International Journal of Advanced Technology and Engineering Exploration, Vol 12(126)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2024.111100193

Research Article

A deep learning approach for recognizing ancient Tamil scripts from historical artifacts

A. Umamageswari[1*], S. Deepa[2], L. Sherin Beevi[3] and A. Sangari[4]
Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India[1]
Assistant Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, India[2]
Assistant Professor, Department of Computer Science and Engineering, R.M.D Engineering College, Kavaraipettai, Tamilnadu, India[3]
Associate Professor, Department of Electrical and Electronics Engineering, Rajalakshmi Engineering College, Thandalam, Tamilnadu, India[4]

Received: 21-February-2024; Revised: 28-April-2025; Accepted: 12-May-2025

Abstract
The preservation and interpretation of ancient scripts are essential for uncovering the rich cultural heritage and historical knowledge of early civilizations. Tamil, one of the world's oldest languages, contains a vast repository of information preserved in inscriptions, manuscripts, and other historical artifacts. This research proposes a novel methodology for the recognition and deciphering of ancient Tamil words from historical documents and artifacts using image processing techniques and deep learning (DL) algorithms. The proposed framework consists of three main stages: pre-processing, feature extraction, and DL-based recognition. A region-based convolutional neural network (RCNN) architecture is employed to automatically learn and identify the intricate patterns and structural elements of ancient Tamil characters. To enhance feature extraction for irregularly shaped characters, this work introduces Adaptive region of interest (ROI) pooling, which dynamically adjusts to variations in stroke patterns and inscription styles, thereby improving recognition accuracy. The RCNN is trained on a large annotated dataset of ancient Tamil word images, with labels verified by domain experts to ensure data accuracy and reliability. Extensive experiments were conducted on a diverse dataset comprising stone inscriptions, palm leaf manuscripts, and clay tablets. The proposed approach achieved a high recognition accuracy of 98.6%, demonstrating robust performance even under challenging conditions such as stylistic variations, surface degradation, and image noise. This method significantly contributes to the preservation of cultural heritage by enabling the digitization and accessibility of historical texts and inscriptions. By safeguarding valuable linguistic and cultural knowledge, the proposed system ensures its availability for future scholarly research and education.

Keywords
Ancient Tamil, Character recognition, Segmentation, RCNN, Deep learning, Computer vision.

1.Introduction
In today's rapidly evolving technological landscape, the fusion of machine learning and linguistic heritage has paved the way for innovative applications that bridge the gap between the past and the present. One such remarkable endeavor is the field of Tamil ancient word recognition, where cutting-edge machine learning techniques intersect with the rich tapestry of the Tamil language's historical evolution [1].

Traditional methods of transcription and analysis of these scripts are time-consuming and rely on specialized epigraphists. In recent years, image processing and deep learning (DL) techniques have emerged as promising approaches to automate the recognition of ancient scripts. By combining optical character recognition (OCR) and convolutional neural networks (CNN), these advanced techniques enable computers to identify and interpret complex patterns in historical Tamil inscriptions, thereby facilitating the digital preservation of ancient knowledge [2, 3]. Tamil, one of the world's oldest

*Author for correspondence

languages with a lineage spanning millennia, carries within its script a treasure trove of wisdom, stories, and cultural nuances [4].

The complexity and variety present in ancient Tamil scripts pose a challenge for deciphering and understanding its context. Machine learning, a branch of artificial intelligence (AI), equips us with tools to analyze patterns, discern meanings, and extract knowledge from vast volumes of textual data [5, 6]. By training algorithms on meticulously curated datasets of ancient Tamil inscriptions, manuscripts, and texts, we embark on a journey to create models capable of recognizing and interpreting the intricate characters and words that have endured the test of time. Recognizing ancient Tamil words using image processing and DL is a challenging yet fascinating task [7, 8].

The objective of this paper is to develop an efficient DL-based system for recognizing and deciphering ancient Tamil words from historical artifacts using image processing and region-based convolutional neural network (RCNN) architecture. It aims to improve recognition accuracy by addressing challenges such as script degradation, irregular stroke patterns, and stylistic variations. The ultimate goal is to support the digitization and preservation of Tamil cultural heritage through automated script analysis.

In recent years, advancements in recognition technologies—particularly computer-based methods—have demonstrated significant potential in assisting archaeologists with the accurate identification of ancient Tamil characters and words. The pursuit of deciphering ancient scripts such as Tamil through image processing and DL is driven by a multifaceted motivation: the preservation and interpretation of cultural heritage, the extraction of historical insights, the advancement of linguistic research, and the integration of cutting-edge technology into historical inquiry. Additionally, these efforts provide valuable educational resources for learners in history, linguistics, and interdisciplinary fields. This convergence of technology and the humanities reflects a growing commitment to uncovering the past using modern tools, offering enhanced recognition capabilities, greater efficiency, and more reliable results.

The structure of the paper is as follows: Section 1 presents the introduction, while Section 2 outlines the literature survey. Section 3 details the algorithms and proposed methods. Section 4 discusses the experimentation and result analysis, along with a detailed comparison with existing techniques. Finally, Section 5 concludes the study and highlights directions for future work.

2.Literature review

This section provides a brief overview of the significance of OCR technology for the Tamil script, particularly in the context of digitizing historical and cultural texts. It emphasizes the unique challenges associated with the Tamil script and underscores the necessity for advanced OCR techniques. Furthermore, it discusses the historical and cultural importance of Tamil literature and the growing need to leverage modern technology for its effective digitization and analysis.

The recognition of handwritten Tamil characters using neural networks represents a significant area of research within the fields of AI and machine learning. This involves the development of models capable of automatically identifying and classifying handwritten characters in the Tamil script. Neural networks—particularly CNNs—have demonstrated promising results in various character recognition tasks, including those involving the com-plex structure of Tamil characters [9, 10].

The preservation and digitization of historical artifacts and documents play a crucial role in maintaining cultural heritage and enabling scholarly research. Ancient Tamil scripts, found on delicate palm leaves and enduring stone inscriptions, hold a treasure trove of information about the language, culture, and history of Tamil-speaking civilizations [11]. However, deciphering and digitizing these scripts manually is a time-consuming and challenging task due to the complexity of characters, variations in handwriting, and potential degradation over time [12].

In recent years, the field of AI and machine learning has offered transformative solutions to automate the recognition and analysis of ancient scripts. AI-based character recognition systems leverage cutting-edge technologies such as CNNs, recurrent neural networks (RNNs), and advanced image processing techniques to decipher and transcribe characters from palm leaves and stone inscriptions [13]. This combination and linguistics open up new avenues for unlocking the insights hidden within these ancient artifacts.

Recognizing characters from palm leaves and stone inscriptions presents unique challenges [14]. The ancient Tamil script includes intricate ligatures, diacritics, and variations in character forms, making manual transcription arduous and error-prone. Additionally, factors such as variations in handwriting styles, potential deterioration of inscriptions, and the sheer volume of characters further complicate the task [15]. AI-based character recognition offers an innovative approach to overcoming these challenges. By harnessing the power of machine learning algorithms and neural networks, researchers and historians can automate the process of character identification and transcription. This not only accelerates the digitization of ancient texts but also ensures higher accuracy and consistency, leading to more reliable outputs for linguistic analysis, historical research, and cultural preservation [16].

A DL approach, specifically using CNNs, has been explored for recognizing characters from ancient Tamil palm leaf manuscripts [17]. Another study investigates the application of DL techniques, including CNNs, for recognizing handwritten Tamil characters, with potential extensions to ancient scripts [18]. Various machine learning methods have also been employed to recognize Tamil characters in ancient inscriptions, contributing significantly to the field of character recognition [19]. Additionally, some works have focused on the use of support vector machines (SVMs) for handwritten Tamil character recognition, which may also be applicable to ancient scripts [20]. Recent studies have proposed CNN-based methods for character recognition in ancient Tamil manuscripts, highlighting the challenges posed by variations in handwriting styles [21]. Furthermore, pattern recognition techniques have been applied for the automatic transcription of characters from Tamil inscriptions [22].

The work investigates the effectiveness of DL approaches in recognizing ancient scripts. It explores CNNs and RNNs for classification and transcription tasks, providing insights into the application of modern machine learning techniques in deciphering ancient texts [23]. A new approach proposed enhanced image processing techniques tailored for analyzing ancient inscriptions. Their work addresses challenges such as noise removal, character segmentation, and feature extraction, offering practical solutions for improving the accuracy of ancient script recognition systems [24]. A neural machine translation (NMT) approach was suggested for automatically translating cuneiform texts. By leveraging the power of DL, a method demonstrates significant improvements in the accuracy and efficiency of cuneiform translation, contributing to the broader field of ancient language processing [25]. A researcher conducted a comparative study of DL models for document image analysis, including applications in ancient script recognition. Their comprehensive analysis sheds light on the strengths and limitations of different neural network architectures, guiding researchers in selecting suitable models for their specific tasks [26]. Digital preservation strategies were presented for ancient manuscripts, focusing on the challenges of preserving fragile and deteriorating documents. Through a case study approach, the effectiveness of digitization techniques and data management protocols in safeguarding valuable historical artifacts for future generations has been demonstrated [27].

A semi-supervised learning approach was used for ancient script recognition when training data is limited. By leveraging both labeled and unlabeled samples, their method achieves competitive performance, offering practical solutions for researchers working with scarce historical datasets [28]. Deep fusion networks are used for multi-modal ancient document analysis, combining text and visual information for improved understanding and interpretation of historical texts. Their approach integrates text recognition with image processing, enabling comprehensive analysis of diverse ancient manuscripts.

The review analysis of the selected papers provides a comprehensive overview of recent advancements in ancient script recognition and decipherment, highlighting a diverse range of methodologies and applications. Key themes include the widespread adoption of DL approaches, advancements in image processing techniques tailored for ancient manuscripts, cross-domain applications of machine learning techniques, interdisciplinary collaboration, and efforts to improve the robustness and reliability of recognition systems. These findings highlight the interdisciplinary nature of research in digital humanities and demonstrate the potential for advanced computational techniques to contribute significantly to historical scholarship and cultural heritage preservation. Continued research in this field promises further breakthroughs in understanding and interpreting ancient texts, enriching our understanding of human history and culture. Although existing research has significantly advanced the recognition of ancient characters across various centuries, no technology has yet been developed to accurately decipher ancient Tamil words along with their semantic meanings. Furthermore, advanced technologies such as augmented reality (AR) and virtual reality (VR) have not been effectively applied to ancient scripts to facilitate easy reading and interpretation in modern Tamil or English. To address this gap, the proposed work attempts to recognize complete words from images of ancient Tamil stone inscriptions, thereby aiding in the understanding of historical heritage and cultural context.

3.Methods

This section outlines the methodology employed and provides a detailed description of the datasets used in this study.

Stone inscription and palm leaves dataset

The dataset for ancient Tamil word recognition was sourced from GitHub and comprises digitized records from various temple stone inscriptions across centuries [29]. It includes 65 classes of ancient Tamil characters, with each class containing 50 distinct images, making it well-suited for training DL models aimed at recognizing and digitizing ancient scripts. One of the techniques used in the dataset preparation is e-stampage, which involves creating inked impressions of inscriptions for detailed study and long-term preservation.

Additionally, this study references the SleukRith Set, a structured dataset developed specifically for Khmer palm leaf manuscripts. It contains annotated data from 657 pages, divided into three categories: isolated characters, words, and lines. Isolated characters are manually segmented with polygon boundaries and labeled with their corresponding Unicode sequences. These form the basis for constructing words, aligned with the Khmer Unicode's consonant-first, vowel-second structure. Lines are created through manual grouping to reflect the logical flow of the manuscripts. All annotations are stored in XML files with detailed character- and word-level metadata, including coordinates, IDs, and labels. While the SleukRith Set is originally intended for Khmer texts, its annotation strategy and structure serve as a valuable reference for digitizing and preserving similar ancient scripts, including Tamil (*Figure 1*).

For this study, a combined dataset was used—referred to as the Stone Inscription Dataset and the Palm Leaves Dataset—comprising digitized texts from ancient Tamil stone carvings and palm leaf manuscripts.

Dataset specifications:

- **Total samples:** 10,403 images
- **Source materials:** Stone inscriptions and palm leaf manuscripts
- **Language covered:** Ancient Tamil
- **File format:** JPG

Figure 1 Ancient Tamil stone inscription and ancient Tamil palm leaf manuscripts

Figure 2 illustrates the architectural diagram of the proposed system. The input to the system consists of image data, such as digital copies of ancient Tamil text obtained from stone inscriptions or palm leaf manuscripts. For the purpose of this study, only stone inscriptions are considered, although the methodology is applicable to various types of ancient script images. The input image undergoes a series of preprocessing steps to enhance its quality for further processing. These steps include cropping, binarization using the Otsu method, and filtering (to remove salt-and-pepper noise) using Gaussian and median filters, followed by image smoothing to facilitate accurate character recognition.

The pre-processed image is then passed to a contour-based convex hull image segmentation algorithm [30], which isolates individual characters. These segmented characters are subsequently fed into a CNN for character recognition. Finally, recognized characters are matched against an online Tamil dictionary to reconstruct and identify meaningful Tamil words.

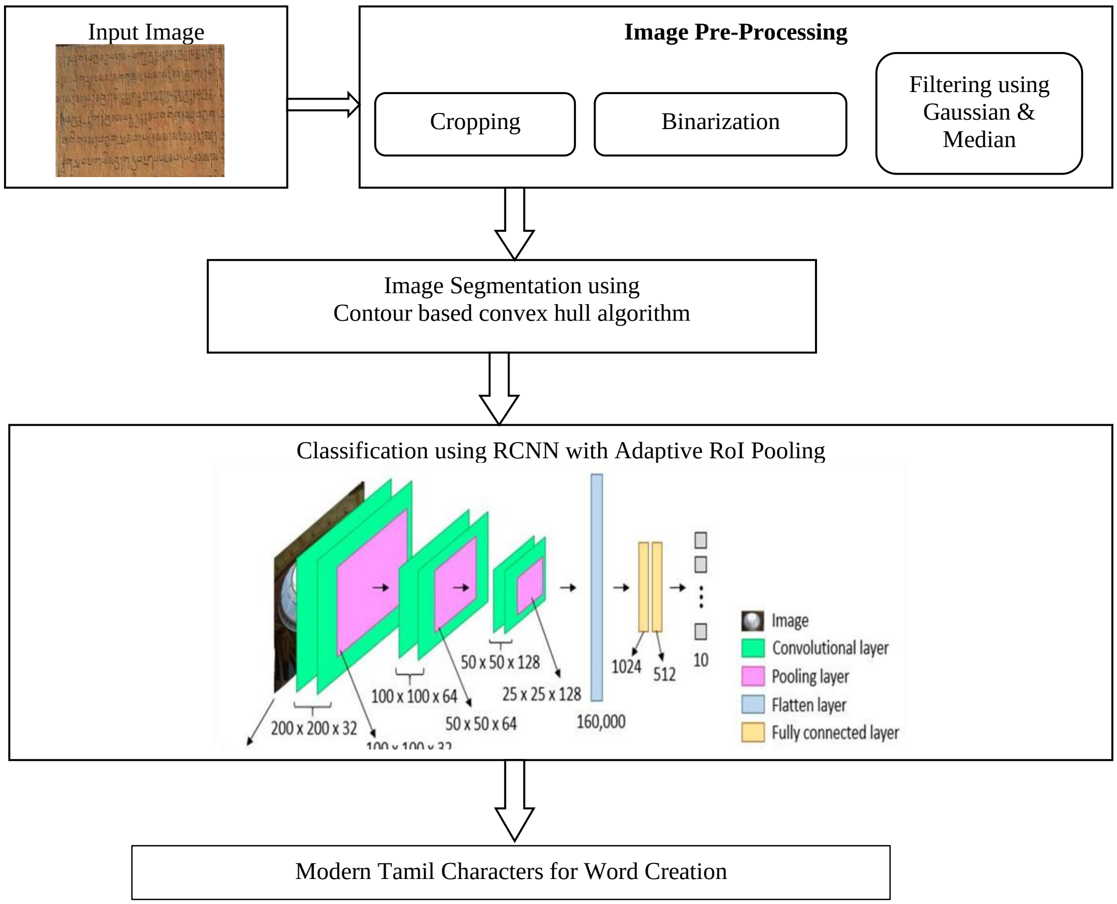


Figure 2 Architecture diagram of the proposed method

Proposed work image pre-processing

The first phase of the proposed system involves image pre-processing, which prepares raw input images for accurate character recognition. The process begins with cropping, where a rectangular region of interest (RoI) is selected from the original image based on specified coordinates. This is followed by binarization using the Otsu method, which automatically determines an optimal threshold to convert the grayscale image into a binary format, separating foreground text from the background. To further enhance the image quality, Gaussian and median filtering techniques are applied. These filters help remove salt-and-pepper noise and smooth the image, preserving essential features while improving overall clarity for downstream character segmentation and recognition.

Binarization using Otsu method

The Otsu's thresholding method is a popular technique for automatically binarizing grayscale images. It's based on finding an optimal threshold that maximizes the separation between the two classes of pixel intensities (foreground and background). Here's a mathematical overview of the Otsu's thresholding algorithm.

Given a grayscale image with pixel intensities in the range [0, L-1], where L is the number of possible intensity levels, the Otsu's algorithm aims to find a threshold T that separates the image into two classes: pixels with intensities less than T (background) and pixels with intensities greater than or equal to T (foreground).

Step 1: Calculate the histogram of pixel intensities in the image, counting the frequency of each intensity value.

Let *p(i)* represent the probability (normalized frequency) of intensity level iii in the image, where this method takes values from 0 to L−1 (where L is the number of gray levels, typically 256 for an 8-bit image).

The histogram *p(i)* is computed using Equation 1.

$$p(i) = n(i)/N \quad (1)$$

Where:

- *n(i)* is the number of pixels with intensity level i,
- *N* is the total number of pixels in the image and equation 1 shows the histogram.

Step 2: *Normalize the histogram so that the sum of all frequencies is 1.*

Step 3: *Compute the cumulative distribution function (CDF) and the cumulative mean intensity up to each intensity level.*

Let:

- *ω0(t)* be the probability of the background class (pixels with intensity ≤t),
- *ω1(t)* be the probability of the foreground class (pixels with intensity >t),
- *μ0(t)* be the mean intensity of the background class,
- *μ1(t)* be the mean intensity of the foreground class.

Equations 2 and 3 show how the class probabilities are calculated:

$$\omega 0(t) = \sum_{t=0}^{L-1}(p(i) \quad (2)$$

$$\omega 1(t) = 1 - \omega 0(t) = \sum_{t=0}^{L-1}(p(i) \quad (3)$$

Equations 4 and 5 show how the class means are calculated:

$$\mu 0(t) = \frac{\sum_{i=0}^{t} i.p(i)}{\omega 0(t)} \quad (4)$$

$$\mu 1(t) = \frac{\sum_{i=t+1}^{L-1} i.p(i)}{\omega 1(t)} \quad (5)$$

Step 4: Calculate the global mean intensity of the entire image, Equation 6 shows the formulae for global mean intensity (Equation 6),

$$\mu T = \sum_{i=0}^{L-1} i.(p(i) \quad (6)$$

Step 5: For each intensity level

- Calculate the probabilities of background and foreground classes up to intensity level
- Calculate the mean intensities of the background and foreground classes up to intensity level Calculate the between-class variance and it is shown in Equation 7.

$$\sigma_B^2(t) = \omega 0(t).\omega 1(t).\mu 0(t).\mu 1(t) \quad (7)$$

Step 6: Find the optimal threshold, The threshold that maximizes the optimal threshold T (Maximum value is 255) and it is shown in Equation 8.

$$t^* = \arg\max t\, \sigma_B^2(t) \quad (8)$$

Filtering using gaussian and median filtering

Gaussian and median filtering are two common image processing techniques used to smooth or denoise images. These filters have various applications in computer vision, image processing, and signal processing. The pro-posed work provides algorithms for both Gaussian and median filtering [31]. Both filtering techniques involve sliding a window (kernel) over the image and performing calculations based on the pixel values within the window. The kernel size and the standard deviation in Gaussian filtering, as well as the neighborhood size in median filtering, are parameters that can be adjusted to control the level of smoothing or denoising applied to the image [32].

Gaussian and median filtering

Gaussian filtering is a smoothing technique that involves convolving the image with a Gaussian kernel, which is a two-dimensional (2D) distribution. This kernel gives higher weight to the central pixels and progressively less to those farther from the center, resulting in a weighted average that effectively reduces Gaussian noise and pre-serves image structure. On the other hand, median filtering replaces each pixel's value with the median of the intensity values of its neighboring pixels. This method is particularly effective in removing salt-and-pepper noise and isolated outliers, making it a preferred choice for denoising while preserving edges [33, 34]. *Figure 3* displays the original input image selected for the pre-processing phase. *Figure 4* illustrates the result after ap-plying the cropping operation to isolate the RoI. *Figure 5* presents the binarized image obtained using the Otsu thresholding method. Finally, *Figure 6* shows the output after applying median and Gaussian filtering, which effectively smooths the image and reduces noise in preparation for subsequent processing steps.

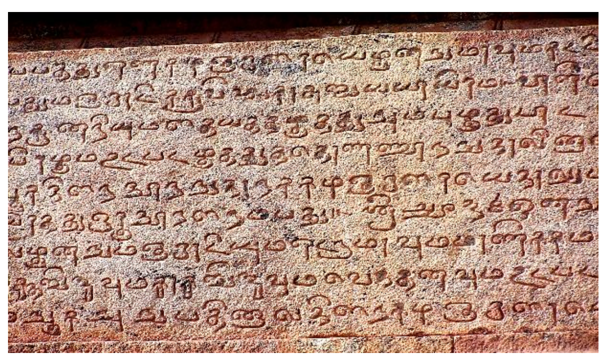


Figure 3 Original input image

Figure 4 Pre-processed image (Cropping)

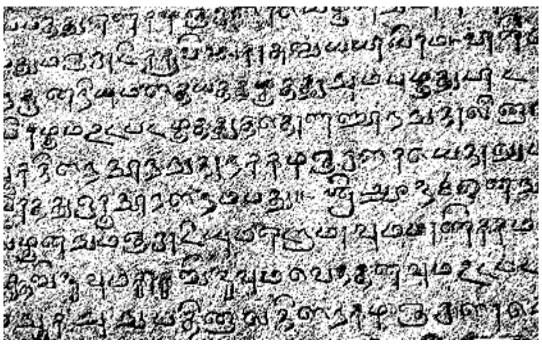


Figure 5 Pre-processed image (Binarization)

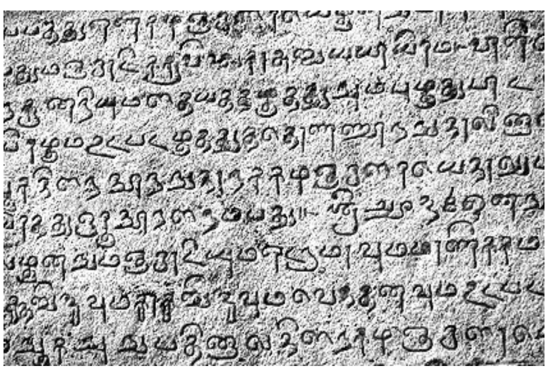


Figure 6 Pre-processing (Filtration)

Image segmentation using contour based convex hull bounding box algorithm

The Convex Hull Bounding Box segmentation algorithm segments objects in an image by calculating the convex hull for each object's contour and then creating a bounding box around the convex hull. This process helps separate and label distinct objects within the image [35, 36]. These steps are explained below in a step-by-step manner as part of the algorithm:

Given an original image I, the objective is to segment the individual objects by computing their convex hulls and generating bounding boxes around each detected region.

Step 1: *Edge Detection: Apply an edge detection algorithm to the original image I to obtain an edge-detected image E.*

Step 2: *Contour Extraction: Extract the contours C of objects from the edge-detected image E using a contour extraction algorithm.*

Step 3: *Convex Hull Calculation: For each contour C:*

a. Compute the convex hull using a convex hull algorithm.

b. Store the vertices of the convex hull as a set of points.

Step 4: *Bounding Box Generation: For each convex hull*

- *Find the minimum and maximum x and y coordinates of the points.*
- *Calculate the width and height of the bounding box.*

Step 5: *Segmentation and Labeling*

Figure 7 shows the output of segmentation process. The *Figure 7* represents that the proposed algorithm correctly segmented all the characters using contour based convex hull bounding box algorithm [37, 38].

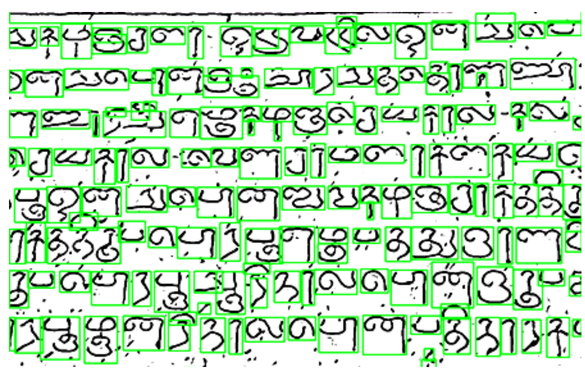


Figure 7 Character segmentation using contour based bounding box Convex Hull Algorithm

Modern Tamil character conversion using RCNN algorithm

Given a pre-processed and segmented image, this work aims to recognize ancient Tamil characters using an RCNN approach. Ancient Tamil scripts often exhibit irregular stroke structures with variations in size, shape, and curvature, making it difficult for traditional fixed ROI pooling methods to effectively capture critical char-acter features. Standard ROI pooling applies a uniform grid, which may not align well with the complex and di-verse patterns found in historical inscriptions. To overcome this limitation, Adaptive ROI Pooling is introduced as an enhancement to the RCNN framework. This technique dynamically adjusts the pooling regions according to the character's shape and structure, ensuring that essential visual features are retained for accurate recognition.

Step 1: Character Proposal

Obtain a set of character proposals (candidate bounding boxes) using a region proposal algorithm:

Char_RoIs=CharacterProposalAlgorithm(segmented_image)

Step 2: Feature Extraction

For each character RoI, extract features using a pre-trained CNN model:

feauresChar_RoI=CNN_Feature_Extractor(Char_RO)

Step 3: Character Classification

Train a character classifier to distinguish between different Tamil characters:

characer_probs=CharacterClassifier(featuresChar_RoI)

Step 4: Non-Maximum Suppression (NMS)

Apply NMS to filter out duplicate character detections and keep only the most confident character RoIs:

selected_Char_RoIs = NMS Char_ RoIs, character_probs)

Step 5: Character Recognition

For each selected character RoI, identify the recognized Tamil character based on the character classifier's output probabilities:

recognized_characters=Argmax(character_probs)

Step 6: Post-Processing and Refinement

Refine the recognized characters using linguistic and contextual analysis, considering the sequence of characters and grammar rules:

final_recognized_characters= Linguistic_Refinement (recognized_characters)

RCNN is a class of DL models designed for character recognition and localization tasks [39, 40]. It was introduced to overcome the limitations of earlier approaches that treated region proposal and classification as two separate processes. RCNN integrates these steps into a single end-to-end framework. The CNN processes each proposed region to extract relevant features, with the complexity dependent on the CNN architecture and the number of candidate regions. These extracted features are then classified into corresponding character categories through the training and evaluation of dedicated classifiers. RCNN's generalizability refers to its ability to perform effectively across diverse character recognition tasks and datasets, even beyond the scope of its original training. This is achieved through techniques such as transfer learning, data augmentation, fine-tuning, handling domain shifts, and leveraging transferable features [41, 42].

Semantic understanding in ancient Tamil word recognition using DL extends beyond simple character or word recognition; it focuses on interpreting the meaning and contextual relevance of the text within a broader historical and linguistic framework. The proposed methods aim to achieve this by incorporating various techniques, including natural language generation (NLG) models. These models are capable of summarizing or explaining the semantic content of the recognized text in a human-readable format. By leveraging a Tamil dictionary and linguistic rules, the system provides meaningful insights into the recognized words, thereby facilitating deeper understanding and interpretation of ancient inscriptions [43–46].

Figure 8 shows the output of the last stage of the proposed method, which identifies the words equivalent to modern Tamil from the image of Ancient Tamil. Since the dataset contains images with distortions and contains a lot of null values, it is not feasible to identify all the words required to make meaningful sentences.

மானாயம்பாறை

ஸ்ரீ ராஜ ராஜ ஈகை

Figure 8 Words identified after character conversion

4.Results and discussion

Measuring the performance of ancient Tamil word and character recognition using RCNN is essential for developing a reliable and effective recognition system. It serves to validate the accuracy of the proposed model, in-form necessary improvements, and ensure that the system meets its intended goals—whether for research, archival, or preservation purposes. The implementation was carried out in Python, using hardware with an Intel Core i5 or Ryzen 5 processor (or higher), featuring quad-core capabilities and a minimum of 8GB random access memory (RAM). Performance evaluation involves several key parameters and metrics that assess the model's accuracy, robustness, and ability to generalize. This process provides an objective measure of the RCNN model's effectiveness and helps determine its suitability for tasks such as scholarly research and the digitization of historical documents.

Character recognition

Figure 9 illustrates the set of modern Tamil characters used for transfer learning. A total of 247 Tamil characters were selected for training purposes. *Figure 10* depicts the ancient Tamil characters targeted for recognition. The RCNN algorithm was trained on 247 distinct character classes, as shown in *Figure 10*. During the training and classification process, an Average Pooling 2D layer with a pool size of 2×2 was employed to compute the mean value of feature map regions. The activation output was then flattened into a vectorized feature representation, followed by two fully connected layers: one with 128 nodes for intermediate character classification and another with 246 nodes for final output classification. The predicted character classes were then aggregated to form the recognized output.

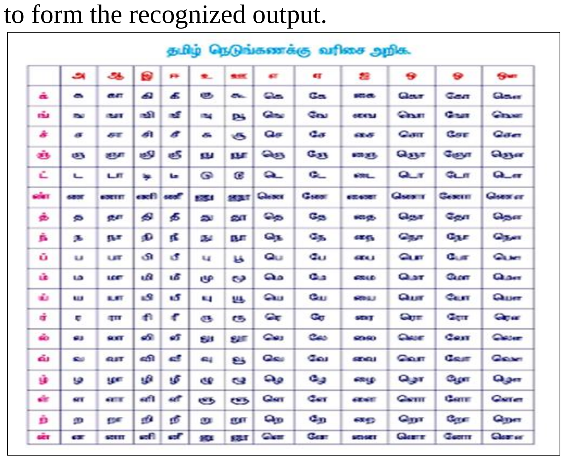


Figure 9 Modern Tamil characters for training

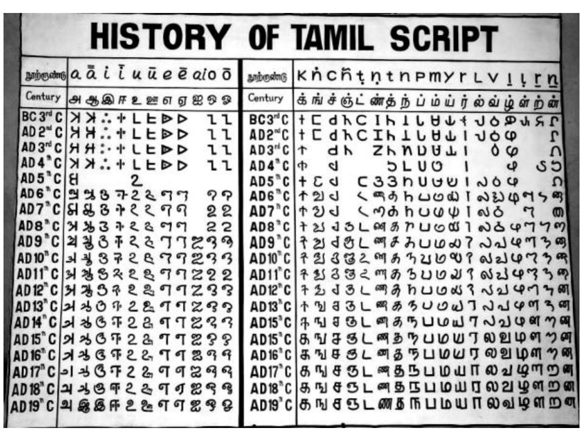


Figure 10 Ancient Tamil characters for recognition

Table 1 presents the key parameters used to assess the model's performance. Two training-to-testing ratios, 80:20 and 70:30, were tested using the dataset. The 70:30 ratio yielded the best recognition results and was therefore adopted for detailed performance evaluation. The model was trained using over 10,000 images and tested on 3,000 images. These datasets comprised a combination of characters extracted from stone inscriptions and palm leaf manuscripts.

Table 1 Parameters taken for performance calculation

Analysis parameters	Size
Size of the Batch	128
Training and Validation Ratio	80:20, 70:30
Epoch	5,15,25,35
Mutation Probability	0.15
Crossover Probability	0.75

Accuracy, precision, recall and F1-score has been considered for the comparison (Equations 9 to12). Accuracy measures the proportion of correctly identified characters out of the total number of samples.

$$\text{Accuracy}=\frac{TP+TN}{TP+TN+FP+FN} \quad (9)$$

Precision measures the proportion of true positive (TP) predictions among all positive predictions made by the mod-el:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (10)$$

Recall (or Sensitivity) measures the proportion of actual positives that were correctly identified:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (11)$$

F1-score is the harmonic mean of precision and recall, offering a balance between the two:

$$\text{F1-score}=2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \quad (12)$$

Where

True Positive (TP): The model correctly identifies a character as the correct class.

True Negative (TN): The model correctly identifies that a character does not belong to a particular class.

False positive (FP): The model incorrectly predicts a character as belonging to a class when it does not.

False Negative (FN): The model fails to identify a character that actually belongs to a class.

Table 2 illustrates the performance of the proposed RCNN model across different training epochs. As the number of epochs increases from 5 to 35, the accuracy improves from 97.2% to 98.6%, indicating better learning, though with a corresponding increase in execution time from 92 to 190 seconds. This reflects a trade-off between model accuracy and computational cost.

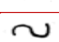
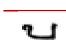
Table 2 Performance of proposed Models based on epochs

Number of epochs	Accuracy (%)	Execution time (Seconds)
5	97.2	92
15	97.6	154
25	98.2	176
35	98.6	190

The proposed RCNN model achieved the accuracy of 98.6%, precision of 96%, and F1-score of 97%, while also maintaining a comparatively lower execution time of 190 seconds, demonstrating both efficiency and effectiveness in ancient Tamil character recognition. *Table 3* presents the sample prediction results of the pro-posed RCNN model, demonstrating accurate recognition of ancient Tamil characters and their corresponding modern equivalents through consistent label mapping. This highlights the model's robust classification capability.

The integration of adaptive ROI pooling and deformable convolutional networks (DCN) further improves recognition by enhancing feature retention and spatial adaptability, especially for irregularly shaped characters. These enhancements contribute to increased accuracy, precision, recall, and F1-score, making the system highly effective for ancient Tamil script analysis.

Table 3 Performance evaluation of sample prediction results

Character expected	Character label	Modern character	Character label
	13	க	13
	24	ர	24
	25	ல	25
	19	த	19
	16	ஞ	16
	15	ச	15
	26	வ	26

The RCNN model demonstrates efficient performance with a training time of 200 seconds and a low inference time of 15 ms, making it suitable for real-time recognition tasks. It maintains moderate memory usage (450 MB) and 3 GB of RAM, with enhanced speed on high-end GPUs like the RTX 2080 Ti. The confusion matrix shows the RCNN model achieved strong performance with 10,257 TP and only 129 FN, indicating high recall. It also recorded just 3 FP and 14 TN, demonstrating excellent precision and overall classification accuracy (*Figure 11*). The ROC curve illustrates the performance of the RCNN model in distinguishing between TPs and FPs during ancient Tamil character recognition. The model achieved an AUC score of 0.88, indicating strong classification capability (*Figure 12*).

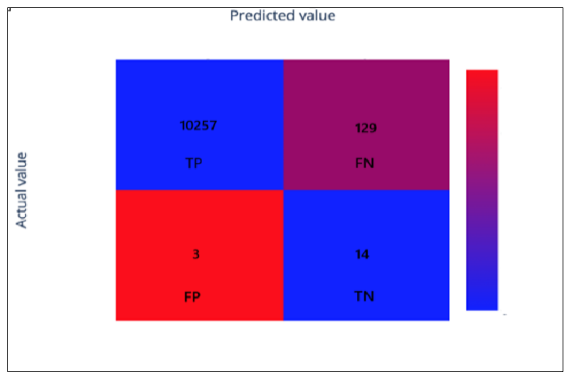


Figure 11 Confusion matrix of RCNN model for ancient Tamil character recognition

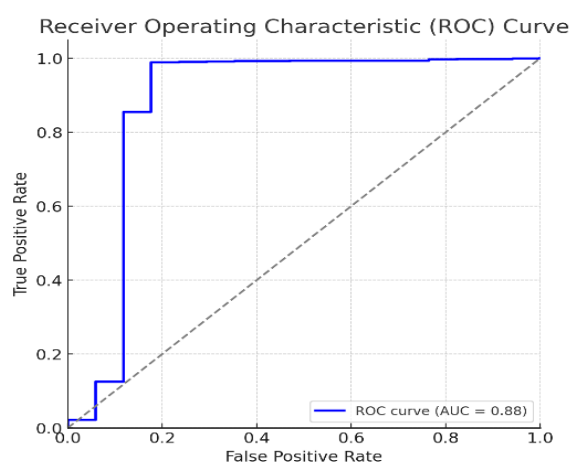
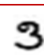


Figure 12 ROC curve for RCNN model performance

Limitations

Deciphering ancient Tamil text using image processing and DL presents several challenges, including the degradation and noise of source materials, variations in writing styles, and a scarcity of annotated data, all of which hinder accurate model training. The inherent complexity of ancient scripts, coupled with the risk of over-fitting due to limited and imbalanced datasets, affects the generalizability of models. Moreover, high computational demands and slow inference times increase resource consumption, while segmentation errors and intricate preprocessing steps introduce further technical hurdles. The ambiguity in interpreting ancient texts, the absence of reliable ground truth, and the need for contextual understanding to address polysemy and homo-graphs further complicate the recognition process. Addressing these limitations necessitates a multidisciplinary approach that integrates insights from computer science, history, and linguistics to enhance data quality, model robustness, and semantic interpretation. *Table 4* highlights the misclassifications that occurred during the char-acter recognition process. These characters are more challenging to identify accurately due to their visual similarities and variations in inscription styles. A complete list of abbreviations is listed in *Appendix I*.

Table 4 Challenging characters predictions in proposed method

S. No.	Actual character		Misclassified character	
1		ல		வ
2		ஞ		
3		க		த

5.Conclusion and future work

Ancient Tamil character and word recognition using the RCNN algorithm has shown promising results and holds significant potential for applications in historical linguistics, digital preservation of ancient texts, and cultural heritage conservation. The proposed method, leveraging RCNN, achieved accurate and efficient recognition of Tamil characters and words in ancient manuscripts and inscriptions, attaining an accuracy of 98.6%, precision of 96%, and an F1-score of 97%. With an execution time of 190 seconds, the RCNN approach enables rapid processing and recognition within large datasets. Its robustness in identifying intricate and complex characters in noisy and degraded images enhances the overall quality of recognition. The integration of Adaptive ROI Pooling further improves spatial adaptability, facilitating accurate recognition of irregularly shaped ancient characters. By digitizing and making historical texts accessible, this method contributes significantly to preserving linguistic and cultural knowledge for future generations. Despite these successes, challenges such as variation in writing styles, font types, and degradation persist. Future research could enhance the model's capability by incorporating advanced techniques like attention mechanisms and transformer architectures.

Acknowledgment

None.

Conflicts of interest

The authors have no conflicts of interest to declare.

Data availability

This study utilizes a combined dataset comprising the Stone Inscription Dataset and the Palm Leaves Dataset. The dataset is available at: https://github.com/Harsh120/Ancient-Tamil-Script-Recognition/tree/master/Input%20Images.

Author's contribution statement

A. Umamageswari: Conceptualization, Investigation, Implementation, Writing – original draft, Writing – review and editing. **S. Deepa**: Data collection, Conceptualization, Writing – original draft, Analysis and Interpretation of results. **L. Sherin Beevi**: Conceptualization, Writing – review and editing. **A. Sangari:** Data collection, Conceptualization, Writing – original draft, Analysis and Interpretation of results.

References

[1] Gautam N, Chai SS. Optical character recognition for Brahmi script using geometric method. Journal of Telecommunication, Electronic and Computer Engineering. 2017; 9(3-11):131-6.

[2] Bag S, Harit G. A survey on optical character recognition for Bangla and Devanagari scripts. Sadhana. 2013; 38:133-68.

[3] Kavitha BR, Srimathi CB. Benchmarking on offline handwritten Tamil character recognition using convolutional neural networks. Journal of King Saud University-Computer and Information Sciences. 2022; 34(4):1183-90.

[4] Ravi J. Handwritten alphabet classification in Tamil language using convolution neural network. International Journal of Cognitive Computing in Engineering. 2024; 5:132-9.

[5] Sabeenian RS, Paramasivam ME, Anand R, Dinesh PM. Palm-leaf manuscript character recognition and classification using convolutional neural networks. In computing and network sustainability: proceedings of IRSCNS 2018 (pp. 397-404). Springer Singapore.

[6] Suganya TS, Murugavalli S. Feature selection for an automated ancient Tamil script classification system using machine learning techniques. In international conference on algorithms, methodology, models and

applications in emerging technologies 2017 (pp. 1-6). IEEE.
[7] Dhivya S, Beulah JR. Ancient Tamil character recognition from stone inscriptions–a theoretical analysis. In 2nd Asian conference on innovation in technology 2022 (pp. 1-8). IEEE.
[8] Subadivya S, Vigneswari J, Yaminie M, Diviya M. Tamilbrahmi script character recognition system using deep learning technique. International Journal of Computer Science and Mobile Computing. 2020; 9(6):114-9.
[9] Bannigidad P, Sajjan SP. Ancient Kannada handwritten character recognition from palm leaf manuscripts using PyTesseract-OCR technique. In international conference on cognitive computing and information processing 2023 (pp. 154-62). Cham: Springer Nature Switzerland.
[10] Manigandan TV, Vidhya V, Dhanalakshmi V, Nirmala B. Tamil character recognition from ancient epigraphical inscription using OCR and NLP. In international conference on energy, communication, data analytics and soft computing 2017 (pp. 1008-11). IEEE.
[11] Rajakumar S, Bharathi VD. 7th century ancient Tamil character recognition from temple wall inscriptions. Indian Journal of Computer Science and Engineering. 2012; 3(5):673-77.
[12] Vidhyavani A, Manoranjitham T. A survey on recognition of ancient Tamil Brahmi characters from epigraphy. Journal of Pharmaceutical Negative Results. 2022; 13(3):1273-7.
[13] Vinoth R, Rajesh R, Yoganandhan P. Intelligence system for Tamil vattezhuttuoptical character recognition. International Journal of Computer Science Engineering and Technology. 2017; 8(4):22-6.
[14] Haritha J, Balamurugan VT, Vairavel KS, Ikram N, Janani M, Indrajith K. CNN based character recognition and classification in Tamil palm leaf manuscripts. In international conference on communication, computing and internet of things 2022 (pp. 1-6). IEEE.
[15] Subramani K, Subramaniam M. Creation of original Tamil character dataset through segregation of ancient palm leaf manuscripts in medicine. Expert Systems. 2021; 38(1):e12538.
[16] Islam MA, Iacob IE. Manuscripts character recognition using machine learning and deep learning. Modelling. 2023; 4(2):168-88.
[17] Sudarsan D, Sankar D. An overview of character recognition from palm leaf manuscripts. In 3rd international conference on smart data intelligence 2023 (pp. 265-72). IEEE.
[18] Srinivasan R, Babu AK, Balasubramanian PL, Sharma R. Ancient science and Tamil heritage: exploring the interdisciplinary connections for research and revival. In conference proceedings 2023. AIP Publishing LLC.
[19] Devi PD, Sathiyapriya V. Brahmi script recognition system using deep learning techniques. In third international conference on inventive research in computing applications 2021 (pp. 1346-9). IEEE.
[20] Durga DK, Uma MP. Recognition and classification of stone inscription character using artificial neural network. In artificial intelligence and technologies: select proceedings of ICRTAC-AIT 2020 (pp. 253-61). Singapore: Springer Singapore.
[21] Wijerathna KN, Sepalitha R, Indika T, Athauda H, Suranjini PD, Silva JA, et al. Recognition and translation of ancient Brahmi letters using deep learning and NLP. In international conference on advancements in computing 2019 (pp. 226-31). IEEE.
[22] Karthikeyan S, Jaganathan I, Rameshbabu KS, Abraham A, Gabralla LA, Sivaraj R, et al. Self-adaptive hybridized lion optimization algorithm with transfer learning for ancient Tamil character recognition in stone inscriptions. IEEE Access. 2023; 11:39621-34.
[23] Magrina MM. Convolution neural network based ancient Tamil character recognition from epigraphical inscriptions. International Research Journal of Engineering and Technology. 2020; 7(04):6130-43.
[24] Vijayalakshmi R, Gnanasekar JM. A review on character recognition and information retrieval from ancient inscriptions. In 8th international conference on smart structures and systems 2022 (pp.1-7). IEEE.
[25] Mahajan H, Narayan P, Reddi SI, Sharmila BK. A systematic approach to handle ancient Tamil script using deep learning techniques. In soft computing and signal processing: proceedings of 3rd ICSCSP 2020 (pp. 57-68). Springer Singapore.
[26] Babu BS, Shetty S, Agarwal A, Sreerama SL, Bhustali VK, Sanjana S. Temple inscriptions recognition and transliteration in Devanagari script. In 2nd international conference on vision towards emerging trends in communication and networking technologies 2023 (pp. 1-6). IEEE.
[27] Rajakumar S, Bharathi VS. Eighth century Tamil consonants recognition from stone inscriptions. In international conference on recent trends in information technology 2012 (pp. 40-3). IEEE.
[28] Meng L, Aravinda CV, Uday KR, Izumi T, Yamazaki K. Ancient Asian character recognition for literature preservation and understanding. In Euro-mediterranean conference 2018 (pp. 741-51). Cham: Springer International Publishing.
[29] Selvapriya B, Rajini V, Sowmya B. Preserving the legacy of ancient Tamil script with deep learning and fuzzy C means algorithm: intelligent approach to digiitization. Journal of Electrical Engineering & Technology. 2025; 20(1):757-71.
[30] Umamageswari A, Deepa S, Raja K. An enhanced approach for leaf disease identification and classification using deep learning techniques. Measurement: Sensors. 2022; 24:1-6.
[31] Arasakumaran U, Johnson SD, Sara D, Kothandaraman R. An enhanced identification and classification algorithm for plant leaf diseases based on deep learning. Traitement Du Signal. 2022; 39(3):1013-18.

[32] Mahajan H, Narayan P, Reddi SI, Sharmila Banu K. A systematic approach to handle ancient Tamil script using deep learning techniques. In soft computing and signal processing: proceedings of 3rd ICSCSP. 2021 (pp. 57-68). Springer Singapore.
[33] Bhuvaneswari S, Kathiravan K. Enhancing epigraphy: a deep learning approach to recognize and analyze Tamil ancient inscriptions. Neural Computing and Applications. 2024; 36(31):19839-61.
[34] Hamdany AH, Omar-Nima RR, Albak LH. Translating cuneiform symbols using artificial neural network. TELKOMNIKA (Telecommunication Computing Electronics and Control). 2021; 19(2):438-43.
[35] Dutta S, Manideep BC, Rai S, Vijayarajan V. A comparative study of deep learning models for medical image classification. In conference series: materials science and engineering 2017 (pp. 1-9). IOP Publishing.
[36] Brown A. Practical digital preservation: a how-to guide for organizations of any size. Facet Publishing; 2013.
[37] Richarz J, Vajda S, Grzeszick R, Fink GA. Semi-supervised learning for character recognition in historical archive documents. Pattern Recognition. 2014; 47(3):1011-20.
[38] Zhu W, Wang X, Li H. Multi-modal deep analysis for multimedia. IEEE Transactions on Circuits and Systems for Video Technology. 2019; 30(10):3740-64.
[39] Nogales A, Delgado-martos E, Melchor Á, García-tejedor ÁJ. ARQGAN: an evaluation of generative adversarial network approaches for automatic virtual inpainting restoration of Greek temples. Expert Systems with Applications. 2021; 180:115092.
[40] Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies 2016 (pp. 1480-9).
[41] Barucci A, Cucci C, Franci M, Loschiavo M, Argenti F. A deep learning approach to ancient Egyptian hieroglyphs classification. IEEE Access. 2021; 9:123438-47.
[42] Khayyat MM, Elrefaei LA. Deep reinforcement learning approach for manuscripts image classification and retrieval. Multimedia Tools and Applications. 2022; 81(11):15395-417.
[43] Gautam N, Dhawas P, Ramteke MA, Madankar M. Application of generative adversarial networks in Brahmi script recognition. In crafting images with generative adversarial networks (GANs) and models 2025 (pp. 221-66). IGI Global Scientific Publishing.
[44] Bi YQ, Jin JQ, Luo Y, Wang ZY. A robust text recognition system for noisy document images. In proceedings of the international conference on frontiers of artificial intelligence and machine learning 2023 (pp. 19-24).
[45] Xint Y, Xu L, Guo J, Li J, Sheng X, Zhou Y. Label incorporated graph neural networks for text classification. In 25th international conference on pattern recognition 2021 (pp. 8892-8). IEEE.
[46] Harish R, Raghavendra Rao GN. Transcription of ancient indian manuscripts through artificial intelligence—current status of technology and the way forward. In international conference on artificial intelligence on textile and apparel 2023 (pp. 339-52). Singapore: Springer Nature Singapore.

Dr. A. Umamageswari is an Associate Professor and the Deputy Head of the Department of Big Data Analytics at SRM Institute of Science and Technology, Ramapuram Campus. She has over 20 years of academic experience and holds a Ph.D. in Computer Science and Engineering with a specialization in Medical Image Processing. She earned her M.Tech from Sathyabama University, Chennai, and her B.E. in Computer Science and Engineering from Sri Venkateswara College of Engineering, Sriperumbudur, affiliated with the University of Madras. Dr. Umamageswari is a recognized Research Supervisor at SRM Institute of Science and Technology. She serves as a member of several Boards of Studies and professional organizations. Among her accolades, she has been honored with the Tamilnadu Chief Minister Medal. She is also an active member of prestigious associations such as IEEE, ACM, ISTE, CSTA, and IAENG.
Email: r.umaramesh@srmist.edu.in

Dr. S. Deepa is an Assistant Professor in the Department of Computer Science and Engineering at SRM Institute of Science and Technology, Chennai. She holds a B.E. in Computer Science and Engineering from Sri Venkateswara College of Engineering and Technology, affiliated with Madras University, an M.E. in Computer Science and Engineering from Jaya Engineering College, affiliated with Anna University, and a Ph.D. from the College of Engineering, Anna University. She is an active member of the Indian Society for Technical Education (ISTE). Her research interests include Image Processing and Pattern Recognition.
Email: deepas1@srmist.edu.in

L. Sherin Beevi, is an Assistant Professor currently serving in the Department of Computer Science and Engineering at R.M.D. Engineering College, Kavaraipettai. She completed her B.Tech in Information Technology in 2007 from St. Xavier's Catholic College of Engineering, Nagercoil, and her M.Tech in Computer Science and Engineering in 2009 from Francis Xavier College of Engineering, Tirunelveli,

affiliated with Anna University, Chennai. She is presently pursuing her Ph.D. in the field of Blockchain with Quantum Computing at Anna University, MIT Campus, Chennai. With over 13.5 years of teaching experience, she has taught both undergraduate and postgraduate programs. Her areas of interest include Computer Networks, Blockchain, Quantum Computing, Data Science, and Deep Learning.
Email: lsb.cse@rmd.ac.in

Dr. A. Sangari is an Associate Professor in the Department of Electrical and Electronics Engineering at Rajalakshmi Engineering College, Chennai. She completed her undergraduate studies at Sri Venkateswara College of Engineering, Sriperumbudur, her postgraduate degree at Rajalakshmi Engineering College, and earned her Ph.D. in Power Electronics from Anna University, Chennai. Her research interests include Multilevel Inverters, Z-Source Inverters, Renewable Energy Systems, and the application of Artificial Intelligence in Power Electronics.
Email: sangari.a@rajalakshmi.edu.in

Appendix I

S. No.	Abbreviation	Description
1	AI	Artificial Intelligence
2	AR	Augmented Reality
3	CNN	Convolutional Neural Networks
4	CDF	Cumulative Distribution Function
5	DL	Deep Learning
6	FN	False Negative
7	FP	False Positive
8	NLG	Natural language Generation
9	NMS	Non-Maximum Suppression
10	NMT	Neural Machine Translation
11	OCR	Optical Character Recognition
12	RAM	Random Access Memory
13	RCNN	Region-Based Convolutional Neural Network
14	RNN	Recurrent Neural Network
15	RoI	Region of Interest
16	SVM	Support Vector Machine
17	TN	True Negative
18	TP	True Positive
19	VR	Virtual Reality
20	2D	Two-Dimensional