Natural Resources for Human Health
eISSN: 2583-1194

Original Research

Received 21 May 2026
Revised 27 June 2026
Accepted 22 July 2026

Natural Resources for Human Health 2026, Vol. 6 (10s): 996–1026

KEYWORDS:
Brahmi Inscriptions, Convolution Neural Networks, U-Net, Segmentation, Transfer Learning.

Segmentation of Ancient Brahmi Inscription Image Using Deep Learning Approach

Sandeep Kaur[1], Deepak Kumar Verma[2], Gurrajan Singh[3], Om Prakash Suthar[2], Chaste Sauveur Uwambazimana[4], Chetankumar Chudasama[2] Rahul Bhandari[5]

[1]Department of CSE, Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India
[2]Department of Computer Engineering, Marwadi University, Rajkot, Gujarat, India
[3]Adobe Systems, Noida, Uttar Pradesh, 201301, India
[4]Washington, District of Columbia, USA
[5]Department of CSE, Chandigarh University, Mohali, Punjab, India

ABSTRACT: Segmentation of inscriptions from Ancient Brahmi scripts poses a significant challenge due to image quality, demand for greater data accessibility, and inadequate segmentation of inscriptions. Ancient Brahmi scripts serve as a progenitor for many modern-day scripts and are inscribed on stones, pillars and tablets in India and neighboring countries. Traditional image segmentation methods utilizing thresholding, edge detection, and clustering have considerably poor performance on inscriptions with artifacts such as cracks, uneven illumination, and varying surface geometries. In order to address the above limitations, we developed a segmentation method using the U-Net model with transfer learning based on pre-trained VGG-16 with heavy data augmentation. Images of inscriptions were divided into patches of size 128×128 to accommodate memory constraints. The proposed method showed significantly improved segmentation of Brahmi characters, with a total of 42 characters correctly segmented, compared to 34 characters with U-Net, and 29 characters with traditional image segmentation methods. Patch-wise and section-wise segmentation were carried out to evaluate the robustness of the segmentation model on real world illumination conditions and varying surface geometries. The proposed method achieved a Dice score of 0.714. In addition, high precision (0.893), recall (0.800), and F1-score (0.844) demonstrated that the segmentation method is robust on inscriptions with cracks and noise. The segmentation method combines transfer learning and data augmentation for segmentation of ancient scripts, and offers a potentially reliable and effective solution for the digitization of archaeological texts and advanced studies in computational epigraphy.

1. INTRODUCTION

Before the invention of paper, ancient civilizations had stones, columns, and leaves as their chief means for writing and communicating text. Analyzing these inscriptions is vital for the study of history, sociology and anthropology. The inscriptions found in ancient structures, buildings and caves are the knowledge etched on pillars, stones and walls. The walls of these structures are often etched using advanced digital imaging techniques and are preserved and recorded more easily in the modern era. The deciphering process, however, is still very manual and often tedious, as historians and scholars must spend a great deal of time interpreting these texts and articulating their meanings and the importance of these inscriptions to their respective cultures [1, 2].

The leaps made in modern imaging and deep learning to analyze texts have left the inscriptions of ancient civilizations relatively unexplored. The analysis of ancient inscriptions and texts is, for the most part, a very manual process, and this is largely due to the poor and often fragmented quality of inscriptions caused by ancient environmental and anthropological factors on the

inscriptions such as cracks and uneven surfaces [3]. The modern imaging techniques and imaging systems that are present today have not advanced enough to fulfil the needs of modern historians and archaeologists to analyze texts from ancient inscriptions.

This research presents innovative solutions to specific issues by employing advanced deep learning algorithms to segment and identify individual characters from inscription images. More specifically, here, we outline a U-Net architecture based on Convolutional Neural Networks (CNN); hereafter, when we refer to U-Net, we will omit full form and diction (i.e. U-Net, U-shaped Convolutional Neural Network [18]) macron. This architecture is designed for segmentation tasks and originally was developed for biomedical imaging, as a layered (encoders and decoders) segmentation architecture. This structure is effective for segmentation tasks. Traditional image processing systems are highly flexible and rely on various algorithms to function; our structure, designed specific to localize and segment Brahmi inscriptions, is highly effective. Text segmentation of images is a highly problematic domain, as character segmentation is further complicated by variable inscriptions, image artifacts, and background noise. In this work, we improve segmentation accuracy by incorporating transfer learning and patching along with appropriate data augmentation techniques to address the severe constraints of existing Brahmi script inscription datasets.

This work offers the following:

- *Enhanced Deep Learning for Brahmi Script:* Proposes a U-Net based CNN for Brahmi inscription segmentation over background noise and artifacts.
- *Transfer Learning Using VGG-16:* Applies VGG-16 for feature extraction and better segmentation, while addressing overfitting due to reduced data availability.
- *Data Augmentation for Increased Variation:* Augmented data were utilized for segmentation, including variations in background illumination and positioning of the inscriptions.
- *Patch Based Training:* Segmented larger inscription images into smaller patches (i.e. 128×128), which reduced the memory requirements and increased usable data for training.
- *High Character Segmentation Accuracy:* Segments 42 out of 49 Brahmi characters, surpassing baseline U-Net (34) and classical image processing (29).
- *Section-wise and Patch-level Consistency:* Demonstrates high Dice score and segmentation quality across all regions and is robust to local lighting and surface variations.
- *Enhanced Performance Relative to Available Techniques:* Exceeds performance of traditional thresholding, edge detection, clustering, and contemporary deep learning techniques, achieving 94.2% accuracy and the highest F1-score.

Although our technique primarily concentrates on Brahmi script inscriptions, it can also be applied to other ancient scripts. The U-Net version of the architecture presented in this paper has a typical encoder-decoder structure as seen in biomedical image segmentation and satellite image analysis. Inscriptions are segmented by the model, which generates an accurate mask for the input image that is subsequently refined by image-processing techniques to extract and scope individual characters [23, 28].

This study was the first to apply U-Net architecture for analyzing inscriptions and demonstrated a valuable middle ground between orthogonal methods. The results show that the combination of transfer learning with data augmentation lessens the impediment of limited datasets and achieved excellent results in character segmentation. The proposed method may play a key role for the conservation and research of ancient scripts and cultures. This may initiate a large number of new research opportunities in the field of historical text research and computational archaeology. The cited works [28–38] are of great importance to this research. They may yield a better understanding of segmentation, transliteration, and other components of the Brahmi and other ancient scripts, as well as improved optical character recognition (OCR) systems. In a broader context, these works help facilitate a shift from non-computational methods to an automated framework for historical scripts and their recognition.

The rest of this document does the following: Survey the Brahmi-indic inscription studies, image segmentation studies, and deep learning applications to ancient scripts in study 2. Present the proposed U-Net model adapted with transfer learning and describe the method in study 3. Present the set of annotation, segmentation, and augmentation techniques developed with deep learning in study 3 as well. Describe the design of the experiments and analyze the results of the experiments both qualitatively and quantitatively in study 4. Present the results of the experiments and the proposed model's strengths and weaknesses in study 5. Present the practical implications of the proposed model and the ancient inscription analysis in study 5. Discuss the

contributions of the paper as well as possible future study directions in computer epigraphy and the preservation of digital heritage in study 6.

2. RELATED WORK

OCR facilitates the extraction and conversion of computer-based systems from printed or handwritten sources such as images or scanned documents in digital format [2]. Pattern recognition, image processing, and machine learning techniques have been developed to enable the accurate character recognition of most modern-day scripts. Segmentation is a critical phase of an OCR system that breaks the entire document into subparts involving line-level, word-level, and character-level segmentation. The main aim of segmentation is to label and extract desired regions from an image. Segmentation techniques range from various image processing methods, such as edge detection, region-based clustering, thresholding, and neural-network-based segmentation [3]. The image processing method works very well for printed text segmentation; however, when used for handwritten text and especially ancient inscription images, segmentation is significantly affected by variations and noise in the input image. Although significant research has been conducted on the segmentation of modern-day scripts [4, 5], there has hardly been any notable research on Brahmi inscription segmentation. The traditional image-processing-based projection profile segmentation approach has been used for Brahmi inscription segmentation [6]; however, this approach only performs well if the inscription has cracks or variable illumination across the image surface owing to the inscription curvature. Template matching to recognize the Tamil Bamini font was presented in [7], where a template database was initially created. The input document is then converted into a grayscale and binary format. Horizontal projection segment lines are followed by connected component labeling to separate individual characters. Again, this process works well for printed text images but does not segment accurately if the input text image has cracks and variable illumination.

Furthermore, methods using Estampage and techniques such as histogram projection, run-length encoding, and Hough transform methods for printed text segmentation and hybrid CNN and SVM models have been proposed for inscription images to classify the text and non-text regions of the image by utilizing several overlapping patches of Estampage images [8]. However, it has limitations because it requires characters to be surrounded by a specific background, such as white. CNN has grown significantly in popularity in computer vision, and various CNN-based models/architectures have been created, such as ResNet, LeNet, and GoogLeNet, which have performed exceptionally well in computer vision tasks [9]. CNN models have frequently been employed for image classification, and deep CNNs have been developed to outperform traditional CNN architectures in image classification tasks [10]. Furthermore, novel and modified CNN architectures have been developed, such as U-Net, which generates a mask for an input image that highlights various objects in images, and the generated mask is used for segmentation. The U-Net architecture includes additional layers, including deconvolution and concatenate layers; thus, the output and input images have exact dimensions. The U-Net architecture performs exceptionally well in segmenting biomedical images [11]. However, using CNN for segmentation requires new and improved loss functions, as the traditional classification-based loss function does not efficiently train a CNN for segmentation tasks [12]. The authors surveyed various deep learning techniques for image segmentation [13], which showed that deep learning and CNN-based methods are gaining prominence in ancient image segmentation. However, a critical requirement of CNN-based models is a sufficient dataset that can be used to train the CNN model. This is a limitation for ancient scripts, such as Brahmi, as no standard dataset is available. In such cases, data augmentation techniques are employed to generate more data from a limited dataset [14].

To overcome the limitations of a large dataset, one of the techniques proposed is character spotting, which requires a user to mark a character on the inscription and link it to a previously defined Unicode character. The HOG feature set of the spotted characters is then used to identify similar characters from the inscription image and mark them as spotted. The procedure was continued until all inscription image characters were marked [15]. However, this is not extendible in most inscriptions, such as the Asoka pillar shown in Figure 1, where each character has different illumination caused by curvatures and sunlight because a single engraved character has variable shading and brightness across the character surface. Moreover, this technique requires human intervention, which is occasionally possible. Another critical aspect of training a convolutional network for an image-processing task is the memory requirement. The memory requirement increases significantly when the input image exceeds 1k in size. To overcome this limitation, patches of the input image can be used for training rather than the entire or downsampled image. In addition to optimizing the memory requirements, this also creates additional training data from a single input image [16]. The accuracy of U-Net segmentation has been significantly improved by utilizing a transfer learning approach in which a pre-trained network, such as VGG16, can serve as the backbone of U-Net's encoder layer. This also overcomes the limitation

of a substantial amount of data for model training, as the network starts with a pre-trained set of weights rather than random weights [17, 18]. A detailed study of the CNN architecture was presented in [19], where the network employed only convolution, pooling, and activation functions to generate a pixel-to-pixel map for semantic segmentation tasks. In addition, the author highlighted that transfer learning can further improve segmentation accuracy by utilizing pre-trained classification networks such as VGG and GoogLeNet.

In addition to medical image segmentation, UNet-based CNN models have shown promising performance for identifying holes in the medium-density fiberboard (MDF) [20] and analysis of digital images of rock to identify various layers present in rock [21]. However, inscription images differ from MDF boards and rock images because of cracks, smudges, and variable lighting. However, studies in related work indicate that U-Net may be used to carry out ancient image segmentation tasks. Moreover, processing the segmented characters can further extract individual character strokes using techniques such as Maya glyph extraction from ancient documents [22]. Segmentation is critical for developing a character recognition system for any language or script. Further, a comparative study of the reviewed literature for the segmentation of Brahmi inscriptions is presented in Table 1, which reveals that most existing image processing methods work well for printed text but struggle with handwritten and ancient inscription images owing to variations and noise. Furthermore, CNN-based approaches have shown remarkable performance and become popular. Specifically, U-Net-based architectures are the most suitable because they can work with limited training data, generate pixel-level segmentation masks, and perform well in challenging scenarios. Moreover, incorporating transfer learning and pre-trained weights (such as VGG16 trained on ImageNet) may significantly improve the performance of ancient inscription image segmentation.

Table 1. Comparative Analysis of Reviewed Literature on Brahmi Inscription Segmentation

Ref. No.	Contribution	Methodology	Research Gap	Remark
[41]	Analyzes various segmentation methods for Brahmi inscriptions (precisely the Rumandei inscription).	horizontal and vertical projection profiles	There is no uniform method, Limited accuracy.	Need for uniform segmentation methods. Advanced character recognition techniques.
[42]	Segmenting lines and characters in degraded Brahmi script images.	Line Segmentation.	Limited accuracy. Limited exploration of isolated symbol recognition techniques.	Limited segmentation accuracy. Explore advancements in character recognition systems.
[43]	Image processing for segmenting Brahmi letters.	Image Processing Techniques, CNN	No automatic system for translating Brahmi to Sinhala exists. Manual translation.	Inappropriate, non-scalable.
[44]	Organizes and visualizes scattered Brahmi script using VisualEyes.	Data Collection and Aggregation, Use of VisualEyes Tool.	Data dependency. Scope of Multimedia Integration.	It may be expanded to other scripts.
[45]	A clustering approach for segmenting characters and text lines in epigraphical scripts, including challenges like skewed documents and uneven spacing.	Nearest Neighbor Clustering Method, Handling Skewed Documents	Complexity of Epigraphical Scripts. Scalability.	Integration of more advanced clustering algorithms.

[46]	Deep learning-based segmentation for digitization and preservation of Tamil inscriptions.	Convolutional Neural Networks (CNNs), U-Net, Attention-Based Mechanisms.	Noise handling. Image degradation.	Use advanced DL approaches to improve performance.

2.1 Brahmi script and inscriptions

Many present-day Asian scripts have roots in the Brahmi script [23], as Brahmi is considered a parent script that led to the evolution of scripts such as Bengali, Devanagari, Kannada, Tamil, Telugu, Gurmukhi, and Odia. The Brahmi script is an alpha-syllabic writing system because its vowels and consonants are distinguishable from writing in horizontal lines from left to right. Figure 2 shows the primary characteristics of the Brahmi script. The first nine letters of the Brahmi script are vowels, and the remaining letters are consonants. Brahmi script came into existence during Ashoka's reign.

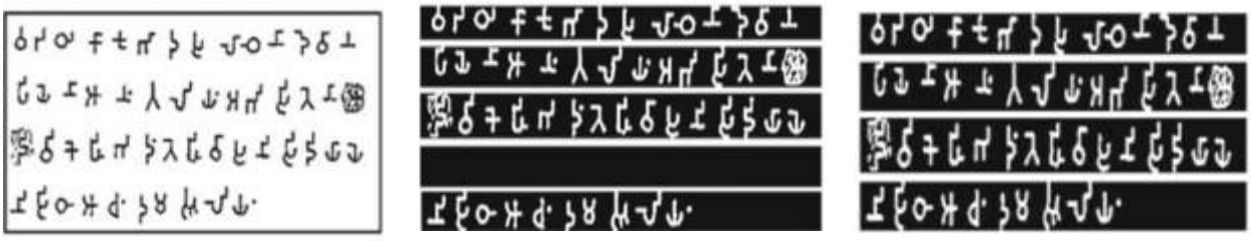
Ancient scriptures, such as the teachings of the Buddha, have been recorded in stone inscriptions using the Brahmi script. These inscriptions, known as the Ashoka edicts, are present throughout the Indian subcontinent. Inscriptions and manuscripts are significant because they are an essential part of cultural heritage and have enabled the preservation of ancient knowledge, history, and culture across centuries. Scholars, religious visitors, and historians spend significant time visiting these sites and deciphering ancient texts. Owing to the variations in scripts and symbols, archaeological experts need considerable effort to read inscriptions and manually identify symbols and characters. Therefore, there is a critical need to preserve these ancient inscriptions and promote ancient heritage.

2.2 Inscription images

Tourists and scholars visiting these inscription sites have captured camera images of ancient inscriptions. Although this enables the preservation and sharing of the inscription text, understanding the text remains challenging. Image segmentation can extract the individual text characters from the inscription images and help develop an optical recognition system for Brahmi. Moreover, tourists and religious visitors can use the segmentation process to extract individual script characters from the inscription images and help them understand these invaluable ancient texts.

2.3 Challenges in segmenting inscription image

Inscriptions are subjected to extreme environmental conditions, leading to cracks, smudges, and other deformities. This is evident in photographs of the inscriptions researchers and historians have captured from pillars and stones. Additionally, some of these photographs may require higher resolution and cause problems owing to their numerous visual flaws. The cuts and cracks in the inscriptions that appear as part of the image's characteristics are typical flaws. Additionally, inscription curvature and sunlight can lead to shadows and bright spots on some characters. Figure 1 displays a picture of the Brahmi pillar inscription and Brahmi script on the Ashoka pillar in Sarnath, which has some breaks, stains, and brightness variations caused by the daylight and camera angle. Figure 2 shows the printed Brahmi character set.



(a) (b) (c)

Figure 1: (a) Digital estampage image of Brahmi script, (b) line images after successful segmentation,

(c) line images after successfully removing blank lines. Source ASI, Mysore, India.

2.4 Existing methods for image segmentation

The segmentation of specific objects in an image is the first step towards developing computer vision-based applications. Optical character recognition systems depend on segmentation accuracy to extract and identify individual characters from

textual images. This study used the U-Net architecture to explore image-processing methods involving thresholds, morphological operations, contour plotting, and deep learning techniques.

2.5 Image Processing-based Techniques

2.5.1 Adaptive Thresholding

Segmentation by thresholding involves assigning each pixel value to one of two classes depending on a specific threshold value (h). A threshold function f assigns a set of intensities to the pixel values in an image such that:

$$f(x) = \begin{cases} x_1, & x < h \\ x_2, & x \geq h \end{cases} \quad (1)$$

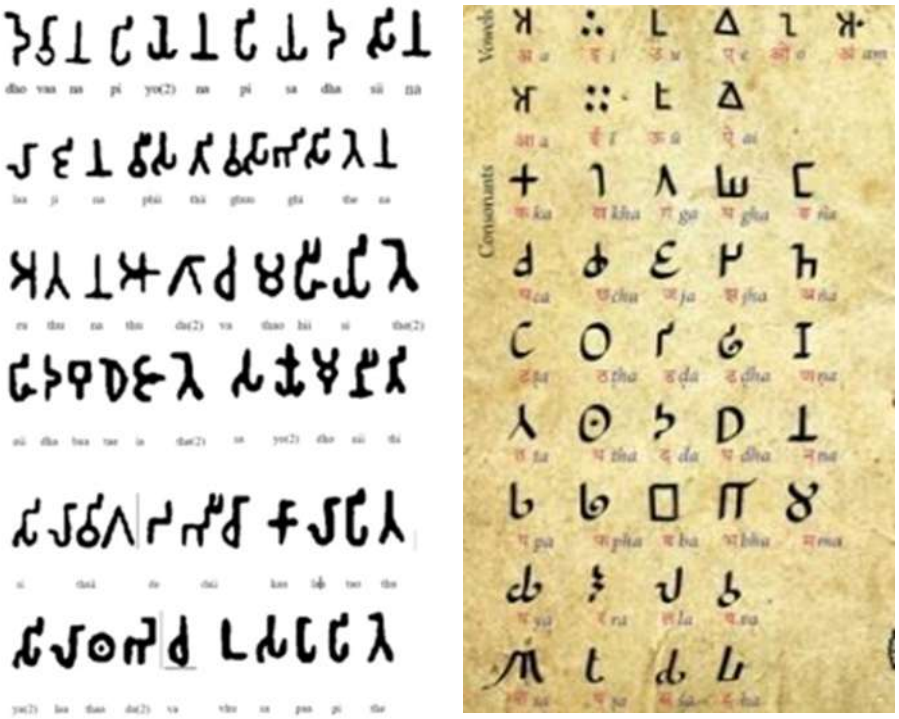
Where $0 < h \leq \max(x)$

Specific values of x_1 and x_2 can yield different results. For example, binarization can be obtained by setting $x_1 = 0$ and $x_2 = 1$.

2.5.2 Morphological modification

A morphological operation is performed on a binary image to reduce the impact of noise, such as white noise, or to join broken sections in the foreground. Based on a kernel or structuring element, two basic operations–erosion followed by dilation–are performed on the threshold image. The kernel is generally a square matrix of odd dimensions with each element set to one. In both operations, the kernel slides over the image, and the final pixel centered in the kernel is determined based on the cumulative pixel values under the kernel. The erosion operation sets a pixel value to '1' if all pixel values under the kernel equal 1. Dilation operation sets pixel value to '1' if any pixel under kernel has a value equal to 1.

Adaptive thresholding is used to handle varying intensities across the images. The global threshold considers a specific threshold for the entire image, whereas the adaptive thresholding determines the optimal threshold value for a small set of neighboring pixels. This automatically adjusts the threshold value across various image sections based on the local pixel intensity.



(a) (b)

Figure 2: (a) Brahmi printed characters [39], (b) Character and vowels of Brahmi script [40]

2.5.3 Contours

The set of points connecting the boundary pixels of an object is known as a contour. The contour tracing method can be used to separate specific foreground pixels from the background such that all image pixels having the same color and intensity can be bounded by drawing a contour over the original image. Various contour detection methods based on pixel, edge, and region-based detection are presented in [24]. The process used in this study included converting the image to grayscale and then applying thresholding to translate the image into a binary format. While retrieving the contours, only the external contours are extracted; that is, any contour included inside another was ignored. Finally, only the corner-most contour points are used to create a bounding rectangle over the area of interest.

2.6 Convolutional Neural Network

A subclass of feed-forward neural networks is the convolutional neural network (CNN), which is widely used for image analysis and processing. In recent years, CNN has exceeded human-level accuracy in image recognition and analysis tasks. A CNN differs from a regular feedforward network in terms of having individual neurons arranged in a multidimensional matrix format: width, height, and channel. Multiple filters are placed in various layers, inputting every layer from the preceding layer or layers. These layers act as feature extractors for the input image to identify its unique features or patterns from the input image. In this study, we used supervised learning by submitting the input image and segmentation mask to train the network to develop a model that maps an inscription image to an output mask. The entire CNN consists of multiple layers; the initial layers extract the low-level features, and the intermediate or later layers capture the high-level features of the input image.

2.6.1 Convolution and Pooling

These are critical components of a Convolutional Neural. The convolution operation uses a kernel that can be considered a 2D square matrix with randomly initialized values or weights. The kernel slides over the entire image, moving a fixed number of pixels each time, called a stride. The output is determined at each stride through kernel convolution with the overlapping sections of the image. Mathematically, the convolution of the two functions v and w is defined as

$$(v \star w)(x) = \int v(t)w(x-t)\,dx \tag{2}$$

Where x is in R^n for $n > 1$

Considering v as a 2-dimensional image and kernel w as a 2-dimensional square matrix with randomly initialized values, the convolution operation can be defined as:

$$(V \star W)(i,j) = \sum_m \sum_n V(m,n)W(i-m, j-n) \tag{3}$$

Each layer in the network contains multiple kernels applied to each input channel to produce values in one of the output channels, which are the input to the next layer. The second most fundamental CNN layer is the pooling layer, also called the subsampling layer, which is applied after the convolutional layer. Using a statistical function on the kernel window, the pooling layer downsamples the feature map created by the convolution layer, such as the average or maximum value of the pixels under the kernel window. The pooling layer can be considered a compression layer that selects the essential features to be passed to the next layer.

2.6.2 Up-Convolution

The convolutional layer maps a larger input image to a relatively small output. It performs the opposite operation of converting a low-size feature map to a larger-size output. This layer is used in the decoder section of the U-Net CNN architecture. It effectively upsamples the input feature by performing a convolution with an input feature matrix with additional rows and columns having values equal to zero and added around the actual feature values.

2.6.3 Loss function and optimizer

The loss function calculates how much the network's prediction deviates from the actual, or ground truth, data. This was used during the training phase to determine the quality of the prediction output. For UNet-based segmentation, the output consists of N values (corresponding to each input image pixel value), with each value suggesting whether the specific pixel belongs to a particular class (background or foreground).

- The difference between the probability distribution of the anticipated pixel values and actual pixel values was computed using cross-entropy, as mentioned in the equation below:

$$L(y_t, y_p) = -(y_t \log(y_p) + (1 - y_t) \log(1 - y_p)) \quad (4)$$

Where y_t = ground truth value and y_p = predicted output.

- Dice loss (D_l) is another crucial factor to consider when comparing the correctness of the anticipated output mask to the actual data [25].

$$D_l(y_t, y_p) = 1 - \left((2y_t y_p + 1)/(y_t + y_p + 1)\right) \quad (5)$$

2.6.4 Activation Function

The activation or transfer function converts a node's output to a specific range, such as (0 to 1) or (-1 to 1).

- Sigmoid activation function: This is a nonlinear, monotonic, and continuous function that takes an input value, converts it to the range of– 0 to 1, and is used for the output layer. The differentiable sigmoid function enables backpropagation of the loss to the kernel weights. The activation function f is represented as

$$f(z) = 1/(1 + e^{-z}) \quad (6)$$

- The rectified linear unit (ReLU) is another activation function used at the output of the convolution layer before sending data to the next layer. This introduces nonlinearity into the network. The ReLU function is represented as

$$ReLU(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (7)$$

2.6.5 U-Net: CNN architecture for Image Segmentation

U-Net is a convolutional neural network-based architecture developed to segment biomedical images [11]. Compared with the conventional CNN, which generates an output that is a single class label, the U-Net architecture generates an output of the exact dimensions as the input image dimension, with each pixel assigned a suitable class value. This was further used to segment specific areas from the image and localize the individual objects in the source image. This is a fully convolutional network because U-Net has no dense layers. Multiple convolution and pooling layers constitute the network. There are three primary components of the U-Net design.

- Encoder: This is the first half of the network that processes the input image. The encoder block, also known as the contraction block, contains multiple sub-blocks of the convolution layer (3 × 3, ReLU) and max pooling layer (2 × 2), such that in each sub-block, the image feature map is downsampled, and the feature channels are doubled. This part encodes the image into suitable features and acts as a feature extractor. Transfer learning assigns pre-trained kernel weights to the encoder.

- Bottleneck: The layer connecting the U-Net encoder and decoder sections is called the bottleneck layer. This layer further reduces the feature parameter size but increases its depth. It is a compressed representation of the input data and holds sufficient information for the decoder block to reconstruct the input image from the compressed feature set.

- Decoder: The decoder block, also known as the expansion path, is in the second half of the network. It contains an up-convolution layer in which feature map up-sampling is performed, followed by convolution (2 × 2), which reduces the number of feature map channels by half. This part of the U-Net localizes the features extracted in the encoder part and projects them to the pixel space to classify individual pixels.

One of the essential advantages of U-Net is that even in scenarios where limited training images are available, such as biomedical images, the network performs very well. Another critical structural element of U-Net is a concatenation link (or skip connection) that concatenates the contraction path feature map to the corresponding expansion path feature map that passes through the convolution layer (3 × 3 ReLU). The final layer used a 1 × 1 convolution operation that assigned each feature map to a suitable class. The U-net architecture developed for the Brahmi inscription image segmentation is shown in Figure 3.

Natural Resources for Human Health
eISSN: 2583-1194

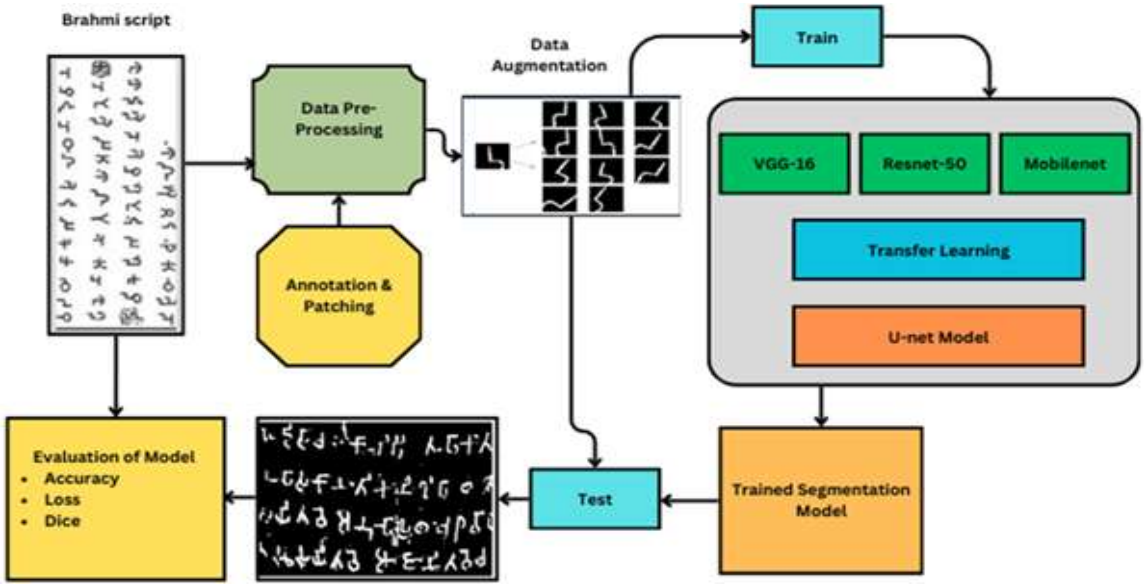


Figure 3: U-Net: CNN architecture for Image Segmentation

3. METHODOLOGY

Figure 3 shows how a U-Net model was created and used to segment the Brahmi script dip inscriptions, which is the basis of the proposed methodology. It is an encoder-decoder architecture in which an encoder first generates and organizes image features using various convolution and pooling layers and a decoder then enlarges those features to create the segmentation mask at the pixel level. Use of transfer learning with the pre-trained VGG-16 weights improved the performance of the encoder in terms of capturing different patterns present in the noisy and poorly illuminated ancient inscriptions. The figure also illustrates the model's ability to handle large images by splitting them into smaller patches and thus saving memory when training the model. Using this U-net helps to accurately pinpoint the characters to be identified while taking into account the cracks, smudges, and shadows which are often present on the images, thus aiding the methodology to achieve its goal of accurately segment and recognize Brahmi characters [32].

To provide a concise representation of the proposed methodology, Algorithm 1 summarizes the complete workflow of the transfer learning-based U-Net framework for Brahmi inscription segmentation. The workflow illustrated in Figure 3 and improves the reproducibility of the proposed approach.

Algorithm 1. Proposed Transfer Learning-Based U-Net Framework

Input:	Raw Brahmi inscription images (**I**), Ground-truth annotation masks (**M**), Patch size (**P = 128 × 128**), Pre-trained VGG16 encoder weights (**WVGG**)
	I : Set of inscription images
	M : Corresponding binary masks
	P : Patch size
	WVGG : Pre-trained VGG16 encoder weights
Output:	Predicted segmentation masks (**S**) and Localized Brahmi characters (**C**)
Procedure	
1:	*PREPROCESS(I)*

2: $(I^* \leftarrow \Phi(I))$

3: $GENERATEMASK((I^*))$

4: $(M \leftarrow \Gamma(I))$

5: $PATCHEXTRACTION((I, M))$

6: $(P_{img} \leftarrow f(I^*, P))$

7: $(P_{mask} \leftarrow f(M^*, P))$

8: **for** each image patch $(P_i \in P_{img})$ **do**

9: $(P_i \leftarrow Rotation(P_i))$

10: $(P_i \leftarrow HorizontalFlip(P_i))$

11: $(P_i \leftarrow Shearing(P_i))$

12: $(P_i \leftarrow Zoom(P_i))$

13: $(P_i \leftarrow Translation(P_i))$

14: **end for**

15: Initialize encoder $(E \leftarrow W_{VGG})$

16: Initialize decoder *(D)*

17: Construct segmentation model

18: $(U \leftarrow (E, D))$

19: Initialize Adam optimizer $((\eta = 0.001))$

20: **for** epoch = 1 to (E_{max}) **do**

21: **for** each mini-batch (B) **do**

22: $(\hat{S} \leftarrow U(B))$

23: $(L_{BCE} \leftarrow BinaryCrossEntropy(\hat{S}, M))$

24: $(\theta \leftarrow \theta - \eta \nabla L_{BCE})$

25: **end for**

26: **end for**

27: **for** each testing patch (P_i) **do**

28: $(\hat{S}_i \leftarrow U(P_i))$

29: **end for**

30: Reconstruct complete segmentation mask

31: $(S \leftarrow \Psi(\hat{S}_i))$

32: Extract contours

33: $(C \leftarrow \Omega(S))$

34: Compute evaluation metrics

35: Accuracy $(\leftarrow Acc(S, M))$

36: Precision $(\leftarrow Pre(S, M))$

37: Recall $(\leftarrow Rec(S, M))$

38: F1-score $(\leftarrow F1(S, M))$

39: Dice $(\leftarrow Dice(S, M))$

40: **return** $(S, ; C, ; Accuracy, ; Precision, ; Recall, ; F1\text{-}score, ; Dice)$

end Procedure

As presented in Algorithm 1, the proposed framework systematically integrates image preprocessing, patch-based learning, transfer learning, and contour-based character localization into a unified segmentation pipeline. The subsequent subsections describe each stage in detail, including the network architecture, data augmentation strategy, and training procedure adopted for accurate Brahmi inscription segmentation. Figure 3 illustrates the overall architecture of the proposed framework, whereas Algorithm 1 summarizes the complete computational workflow employed for training and inference.

3.1 Incorporating Transfer Learning with Pre-trained Weights

The first modification was employing transfer learning techniques on the U-Net architecture to improve the speed and correctness of the model. Instead of starting with a random initialization of the U-Net weights, the pre-trained weights of the VGG-16 network, a well-known Convolutional Neural Network (CNN), were employed. The VGG-16 pre-trained network weights are available publicly because the VGG-16 network was trained on the ImageNet dataset, which is a large, fully annotated dataset of images for Computer Vision. The pre-trained weights for the U-Net encoder section were beneficial because they provided advanced feature extraction and eliminated the need for the VGG-16 network to learn low-level features first.

This adjustment was beneficial for a number of reasons:

- The pre-trained weights of VGG-16 provided a good starting point for the U-Net encoder section to learn features that are domain specific to Brahmi inscriptions. Since the Brahmi dataset is small, transfer learning improved the model by decreasing the effects of overfitting.
- The adjustment used the encoder-decoder framework of U-Net, which was beneficial for improving the representation of features across layers.
- The adjustment improved the ability to learn features of Brahmi script characters from images that may contain noise and poor illumination.

3.2 Data Augmentation for Dataset Expansion

The second adjustment aimed to improve the scarcity of the dataset and was the implementation of data augmentation. The steps of augmentation included the following:

- Horizontal Flipping: Mirroring the images along the x-axis.
- Allowable Random Rotation: Accounts for angular distortion of inscriptions by random rotation (0-20 degrees).

- Allowed Shear Transformation: Factor in the gradual wear of inscriptions causing distortion of character shapes by applying a shear transformation (20% range).
- Allowed Zoom: Snapshots capturing varying degrees of focus and distance can be simulated by zooming in (20% range).
- Allowed Shift: Represent characters in varying positions by shifting the height and width of the image.
- The chosen augmentation technique:
- Allows Growth of Dataset: Offers the ability to train deep-learning models on significantly larger datasets by using a small number of original images.
- Allows Robust Model: Exposes the model to a wide variety of changes in texture, lighting, and the appearance of characters in order to strengthen its ability to generalize.
- Allows Real-World Simulation: Variability in the dataset is introduced to simulate the occurrence of real-world conditions, such as variable lighting and undulating surfaces.

3.3 Objective and Impact of Modifications

Considerable effort has been taken in making modifications to tackle segmentation of the Brahmi inscriptions in the image. The model has utilized transfer learning to focus on advanced feature extraction, while data augmentation has been used to ensure the model is trained on a wide, varied, and realistic dataset. The combination of these effects has made the U-Net model not only robust to the variations and imperfections of ancient inscriptions, but has also made the segmentation model accurate.

3.4 Dataset Preparation

Images of inscriptions coupled with their corresponding annotation images that highlight areas of interest for the inscriptions on images constitute the training data for the U-Net model. A sample inscription image is the 'Lumbini pillar inscription' from Nepal, shown in Figure 4. This image was the training image for U-Net. The image was preprocessed, manually annotated, and segmented into smaller sub-images to create a U-Net training dataset. The following subsections elaborate the procedure.

Figure 4: Sample Lumbini inscription image used for training the CNN model

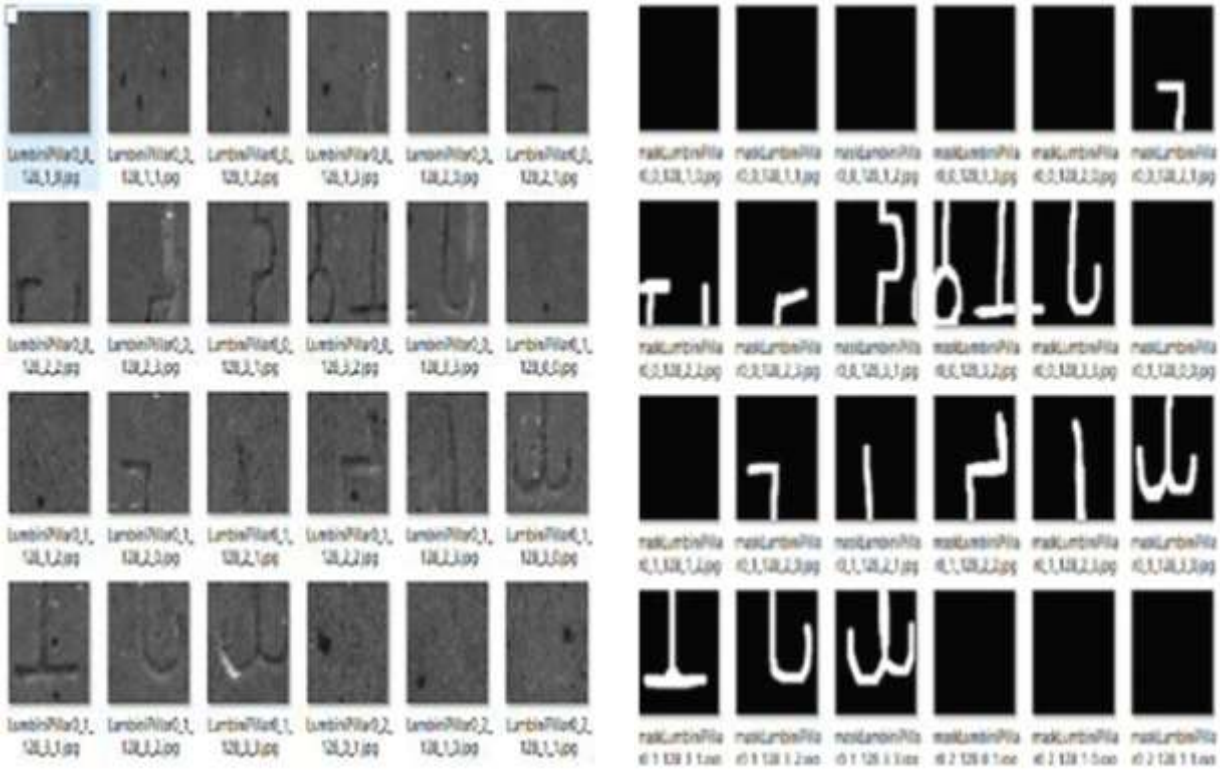
3.5 Image Annotation

The annotated image (called a mask) is manually prepared by marking the pixels corresponding to the Brahmi characters as foreground pixels and the remaining region as the background. As shown in Figure 4, the image has various defects, such as the left side being darker than the right side, a dark smear across the horizontal section, and multiple black spots on the entire image. These defects pose a significant challenge in image processing techniques to separate interest areas from the rest of the background. However, only specific foreground pixels can be marked as regions of interest through manual annotation, ignoring defects such as cracks, spots, and variable brightness across the image cross-section.

3.6 Patching

The inscription images, like the Lumbini inscription image, are large, e.g., the mentioned image has dimensions of 3.2k x 2.3k. Such a large image is unsuitable for direct input to the CNN network, as the memory requirement is significantly high and processing time is also increased. Therefore, the image was split into patches of 128 × 128 pixels. Medical image segmentation

has been performed using patch-based training for Convolutional neural networks, and the results have been promising [16]. Splitting the Lumbini image into multiple patches not only aids in training within reasonable memory (<16GB) but also generates additional training data from a single inscription image. All training images were split into 128x128 size patches along with the annotated images. The images and masks must be accurately divided into patches and named so that the training and testing processes load the image patch and corresponding mask. The annotated image is the ground-truth image that separates the background and foreground pixels into different classes. A set of patch images and accompanying masks is shown in Figure 5. The inscription pixel in the patch was marked as a foreground pixel in the mask, and the remaining area was marked as the background. In this manner, the annotated mask clears the foreground and background pixels. The patch creation process is implemented so the image and the corresponding mask patch are numbered similarly.



(a) (b)

Figure 5: (a) Inscription image patches (b) the corresponding mask patches with similar naming

3.7 Data Augmentation

Additional training data were generated from the existing training data using a data augmentation technique to increase the dataset size. Additional data help improve the accuracy of the model. Data augmentation works well for this specific research because the purpose is to extract the inscription pixels from the background. Augmentation consists of the following operations performed on the input image patches and the corresponding mask patches:

- Flip – Flipping the data horizontally.
- Rotation – The image is randomly rotated between 0 to 20 degrees.
- Shear: This transformation maps all image points from the current coordinate to a new coordinate set. For example, a square can be transformed into a parallelogram and a circle into an ellipse. This study used a shear range of 20%.
- Zoom enables zooming inside the image, and a zoom range of 20% is used.
- Width and height shift: The image was translated horizontally or vertically within a fraction of its total height and width.

These translations can create space around the image area after the translation. The space beyond the image boundary is filled with the image's edge values; the edge pixel values are extended in the vacant area created after the augmentation is applied. The augmented image set from an individual image patch and mask is shown in Figures 6 and 7, respectively.

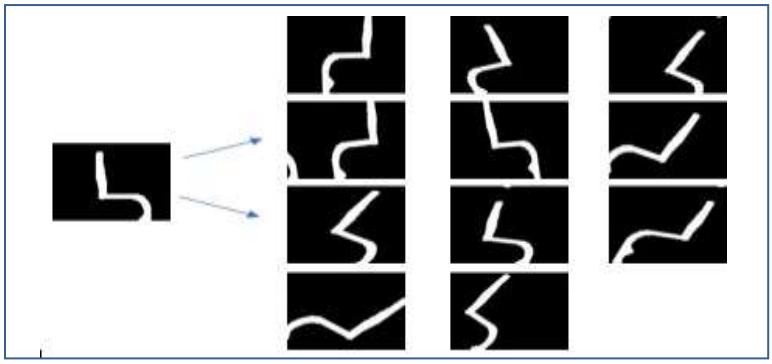


Figure 6: Image patch augmentation

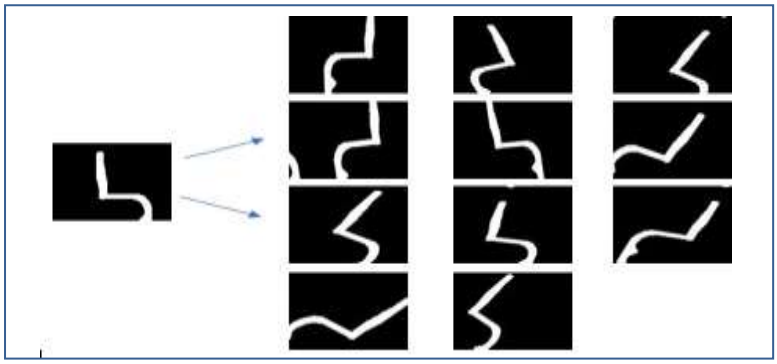


Figure 7: Corresponding patch augmentation

3.8 Proposed U-Net-based CNN Method

This research proposes a hybrid deep learning framework that integrates U-Net, VGG, and ImageNet architectures to consistently segment Brahmi script characters satisfactorily. The U-Net-based design is a significant development because it is the first time a neural network has been employed as a tool for char segmentation of parchment-based inscriptions on ancient fragments while dealing with noise, cracks, shadows, and illumination problems. The architecture of the proposed hybrid model is as follows:

- Model 1: UNet: The segmentation procedure utilizes the U-Net model architecture delineated in Figure 3 as the baseline structure for segmentation. The model's architecture comprises an encoder and decoder module for image segmentation tasks. This model is appropriate because it extracts and rebuilds the feature maps.
- Model 2: Base U-Net with a VGG backbone and ImageNet weight—The VGG-16 network learned on ImageNet was utilized for model kernels to refine the baseline model using one-epoch transfer learning. This approach significantly mitigates the effects of data scarcity. Cross-entropy was selected as an appropriate loss function for the weight updates and backpropagation. Furthermore, the model's accuracy was measured with the help of the Dice-Loss function.

4. EXPERIMENT AND RESULTS

Experiments were performed with Python for image processing and the U-Net model using the OpenCV library for conventional image processing and Keras for U-Net processing. Each model was validated using the Brahmi inscription image from the Asoka pillar, as shown in Figure 8.

Natural Resources for Human Health
eISSN: 2583-1194

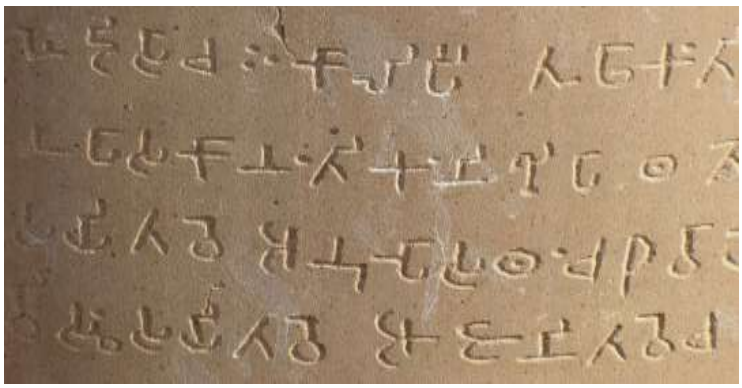
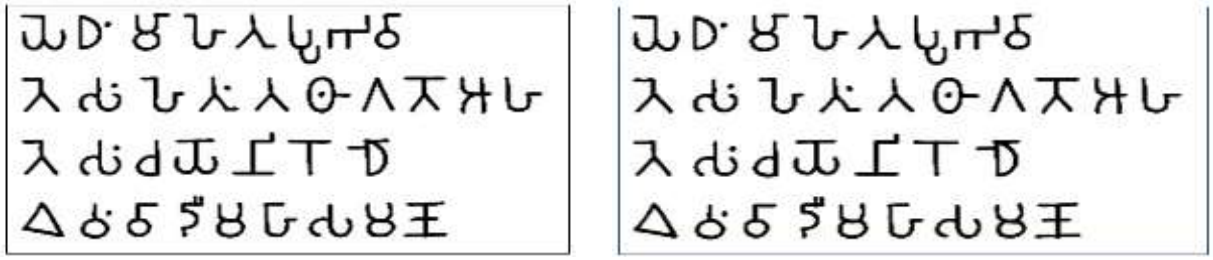


Figure 8: Brahmi inscription from Asoka pillar.

Another measure of accuracy is the number of Brahmi characters segmented by the CNN model from the total Brahmi characters shown in the Ashoka inscription image. The segmentation results obtained by applying both methods to the Asoka Pillar inscription image of Sarnath are discussed below:

4.1 Experimental results of image processing method

Image processing was performed on printed text and inscription images. Figures 9(a) and 10(a) show the input printed and inscription text images. The adaptive threshold block size and neighborhood pixel used to calculate the threshold must be adjusted based on the input image size. For an input image of size 1280 × 960, Figures 9(b) and 10(b) display the impact of applying adaptive thresholding with a block size of 50, followed by morphological erosion and dilation.



(a) (b)

Figure 9: (a) printed text image (b) corresponding threshold text image

A bounding rectangle was generated for the individual characters using contours, and the final images with bounding rectangles are shown in Figures 11 and 12. The segmentation results for the printed Brahmi character image were highly accurate, whereas, for the Brahmi inscription image, there were several inaccuracies, such as partially captured and missed characters.

(a) (b)

Figure 10: (a) The Brahmi inscription image (b) the corresponding threshold images

Figure 11: Segmentation result of printed text image

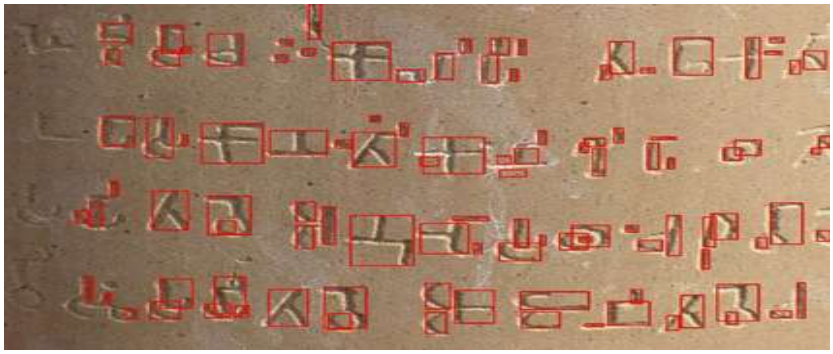


Figure 12: Segmentation result of inscription image

4.2 Experimental Results of U-Net-Based Method

The CNN-based model trained on the Lumbini data processed the Ashoka inscription images. The primary benefit of the CNN method is that it does not require manual modification compared to the image processing method. The training dataset consists of an inscription image as the input and a corresponding mask or foreground image as the ground truth training data preparation steps.

- Annotate an inscription image to mark the foreground pixel and generate a mask.
- The large-size inscription images are split into smaller 128 × 128 patches and the masks.
- The CNN model was trained using 2000 images and their corresponding masks, with a 90%: 10% training and validation split. The model was trained using the following parameters.
- Batch size = 16
- Optimizer = Adam [27]
- learning rate = 0.001

The sigmoid function was used for output activation, and the loss function employed in the model was binary cross-entropy. In addition, the Dice coefficient was used to calculate the accuracy of the generated mask. The experimental results for the training and validation of the two models are displayed in Figures 13 and 14, respectively, and Table 2 shows the comparison after 500 epochs. The training time required for both models was approximately 36 hours using the 2000 image dataset. U-Net with a VGG backbone was validated better than U-Net with random weight initialization.

As the model was trained using a specific image size of 128 × 128, the test image was also fed to the U-net model by splitting it into patches, and the output generated was combined to form a mask for the entire inscription image. The output masks generated by both models are shown in Figures 15 and 16. Finally, to identify the contours for each generated mask, the bounding rectangle image is shown in Figures 17 and 18 for U-net and U-net with VGG, respectively. Such improvements in the U-Net architecture notably upgraded the recognition of ancient Brahmi inscriptions, including pre-trained VGG-16 weight-

transfer learning and data augmentation. These designs address the problems of cracks, smudge, and non-uniform illumination more successfully, focusing on achieving better accuracy in the case of scanty training data. Using traditional models, the model segmented 42 out of 49 Brahmi characters instead of 29. The model attained a Dice coefficient of 0.714, superior to the baseline U-Net value of 0.612. The results establish the role of advancements in transfer learning procedures and augmentation in ancient script segmentation tasks.

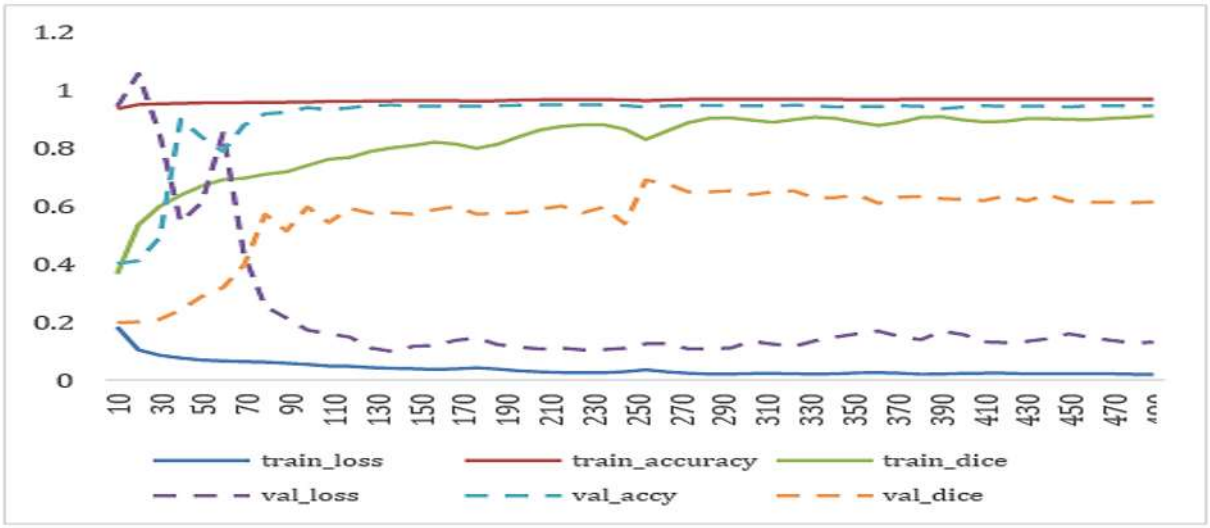


Figure 13: U-net training and validation results.

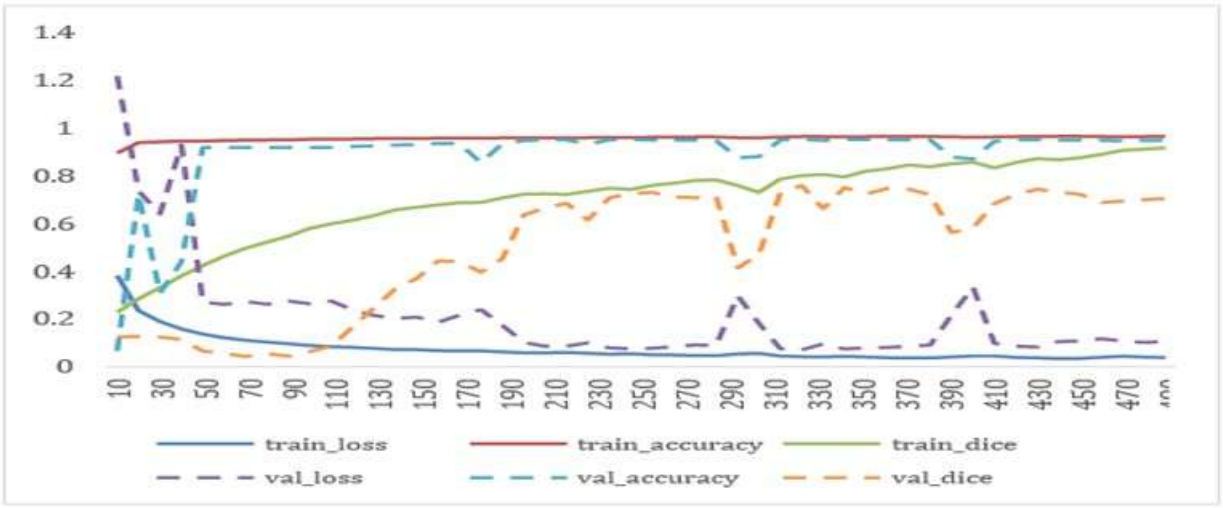


Figure 14: U-net with transfer learning and VGG16 backbone results.

Table 2. Performance matrix of both the models

Model	Training			Validation		
	Loss	Accuracy	Dice	Loss	Accuracy	Dice
U-Net	0.036	0.960	0.90	0.122	0.915	0.612
U-Net++	0.019	0.965	0.80	0.096	0.942	0.714

Natural Resources for Human Health
eISSN: 2583-1194

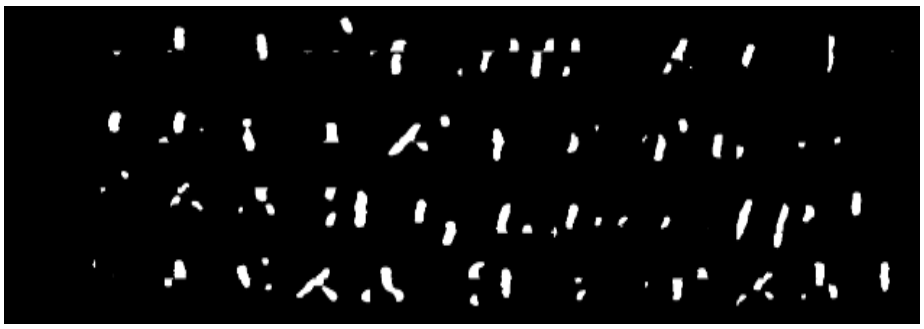


Figure 15: Mask generated by UNet with random weights initialization.

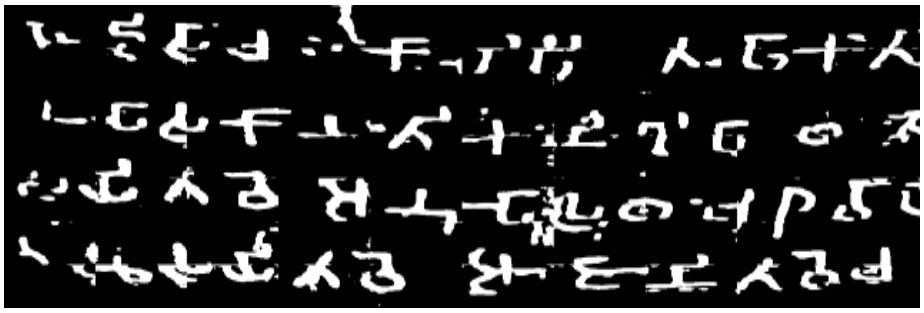


Figure 16: Mask generated by UNet with VGG16 backbone and with weights initialized by VGG16 trained on ImageNet.

Figure 17: A bounding box for UNet is created by applying contour plotting over the mask generated by the CNN model

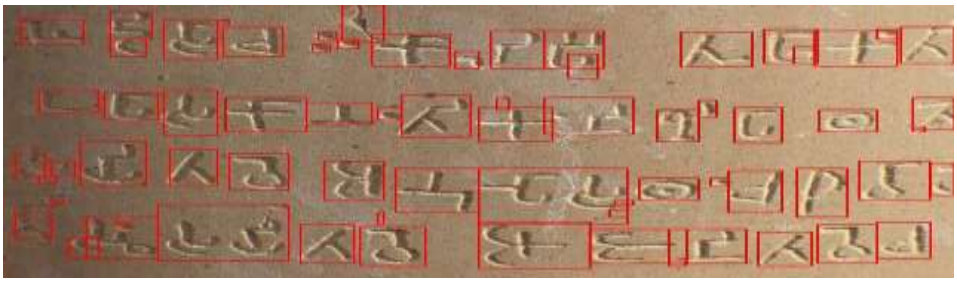


Figure 18: Bounding box for UNet VGG by applying contour plotting over the mask generated by the CNN model

Considering the output of the U-Net with the VGG backbone and comparing it with the image processing output, Figure 19 shows that the number of script characters captured by the CNN is more significant than those identified through the image processing method.

Natural Resources for Human Health
eISSN: 2583-1194

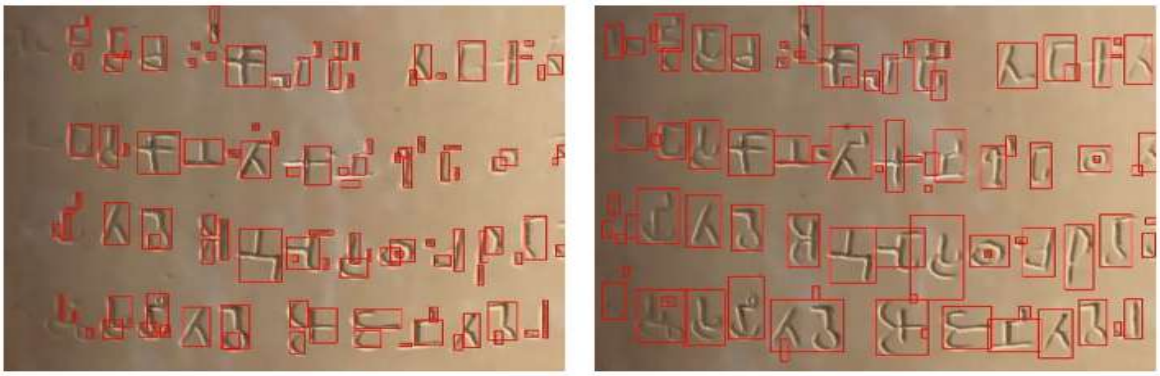


Figure 19: Comparison of generated bounding box (a) image processing (left) and (b) CNN (right)

A detailed analysis of the various cross-sections of the inscription image shows that segmentation mask generation is significantly better for the CNN-based method than for the image-processing method. Figures 20, 21, and 22 show more detailed results of the various sections of the inscription image and the corresponding bounding rectangle generated by the image processing and CNN.

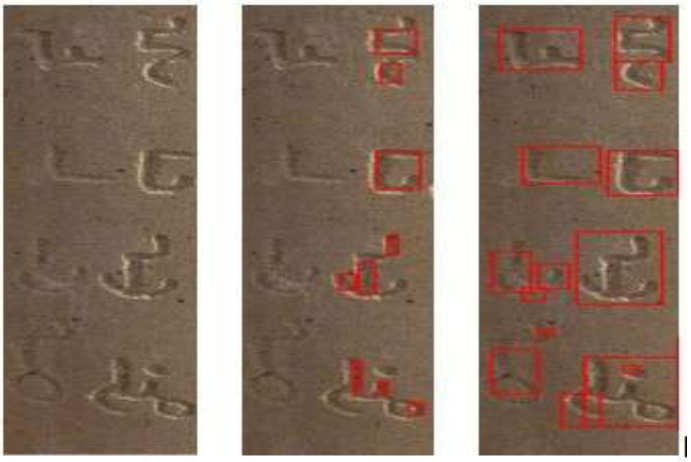


Figure 20: The leftmost section of the inscription image has very low illumination due to the curvature of the pillar, and the image processing method does not capture the entire left portion of the inscription. With the CNN model, two characters are fully captured from the first column, and all the characters are captured in the second column.

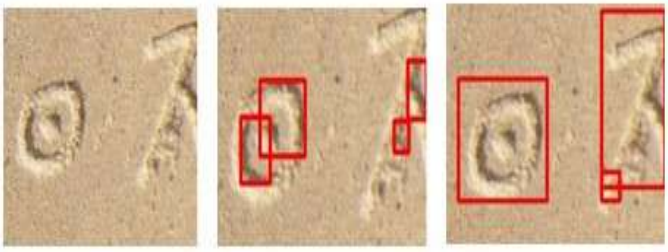


Figure 21: The middle section of the inscription is brightly illuminated, and due to engravings, shadows are formed. Both characters have very bright and shadow regions, which results in the image processing method treating these as two different characters. In contrast, the CNN-based model can recognize these as a single character.

Natural Resources for Human Health
eISSN: 2583-1194

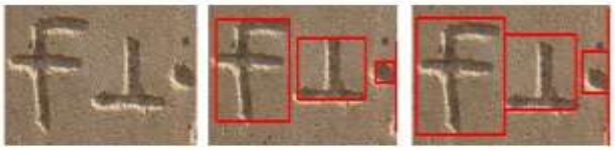


Figure 22 shows a section of the inscription where the results are similar for both the image processing and CNN-based methods. Both characters displayed have consistent color, and the shadow formed in the engraving is consistent because due of the curvature of the pillar and the direction of the sunlight.

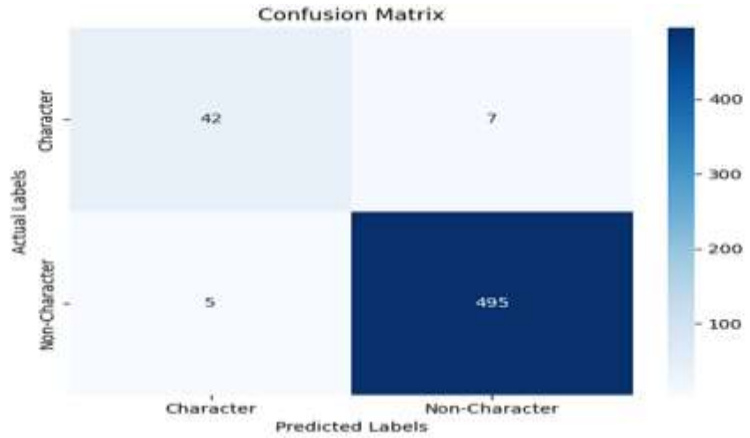


Figure 23 Confusion Matrix for Brahmi Character Segmentation

Figure 23 shows the performance of the Brahmi character segmentation model. It provides information regarding true positives, which are the characters that were correctly identified (42 in this case); false negatives, missing characters (7); and false positives, which are non-characters that were incorrectly identified as characters (5), along with true negatives, which are the non-characters that were correctly identified (495 in this case). This provides a comprehensive picture of the precision of the model and the division of errors. The enhancements introduced to the U-Net architecture yielded significant improvements in the segmentation of the ancient Brahmi inscriptions.

4.2.1 Training and Validation Performance

Analysis of training and validation performance of the models proposed in this research was done to understand the ability of the models in segmenting characters of Brahmi inscriptions. Two implementations of U-Net were analyzed. One was the baseline U-Net, which was implemented with random weight initialization. The other was U-Net with a pre-trained VGG16 backbone.

As shown in Table 3, U-Net + VGG proposed by the authors achieved a validation accuracy of 94.2%, which is 2.7% higher than the baseline U-Net. The improvement is due to the pre-trained encoder, which offers advanced hierarchical feature representation and assists in faster convergence for cases where only a few annotated images of Brahmi inscriptions are available. The higher Dice coefficient indicates that U-Net + VGG achieved a better characterization of the boundary of Brahmi characters.

Table 3. Training and Validation Results of U-Net Models

Model	Training			Validation		
	Loss	Accuracy	Dice	Loss	Accuracy	Dice
U-Net	0.036	0.960	0.900	0.122	0.915	0.612
U-Net+VGG	0.019	0.965	0.800	0.096	0.942	0.714

To better understand the U-Net + VGG model and its segmentation, Table 4 includes an analysis of the model's precision, recall, and F1-score. The U-Net + VGG model achieved a precision of 0.893, a recall of 0.800, and an F1-score of 0.844, outperforming the baseline U-Net (0.853, 0.698, 0.768). From the reported measurements, the proposed model achieved a better (i) segmentation of characters and (ii) reduction of false positives. Thus, the proposed model achieved better overall segmentation.

Table 4. Precision-Recall-F1 Comparison

Model	Precision	Recall	F1-Score
U-Net	0.853	0.698	0.768
U-Net + VGG	0.893	0.800	0.844

Table 5 presents an analysis of error rates based on the False Omission Rate (FOR) and False Discovery Rate (FDR). FOR decreased from 0.302 of baseline U-Net to 0.200 of U-Net + VGG. FDR decreased from 0.147 to 0.107. These findings show the proposed model is better at dealing with both missed characters (which would be false negatives) and characters that were incorrectly identified as present (which would be false positives).

Table 5. Error Rate Metrics

Model	False Omission Rate (FOR)	False Discovery Rate (FDR)
U-Net	0.302	0.147
U-Net + VGG	0.200	0.107

The combination of the training/validation metrics, the precision-recall assessment, and the evaluation of error rates indicate that the U-Net + VGG backbone offers an important advancement to segmentation of Brahmi ancient inscriptions. The improvement is due to representation of higher quality of features by the pre-trained weights of VGG, which deal better with the cracks and the quality of the lighting as well as the subtle variations of the characters present in the images of the inscriptions.

4.2.2 Character Segmentation Results

Character segmentation results refer to the capacity of each method to successfully isolate individual Brahmi characters from the Asoka pillar inscription. Segmentation results across three methods are summarized in Table 6. These three methods correspond to traditional image processing, baseline U-Net, and proposed U-Net with VGG backbone. As inferred from the results, the proposed U-Net + VGG model outperformed the other two methods with 42 of the 49 characters identified, compared to 34 characters identified by the baseline U-Net and 29 characters identified by the image processing method. The proposed method also outperformed the baseline methods in terms of the number of fully segmented characters, with only 2 characters partially segmented, as opposed to 6 and 10 by the U-Net method and the image processing method, respectively. The number of characters missed using the image processing method (20) and U-Net method (15) also decreased substantially to 7 using the proposed method. These results imply that segmentation of character images that are smeared, unevenly illuminated, or contain cracks, can be improved by using transfer learning and deep feature extraction.

Table 6. Character Segmentation Accuracy Across Methods

Model	Characters in Inscription	Correctly Segmented	Partially Segmented	Missed Characters
Image Processing	49	29	10	20

U-Net	49	34	6	15
U-Net + VGG (Proposed)	49	42	2	7

4.2.3 Comparative Improvements

Table 7 provides a concise comparison of improvements achieved by the proposed model across key metrics.

- Segmented Characters: Proposed U-Net + VGG successfully segments 42 characters, demonstrating significant improvement from the baseline methods.
- Validation Accuracy and Dice: Validation accuracy improved from 0.915 (U-Net) to 0.942 for the proposed method, while the Dice coefficient also improved from 0.612 to 0.714, demonstrating an improved match between the predicted and actual mask.
- Robustness to Noise/Cracks and Illumination: The new method shows superior robustness to noise and illumination compared to other methods as the baseline U-Net shows moderate robustness and image-processing methods show poor robustness.

This evaluation shows that the U-Net + VGG backbone improves segmentation metrics and contributes to the model's robustness, thus the model's suitability increases for the analysis of ancient inscriptions.

Table 7. Comparative Gains Across Methods

Metric	Image Processing	U-Net	U-Net + VGG
Segmented Characters	29	34	42
Validation Accuracy	–	0.915	0.942
Validation Dice	–	0.612	0.714
Robustness to Noise/Cracks	Low	Moderate	High
Robustness to Illumination	Low	Moderate	High

4.2.4 Epoch-wise Performance Trends

To examine the convergence behavior, the models' performance was evaluated at different training epochs, as shown in Table 8.

- Baseline U-Net achieved a validation accuracy of 0.915 and a Dice coefficient of 0.612 at epoch 500, both metrics improved in a linear fashion.
- In contrast, the U-Net + VGG model achieved a higher accuracy (0.932) and Dice (0.689) at epoch 300. This is an example of convergence at an increased speed, a benefit of transfer learning.
- Applying the proposed model at epoch 500 achieved an accuracy of 0.942 and a Dice coefficient of 0.714, showing the model sustained and improved performance. The trends in this section shows that the use of pre-trained weights aids in learning and improves generalized performance, a benefit to domains where the labeled data is scarce.

- **Table 8.** Epoch-wise Validation Performance of U-Net Models

Epochs	U-Net Accuracy	U-Net Dice	U-Net + VGG Accuracy	U-Net + VGG Dice
100	0.872	0.521	0.896	0.602

200	0.893	0.557	0.915	0.650
300	0.902	0.583	0.932	0.689
400	0.910	0.598	0.938	0.701
500	0.915	0.612	0.942	0.714

4.2.5 Section-wise Performance Across Inscription Image

To assess performance under varying illumination and structural challenges, the inscription image was sectioned into three regions which are (leftmost, middle, right). Table 9 displays the performance of each method.

- Leftmost Section: Image processing failed under poor illumination to identify two characters. The baseline U-Net partially identified one character, whereas the U-Net + VGG model successfully identified and segmented both characters. This shows U-Net + VGG is robust in low illumination scenarios.
- Middle Section: In response to fragmenting characters due to the presence of shadows in the image, baseline U-Net performed partial character recognition. In contrast, the model in this study accurately recognized the character in its entirety, and was therefore better able to resolve the problem of character fragmentation due to the presence of shadows.
- Right Section: The region of interest was uniformly illuminated and the engravings were clear, allowing all three methods to perform equally.

The section-wise analysis showed that the U-Net + VGG model, in all the cases, managed difficult regions better, thus accommodating the changes in illumination and also the curvature and imperfections in the surface, which often create problems for conventional methods.

Table 9. Section-wise Segmentation Results

Section	Image Processing	U-Net Recognition	U-Net + VGG Recognition	Observations
Leftmost Section	Missed 2 chars	Captured 1 char	Captured 2 chars	CNN robust to low illumination
Middle Section	Split 1 char into 2	Partial recognition	Single correct char	CNN resolved shadow-induced fragmentation
Right Section	All chars clear	All chars clear	All chars clear	Similar performance across methods

4.2.6 Comparative Robustness to Image Challenges

To evaluate performance under difficult conditions, the robustness of each segmentation method to common image challenges was assessed. The results of this assessment are in Table 10. Conventional image processing techniques struggled with non-uniform illumination, smudges, cracks, and character fragmentation, though they performed acceptably for smooth and well-defined characters. The baseline U-Net exhibited improved performance over conventional methods, but still showed low resistance to irregularities. In all other cases, the proposed U-Net + VGG model displayed high resistance and robustness, successfully dealing with cracks, smudges, and non-uniform illumination, and demonstrated low character fragmentation. The results also showed that U-Net + VGG model extracted consistent and reliable features even with complex and difficult inscription regions.

Table 10. Robustness Comparison

Challenge	Image Processing	U-Net	U-Net + VGG
Cracks & Smudges	Poor	Moderate	High
Non-uniform Illumination	Poor	Moderate	High
Character Fragmentation	High	Moderate	Low
Smooth Characters	Good	Good	Good

4.2.7 Improvement in Character Coverage

Table 11 shows how character coverage is affected by the proposed method. The image processing method covered 29 of 49 characters, with a coverage of 59.18%. The baseline U-Net coverage improved to 69.38% by identifying 34 characters, indicating an improvement due to segmentation deep learning. The U-Net + VGG model improved this coverage to 85.71 % by identifying 42 characters. This coverage, representing a 26.5 % improvement over the image processing method, shows the advantage of fine-tuning VGG weights and applying VGG data augmentation. The results show the model's improved ability to identify partially broken or difficult-to-detect characters.

Table 11. Character Coverage Improvement (%)

Method	Characters Recognized	Coverage (%)	Improvement Over Image Processing
Image Processing	29 / 49	59.18%	–
U-Net	34 / 49	69.38%	+10.2%
U-Net + VGG (Proposed)	42 / 49	85.71%	+26.5%

4.2.8 Improvement in Character Coverage

Localized segmentation was evaluated by dividing the test images into 128 × 128 patches, with Dice coefficient values calculated for each patch, as shown in Table 12. The baseline U-Net covered patch-wise Dice coefficients of 0.589 to 0.624. The U-Net + VGG model showed patch-wise Dice coefficients of 0.675 to 0.717. All six patches (top-left, top-middle, top-right, bottom-left, bottom-middle, bottom-right) showed improvement. The proposed method improved segmentation quality regardless of regional variation of lighting and texture. The variation showed consistency and reliability of the proposed method.

Table 12. Character Coverage Improvement (%)

Patch Region	U-Net Dice	U-Net + VGG Dice
Top-Left	0.589	0.675
Top-Middle	0.605	0.698
Top-Right	0.618	0.709
Bottom-Left	0.599	0.684
Bottom-Middle	0.610	0.702
Bottom-Right	0.624	0.717

4.2.9 Section-Wise Accuracy Metrics

We segmented the inscription image into three regions to examine performance across differing illumination, engraving curvature, and depth. The accuracy by region is reported in Table 13. The baseline U-Net model scored 0.875, 0.890, and 0.908 for the left, middle, and right regions, respectively. The U-Net + VGG model scored 0.921, 0.935, and 0.941 for the same regions, demonstrating a significant improvement. The U-Net + VGG model scored higher than the baseline U-Net across all regions, showing that the VGG backbone provides transfer learning to improve accuracy and performance across difficult regions, addressing local variation in the inscription image.

Table 13. Section-wise Segmentation Accuracy

Section	U-Net Accuracy	U-Net + VGG Accuracy
Leftmost Section	0.875	0.921
Middle Section	0.890	0.935
Right Section	0.908	0.941

4.2.10 Comparative Performance with Existing Techniques

Considering results in the previous sections, the performance of the U-Net + VGG model was compared with various segmentation methods to include older image processing methods, threshold and edge-based methods, clustering techniques, and contemporary deep learning-based methods to segment images. A number of key segmentation metrics, which include accuracy, precision, recall, the Dice coefficient, F1-score, and the total number of Brahmi characters segmented correctly, are included in Table 14 to facilitate a comparison of performances.

The results show that Otsu's method and adaptive thresholding offered poor performance due to variable conditions resulting in poor segmentation of characters (21–23) with Otsu's method and adaptive thresholding achieving segmentation with 72.5% and 75.1% accuracy respectively. Other methods based on edges (Canny) and clustering (K-means, Watershed) offered slight improvements (78.3% to 82.0% accuracy) but still suffered from the same conditions of shadow, over-segmentation and eroded characters. Other methods based on traditional image processing techniques offered segmentation with 85% accuracy (29 characters) but failed at solving the segmentation of complex inscriptions.

Out of the deep learning methods, RAS-LSTM (Seq2Seq) and segmentation based on GANs (GANsum) offered improvements in both accuracy and F1 with GANsum achieving 87.3% with an F1 of 0.644 and RAS-LSTM (Seq2Seq) at 88.7% and an F1 of 0.657. The base U-Net offered improvements achieving 91.5% with 34 segmented characters. U-Net++ also offered improvements with 92.8% and 36 segmented characters and showed the impact of using a deep architecture with skip connections for feature reuse.

The U-Net + VGG model achieved the greatest accuracy of 94.2%, with a Dice coefficient of 0.714, a precision of 0.893, a recall of 0.800, and an F1 of 0.844 and segmented 42 of the 49 Brahmi characters and achieved a superior balance of accuracy and recall and robustness. The model is extremely versatile and highly effective due to the use of the pre-trained VGG module for advanced feature extraction and offered the capability of segmentation under extremely difficult conditions; specifically, non-uniform illumination, cracks, and shadow-cast fragmented characters. The results demonstrate that this model is highly effective and generalizable and surpasses all traditional image processing techniques and current state-of-the art methods of deep learning for segmentation of ancient scripts.

Table 14. Comparison of Proposed U-Net + VGG with Existing Segmentation Methods

Method	Accuracy (%)	Dice Coefficient	Precision	Recall	F1-Score	Characters Segmented (out of 49)	Remarks
Otsu Thresholding	72.5	0.412	0.610	0.421	0.499	21	Sensitive to illumination, poor for cracks
Adaptive Thresholding	75.1	0.439	0.634	0.436	0.517	23	Better than Otsu, still fragile under noise
Canny Edge Detection	78.3	0.471	0.652	0.462	0.542	25	Detects outlines, fails on eroded characters
K-means Segmentation	80.6	0.498	0.670	0.481	0.560	26	Cluster-based, weak with uneven illumination
Watershed Segmentation	82.0	0.525	0.681	0.502	0.576	27	Over-segmentation issue, shadows misclassified
Traditional Image Processing	85.0	0.552	0.710	0.521	0.600	29	Baseline from inscription dataset
RAS-LSTM (Seq2Seq)	87.3	0.576	0.735	0.570	0.644	31	Performs well with structured datasets
GAN-based Segmentation (GANsum)	88.7	0.590	0.748	0.589	0.657	32	Effective, but unstable training
Baseline U-Net	91.5	0.612	0.853	0.698	0.768	34	Strong baseline, overfitting observed
U-Net++	92.8	0.655	0.871	0.739	0.800	36	Deeper skip connections improve feature reuse
Proposed U-Net + VGG	**94.2**	**0.714**	**0.893**	**0.800**	**0.844**	**42**	Best balance of accuracy, recall & robustness

Compared with traditional methods, the proposed framework shows excellent robustness as feature extraction is done directly from the annotated images of the inscriptions. Most methods rely on handpicked features of the images. The proposed framework innovatively builds upon the existing methods by introducing a deep learning framework for inscriptions segmentation. By relying on a VGG16-based encoder, superior control of feature reuse is achieved and hence greater segmentation accuracy for small data volumes.

Overall, the experimental results consistently demonstrate that integrating transfer learning, patch-based training, and data augmentation enables more reliable segmentation of ancient Brahmi inscriptions than both conventional image-processing methods and baseline deep learning models. The improvements are evident across multiple quantitative metrics, visual analyses, and section-wise evaluations, confirming the effectiveness and robustness of the proposed framework under challenging inscription conditions.

5. DISCUSSION

In the Asoka pillar Brahmi inscription, 49 Brahmi characters were inscribed. Based on the bounding rectangles obtained from both methods, we observed that the image-processing method could extract 29 characters and the CNN-based method could extract 42 characters accurately. Therefore, traditional image processing techniques, such as printed text images, are better for processing images with transparent background and foreground separation. However, character segmentation using conventional image processing techniques is only partially accurate for an inscription image that contains several inconsistencies across the cross-section, such as cracks, variable illumination, shading due to sunlight, and the curvature of the inscription. U-Net-based CNN has demonstrated excellent results for inscription image segmentation. Character segmentation suits sections with shadow effects and bright sections. The proposed U-net model used in this study consisted of the VGG-16 backbone trained using the ImageNet dataset; that is, the U-net expansion path was initialized with weights corresponding to VGG-16 trained on the ImageNet dataset. This helped to train the model for inscription images, even with less training data. The proposed model can be further improved and enhanced by training using Brahmi inscription image data collated and hosted on various web sources.

6. CONCLUSION

The research presented a Brahmi Script Segmentation method utilizing hybrid U-Net models augmented with deep learning tools, signifying critical advancements toward the digitization and safeguarding of ancient manuscripts. Compared with the traditional U-Net model, the proposed hybrid U-Net model showed superior performance across the board in segmentation tasks involving images of ancient manuscripts with noise, parchment shadows, cracks, and inconsistent exposure. The use of transfer learning in the hybrid U-Net model was also crucial in the segmentation task, working around the dataset constraints, and maximizing the segmentation results. The methodology was verified with experiments. The proposed method achieved a Dice coefficient of 0.714 compared with the baseline U-Net Dice coefficient of 0.612. The model also achieved greater segmentation results in terms of validation accuracy (0.942 vs. 0.915), and F1-Score (0.844 vs. 0.768), in addition to lower segmentation error rates in terms of the False Omission Rate (0.200 vs. 0.302) and the False Discovery Rate (0.107 vs. 0.147). With regard to the 49 target characters, the hybrid U-Net model achieved successful segmentation of 42 of the target characters, in contrast to the baseline model segmentation of only 29 target characters. The results have shown the applicability of deep learning in the field of ancient script analysis and have considerable potential in the domains of computational archaeology and cultural heritage preservation.

6.1 Future Research Direction

The Future research can focus on enhancing model architectures, expanding datasets, and integrating advanced imaging and interpretability techniques for ancient script analysis which are explained as follows:

- *Enhanced Backbone Networks:* Explore deeper and more advanced CNN architectures to further improve feature extraction and segmentation accuracy for ancient scripts.
- *Expanded Dataset Collection:* Curate larger datasets of Brahmi and other ancient inscriptions from multiple sources to reduce reliance on augmentation and improve generalization.
- *Multi-Script Segmentation:* Extend the model to segment and recognize multiple ancient scripts beyond Brahmi, supporting comparative epigraphy studies.
- *Integration of 3D Imaging:* Incorporate 3D scanning and depth information to handle complex surfaces and better capture inscriptions on curved or eroded artifacts.
- *Explainable AI (XAI):* Develop interpretability methods to visualize how the model identifies characters, aiding scholars in verifying results and understanding model decisions.
- *Real-Time Mobile/Edge Deployment:* Optimize the model for lightweight deployment on portable devices for field-based epigraphic research and rapid analysis.
- *Cross-Modal Learning:* Combine image data with historical context, linguistic databases, or textual references to enhance character recognition and script understanding.

6.2 Potential Industrial Applications

The proposed segmentation framework offers practical applications in archaeology, cultural heritage digitization, education, and AI-assisted historical text preservation which are explained as follows:

- *Archaeology and Epigraphy:* Automated analysis of inscriptions to support historical research, preservation, and documentation of ancient artifacts.
- *Cultural Heritage Digitization:* High-accuracy segmentation for digitizing museum collections, monuments, and stone pillars for public archives.
- *Education & Research Tools:* Development of software for scholars, students, and historians to study ancient scripts interactively.
- *Digital Libraries & Archives:* Integration with digital libraries for searchable and annotated inscription repositories.
- *Augmented Reality Applications:* Overlay reconstructed or translated text on physical artifacts for educational exhibits or tourist guides.
- *Artificial Intelligence-Assisted Translation:* Assist linguists in transliterating or translating ancient scripts using segmented character datasets.
- *Historical Text Preservation:* Aid in creating high-fidelity digital copies of deteriorating inscriptions for long-term preservation and restoration projects.

ACKNOWLEDGEMENTS

The authors are grateful to the Archeological Survey of India (ASI), Mysore, India for providing access to the Brahmi inscription images used in this study. The authors also thank their respective institutions for the support and resources provided during the development and evaluation of the proposed deep learning framework.

AUTHOR CONTRIBUTIONS

Sandeep Kaur: Conceptualization, methodology, **Deepak Kumar Verma:** Methodology, supervision, validation, , and critical revision of the manuscript, **Gurrajan Singh:** Data analysis, experimentation, **Om Prakash Suthar:** Methodology, deep learning model analysis, validation, manuscript review, and editing, **Chaste Sauveur Uwambazimana:** data analysis, interpretation, **Chetankumar Chudasama:** visualization, and manuscript review, **Rahul Bhandari:** validation, and critical review of the manuscript.

CONFLICT OF INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have been perceived to influence the work reported in this paper.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

DATA AVAILABILITY STATEMENT

No external datasets were used in this study.

ORCID

Sandeep Kaur:	0000-0002-9545-7470
Deepak Kumar Verma:	0000-0002-4745-1752
Gurrajan Singh:	0009-0001-8955-3837
Om Prakash Suthar:	0000-0002-7840-481X
Chaste Sauveur Uwambazimana:	NA
Chetankumar Chudasama:	0000-0003-4816-0196
Rahul Bhandari:	0000-0001-9764-0466

ABBREVIATIONS

OCR - Optical Character Recognition

CNN - Convolutional Neural Network

U-Net–U-Net architecture (a type of CNN used for image segmentation).

HOG - Histogram of Oriented Gradients

MDF - Medium Density Fiberboard

VGG16 - Visual Geometry Group 16 (a pre-trained CNN model)

ResNet - Residual Network (a type of deep neural network)

LeNet - LeNet architecture (a kind of CNN used for digit recognition)

GoogLeNet - Google LeNet architecture (a type of deep CNN)

MTP - Media Transfer Protocol

PTP - Picture Transfer Protocol

SVM - Support Vector Machine

t-SNE - t-Distributed Stochastic Neighbor Embedding

PCA - Principal Component Analysis

REFERENCES

[1] R. Mokashi and P. V. Samel, Brahmi Inscriptions from Kondane Caves, *Ancient Asia* **89**(3) (2017), seven pages. http://doi.org/10.5334/aa.114

[2] S. Kaur and B. B. Sagar, Brahmi character recognition based on SVM (support vector machine) classifier using image gradient features, *Journal of Discrete Mathematical Sciences and Cryptography* **22** (2019), 1365-1381. https://doi.org/10.1080/09720529.2019.1692445

[3] K Jeevitha, A. Iyswariya, V. R. Kumar, S. M. Basha, and V. P.Kumar, A Review on various Segmentation techniques in image processing, *European Journal of Molecular & Clinical Medicine* **7**(4) (2020),1342-1348.

[4] L. F. da Silva, A. Conci and Angel Sanchez, Word-level segmentation in printed and handwritten documents, In *Proceedings of the IEEE 2011 18th International Conference on Systems, Signals, and Image Processing,* IEEE, 2011, Sarajevo, Bosnia and Herzegovina, 1-4.

[5] J. Jo, H. I. Koo, J. W. Soh and N. I. Cho, Handwritten Text Segmentation via End-to-End Learning of Convolutional Neural Networks, *Multimedia Tools Applications* **79** (2020), 32137–32150. https://doi.org/10.1007/s11042-020-09624-9

[6] A.P. Singh and A.K. Kushwaha, Analysis of Segmentation Methods for Brahmi Script, *DESIDOC Journal of Library & Information Technology* **39**(2) (2019), 109-116. https://doi.org/10.14429/djlit.39.2.13615

[7] D. Pugazhenthi and S. A. Vallarasi, Offline Character Recognition of Printed Tamil Text using Template Matching Method of Bamini Tamil Font, I*ndian Journal of Science and Technology* **8** (35) (2015), four pages. https://doi.org/10.17485/ijst/2015/v8i35/86811

[8] P. Preethi, H.R. Mamatha and H. Viswanath, Study of using hybrid deep neural networks in character extraction from images containing text, *Trends in Computer Science and Information Technology* **6** (2) (2021), 45- 52. https://dx.doi.org/10.17352/tcsit.000039

[9] S. Khan, H. Rahmani, S. A. A. Shah and M. Bennamoun, *Convolutional Neural Network. A Guide to Convolutional Neural Networks for Computer Vision*, Morgan & Claypool, 2018.

[10] K. Simonyan and A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv:1409.1556*, 2014. https://arxiv.org/abs/1409.1556

[11] O. Ronneberger, P. Fischer and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Navab, N., Hornegger, J., Wells, W., Frangi, A. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, MICCAI 2015, Lecture Notes in Computer Science()*, **vol 9351**. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28

[12] S. Jadon, A survey of loss functions for semantic segmentation, In *Proceedings of the 2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB),* IEEE, 2020, Via del Mar, Chile, seven pages. https://doi.org/10.1109%2Fcibcb48159.2020.9277638

[13] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz and D. Terzopoulos, Image Segmentation Using Deep Learning: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44** (7) (2022), 3523-3542. https://doi.org/10.1109/TPAMI.2021.3059968

[14] C. Shorten and T.M. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning, *J Big Data* **6**(60) (2019). https://doi.org/10.1186/s40537-019-0197-0

[15] S.M. Aswatha, A.N. Talla, J. Mukhopadhyay, and P. Bhowmick, A Method for Extracting Text from Stone Inscriptions Using Character Spotting, In *Jawahar, C., Shan, S. (eds) Computer Vision - ACCV 2014 Workshops. ACCV 2014. Lecture Notes in Computer Science ()*, **Vol. 9009**, 2015, Springer, Cham. https://doi.org/10.1007/978-3-319-16631-5_44

[16] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, Patch-Based Convolutional Neural Network for Whole Slide Tissue Image Classification, In *Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, Las Vegas, NV, USA, 2424-2433. https://doi.org/10.1109/CVPR.2016.266

[17] R. Zhang, L. Du, Q. Xiao, and J. Liu, Comparison of Backbones for Semantic Segmentation Network, *Journal of Physics: Conference Series* **1544**(012196) (2020), Suzhou, China. https://doi.org/10.1088/1742- 6596/1544/1/012196

[18] A. Pravitasari, N. Iriawan, M. Almuhayar, T. Azmi, I. Irhamah, K. Fithriasari, S.W. Purnami and W. Ferriastuti, UNet-VGG16 with transfer learning for MRI-based brain tumor segmentation, *TELKOMNIKA (Telecommunication Computing Electronics and Control)* **18**(3) (2020), 1310-1318. http://dx.doi.org/10.12928/telkomnika.v18i3.14753

[19] E. Shelhamer, J. Long and T.Darrell, Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**(4) (2017), 640–651. https://doi.org/10.1109/TPAMI.2016.2572683

[20] S. Karimpouli and P. Tahmasebi, Segmentation of digital rock images using deep convolutional autoencoder networks, *Computers & Geosciences* **126** (2019), 142-150. https://doi.org/10.1016/j.cageo.2019.02.003

[21] R. Augustauskas, A. Lipnickas and T. Surgailis, Segmentation of Drilled Holes in Texture Wooden Furniture Panels Using Deep Neural Network. *Sensors 21* **11**(2021), Article 3633. https://doi.org/10.3390/s21113633

[22] R. Hu, J. M. Odobez and D. G. Perez, Extracting Maya Glyphs from Degraded Ancient Documents via Image Segmentation, *J. Comput. Cult. Herit.* **10** (2) (2017), 23 pages. https://doi.org/10.1145/2996859

[23] A.K. Datta, A Generalized Formal Approach for Description and Analysis of Major Indian Scripts, *IETE Journal of Research* **30**(6) (1984), 155-161. https://doi.org/10.1080/03772063.1984.11453262

[24] X.Y. Gong, H. Su, D. Xu, Z.T. Zhang, F. Shen, and H.B. Yang, An Overview of Contour Detection Approaches, *International Journal of Automation and Computing* **15** (2018), 656–672. https://doi.org/10.1007/s11633-018-1117-z

[25] C.H. Sudre, W. Li, T. Vercauteren, S. Ourselin and M. Jorge Cardoso, Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations, In Cardoso, M. *et al. Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support. DLMIA ML-CDS 2017 2017. Lecture Notes in Computer Science*, **Vol. 10553** (2017), Springer, Cham. https://doi.org/10.1007/978-3-319-67558-9_28

[26] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. F. Fei, ImageNet: a Large-Scale Hierarchical Image Database. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2019, Miami, FL, USA 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[27] D. P. Kingma and J. L. Ba, Adam: A Method for Stochastic Optimization, In *Proceedings of the 3rd International Conference for Learning Representations*. San Diego, 2014, 15 pages. https://doi.org/10.48550/arXiv.1412.6980

[28] S. Kaur and B. B. Sagar, Efficient, scalable template-matching technique for ancient Brahmi script image, *Computers, Materials & Continua* **74**(1) (2023), 1541–1559. https://doi.org/10.32604/cmc.2023.032857

[29] H.M. Kathiriya and M.M. Goswami, Word spotting techniques for Indian scripts: a survey, in *International Conference on Innovations in Power and Advanced Computing Technologies (i-PACT 2017)*, pp. 1–5, 2017.

[30] M. Mathew, A.K. Singh, and C.V. Jawahar, Multilingual OCR for Indic scripts, in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, Santorini, pp. 186–191, 2016.

[31] A. Kunchukuttan, R. Puduppully, and P. Bhattacharyya, Brahmi-Net: a transliteration and script conversion system for languages of the Indian subcontinent, in *Proceedings of NAACL-HLT*, pp. 81–85, 2015.

[32] N. Gautam and S.S. Chai, Optical character recognition for Brahmi script using geometric method, *Journal of Telecommunications, Electronics and Computer Engineering* 9 (2017), 131–136.

[33] N. Warnajith, D. Bandara, N. Bandara, A. Minati, and S. Ozawa, Image processing approach for ancient Brahmi script analysis (Abstract), *University of Kelaniya*, Colombo, Sri Lanka, p. 69, 2015.

[34] K.B.M.R. Batuwita and G.E.M.D.C. Bandara, New segmentation algorithm for individual offline handwritten character segmentation, in *Fuzzy Systems and Knowledge Discovery (FSKD)*, Lecture Notes in Computer Science, vol. 3614, Springer, Berlin, Heidelberg, 2005.

[35] R. Kumar and A. Singh, Detection and segmentation of lines and words in Gurmukhi handwritten text, in *IEEE 2nd International Advance Computing Conference*, pp. 353–356, 2010.

[36] H.K. Anasuya Devi, Thresholding: a pixel-level image processing methodology preprocessing technique for an OCR system for the Brahmi script, *Ancient Asia* 1 (2006), 161–165.

[37] C.H. Patil and S.M. Mali, Segmentation of isolated handwritten Marathi words, in *National Conference on Digital Image and Signal Processing*, pp. 21–26, 2015.

[38] A.S. Nagane and S.M. Mali, Segmentation of special character "∴" from degraded Brahmi script documents, in *Two Days 2nd National Conference on 'Innovations and Developments in Computational & Applied Science' (NCIDCAS-2018)*, pp. 65–67, 2018.

[39] P. Rajnish, K.P. Kamath, B. Kumar, M. Nishanth, and P. Preethi, Improving the quality and readability of ancient Brahmi stone inscriptions, in *2023 2nd International Conference for Innovation in Technology (INOCON)*, Bangalore, India, 2023, pp. 1–8. https://doi.org/10.1109/INOCON57975.2023.10101244.

[40] R.G. Salomon, Forgotten Buddhist Literatures as Revealed by Manuscript Discoveries in India and Beyond, *Philological Encounters* 1, no. aop (2024), 1–24.

[41] A. Pratap Singh, A. Kumar, and Kushwaha, Analysis of Segmentation Methods for Brahmi Script, *DESIDOC Journal of Library & Information Technology* 39(2) (2019), 109–116. doi: 10.14429/DJLIT.39.2.13615.

[42] A. S. Nagane and S. M. Mali, Segmentation of Characters from Degraded Brahmi Script Images, in *Advances in Computing and Artificial Intelligence* (2020), 326–338. doi: 10.1007/978-981-15-4029-5_33.

[43] K.A.S.A. N. Wijerathna, R. Sepalitha, T. Indika, H. Athauda, P.D. Suranjini, J.A.D.C. Silva, and A. Jayakodi, Recognition and Translation of Ancient Brahmi Letters Using Deep Learning and NLP, in *2019 IEEE International Conference on Advances in Computing (ICAC)* (2019). doi: 10.1109/ICAC49085.2019.9103340.

[44] A. Kumar and R. Kumari, Visualizing the Lost Script of South Asia Under the Umbrella of the Digital Humanities Tool VisualEye, *DESIDOC Journal of Library & Information Technology* 43(1) (2023), 11–20. doi: 10.14429/djlit.43.01.18625.

[45] K.S. Murthy, G.H. Kumar, P. Shivakumar, and P. Ranganath, Nearest Neighbor Clustering-Based Approach for Line and Character Segmentation in Epigraphical Scripts, *Indian Conference on Pattern Recognition and Image Analysis* (2004).

[46] K. Poornimathi, V. Muralibhaskaran, and L. Priya, A Comparative Study on Deep Learning-Based Segmentation Techniques for Tamil Inscriptions, in *IEEE International Conference on Systems, Man, and Cybernetics (i-SMAC)* (2024), 1976–1982. doi: 10.1109/i-smac61858.2024.10714635.