Implementation of Preprocessing Techniques for Precise Classification of Ancient Kannada Epigraphs

L. S. Anusha
Department of ECE, RV College of Engineering, Bangalore, India | Visvesvaraya Technological University, Belagavi, Karnataka, India
anushashekar@rvce.edu.in (corresponding author)

Abhay Anandrao Deshpande
Department of ECE, RV College of Engineering, Bangalore, India | Visvesvaraya Technological University, Belagavi, Karnataka, India
abhayadeshpande@rvce.edu.in

Received: 17 March 2025 | Revised: 17 April 2025 and 25 April 2025 | Accepted: 27 April 2025

 DOI: https://doi.org/10.48084/etasr.10996

ABSTRACT

The Dravidian language Kannada is most spoken in the state of Karnataka, and due to its extensive library of epigraphs, including old manuscripts and inscriptions, it is regarded as a repository of knowledge. To make this knowledge more accessible, efforts are underway to digitize documents for optimized usage and storage using Optical Character Recognition (OCR). However, often these epigraphs are in poor condition, and the quality of the image being fed to the OCR model may not be good enough to achieve high accuracy of recognition and classification. Preprocessing techniques are used to improve dataset quality. Preprocessing methods, including binarization, smoothing, edge detection, and segmentation, help to increase the model's interpretability, decrease overfitting, and train it more quickly and with fewer resources. When applied to epigraphs, these preprocessing approaches significantly increase image quality and minimize noise, making it easier to identify and digitize the text.

Keywords-data preprocessing; binarization; edge detection; segmentation; smoothing

I. INTRODUCTION

Approximately 44 million people speak Kannada, a Dravidian language, primarily in the state of Karnataka in southern India. It is one of India's oldest languages, as it has a long literary history that dates to the ninth century. The script used in Kannada is distinctive and descends from the ancient Brahmi script. Sanskrit and other Indo-Aryan languages, as well as other Dravidian languages, have all influenced and been influenced by the language. The old Kannada script has evolved over time and has been used to write inscriptions, manuscripts, and other important documents. The study of ancient Kannada script is important to understand the history, culture, and language of Karnataka and the wider region. However, it can be challenging because the old Kannada script is significantly different from the modern one, which has evolved over time and across multiple dynasties.

The process of locating and deciphering extinct scripts or writing systems is known as ancient script recognition. Common examples of ancient artifacts with such scripts include pottery, coins, tablets, and manuscripts. Understanding the history, culture, and language of ancient civilizations requires a thorough understanding of ancient scripts. Modern developments in artificial intelligence and machine learning have made it possible to automate the process of script recognition [1]. A standard Optical Character Recognition (OCR) system is designed to first divide the entire document into text lines, then into words, and finally into individual characters. Then, the required features can be extracted from these characters, which are subsequently recognized and classified.

To identify and translate the script, it is necessary to train computer algorithms to recognize patterns and characters. However, due to the poor preservation of many ancient scripts, ancient script recognition continues to be a complex and difficult field. In [2], ancient text character recognition was performed using a deep learning selection autoencoder technique for character level recognition, achieving a binarization accuracy of 74.24%, while segmentation using the Seam Carbel method achieved an accuracy of 70%. The various preprocessing techniques used in character recognition systems with various types of images, ranging from simple handwritten form-based documents and documents with colored and complex backgrounds and varied noise intensities, have been covered in analytical reviews of recent OCR system architectures and technologies [3-5]. Preprocessing involves cleaning and converting raw data into a format that can be analyzed and is a crucial step in data analysis and machine learning. Preprocessing in offline OCR is considerably influenced by both techniques and operating conditions [6]. Preprocessing techniques, such as grayscaling, binarization, edge detection, and smoothing, among others, are used to improve the accuracy of machine-learning models [7]. Preprocessing is significant because it addresses common problems in raw data, such as noise, missing values, and inconsistencies. Therefore, preprocessing algorithms for ancient script recognition can improve the quality, accuracy, and efficiency of the recognition model.

II. METHODOLOGY

Six essential phases make up a standard character recognition model: Data acquisition or dataset creation, preprocessing, segmentation and augmentation, feature extraction, classifier training, and evaluation. Preprocessing is essential for preparing raw data to develop and train machine learning models. It is much more crucial in the case of ancient writings because they are not always discovered or preserved under the best conditions and may not be completely appropriate for a neural network model in its raw form [8, 9]. Preprocessing is further divided into various steps depending on the type of epigraph used. A combination of smoothing and character segmentation has been successfully implemented on stone inscriptions, whereas a combination of line and character segmentation is more successful in handwritten scriptures and manuscripts. Grayscaling transforms a regular RGB image into a grayscale and stores values in a single array. Thus, the calculation in a neural network involves only one convolution, as opposed to performing convolutions on three different arrays for an RGB image, improving computational efficiency. Image binarization is the process of converting a document image into a bilevel one, aiming to distinguish between the foreground and background text in a document [10].

Edge detection is an image processing technique to identify regions in a digital image with abrupt brightness fluctuations or other visual discontinuities. The areas where the brightness of the image varies significantly are the margins or boundaries of the image. It may be essential to crop an image by eliminating or changing its outer edges to enhance framing or composition, call attention to the image's subject, and change the size or aspect ratio. To improve an image, it must be cropped, which means removing unnecessary elements. If necessary, a skew correction is performed to rectify the text skew in an image that has a block of text rotated at an unidentified angle [11, 12]. This can be achieved by finding the text block in the image, calculating the text's rotational angle, and then rotating the image to account for the skew. A method of digital image processing known as image smoothing reduces and suppresses image noise. Line segmentation is used to identify and segment lines in the image or text block. Character segmentation also identifies and segments the characters that are present in the text in the image.

A. Dataset Creation and Preprocessing

To build a robust dataset for the classification of ancient Kannada stone inscriptions, a comprehensive process was followed that involved capturing, preprocessing, and organizing the images of Kannada manuscripts from historical places across Karnataka. The following steps outline the creation of the dataset.

1) Image Capture

Images were captured by visiting historical sites known for their ancient Kannada stone inscriptions. These sites included:

- Hampi, in the Bellary district of Karnataka.
- Belur, in the Hassan district of Karnataka.
- Somnath Temple, in the Mysore district of Karnataka.

A total of 2000 high-resolution images were captured using a 50-megapixel camera, representing a diverse collection of stone inscriptions found across these locations.

2) Image Preprocessing

After capturing the images, a series of preprocessing techniques were applied to prepare them for segmentation:

- Grayscale conversion: The images were converted to grayscale to simplify the subsequent processing steps.
- Noise reduction: Gaussian filters were used to reduce noise and smooth out any irregularities caused by the stone surface.
- Thresholding: Otsu's binarization was applied to create a binary image, making it easier to distinguish characters from the background.
- Resizing: Each image was resized to 52×52 pixels to standardize the dataset and make it suitable for training Convolutional Neural Networks (CNNs).

3) Character Segmentation

Once the images were preprocessed, individual characters were extracted using horizontally projected techniques. The segmentation process isolated each Kannada character from the inscriptions, which were then resized to a uniform size of 52×52 pixels.

4) Unicode Mapping

Each segmented Kannada character was assigned to its corresponding Unicode value from the Kannada Unicode block (U+0C80 to U+0CFF). This mapping ensured that the characters were properly labeled for future supervised learning tasks.

Preprocessing techniques can aid in the extraction of significant features and reduce the computational burden of subsequent processing steps by removing noise and artifacts and enhancing relevant information and image quality [13]. Additionally, preprocessing can standardize image dimensions and orientation, adjust for variations in lighting, and improve the visual appeal of the image[14].

B. Gray Scaling

There are several commonly used methods to convert an RGB image into a grayscale one, such as the average and weighted methods. This study used the average method, taking the average values of R, G, and B for the grayscale value [15].

$$Grayscale = (R + G + B)/3 \quad (1)$$

C. Binarization

The primary objective of binarization is to obtain the characteristics of the original image and eliminate any noise [16]. Image thresholding is used to binarize an image based on pixel intensities. As a result, each pixel will hold fewer data, simplifying subsequent computations [17]. A thresholding technique requires a threshold and a grayscale image to output a binary image. If a pixel's intensity in the input image exceeds a threshold, the associated output pixel is marked as white (foreground), and if it is less than or equal to the threshold, the corresponding output pixel is marked as black (background). The steps that Otsu thresholding algorithms usually take are as follows.

- Processing the supplied image.
- Obtain the pixel distribution histogram for the image.
- Find the threshold value T.
- Replace image pixels with white in areas where saturation is higher than T and black in areas where the contrary is true.

By minimizing the variance for each class, Otsu's method analyses the image histogram and segments the objects [18]. In most cases, this technique produces the desired results for bimodal images. The basic approach consists of dividing the image histogram into two clusters using a threshold that is chosen by minimizing the weighted variance of these classes. The general algorithm's pipeline for the between-class variance maximization option can be represented in the following way:

- Calculate the histogram and intensity level probabilities.
- Initialize ω_i (0), μ_i (0).
- Iterate over possible thresholds: $t = 0$, maximum intensity.
  - Update the values of ω_i, μ_i, where ω_i is a probability and μ_i is a mean of class i.
  - Calculate the between-class variance value $\sigma_b^2(t)$
- The final threshold is the maximum $\sigma_b^2(t)$ value.

D. Edge Detection and Cropping

Edge detection involves locating the borders or edges of items in an image. This assists in recognizing the key elements of an image, making it a crucial step in image processing and computer vision. In edge detection, high-intensity gradients in an image are identified and designated as edges, helping to improve the accuracy and efficiency of the subsequent processing steps [19]. The Canny edge detection model was the most efficient for the model used to recognize manuscripts and inscriptions, which is widely utilized in several disciplines. An image edge recognition technique based on the Canny algorithm was presented to eliminate pepper salt noise and extract edge information of the area of interest [20]. This is a multi-stage algorithm to locate and/or identify different types of edges. The key steps used in implementing the Canny algorithm are as follows:

- The image should be in grayscale.
- Edge identification using derivatives is noise-sensitive, so reducing noise is crucial.
- Determine the gradient to establish the edge's direction and intensity.
- Non-maximum suppression is applied to soften the edges of the image.
- Use a twofold threshold to differentiate between the images' strong, weak, and irrelevant pixels.
- Hysteresis edge tracking helps weak pixels become strong ones when there is a strong pixel nearby.

E. Smoothing

Smoothing is performed before segmenting the text within the dataset for epigraphs in the form of stone inscriptions. Smoothing is used to create images that are less noisy and pixelated [21]. Most smoothing techniques use low-pass filters, but this study used a kernel, a moving collection of pixels, by averaging or median the group of pixels to smooth an image. Picture smoothing aims to maintain image quality while reducing noise without affecting its main aspects. These noises could be additive, impulsive, multiplicative, or of any other form. By removing noise, smoothing the inscriptions and epigraphs makes it possible to segment text more effectively.

F. Line and Character Segmentation

Line-level segmentation aims to divide the image into lines. The idea is that the rows that correspond to each line of text have a higher proportion of foreground pixels, which, when the binary image is horizontally projected, corresponds to higher peaks in the histogram. The rows that show the intervals between the lines, which correlate with lower histogram peaks, include many background pixels. It is feasible to segment lines by selecting rows as the segmenting lines that correspond to the lower peaks of the histogram [22]. Character segmentation divides an image of a group of characters into smaller images that represent the symbols individually. At this level of segmentation, a single word that was previously segmented and is made up of a string of characters is shown as an image. Thus, text line segmentation is crucial to the overall effectiveness of a document recognition system [23].

Character segmentation is an important phase in OCR, as the separation of broken characters is a crucial factor in determining how well a recognition system works [24]. Character-level segmentation aims to separate the characters in the image's text into separate groups. This level of segmentation may or may not be necessary depending on the circumstances surrounding the OCR application. If OCR technology is used on text and words that have separate characters, character-level segmentation is not required. Since a uniform space is maintained between the letters within a word, no matter how small it may be, characters can be segmented in the previous stage (by selecting a very low threshold). On the other hand, if the text and characters inside a word are joined (cursive handwriting), character-level segmentation must be performed. To determine the segmentation sites, the segmentation process uses geometry and shape. Word image thinning is then used to obtain the width of a pixel's worth of stroke and locate the ligatures of Kannada letters [25]. Character-level segmentation is crucial for Kannada scripts to recognize the more intricate and densely located characters. In contemporary Kannada, it is possible to recognize segmented characters and process, categorize, and store them correctly.

III. RESULTS AND DISCUSSION

The above-mentioned preprocessing techniques were performed on different noise-level images.

A. Input Image

Capturing high-quality images of stone inscriptions is a crucial first step. Figure 1 presents a low-noise stone inscription, whereas Figures 2 and 3 present a moderate and a high-noise input image.

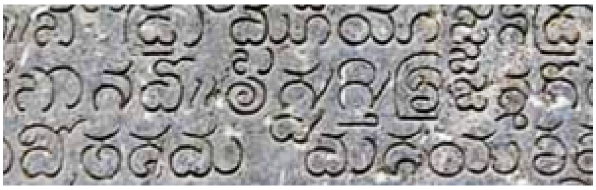

Fig. 1. Input image with low-noise stone inscription.

Fig. 2. Input image with moderate-noise stone inscription.

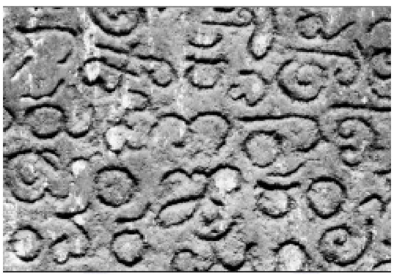

Fig. 3. Input image with high-noise stone inscription.

B. Gray Scale Conversion

Figure 4 presents a low-noise grayscaled stone inscription, whereas Figures 5 and 6 present grayscaled images of moderate and high-noise grayscale stone inscriptions.

Fig. 4. Grayscale image with low-noise stone inscription

Fig. 5. Grayscale image with moderate-noise stone inscription.

Fig. 6. Grayscale image with high-noise stone inscription

C. Binarization

Figure 7 represents a low-noise binarized image of a stone inscription, whereas Figures 8 and 9 present binarized images of moderate and high-noise stone inscriptions, respectively.

D. Edge Detection

Figure 10 shows a Canny edge detection image for a low-noise stone inscription, whereas Figure 11 shows a Canny edge detection image for a moderate noise stone inscription and Figure 12 shows a Canny edge detection image for a high-noise stone inscription.

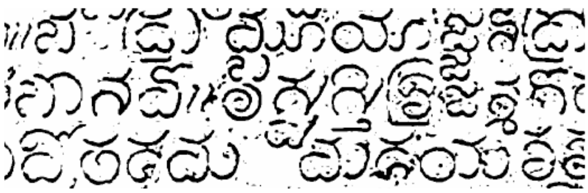

Fig. 7. Binarized image of a low-noise stone inscription.

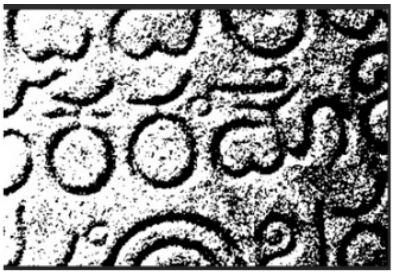

Fig. 8. Binarized image of a moderate-noise stone inscription.

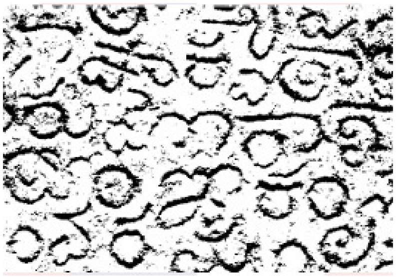

Fig. 9. Binarized image of a high-noise stone inscription.

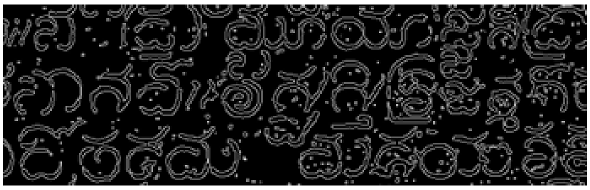

Fig. 10. Canny edge detection image of a low-noise stone inscription.

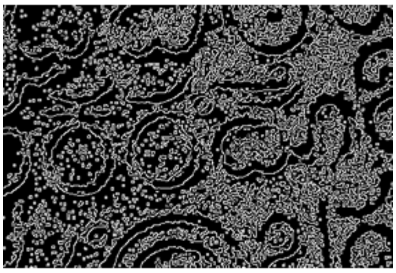

Fig. 11. Canny edge detection image for a moderate-noise stone inscription.

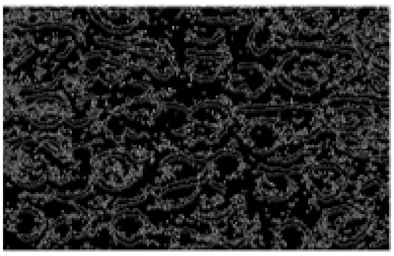

Fig. 12. Canny edge detection image of a high-noise stone inscription.

E. Smoothening

Figure 13 presents a smoothened image for a low-noise inscription, whereas Figures 14 and 15 shows smoothened stone inscriptions of moderate and high noise, respectively.

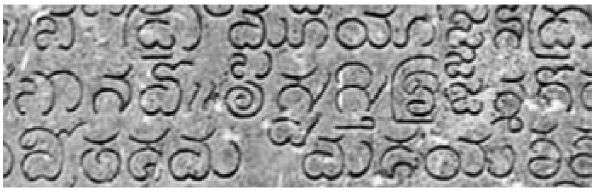

Fig. 13. Smoothened image of low-noise stone inscription.

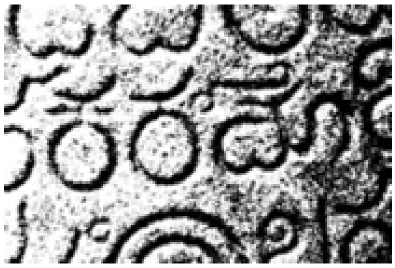

Fig. 14. Smoothened image of moderate-noise stone inscription.

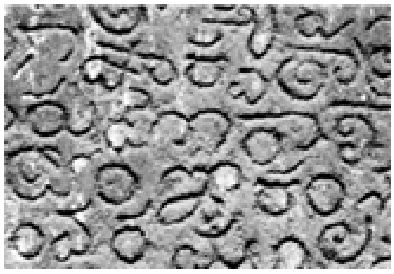

Fig. 15. Smoothened image of high-noise stone inscription.

F. Character Segmentation

Figure 16 presents a character-segmented image for a low-noise stone inscription, whereas Figures 17 and 18 present character-segmented images for moderate and high-noise stone inscriptions, respectively.

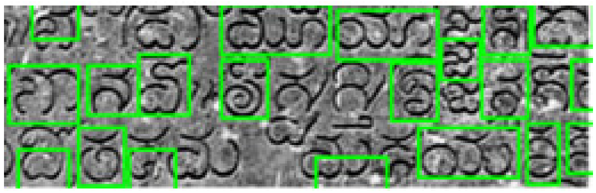

Fig. 16. Character segmentation image of low-noise stone inscription.

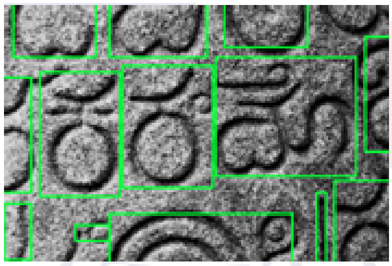

Fig. 17. Character segmentation image of moderate-noise stone inscription.

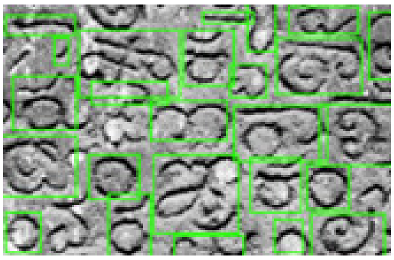

Fig. 18. Character segmentation image of high-noise stone inscription.

IV. MODEL SUMMARY

To evaluate the model's performance on low, medium, and high-noise Kannada stone inscriptions, image degradation was assessed quantitatively by measuring Noise Level, PSNR (Peak Signal-to-Noise Ratio), and SSIM (Structural Similarity Index Measure). These metrics are critical in understanding how noise impacts accuracy, particularly under high noise conditions.

1) Noise Levels in Low, Medium, and High-Noise Kannada Stone Inscriptions

- Low Noise: Very low noise level, possibly about 10% Gaussian noise.
- Moderate Noise: Moderate noise level, about 15% Gaussian noise.
- High Noise: High noise level, typically more than 30% Gaussian noise, which can severely obscure the characters.

2) PSNR (Peak Signal-to-Noise Ratio)

PSNR is used to measure the quality of the degraded image compared to the original. Higher values indicate less degradation.

3) SSIM (Structural Similarity Index Measure)

SSIM assesses perceptual similarity, comparing the luminance, contrast, and structure of the original and degraded images. SSIM ranges from -1 to 1, with values closer to 1 indicating greater similarity.

4) Quantified Analysis of Kannada Stone Inscriptions for Various Noises

- Low Noise (10% Noise Level)
  - PSNR: 19.98 dB
  - SSIM: 0.084
  - Interpretation: A low noise level causes very little degradation and retains the same character structure.
- Moderate Noise (15% Noise Level)
  - PSNR: 17.65 dB
  - SSIM: 0.064
  - Interpretation: A moderate noise level causes noticeable degradation but retains some character structure.
- High Noise (30% Noise Level)
  - PSNR: 15.36 dB
  - SSIM: 0.031
  - Interpretation: A high noise level severely degrades the image, making character details indistinct.

TABLE I. COMPARATIVE ANALYSIS OF KANNADA STONE INSCRIPTION FOR VARIOUS NOISES

Noise	PSNR	SSIM
Low noise	20.98	0.084
Moderate noise	17.65	0.064
High noise	15.36	0.031

The experimental results show that PSNR falls below 20 dB and SSIM falls below 0.1 (as in fully degraded conditions) for high-noise images, which may require much more preprocessing than the scriptures with low noise levels and comparatively better quality.

V. CONCLUSION

This paper discussed the importance of ancient script recognition and the challenges it poses, particularly in the case of the old Kannada script. The proposed solution for ancient scripts consists of various preprocessing techniques, such as gray scaling, binarization, edge detection and cropping, smoothing, line segmentation, and character segmentation. This technology can be used for various ancient epigraphs depending on noise levels, which means that all preprocessing steps may not need to be performed on all images. Since the key motive behind developing this solution was to improve the quality of the image before it is fed into a recognition model, it can be concluded that inscriptions with higher noise levels, or belonging to older dynasties, may require much more preprocessing than those with low noise levels and comparatively better quality. The techniques and methods mentioned above were successful in improving image quality, making it computationally more efficient when fed to a neural network. They can be used in recognition models not just for ancient Kannada epigraphs but also other epigraphs or ancient scripts, which may be using neural networks such as CNNs or any machine learning algorithms. The emphasis is likely to be on enhancing accuracy, adaptability, and integration with other modalities and techniques, as ancient Kannada script recognition continues to be essential for the preservation of knowledge and culture. With the development of machine learning and artificial intelligence techniques, more sophisticated preprocessing methods are anticipated to recognize the unique qualities of each image and learn more intricate features from them. These developments are likely to increase accuracy rates and make OCR systems more reliable, which will ultimately help businesses, organizations, and people who use OCR technology for a variety of purposes.

REFERENCES

[1] S. N. Karnam, V. S. Vaddagallaiah, P. K. Rangnaik, A. Kumar, C. Kumar, and B. M. Vishwanath, "Precised Cashew Classification Using Machine Learning," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17414–17421, Oct. 2024, https://doi.org/10.48084/etasr.8052.

[2] S. Chadha, S. Mittal, and V. Singhal, "Ancient Text Character Recognition Using Deep Learning," *International Journal of Engineering Research and Technology*, vol. 13, no. 9, Sep. 2020, Art. no. 2177, https://doi.org/10.37624/ijert/13.9.2020.2177-2184.

[3] G. Kumar, P. K. Bhatia, and Indu, "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition," 2013, https://doi.org/10.13140/RG.2.1.3896.7842.

[4] P. Mishra, P. Pai, M. Patel, and R. Sonkusare, "Extraction of Information from Handwriting using Optical Character recognition and Neural Networks," in *2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, Nov. 2020, pp. 1328–1333, https://doi.org/10.1109/iceca49313.2020.9297418.

[5] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, https://doi.org/10.1109/access.2020.3012542.

[6] R. Dey, R. C. Balabantaray, S. Mohanty, D. Singh, M. Karuppiah, and D. Samanta, "Approach for Preprocessing in Offline Optical Character Recognition (OCR)," in *2022 Interdisciplinary Research in Technology and Management (IRTM)*, Kolkata, India, Feb. 2022, pp. 1–6, https://doi.org/10.1109/irtm54583.2022.9791698.

[7] S. Joseph and J. George, "Handwritten Character Recognition of MODI Script using Convolutional Neural Network Based Feature Extraction Method and Support Vector Machine Classifier," in *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, Nanjing, China, Oct. 2020, pp. 32–36, https://doi.org/10.1109/icsip49896.2020.9339435.

[8] M. Brisinello, R. Grbic, M. Pul, and T. Andelic, "Improving optical character recognition performance for low quality images," in *2017 International Symposium ELMAR*, Zadar, Croatia, Sep. 2017, pp. 167–171, https://doi.org/10.23919/elmar.2017.8124460.

[9] G. V. S. S. K. R. Naganjaneyulu, A. V. Narasimhadhan, and K. Venkatesh, "Performance evaluation of OCR on poor resolution text document images using different pre processing steps," in *TENCON 2014 - 2014 IEEE Region 10 Conference*, Bangkok, Thailand, Oct. 2014, pp. 1–4, https://doi.org/10.1109/tencon.2014.7022357.

[10] W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image Preprocessing for Improving OCR Accuracy," in *2007 International Conference on Perspective Technologies and Methods in MEMS Design*, Lviv, Ukraine, May 2007, pp. 75–80, https://doi.org/10.1109/memstech.2007.4283429.

[11] C. Chinara, N. Nath, S. Mishra, S. K. Sahoo, and F. A. Ali, "A novel approach to skew-detection and correction of English alphabets for OCR," in *2012 IEEE Student Conference on Research and Development (SCOReD)*, Pulau Pinang, Malaysia, Dec. 2012, pp. 241–244, https://doi.org/10.1109/scored.2012.6518646.

[12] M. Sarfraz, A. Zidouri, and S. A. Shahab, "A Novel Approach for Skew Estimation of Document Images in OCR System," in *International Conference on Computer Graphics, Imaging and Visualization (CGIV'05)*, Beijing, China, pp. 175–180, https://doi.org/10.1109/cgiv.2005.6.

[13] Md. M. Hasan, H. Ali, Md. F. Hossain, and S. Abujar, "Preprocessing of Continuous Bengali Speech for Feature Extraction," in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, Jul. 2020, pp. 1–4, https://doi.org/10.1109/icccnt49239.2020.9225469.

[14] A. V. Kottath and S. Shri Bharathi, "Image Preprocessing Techniques in Skin Diseases Prediction using Deep Learning: A Review," in *2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA)*, Coimbatore, India, Sep. 2022, pp. 1–6, https://doi.org/10.1109/icirca54612.2022.9985547.

[15] M. I. Shah and C. Y. Suen, "Word Spotting in Gray Scale Handwritten Pashto Documents," in *2010 12th International Conference on Frontiers in Handwriting Recognition*, Kolkata, India, Nov. 2010, pp. 136–141, https://doi.org/10.1109/icfhr.2010.28.

[16] P. Ranjitha and T. D. Shreelakshmi, "A Hybrid Ostu based Niblack Binarization for Degraded Image Documents," in *2021 2nd International Conference for Emerging Technology (INCET)*, Belagavi, India, May 2021, pp. 1–7, https://doi.org/10.1109/incet51464.2021.9456150.

[17] B. Vishnudharan and K. Anusudha, "Reclamation of information from degraded and damaged document images by image binarization method," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, Mar. 2017, pp. 1–4, https://doi.org/10.1109/iciiecs.2017.8276100.

[18] R. G. G. Acuna, Junli Tao, and R. Klette, "Generalization of Otsu's binarization into recursive colour image segmentation," in *2015 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, Auckland, New Zealand, Nov. 2015, pp. 1–6, https://doi.org/10.1109/ivcnz.2015.7761549.

[19] S. Wang, K. Ma, and G. Wu, "Edge Detection of Noisy Images Based on Improved Canny and Morphology," in *2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Yunlin, Taiwan, Oct. 2021, pp. 247–251, https://doi.org/10.1109/ecice52819.2021.9645601.

[20] Y. Li and B. Liu, "Improved edge detection algorithm for canny operator," in *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Chongqing, China, Jun. 2022, pp. 1–5, https://doi.org/10.1109/zitaic54216.2022.9836608.

[21] P. Li, H. Wang, M. Yu, and Y. Li, "Overview of Image Smoothing Algorithms," *Journal of Physics: Conference Series*, vol. 1883, no. 1, Apr. 2021, Art. no. 012024, https://doi.org/10.1088/1742-6596/1883/1/012024.

[22] B. Gatos, G. Louloudis, and N. Stamatopoulos, "Segmentation of Historical Handwritten Documents into Text Zones and Text Lines," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Greece, Sep. 2014, pp. 464–469, https://doi.org/10.1109/icfhr.2014.84.

[23] K. Mullick, S. Banerjee, and U. Bhattacharya, "An efficient line segmentation approach for handwritten Bangla document image," in *2015 Eighth International Conference on Advances in Pattern Recognition (ICAPR)*, Kolkata, India, Jan. 2015, pp. 1–6, https://doi.org/10.1109/icapr.2015.7050679.

[24] R. J. Shah and T. V. Ratanpara, "Challenges of broken characters in character segmentation method for Gujarati printed documents," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, India, Mar. 2015, pp. 1–5, https://doi.org/10.1109/iciiecs.2015.7193263.

[25] A. A. A. Ali and M. Suresha, "An Efficient Character Segmentation Algorithm for Recognition of Arabic Handwritten Script," in *2019 International Conference on Data Science and Communication (IconDSC)*, Bangalore, India, Mar. 2019, pp. 1–6, https://doi.org/10.1109/icondsc.2019.8817037.