International Research Journal of Multidisciplinary Scope (IRJMS), 2026; 7(1): 1627-1640 IRJMS

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2026.v07i01.08737

Global and Local Feature Extraction Using Deep Learning Technique for Keyword Spotting in Ancient Tamil Inscriptions

Vidhyavani A*, Manoranjitham T

Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Chennai – 603203, Tamil Nadu, India. *Corresponding Author's Email: va3647@srmist.edu.in

Abstract

Ancient Tamil scripts preserved in stone inscriptions represent valuable cultural, historical, and social heritage. However, extracting meaningful information from these inscriptions remains a challenging task due to severe degradation, erosion, noise, and complex character structures. Word spotting (WS) has emerged as an effective alternative to traditional optical character recognition (OCR) systems for information retrieval from such degraded documents. This paper proposes a hybrid deep learning framework for word spotting in ancient Tamil inscription images. The proposed system consists of pre-processing, feature extraction, script identification, and word spotting stages. To capture comprehensive information from inscription images, both local and global features are extracted. Local visual features, such as stroke patterns and character shapes, are extracted using a self-attention convolutional neural network (SA-CNN), while global contextual features are learned using a stacked long short-term memory and bidirectional gated recurrent unit (S-LSTM–Bi-GRU) model to capture sequential dependencies. The extracted local and global features are fused through a concatenation layer to form a unified feature representation, which is then classified using a softmax layer for accurate word spotting. The effectiveness of the proposed approach is evaluated using a real-time dataset of ancient Tamil inscriptions. Experimental results demonstrate that the proposed framework achieves an accuracy of 96.6% and a precision of 98.1%, outperforming several existing word spotting methods. The results indicate that integrating local and global feature representations significantly enhances the robustness and reliability of word spotting in degraded inscription images, making the proposed approach suitable for real-world epigraphically analysis.

Keywords: Ancient Tamil Scripts, Global Features, Local Features, Self-Attention Convolutional Neural Network, Word Spotting.

Introduction

Tamil is a classical language that is written and spoken primarily by the people of Tamil Nadu using the Tamil script. The Tamil alphabet consists of 12 vowels, one Āytha Ezhuthu, 18 consonants, and 216 consonant–vowel combinations, resulting in a total of 247 characters. Although several research studies have been conducted on Tamil Character Recognition (TCR), limited work has been reported on Tamil keyword spotting (1). Temple inscriptions serve as invaluable cultural artifacts that provide rich insights into early civilizations, including their belief systems, rituals, administrative practices, and everyday life. Commonly engraved on temple walls and other sacred structures, these inscriptions reflect the artistic, religious, and linguistic heritage of societies that existed in the past. The study and interpretation of such inscriptions, known as epigraphy, enable researchers to reconstruct ancient history and understand cultural evolution. Ancient historical scripts contain crucial information for historians, archaeologists, and researchers; however, extracting this information remains a significant challenge due to script degradation and aging effects (2). With recent advancements in Information and Communication Technology (ICT), scripts and documents can be converted into digital formats, facilitating storage, preservation, and transmission. As a result, large repositories of public and private document images have been created. Nevertheless, digitization alone is insufficient for effective information retrieval from document images (3–5). In many cases, complete restoration and recognition of ancient scripts are difficult due to inadequate preservation, variations in writing

(Received 19th October 2025; Accepted 13th January 2026; Published 31st January 2026)

styles, and archaic language forms. Furthermore, Optical Character Recognition (OCR) systems face limitations when applied to historical inscriptions. While OCR achieves high accuracy for modern printed text, its performance significantly degrades for ancient Tamil inscriptions. Major challenges in Tamil OCR include character segmentation and the recognition of complex character structures (6–9).

To overcome these limitations, word spotting (WS) has emerged as a promising alternative, particularly for historical documents. The objective of WS is to retrieve word images from a collection of documents that match a given query. Unlike text recognition methods that require complete document transcription, WS focuses on detecting the presence of specific words within the document images (10–12). WS can be categorized into two types: query-by-string (QbS), where a textual string is used as input, and query-by-example (QbE), where an example word image is provided as the query (13). By avoiding full character segmentation, WS is more suitable for degraded and irregular ancient inscriptions than OCR-based approaches.

Many existing WS methods assume prior knowledge of the script and are unable to effectively handle mixed or misidentified scripts (14). To address arbitrary and multilingual scripts, unified recognition frameworks have been proposed that eliminate the need for script-specific processing. These approaches offer greater generalization and flexibility across multilingual datasets. Nevertheless, script identification (SI) is often treated as a pre-processing step to determine the script type before further analysis (15).

Recently, SI has been applied in various domains, including video analysis, natural scene text recognition, document analysis, and document retrieval. A typical WS framework involves pre-processing, segmentation, feature extraction, and classification stages. Traditional machine learning-based methods rely on handcrafted features such as invariant moments (IM), Histogram of Oriented Gradients (HOG), and Scale-Invariant Feature Transform (SIFT) (16). In contrast, deep neural networks (DNNs), particularly Convolutional Neural Networks (CNNs), have demonstrated superior performance by automatically learning discriminative features and determining whether a word segment is present in a script (17).

In this study, the terms keyword spotting and word spotting (WS) identification are used interchangeably, whereas inscription identification refers to the broader process of analyzing and interpreting inscription content.

Motivation

The feature extraction mechanism is a core component of word spotting (WS) systems, as it directly influences recognition accuracy. Conventional WS models rely on handcrafted descriptors to extract discriminative features; however, their performance is limited when applied to degraded historical inscriptions. Recent advances in deep learning (DL) have led to the adoption of models such as Convolutional Neural Networks (CNNs), Residual Networks (ResNet), and Recurrent Neural Networks (RNNs) for WS tasks. Although CNN and ResNet architectures are effective in learning spatial representations, they have limited capability in capturing long-term contextual dependencies. Conversely, RNN-based models process sequential information directly but often fail to effectively capture local spatial features. Moreover, CNNs tend to extract only localized feature maps, which may be insufficient for modelling global word-level information.

Bidirectional Gated Recurrent Units (Bi-GRU) mitigate gradient vanishing issues, enhance feature propagation, and reduce the number of training parameters. Stacked Long Short-Term Memory (LSTM) networks are particularly effective in learning sequential and temporal dependencies. Motivated by these observations, this work employs a hybrid deep learning framework that integrates a self-attention-based CNN for local feature extraction with a stacked LSTM-based Bi-GRU architecture for global feature representation.

Contributions of the Proposed Work

The major contributions of this work are summarized as follows:

a) An automated word spotting (WS) framework is proposed for ancient Tamil temple inscriptions.
b) Multiple pre-processing techniques are employed to standardize inscription images and enhance detection performance.
c) Local and global features are effectively extracted using a self-attention convolutional

neural network (SA-CNN) and a stacked LSTM-based Bi-GRU (S-LSTM–Bi-GRU) architecture.

Several studies have explored deep learning-based WS approaches. A multi-stage WS model combining CNNs for local feature extraction and bidirectional LSTMs for global feature learning was proposed for handwritten and printed documents, achieving a mean average precision of 89% through compact bilinear pooling (5). A transfer learning-based CNN model for Tamil character recognition using stroke width transform segmentation achieved F-score and recall values of 95.1% and 95%, respectively (10). PU-Net-based architectures utilizing transfer learning and PHOC encoders have been proposed for handwritten document WS without pre-processing or data augmentation (18). An end-to-end Mandarin WS system based on convolutional recurrent neural networks with a connectionist temporal classification (CTC) loss achieved false rejection rates of 5.22% and 6.35% for 13 and 20 keywords, respectively, on the AISHELL-2 dataset (19). An auto encoder-based WS method applied to handwritten Gujarati documents achieved recall and precision values of 50% and 67.95%, respectively (20).

Monte Carlo dropout-based CNN models have been introduced to estimate feature uncertainty in both query-by-example and query-by-string WS tasks, with cosine similarity used for matching across multiple benchmark datasets (21). Multi-script WS frameworks combining bidirectional LSTM and hidden Markov models have also been explored for printed and handwritten documents (22). Cross-language WS approaches using character mapping and script similarity measures have demonstrated improved recognition accuracy across diverse scripts (23). Zone-based WS methods employing pyramid histogram of oriented gradients achieved mean average precision values of 72.6% and 73.1% for Devanagari and Bangla documents, respectively (24).

Joint embedding approaches integrating CNNs and RNNs have been proposed for word recognition and WS across multiple datasets and languages (25). Segmentation-free WS methods using wave kernel signatures for ancient Bangla documents achieved precision and recall values of 96.5% and 98.8%, respectively (26). Offline handwritten WS frameworks incorporating generative adversarial networks for data generation, histogram of oriented gradients for feature extraction, and LSTM-based classification achieved recognition rates of 98.6% on benchmark datasets (27).

Despite these advancements, most existing WS methods are designed for specific scripts, limiting their generalization capability. Additionally, several approaches rely on manual intervention, resulting in increased computational complexity and reduced scalability for large datasets. To overcome these limitations, the proposed work focuses on accurate and scalable keyword spotting in ancient Tamil temple inscriptions using a robust hybrid deep learning framework that effectively integrates both local and global contextual information.

Methodology

This research paper introduces an innovative algorithmic approach for WS with a specific emphasis on the Tamil ancient script. The proposed approach integrates sophisticated pre-processing algorithms, and DL models to tackle the challenges presented by noisy inscriptions, script changes, and the understanding of historical context. Figure 1 presents the proposed WS methodology. After the pre-processing stage, the local and global feature extraction processes are carried out. Here, the SA-CNN is used for extracting the local features and the S-LSTM- Bi-GRU is used for extracting the global features.

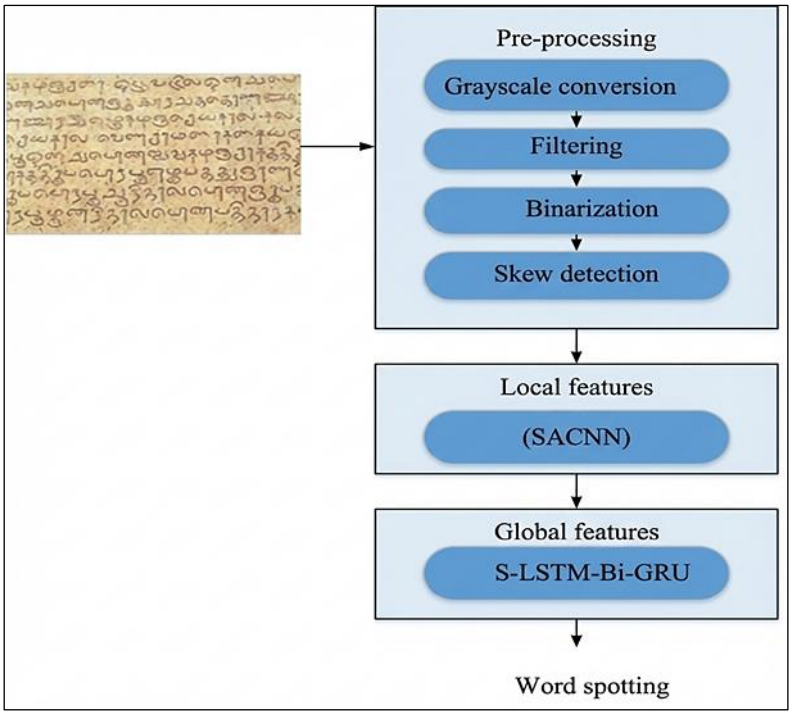

Figure 1: Block Diagram of the Proposed SA-CNN and S-LSTM–Bi-GRU based Keyword Spotting Framework for Ancient Tamil Inscriptions

Pre-processing

The pre-processing stage is employed to minimize differences in inscriptions. In this work, different fundamental preprocessing stages are incorporated. They are greyscale conversion, filtering, binarization and skew detection. Initially, the input RGB image is converted into grayscale conversion, and then noise is removed using the wiener filtering. Then binarization is used to convert the grey image into black and white image by the thresholding. Finally, skew detection processes are carried out for checking an angle of orientation.

Feature Extraction

After the pre-processing stage, the utilization of distinct features with varying abstraction stages for every process. In the task of script identification, both global (high level) and local (low level) features contribute to the identification. In this work, the local features are extracted using the SACNN and the global features are extracted using the S-LSTM- Bi-GRU. Then, these two features are combined for the word recognition process.

Global Feature Extraction

The LSTM captures long-term sequential dependencies, while the Bi-GRU efficiently models bidirectional contextual information. Their combination enhances global feature learning.

Local Feature Extraction

The **local features** captured by the SA-CNN, including stroke patterns, edge orientations, character contours, and spatial texture details relevant to ancient Tamil inscriptions.

Feature Fusion

Feature fusion combines local visual features and global contextual features into a unified representation, enabling more accurate and robust word spotting from degraded ancient Tamil inscription images.

SACNN: The DL model SACNN has layers like convolution layer, pooling, self-attention and FC (fully connected) layer.

Convolution layer: In a CNN, the convolution layer is the major component, it has a collection of separate filters, and the filters are individually convolved with the image to generate feature maps. If the image with size of $m \times n$ having the filter size of $a \times b$ (width x height) then the resultant feature map $O_a \times O_b$ is given as in Equations [1] and [2]:

$$O_a = \frac{m - a + 2p_a}{s_a} + 1 \quad [1]$$

$$O_b = \frac{m - b + 2p_b}{s_b} + 1 \quad [2]$$

where p_a and p_b depicts the zero-padding and s_a and s_b are the stride in the a and b dimensions.

The generation of each feature map's output involves convolving the input maps I_k with a linear filter, incorporating a bias term b_k^l, and then using a nonlinear function. It is given as in Equation [3]:

$$Z_k^l = f\left(\sum_{j \in I_k} Z_k^{l-1} * W_{jk}^l + b_k^l\right) \quad [3]$$

where l is the total layers and W_{jk} is the convolutional kernel. $f()$ is the activation function and plays a crucial role in a CNN, enabling it to learn and execute more complex tasks. In this work, ReLU activation function is considered and it is defined by using Equation [4] and [5]:

$$f(Z) = \max(0, Z) \quad [4]$$

$$f(Z) = \begin{cases} Z, & Z \geq 0 \\ 0, & Z < 0 \end{cases} \quad [5]$$

Pooling layers: In a CNN, a pooling layer is introduced between consecutive convolutional layers. The purpose of these layers is to progressively decrease the spatial size. This reduction in size helps control overfitting by reducing the number of parameters and complexity in the network. Additionally, these layers contribute to making the CNN translation invariant. Operating independently on each input layer, the pooling layer spatially resizes the input by the pooling operation.

Self-attention layer: The SA layer can acquire the capability to emphasize the most significant features in the images. Further, this layer understands the attention weights of every input's part (Equation [6] and [7]).

$$a_k = \frac{\exp\left(v_t^T v\right)}{\sum_{t=1}^{n=1} \exp\left(v_t^T v\right)} \quad [6]$$

$$v_t = \tanh(U h_t + b) \quad [7]$$

where v and U are the learning variables and a_k is the attention weight.

S-LSTM- Bi-GRU: The DL model S-LSTM- Bi-GRU has layers like S-LSTM and Bi-GRU which extracts the local features. Figure 2 states the structure of the DL model S-LSTM- Bi-GRU.

LSTM: In LSTM, the past information is retained by the neuronal state updation through the input gate i_p, output gate o_p, and forget gate f_p, regulating the weight of historical information. The incorporation of three thresholds in LSTM effectively addresses the issue of gradient vanishing. This mechanism filters out relatively unimportant information and, consequently, reduces the training time.

The f_p is responsible for deciding how many units G_{p-1} from the prior time should be preserved for the present time G_p and it is computed by using Equation [8]:

$$G_p = G_{p-1} + f_p + i_p \times u_p \quad [8]$$

The input of the f_p incorporates the input data Z_p and output Y_{p-1}. At last, f_p is determined through the σ (activation function sigmoid). The calculation formula is expressed in Equation [9].

$$f_p = \sigma\left(W_f\left[Y_{p-1}, Z_p\right] + b_f\right) \quad [9]$$

The i_p is responsible to define which input in Y_t can be recorded in the neurons and it is split into i_p and u_p and is given by using Equation [10] and [11].

$$i_p = \sigma\left(W_i\left[Y_{p-1}, Z_p\right] + b_i\right) \quad [10]$$

$$u_p = \tanh\left(W_u\left[Y_{p-1}, Z_p\right] + b_u\right) \quad [11]$$

The o_p regulates the neuron's output state and facilitates the transfer of the next neuron state and it is computed by using Equation [12]:

$$o_p = \sigma\left(W_o\left[Y_{p-1}, Z_p\right] + b_o\right) \quad [12]$$

where W_f, W_i, W_o and W_u are the weighting matrices; b_f, b_i, b_o and b_u are the bias values. The neuron's output Y_p is given by using Equation [13]:

$$Y_p = o_p \times \tanh(G_p) \quad [13]$$

Bi-GRU: In contrast to LSTM, Bi-GRU lacks a memory cell and is equipped with only two gates like reset gate r^k and update gate z^k. The activation of Bi-GRU h_t^k at time t is the linear way of candidate's activation $\hat{h}_t^k$ and prior activation h_{t-1}^k. The h_t^k of the k^{th} Bi-GRU is computed by using Equation [14]:

$$h_t^k = (1 - z_t^k)h_{t-1}^k + z_t^k \times \hat{h}_t^k \quad [14]$$

For computing the z^k at time t, this work consider the prior hidden phase h_{t-1}^k and the present input y_t is given by using Equation [15]:

$$z_t = \sigma\left(W_z y_t + V_z h_{t-1}\right) \quad [15]$$

The candidate's activation $\hat{h}_t^k$ is computed by using Equation [16]:

$$\hat{z}_t = \tanh\left(W y_t + r^k V h_{t-1}\right) \quad [16]$$

The r^k is utilized for determining the total information for forgetting from the past and it is computed by using Equation [17]:

$$r_t = \sigma\left(W_r y_t + V_r h_{t-1}\right) \quad [17]$$

Bi-GRU takes into account the reverse sequence through the integration of both forward and reverse GRUs. This involves employing two distinct hidden layers $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ to concurrently process data from both the forward and reverse directions. It is computed by using Equation [18]:

$$h_t = \left[\overrightarrow{h}_t, \overleftarrow{h}_t\right] \quad [18]$$

For enhancing the classification performance, this work presents a S-LSTM- Bi-GRU which reduces the training time. Initially, S-LSTM layers are employed due to having more parameters and

achieving higher classification performance. The utilization of a S-LSTM typically results in better local feature extraction when compared to a single layered LSTM. Then, the S-LSTM model is integrated with the Bi-GRU that reduces the computational time and training time.

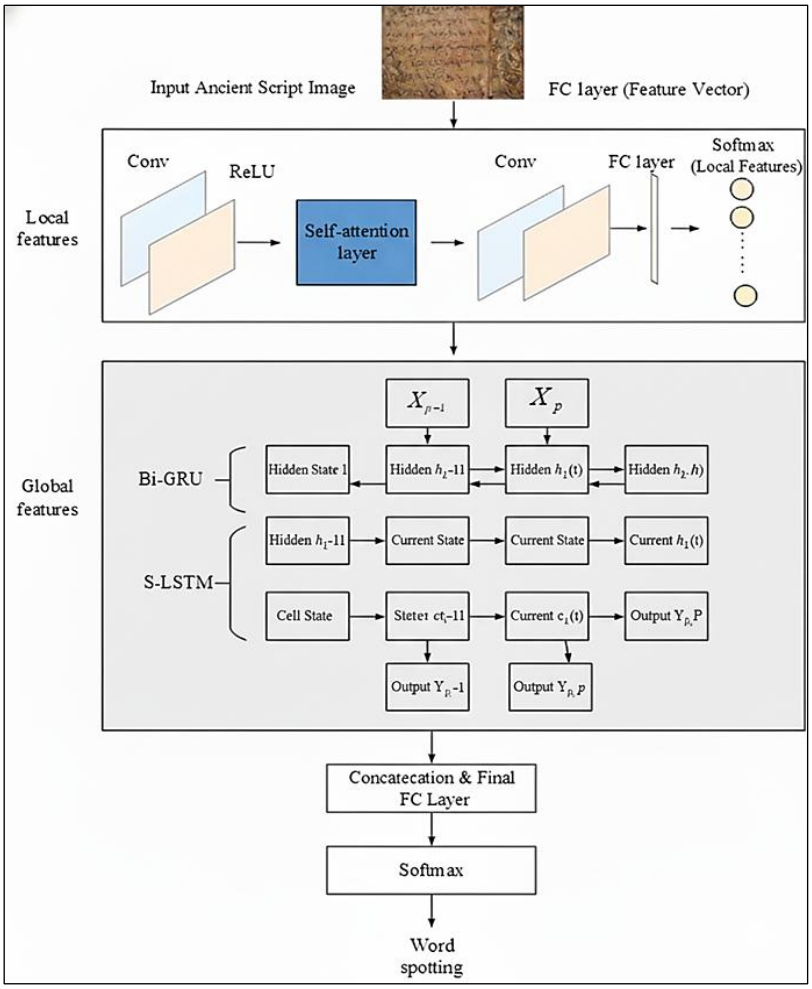

Figure 2: Detailed Architecture of the Proposed Model Showing Local Feature Extraction using SA-CNN and Global Feature Learning using S-LSTM–Bi-GRU

Results

The present study is executed on a system equipped with an Intel Core i7 processor operating at a clock speed of 2.20 GHz and 8 GB RAM. The approaches are evaluated using MATLAB 2020, incorporating the Image Processing Toolbox. Table 1 presents the parameters utilized for the demonstration process.

Table 1: Parameters for the Demonstration Process

Parameters	Values
Size of batch	128
Learning rate	0.001
Epochs	300
Optimizer	Adam
Loss function	Cross entropy
Activation	ReLU

Dataset Detail

For compiling the dataset, we sought sources of data containing Tamil ancient characters images engraved in temples. This data included archives from Tamil temples, historical databases focusing on inscriptions of the South, research organizations relevant to Tamil Nadu and its neighbouring places. Figure 3 delineates the sample images of the collected dataset.

Figure 3: Sample Images of the Collected Dataset

Performance Measures

To assess the effectiveness of the suggested system, multiple evaluation metrics have been taken into account, including, accuracy, F1-score, precision, recall, specificity and AUC (area under the curve). The following four crucial terms are considered for evaluation. G_p, L_p and $G_n L_n$ are the true, false positives and true, false negatives.

Accuracy: It the ratio of correctly predicted instances to the total number of instances and it is given by using Equation [19]:

$$A = \frac{G_p + L_n}{G_p + L_n + G_n + L_p} \quad [19]$$

Precision: It is the ratio of correctly identified positives. It is calculated as the ratio of accurate positive results to the total number of predicted positive results, as given in Equation [20].

$$\mathrm{Pre} = \frac{G_p}{G_p + L_p} \quad [20]$$

Recall: It is the ratio of actual positive values that are determined accurately. It is calculated as the ratio of accurate positive results to the total number of instances determined as positive, as given in Equation [21].

$$A = \frac{G_p}{G_p + L_n} \quad [21]$$

F1-score: This measure combines $\mathrm{Pr}e$ and Re in relation to a specific positive class. It is calculated as the weighted mean of Pre and Re, as demonstrated in Equation [22]:

$$\mathrm{F1-score} = 2 \times \frac{\mathrm{Pre} \times \mathrm{Re}}{\mathrm{Pre} + \mathrm{Re}} \quad [22]$$

Specificity: It refers to the ratio of instances with a true negative outcome that are accurately classified as negative, as demonstrated in Equation [23]:

$$\mathrm{Sp} = \frac{G_n}{G_n + L_p} \quad [23]$$

The calculation of AUC (area under the curve) involves utilizing the ROC (Receiver Operating Characteristic) curve, using the TPR (True Positive Rate) and FPR (False Positive Rate).

TPR: It is the proportion of positive points of data that are accurately identified as positive, relative to the total number of positive points of data, as demonstrated in Equation [24]:

$$\mathrm{TPR} = \frac{G_p}{G_n + L_p} \quad [24]$$

FPR: It is the proportion of negative points of data that are wrongly identified as positive, relative to the total number of negative points of data, as demonstrated in Equation [25]:

$$FPR = \frac{L_p}{G_n + L_p} \quad [25]$$

assess quantitative measures of various DL models.

Qualitative Analysis

In this section, we conduct a qualitative analysis of Tamil inscription identification. Following that, we

Figure 4: Analysis of (A) Input image, (B) Greyscale, (C) Filtered, (D) Binary image, (E) WS and (F) Meaningful word

Figure 4 depicts the analysis of input images, Greyscale, Filtered, Binary image, WS and Meaningful word images. The findings provided in this Figure demonstrate the model's proficiency in recognizing and comprehending particular historical characters and words in the inscriptions.

Qualitative Analysis

In this section, we conduct a quantitative analysis of Tamil inscription identification of various DL models. The performances are carried out by varying the training percentages to 50, 60, 70 and 80. The comparison of proposed WS is compared over the approaches like SA-CNN, LSTM, Bi-LSTM, and Bi-GRU.

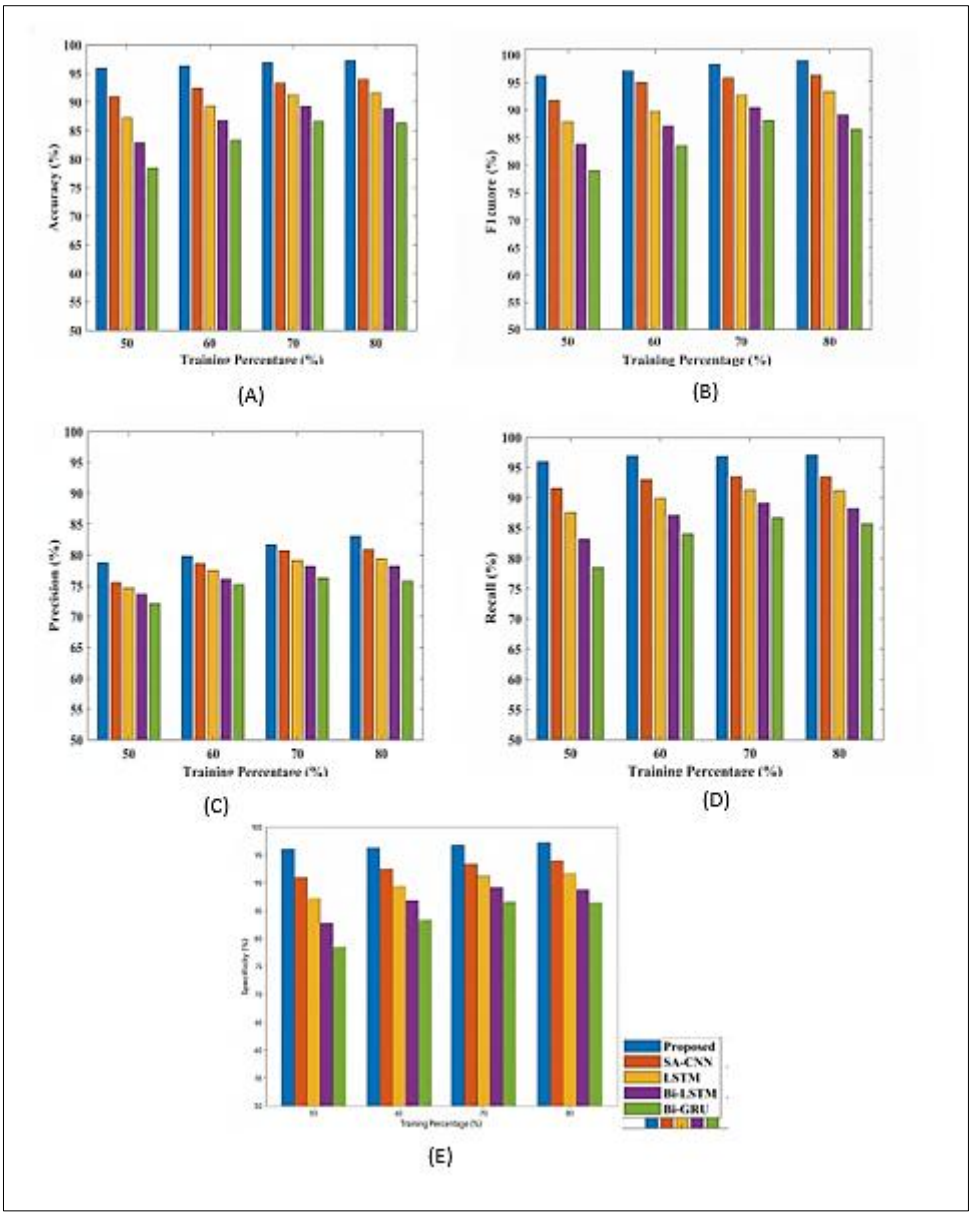

Figure 5: Comparative Analysis of (A) Accuracy, (B) F1-Score, (C) Precision, (D) Recall and (E) Specificity

Table 2: Comparative Analysis

Accuracy					
Training%	SA-CNN	LSTM	Bi-LSTM	Bi-GRU	Proposed
50	90.93	87.18	82.81	78.43	95.93
60	92.42	89.29	86.68	83.28	96.344
70	93.28	91.27	89.26	86.57	96.86
80	93.93	91.58	88.84	86.30	97.26
F1-score					
Training%	SA-CNN	LSTM	Bi-LSTM	Bi-GRU	Proposed
50	93.13	90.11	86.63	83.02	96.86
60	94.28	91.85	89.75	87.14	97.32
70	94.84	93.23	91.60	89.56	97.57
80	95.19	93.37	91.16	89.16	97.83
Precision					
Training%	SA-CNN	LSTM	Bi-LSTM	Bi-GRU	Proposed
50	88.16	94.74	92.83	90.41	97.79
60	90.50	95.61	93.88	92.55	97.74
70	92.60	96.25	95.24	94.22	98.30
80	92.90	96.96	95.67	94.30	98.63
Recall					
Training%	SA-CNN	LSTM	Bi-LSTM	Bi-GRU	Proposed
50	78.45	91.58	87.54	83.16	95.95
60	84.03	92.99	89.91	87.11	96.91
70	86.71	93.47	91.30	89.13	96.85

80	85.71	93.48	91.17	88.23	97.05
			Specificity		
Training%	SA-CNN	LSTM	Bi-LSTM	Bi-GRU	Proposed
50	78.35	87.09	85.07	80.71	95.80
60	78.55	88.80	85.34	83.94	92.69
70	85.78	92.19	91.10	90.01	96.91
80	90	96.74	94.15	92.68	98.52

Figure 5 and Table 2 present the comparative analysis of various approaches by varying the training percentages. Figures 5 (A) to (E) illustrate the performance of accuracy, F1-score, precision, recall, and specificity for various WS approaches, comparing methods like SA-CNN, LSTM, Bi-LSTM, and Bi-GRU against the proposed WS model. In Figure 5 (A), accuracy values for SA-CNN, LSTM, Bi-LSTM, Bi-GRU, and the proposed WS are 93.93%, 91.58%, 88.84%, 86.30% and 97.26%, respectively when the training percentage is 80. Likewise, Figure 5 (B) shows that the F1-score value achieved by the proposed WS is 97.57%, when the training percentage is 70, surpassing existing approaches SA-CNN, LSTM, Bi-LSTM, and Bi-GRU by 3.88%, 4.89%, 5.1% and 8%, respectively. Then, in Figure 5 (C) and (D) the precision and recall values achieved by the proposed WS are 97.79% and 95.95% when the training percentage is 50. Finally, in Figure 5 (E), the specificity value achieved by the proposed segmentation is 98.52%, outperforming existing approaches like SA-CNN, LSTM, Bi-LSTM, and Bi-GRU by 9.6%, 6.7%, 7.12%, and 8.6% respectively. In all comparisons, the proposed WS demonstrates superior performance. The hybridization of opposition SA-CNN with S-LSTM- Bi-GRU to the improved outcomes of the proposed WS. Conversely, existing approaches exhibit lower performance due to their high complexity, and have overfitting issues.

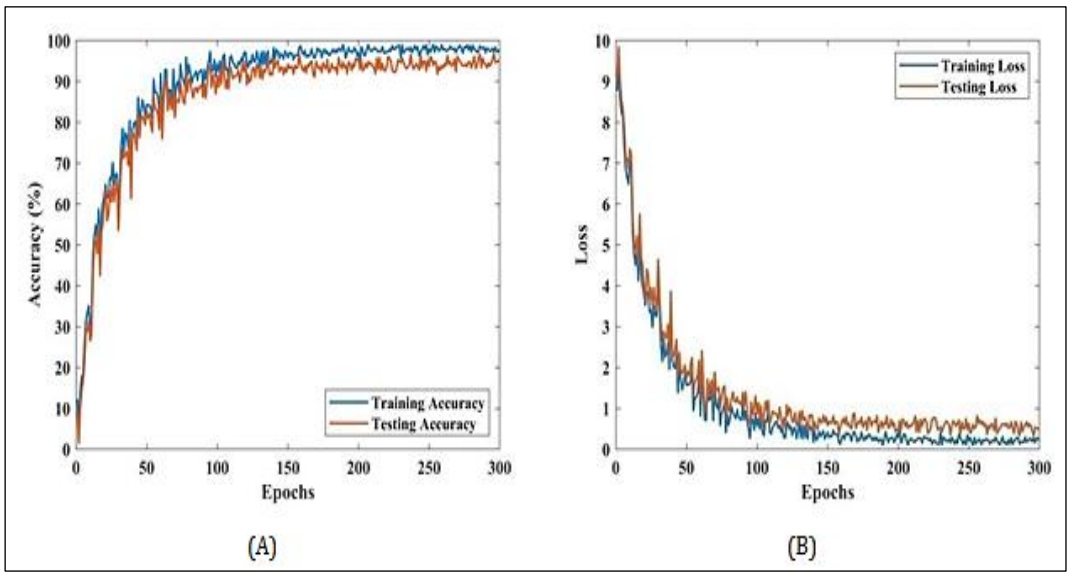

Figure 6: (A) Accuracy, (B) Loss Curves of the Proposed SA-CNN- S-LSTM- Bi-GRU

Figure 6 illustrates the accuracy-loss curves of the proposed SA-CNN- S-LSTM- Bi-GRU. The evaluation covers epochs ranging from 1 to 300, focusing on both training and testing performances. In Figure 6 (A), it is noticeable that both training and testing accuracies are constant beyond the 150th epoch. Moreover, in Figure 6 (B), the training and testing loss is constant after the 150th epoch. Hence, it is proved that the model is not under-fit or over-fit and well adaptable for WS spotting.

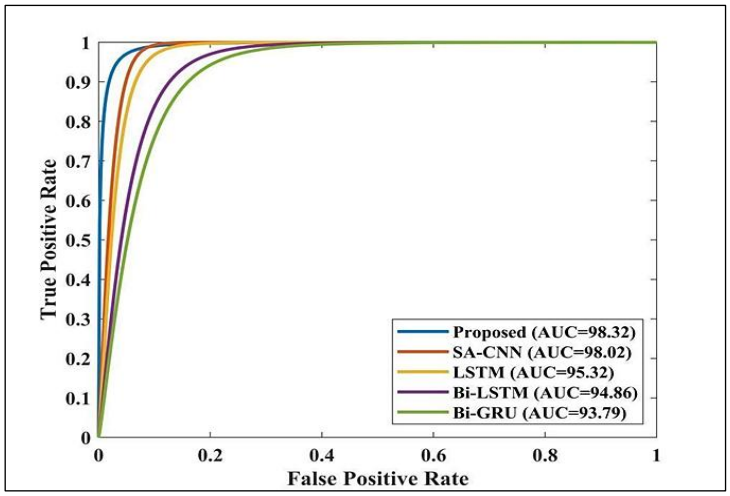

Figure 7: ROC Curve of the Proposed SA-CNN- S-LSTM- Bi-GRU

Figure 7 illustrates the ROC curve of the proposed SA-CNN- S-LSTM- Bi-GRU. AUC is utilized in binary classification problems and signifies the probability that the classifier will prioritize a randomly chosen positive sample over a randomly chosen negative sample in terms of ranking. In Figure 7, the AUC values achieved by the CNN, LSTM, Bi-LSTM, Bi-GRU, and the proposed WS are SA-CNN, LSTM, Bi-LSTM, Bi-GRU, and the proposed WS are 93.79%, 95.32%, 94.86%, 98.02% and 98.32% respectively.

Discussion

The proposed hybrid deep learning framework achieved an accuracy of 96.6% and a precision of 98.1%, demonstrating superior performance compared to several existing word spotting and inscription recognition approaches reported in the literature. Earlier studies have primarily relied on either convolutional neural networks or sequence-based models independently, which often limits their ability to capture both fine-grained visual patterns and long-range contextual dependencies in degraded inscription images (5, 18, 22). In contrast, the proposed method integrates local feature extraction using SA-CNN with global contextual modeling through S-LSTM–Bi-GRU, enabling a more comprehensive representation of ancient Tamil word structures. The improved performance is consistent with recent findings that highlight the effectiveness of combining convolutional and recurrent architectures for word spotting tasks (21, 23). However, unlike previous works that focus mainly on handwritten or printed documents, the present study specifically addresses stone inscription images, which are more challenging due to erosion, noise, and irregular character shapes. The observed performance improvement can be attributed to the feature fusion strategy, which effectively combines spatial and sequential information, thereby enhancing robustness under degraded conditions. These results confirm that the proposed approach not only aligns with existing deep learning trends but also extends their applicability to complex epigraphical datasets, making it suitable for real-world inscription analysis.

Conclusion

This study presents an innovative model SA-CNN-S-LSTM- Bi-GRU for WS recognition on the historical inscriptions in the ancient Tamil script. That is the local features were extracted by the SA-CNN and the global features were extracted by the S-LSTM- Bi-GRU. This hybrid model guarantees high efficiency in WS recognition, contributing to a more profound understanding of the historical contents embedded in the inscriptions. The experimentation was carried out by varying the training percentages from 50 to 80%. In all comparative performances, the proposed SA-CNN-S-LSTM outperformed the conventional WS approaches. Moreover, the model's flexibility in WS recognition across various scripts underscores its potential suitability for inscriptions containing multi-lingual components. This analysis has established better research for future progress in the realms of historical findings. It has significant promise for the protection and investigation of the diversity of cultural traditions encompassing various historical scripts.

Abbreviations

ACC: Accuracy, AGCN: Adaptive graph convolutional network, Bi-GRU: Bidirectional gated recurrent unit, CNN: Convolutional neural network, GNN: Graph neural network, LSTM: Long short-term memory, OCR: Optical character recognition, PRE: Precision, SA-CNN: Self-attention convolutional neural network.

Acknowledgement

The authors would like to express their sincere gratitude to the experts and researchers who provided valuable guidance during the development of this work. The authors also acknowledge the support of the Department of Computer Science & Engineering, SRM Institute of Science and Technology, for providing necessary facilities and resources to carry out this research.

Author Contributions

Vidhyavani A: conceptualization, methodology design, dataset preparation, experiments, model development, manuscript writing, overall supervision, validation, performance evaluation, proofreading, assisting with result interpretation, Manoranjitham T: technical review, verification of experimental procedures, literature support, manuscript editing. All the authors reviewed and approved the final version of the manuscript.

Conflict of Interest

The authors declare that **there is no conflict of interest** regarding the publication of this manuscript.

Declaration of Artificial Intelligence (AI) Assistance

Declaration of Artificial Intelligence (AI) Assistance This manuscript was written by the authors without the use of generative AI or AI-assisted technologies. All content is original and has been created by the authors themselves.

Ethics Approval

This study does not involve human participants, animals, or sensitive personal data. Ethical approval was therefore **not required** for this research. All procedures followed were in accordance with institutional guidelines for academic research.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

1. Lincy RB, Gayathri R. Optimally configured convolutional neural network for Tamil handwritten character recognition by improved lion optimization model. Multimedia Tools Appl. 2021;80:5917–43.
2. Vygon R, Mikhaylovskiy N. Learning efficient representations for keyword spotting using deep metric learning. Neural Netw. 2022;148:1–12.
3. Fujikawa Y, Li H, Yue X, Aravinda CV, Prabhu GA, Meng L. Recognition of oracle bone inscriptions using two deep learning models. Int J Digit Humanit. 2023;5(2):65–79.
4. Tabibian S. A survey on structured discriminative spoken keyword spotting. Artif Intell Rev. 2020;53(4):2483–520.
5. Cheikhrouhou A, Kessentini Y, Kanoun S. Multi-task learning for simultaneous script identification and keyword spotting in document images. Pattern Recognit. 2021;113:107832.
6. Handa A, Agarwal R, Kohli N. A multimodal keyword spotting system based on lip movement and speech features. Multimedia Tools Appl. 2020;79:20461–81.
7. Wei Y, Gong Z, Yang S, Ye K, Wen Y. EdgeCRNN: an edge-computing oriented model of acoustic feature enhancement for keyword spotting. J Ambient Intell Humaniz Comput. 2022;13:1–11.
8. Liu L, Yang M, Gao X, Liu Q, Yuan Z, Zhou J. Keyword spotting techniques to improve the recognition accuracy of user-defined keywords. Neural Netw. 2021;139:237–45.
9. van der Westhuizen E, Kamper H, Menon R, Quinn J, Niesler T. Feature learning for efficient ASR-free keyword spotting in low-resource languages. Comput Speech Lang. 2022;71:101275.
10. Bhuvaneswari S, Kathiravan K. Script-specific character recognition: a deep learning framework for analyzing Tamil ancient inscriptions in temples. Int J Adv Comput Sci Appl. 2024;15(2):112–20.
11. Yue X, Li H, Fujikawa Y, Meng L. Dynamic dataset augmentation for deep learning-based oracle bone inscriptions recognition. ACM J Comput Cult Herit. 2022;15(4):1–20.
12. Krithiga R, Varsini SR, Joshua RG, Kumar CUO. Ancient character recognition: a comprehensive review. IEEE Access. 2023;11:45678–95.
13. Dutta K, Krishnan P, Jawahar CV. Word spotting and recognition in Indic scripts using deep learning. Pattern Recognit Lett. 2019;121:58–65.
14. Singh H, Sharma RK, Singh VP, Kumar M. Recognition of online handwritten Gurmukhi characters using recurrent neural network classifiers. Soft Comput. 2021;25:6329–38.
15. Ruwanmini S, Dias K, Niluckshini C, Nandasara T. Sinhala inscription character recognition model using deep learning technologies. Int J Adv ICT Emerg Reg. 2023;16(1):1–10.
16. Giraldo JSP, Jain V, Verhelst M. Efficient execution of temporal convolutional networks for embedded

keyword spotting. IEEE Trans VLSI Syst. 2021;29(12):2220–8.
17. Parnami A, Lee M. Few-shot keyword spotting using metric learning approaches. IEEE/ACM Trans Audio Speech Lang Process. 2023;31:1450–62.
18. Boudraa O, Michelucci D, Hidouci WK. PUNet: novel and efficient deep neural network architecture for handwritten documents word spotting. Pattern Recognit Lett. 2022;155:19–26.
19. Yan H, He Q, Xie W. End-to-end keyword spotting using CRNN with CTC loss. IEEE/ACM Trans Audio Speech Lang Process. 2021;29:2637–49.
20. Khushali KB, Goswami MM, Mitra SK. Deep learning-based handwritten word spotting for low-resource Indic languages. Multimedia Tools Appl. 2021;80:18245–63.
21. Daraee F, Mozaffari S, Razavi SM. Handwritten keyword spotting using deep neural networks and certainty prediction. Comput Electr Eng. 2021;92:107111.
22. Cheikhrouhou A, Kessentini Y, Kanoun S. Hybrid HMM-BLSTM system for multi-script keyword spotting in printed and handwritten documents. Neural Comput Appl. 2020;32:9201–15.
23. Bhunia AK, Roy PP, Mohta A, Pal U. Cross-language framework for word recognition and spotting of Indic scripts. Pattern Recognit. 2018;79:12–31.
24. Bhunia AK, Roy PP, Sain A, Pal U. Zone-based keyword spotting in Bangla and Devanagari documents. Multimedia Tools Appl. 2020;79:27365–89.
25. Mhiri M, Desrosiers C, Cheriet M. Word spotting and recognition via a joint deep embedding of image and text. Pattern Recognit. 2019;88:312–20.
26. Das S, Mandal S. Segmentation-free word spotting in historical Bangla handwritten document using wave kernel signature. Pattern Anal Appl. 2020;23:593–610.
27. Farooqui FF, Hassan M, Younis MS, Siddhu MK. Offline handwritten Urdu word spotting using random data generation. IEEE Access. 2020;8:131119–36.

How to Cite: Vidhyavani A, Manoranjitham T. Global and Local Feature Extraction Using Deep Learning Technique for Keyword Spotting in Ancient Tamil Inscriptions. Int Res J Multidiscip Scope. 2026; 7(1): 1627-1640. DOI: 10.47857/irjms.2026.v07i01.08737