the pixel valuewith its neighbours as shown in Fig.

8. This removes unnecessary specs and noise in the image and allowsfor better character segmentation

Fig. 8: Smoothening of Kannada Stone Inscriptions Images

4.5 Character Segmentation of Hoyasala Stone Inscriptions

The Fig.9 represents the results obtained after character segmentation of Hoyasala Inscriptions. Connected Components algorithm used is successful in character segmentation of Hoyasala Stone Inscriptions

Fig. 9: Character Segmentation

4.6 Recognition of Hoysala Stone inscription: The Hoysala Stone Inscriptions are recognized and it is shown in the Fig. 10

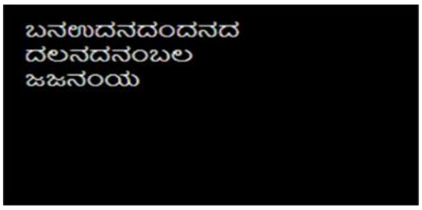

Fig. 10: Result obtained from character segmentation of Kannada stone inscription

A graph depicting model accuracy versus epochs is illustrated in Fig. 11. It demonstrates the relationship between the number of training epochs and the recognition model's accuracy. The y-axis shows the model's accuracy on a particular task or dataset, and the x-axis shows the epochs, which are training process iterations.

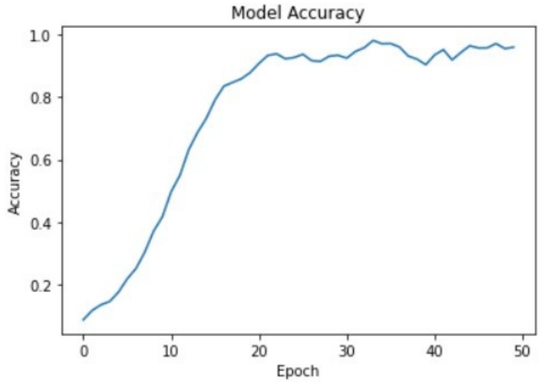

Fig. 11: Epochs v/s Accuracy

As the number of epochs rises, so does the model's accuracy,making it possible for it to recognize characters more precisely. A graph illustrating model loss versus epochs in Fig. 12 demonstrates the relation between the number of training epochs and the loss of the recognition model. The training process's iterations are represented by the epochs on the x-axis, and the model's loss is shown on the y-axis. It can be seen that as the number of epochs rises, the loss likewise falls, which lowers the frequency of incorrect predictions.

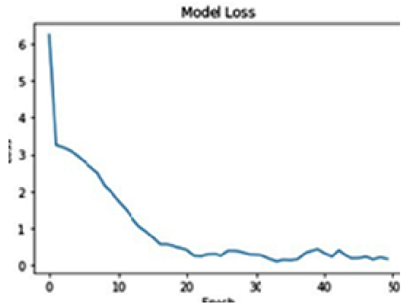

Fig.12: Epochs v/s Loss

5. Comparsion Results:

The recognition rates for different types of inscriptions with varying levels of noise are displayed in Table 2. The palm leaf inscription with low noise achieved a recognition rate of 90.9%, demonstrating an exceptional ability to recognize characters. Stone inscriptions 1 and 2 with low noise attained recognition rates of 88.4% and 82.8%, respectively, displaying good performance but slightly lower than the palm leaf. The stone inscription with high noise obtained a recognition rate of 50.5%, implying significant challenges in accurately processing heavily noisy inscriptions. Lastly, the stone inscription with moderate noise achieved a recognition rate of 75%, indicating reasonable performance in handling moderately noisy data.

Table 2: Comparison of Results

Epigraph	Recognition rate (in %)
Palm Leaf inscription - Low Noise	90.9
Stone inscription 1 - Low Noise	88.4
Stone inscription 2 - Low Noise	82.8
Stone inscription - High Noise	50.5
Stone inscription - Moderate Noise	75

The recognition accuracy attained on degraded documents is shown in Table 3. The Table 3 contains the results of the comparison between the proposed and existing approaches. The projected experimental results in Table 3 make it evident that the proposed method has a greater level of recognition accuracy than the other methods already in use

Table 3: Comparative analysis of degraded printed Kannada character recognition

Sl.No.	Techniques	Total datasets experimented	Accuracy (in%)
1	Neural Network[27]	2450 broken character dataset synthetically generated	98.9
2	FDA(Fit Discriminant Analysis)[28]	250 real datasets from historical, 150x49=7350 Synthetically generated dataset	99.38
3	FLD(Fisher Linear Discriminant Analysis)[29]	21560 both clear and degraded characters(Kannada and English) [Kailasam, Kasturi,Times new Roman,Arial	98.2
4	End point algorithm[30]	100 degraded Kannada characters	89
5	Convolutional Neural Network (Proposed Method).	16100 degraded Kannada characters extracted from old documents	95.9

6. Conclusion

The Optical Character Recognition (OCR) system uses CNN model algorithms for recognizing historical Kannada characters. The system reduces the amount of information contained in the original dataset during the preprocessing step by using algorithms like OTSU thresholding and Gaussian thresholding. It is observed that though there is considerable skew angle present in captured images, the system is capable of not only calculating the skew angle but also correcting it. This helps in a considerable reduction in human effort and makes processing faster. The average accuracy of the OCR system is 95.9%. The OCR system is found to be best suited for leaf manuscript databases using the CNN model. The system could be modified in the future by using more advanced algorithms that can translate historical Kannada words into present-day Kannada words, which could lead to an easier understanding of historical Kannada documents.

Conflicts of interest

The authors declared no conflicts of interest.

References

[1] Rajithkumar B K et al, "Template matching method for recognition of stone inscripted Kannada characters of different time frames based on correlation analysis", International Journal of Electrical and Computer Engineering (IJECE) Vol. 4, No. 5, October 2014, pp. 719~729,ISSN: 2088-8708.

[2] Sridevi T.N, et al. "Deep Convolution Neural Network for Degraded Printed Kannada Character Recognitions". Indian Journal of Computer Science and Engineering. Volume 12 No. 3 May-Jun 2021. DOI: 10.21817/indices/2021/v12i3/211203187

[3] Rajithkumar B K et al, "Extraction of Stone In-scripted Kannada Characters Using Sift Algorithm Based Image Mosaic", International Journal of Electronics & Communication Technology, Volume 5, Issue 2, April - June 2014.

[4] Rajithkumar B K et al., "Era Identification and Recognition of Stone In-scripted Kannada Characters Using Artificial Neural Networks":2nd National Conference on Innovation in Computing and Communication Technology, March, 2014.

[5] Haoming Zhang. "Ancient Stone Inscription Image Denoising and Inpainting Methods Based on Deep Neural Networks". Discrete Dynamics in Nature and Society Vol. 1, 2021. DOI: 10.1155/2021/7675611

[6] Chandrakala, H. T, "Deep Convolution Neural Networks for Recognition of Historical Handwritten Kannada Characters", In Frontiers in Intelligent Computing: Theory and Applications (pp. 69-77). Springer, Singapore. 2021

[7] Thippeswamy, G. "Recognition of Historical Handwritten Kannada Characters Using Local Binary Pattern Features". International Journal of Natural Computing Research (IJNCR), 2020

[8] F. Lombardi, "Deep learning for historical document analysis and recognition—a survey," National Library of Medicine, vol. 10, 2020.

[9] M. R. Gupta, N. P. Jacobson, and E. K. Garcia, “OCR binarization and image pre-processing for searching historical documents,” Pattern Recognition, vol. 40, 2007.
[10] J. Martine, L. Lenc, and P. Kr al, “Building an efficient OCR system for historical documents with little training data,” Neural Computing and Applications, vol. 32, 2020.
[11] R. Manmatha and N. Srimal, “Scale space technique for word segmentation in handwritten documents,” in Scale-Space Theories in Computer Vision, Springer Berlin Heidelberg, 1999.
[12] G. Chen, Q. Chen, X. Zhu, and Y. Chen, “A study of historical documents denoising,” in 2017 10th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics (CISP-BMEI), 2017.
[13] P. Sharan, S. Aitha, A. Kumar, A. Trivedi, A. Augustine, and S. R. K. Sarvadevabhatla, “Palmira: A deep deformable network for instance segmentation of dense and uneven layouts in handwritten manuscripts,” CoRR, 2021.
[14] R. I. Minyue Dai Carrie Yang and M. J. Brown., “Experiments with early modern manuscripts and computer-aided transcription,” Pattern Recognition Letters, 2018.
[15] Kshetry and R. Lal, “Image pre-processing and modified adaptive thresholding for improving OCR,” ArXiv, 2021.
[16] M. Shen and H. Lei, “Improving OCR performance with background image elimination,” in 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2015
[17] T. Blanke, M. Bryant, and M. Hedges, “Open source optical character recognition for historical research,” Journal of Documentation, vol. 68, 2012.
[18] B. J. Bipin Nair, N. Shobharani, N. R. Sreekumar, and G. Ashok, “A two phase denoising approach to remove uneven illumination from ancient note book images,” in 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), vol. 1, 2021.
[19] K. Saddami, K. Munadi, Y. Away, and F. Arnia, “Effective and fast binarization method for combined degradation on ancient documents,” Heliyon, vol. 5, 2019.
[20] C. Tensmeyer and T. Martinez, “Historical document image binarization: A review,” SN Computer Science, vol. 1, 2020.
[21] J. A. S anchez, V. Romero, A. H. Toselli, M. Villegas, and E. Vidal, “A set of benchmarks for handwritten text recognition on historical documents,” Pattern Recognition, vol. 94, 2019.
[22] M. Almeida, R. Lins, R. Bernardino, D. Jesus, and B. Lima, “A new binarization algorithm for historical documents,” Journal of Imaging, vol. 4, 2018.
[23] W. Xiong and L. Zhou, “An enhanced binarization framework for degraded historical document images,” EURASIP Journal on Image and Video Processing, vol. 13, 2021
[24] S. Lu and C. L. Tan, “Script and language identification in noisy and degraded document images,” IEEE transactions on pattern analysis and machine intelligence, vol. 30, 2008.
[25] S. Vijayarani and A. Sakila, “Multi-language script identification from document images,” International Research Journal of Modernization in Engineering Technology and Science, vol. 3, 2021
[26] Kumar, H. S et al., “ Versatile OCR for Documents in any Language Printed in Kannada Script”. ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP),2020.
[27] Monisha, G. S. et al,: “Effective Survey on Handwriting Character Recognition”. In Computational Method and Data Engineering. Springer, Singapore.2021
[28] Sandhya, N., & Krishnan, R. “Broken Kannada character recognition a neural network based approach”, International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT).2016, March: (pp. 2047-2050). IEEE.
[29] Sandhya, N., Krishnan, R., Babu, D. R., & Rao, N. B.,. “An efficient approach for handling degradation in character recognition.”, International Journal of Advanced Intelligence Paradigms,”.2019, 14(1-2), 14-29.
[30] Aradhya, V. M., Kumar, G. H., Noushath, S., &Shivakumara, P. “Fisher linear discriminant analysis based technique useful for efficient character recognition”, Fourth International Conference on Intelligent Sensing and Information Processing,2006, (pp. 49-52). IEEE
[31] Sandhya, N., Krishnan, R., &Babu, D. R. “A novel local enhancement technique for rebuilding Broken characters in a degraded Kannada script”. In 2015 IEEE International Advance Computing Conference (IACC),2015 June (pp. 176-179). IEEE