Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANCIENT TAMIL CHARACTER RECOGNITION USING OPTIMAL THRESHOLDING WITH RESNET-CAPSNET

VIDHYAVANI A[1], DR. T. MANORANJITHAM[2]
[1]Research Scholar, Department of Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai–603203, Tamil Nadu, India.
[2]Professor, Department of Computing Technologies, School of Computing,
SRM Institute of Science and Technology,
Kattankulathur, Chennai–603203, Tamil Nadu, India.

Email: va3647@srmist.edu.in , manorant@srmist.edu.in

ABSTRACT

Ancient Tamil character recognition (TCR) is complex due to its orientation, scale and writing style varying from person to person. Further, it is a challenging task for Epigraphers. Researchers for the ancient Tamil scripts and languages are relatively less when compared to other languages. When the inscriptions are on the stone wall, it is more complex to identify the characters. This research work focuses on the recognition of different Tamil characters using automated segmentation and classification techniques. Initially, the input images are pre-processed and segmented. Here, the segmentation process is carried out by Otsu thresholding. Then, the optimal threshold value is determined by the optimization adaptive salp swarm algorithm (ASSA). Finally, the Tamil characters are recognized by the DL (deep learning) model Residual capsule network (ResNet-CapsNet). The experiment is analyzed via real-time dataset and compared with the other deep learning models. Finally, the proposed ResNet-CapsNet achieved a better overall accuracy of 91.3% and specificity of 88.3% respectively and is suitable for ancient Tamil character recognition.

Keywords: *Ancient Tamil Character, Scripts, Adaptive Salp Swarm Algorithm, Inscriptions, Residual Capsule Network*

1. INTRODUCTION

Inscriptions are an essential resource of information for understanding the cultures and history of the ancient civilization. In India, inscriptions are seen everywhere (rocks, pillars and slabs) in temples [1]. These inscriptions provide more useful historical information based on religious and administrative processes; further, they are precious documented proof for understanding better the precious of life. Compared to other districts, Tamil Nadu has more inscriptions and it holds first place in the Indian Epigraphy survey [2].

Tamil is one of the ancient Indian languages which are predominantly used in Southern India, Srilanka and Malaysia. The specialty of the Tamil language is every sound pronounced has a syllable in Tamil. The economy of characters to represent a word is minimal in the Tamil language [3]. Thus, the work on Tamil script is very useful for the Tamil community around the world [4]. Ancient Tamil inscriptions are in Tamil-Brahmi form. The inscriptions date back to the 3rd century. Epigraphs are translated forms of ancient inscriptions that are deciphered from the inscription on stones, palm leaves etc. The epigraphs contain historical information regarding kingdoms, lifestyle, military strategy, business and medicines [5]. The present Tamil language is not based on the Tamil-Bramhi but on PallavaGranth. The ancient Tamil has been written in a wide range of continuum of scripts [6].

It is not easy to read the inscriptions on stones since they are written in the ancient Chola periods and they can be read only by practice. The medieval inscriptions were mostly in Tamil mixed with Grandha letters [7]. Only if we know the Grandha characters we can be able to read the Tamil stone inscriptions completely. The Historical documents can be read only by experts in language, linguistics, literal and historical backgrounds. The historical documents can be easily understood by giving the basic knowledge and training about the inscriptions and more books must be published based on this [8].

With the advent of image processing, automated script recognition is gaining popularity. The approach to reclassifying characters is in incorporates structure analysis typographically which categorizes characters in the initial step and thus reduces the scope of character recognition [9]. The field of Artificial Intelligence (AI) is essentially used when machines can do tasks that typically require human intelligence [10]. It encompasses Machine Learning (ML) where machines can learn by experience & acquire skills without human involvement [11-13]. Machine Learning (ML) and Deep Learning (DL) are the fastest-growing study areas in AI. So far many types of research have been done in the domain of classification and recognition of characters but yet, require further research in Tamil characters due to century-old or generation based characters [14-15]. The major objectives of the proposed work are:

- To introduce pre-processing, segmentation, feature and classification model for automatic ancient Tamil character recognition from the inscription.
- To introduce adaptive optimal Otsu thresholding for segmenting the Tamil characters and to improve the recognition accuracy.
- To introduce enhanced Residual capsule network deep learning model for classifying and recognizing Tamil characters.

The remainder of the paper is organized as follows: Section 2 presents a recent related work based on ancient TCR, Section 3 presents the proposed ancient TCR model with frameworks, Section 4 is the results and discussions and the entire work is concluded in Section 5.

2. RELATED WORKS

Some of the related works based on ancient TCR from inscriptions and palm leaves are listed below.

Giridhar et al. [18] introduced TCR for ancient Tamil script using DL models. Initially, the image was cropped and Otsu thresholding was applied for the binarization of the image. Then, 2D-CNN was used for the recognition of Tamil characters. Further, this work used Google text-to-speech voice for producing the audio result of digital text and the efficiency obtained by this model was 77.7%. Subadivya et al. [19] introduced Tamil-Brahmi character recognition using the DL model. Initially, the dataset undergoes various pre-processing stages like cropping, normalizing the image, reducing dimensionality and augmentation. Finally, CNN was used for extracting and classifying the features and achieved a better accuracy of 94.6%.

Athisayamani et al. [20] developed bidirectional long short-term memory (Bi-LSTM) for TCR recognition from the palm leaf. Initially, the input image was resized and rotated. Then, a local binary filter (LBF) was used for recognizing the separators like dots, and vertical and horizontal lines. Then, the feature vectors were trained by Bi-LSTM and achieved better accuracy and precision of 95% and 85% respectively. Robert Singh et al. [21] introduced the pattern matching model Enhanced-Speeded Up Robust Feature with Bag of Grapheme (E-SURF-BoG) for identifying the better interesting points. Initially, the image was enhanced, the background was subtracted and cropped. E-SURF-BoG handled all kinds of characters and achieved a high recognition rate and reduced time complexity.

Suganya and Murugavalli [22] introduced an artificial neural network (ANN) with group search with firefly optimization (GS-FF) for ancient script identification. Initially, the noise was removed and the binarization process was carried out. Then, the segmentation process was carried out for segmenting into individual characters. Finally, shape and texture features were extracted and classified using an ANN classifier. Here, the hybrid GS-FF achieved better results than the GS and FF techniques. Kumar and Geetha [23] introduced Advanced Maximally Stable Extremal Regions for TCR. The inscriptions were obtained from Garbarakshambigai temple and the pre-processing stages like image restoration and geometric transformation were carried out. Then foreground characters were extracted and obtained a better recognition rate of 95.59%.

Karunarathne et al. [24] introduced the CNN model for the recognition of Sinhala inscription characters. The processes like binarization, boundary detection, segmentation and thinning were carried out. Once the features were extracted and classified, the post-processing was carried out at the last stage. Giridharan et al. [25] developed multilayer perceptron (MLP) based system for the recognition and information retrieval of Brahmi, Vattezhuthu and Grantha letters from temple epigraphy. This model converted etymological word to equivalent meaningful Tamil words from temple epigraphy. After pre-processing the Brahmi character image, Zoning-based feature extraction was carried out and MLP was used for classification. This model achieved a better recognition rate of 84. 57% on the Brahmi dataset.

3. PROPOSED METHODOLOGY

This work presents an automated enhanced DL for character recognition from the ancient Tamil inscriptions. In this work, after converting the input image to grayscale, Improved Otsu's thresholding algorithm is implemented to remove the foreground text from the noisy background. In the improved Otsu algorithm, the threshold value is adaptively adjusted based on the ratio of pixel grey level value and total pixel numbers, by applying the ASSA based optimization technique. Then, the pre-processed image is segmented into equivalent letter blocks containing an ancient Tamil character. Finally, the DL model ResNet-CapsNet is proposed for recognizing the ancient Tamil characters. Figure 1 represents the framework of the proposed classification model.

3.1 Pre-Processing

Initially, the input RGB image is resized and converted to a grayscale image and the binarization approach is applied to that image. The binarization approach is a method of splitting of pixel values into 0 (black) and 1 (white). When T_h is the threshold of the image $f(a,b)$ and $g(a,b)$ is the threshold image and it is expressed as:

$$g(a,b) = \begin{cases} 1 & when\ f(a,b) \geq T_h \\ 0 & elsewhere \end{cases} \quad (1)$$

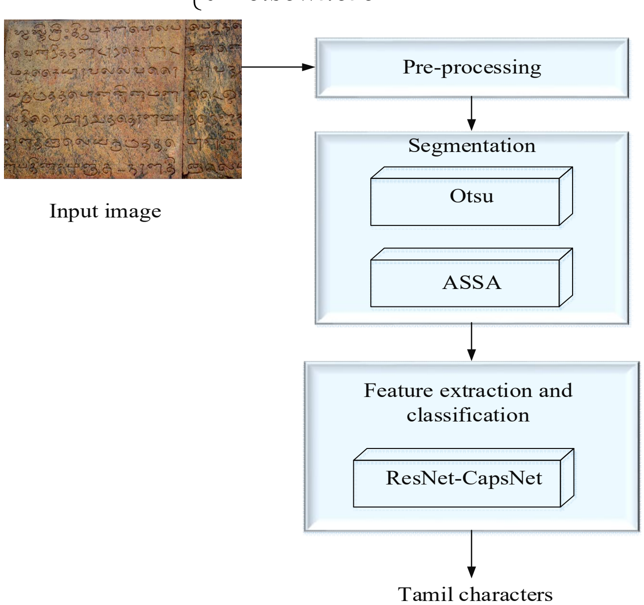


Figure 1: Framework Of The Proposed Classification Model

3.2 Segmentation

It is the second stage in image analysis and it is a challenging process in character recognition. It is a process of differentiating the character, lines and words of the inscription image and it extracts the useful segments for the analysis. In this work, Otsu with ASSA is used for finding the best values of threshold to achieve segmentation.

3.2.1 Otsu thresholding

Otsu thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels on each side of the threshold, i.e. the pixels that either fall in foreground or background. The main role of the threshold value is to find the sum of foreground and background spreads with its minimum. This thresholding is used for determining the optimal threshold in an image by information obtained from the histogram of the image. Let us consider the threshold value as t and the histogram is split as D_0 and D_1 classes. In D_0 part and class, the pixel light intensity value ranges between 0 and $t-1$. Then, for D_1 part and class, the pixel light intensity value ranges between t and $M-1$. For providing the correct thresholding fitness function, it is essential to consider the light intesnity frequency of every pixel in an image and it is expressed as:

$$p_j = \frac{l_j}{L}, p_j \geq 0, \sum_{j=0}^{N} p_j = 1 \quad (2)$$

When one threshold is utilized in the histogram of the image, then the light intensity of two classes are generated and every class has a total frequency and it is given as:

$$u_0 = \sum_{j=0}^{t-1} p_j \quad (3)$$

$$u_1 = \sum_{j=t}^{M-1} p_j \quad (4)$$

where u_0 and u_1 are the threshold and p_j is the total of pixels. Then, the value of p_j is computed by:

$$u_0 = \sum_{j=0}^{t-1} \frac{j \times p_j}{\omega_0} \quad (5)$$

$$u_1 = \sum_{j=t}^{M-1} \frac{j \times p_j}{\omega_1} \quad (6)$$

where ω_0 and ω_1 are the sum of the weights of the threshold values. The sum of the average weights of the threshold values is computed as:

$$u_T = \omega_0 \times \mu_0 + \omega_1 \times \mu_1 \quad (7)$$

The fitness function is the variation between average light intensity and weight values of the threshold. It is expressed as:

$$f = \omega_0(\mu_0 - u_T)^2 + \omega_1(\mu_1 - u_T)^2 \quad (8)$$

Otsu model attempts to maximize the target value by choosing the proper threshold. In this scenario, all the values of the threshold have to be considered. When the number of thresholds is high, then the Otsu takes more time for processing and it is not applicable for segmentation.

3.2.2 Adaptive salp swarm algorithm (ASSA)

The standard SSA [16] initiates by splitting the population into two categories like header and followers. The forepart salp of the chain is the headers and the remaining salps are followers. The position of salp is identified in $m-$ dimensions that indicate the search space. This salp finds for the food shows the goal of the swarm. The position must be updated regularly and the below equation is utilized for performing this process to salp header:

$$y_k^1 = \begin{cases} F_k + f_1((UL_k - LL_k)\, f_2 + LL_k), & f_3 \geq 0 \\ F_k - f_1((UL_k - LL_k)\, f_2 + LL_k), & f_3 < 0 \end{cases}$$

(9)

where y_k^1 and F_k are the leader position and food source in k^{th} dimension. The upper and lower limits are represented as UL_k and ; f_1, f_2 and f_3 are the random variables which ranges between 0 to 1. f_1 is the important parameter and is used for managing the balance between exploration and exploitation and it is computed as:

$$f_1 = 2e\left(-\frac{4t_1}{t_2}\right)^2 \quad (10)$$

where t_1 and t_2 are the present and maximum iterations. Once the position of leaders are updated, followers position is updated using the below expression.

$$z_k^l = \frac{1}{2}\left(z_k^l + z_k^{l-1}\right) \quad (11)$$

where z_k^l is the position of l^{th} follower in k^{th} dimension. However, the standard SSA has the drawbacks like slow convergence and being trapped by the local optima. Hence, to tackle these issues this work presents an adaptive version of SSA and it is called as ASSA. This optimizer enhances the exploitation and exploration capacities by introducing the inertia weight i_w. Further, this optimizer improves the convergence speed during the search space and enhances the segmentation performance. Finally, the position of salp header is updated as:

$$y_k^1 = \begin{cases} i_w F_k + f_1((UL_k - LL_k)\, f_2 + LL_k), & f_3 \geq 0 \\ i_w F_k - f_1((UL_k - LL_k)\, f_2 + LL_k), & f_3 < 0 \end{cases}$$

(12)

3.2.3 Optimal thresholding

In this process, for finding the optimal thresholding, the entire threshold in the image is analyzed from 0 to 255. When the number of thresholds is increased, the number of iterations for finding the threshold also increases and the complexity is also high. Hence, the number of thresholds is computed optimally in the ASSA with less error and less time. This model needs that the problem to be framed accurately and every solution to the problem to be set as a member of SSA and the fitness function must be defined correctly. In this work, every problem's solution is a different threshold in the histogram and is defined as:

$$ASSA_j = \ll ASSA_j^1, ASSA_j^2, ASSA_j^3, \ldots, ASSA_j^D \gg$$

(13)

$$ASSA = \begin{bmatrix} ASSA_1^1 & ASSA_1^2 \ldots\ldots\ldots\ldots ASSA_1^D \\ ASSA_2^1 & ASSA_2^2 \ldots\ldots\ldots\ldots ASSA_2^D \\ \ldots\ldots & \ldots\ldots\ldots \quad\quad \ldots\ldots\ldots\ldots \\ ASSA_n^1 & ASSA_n^2 \ldots\ldots\ldots\ldots ASSA_n^D \end{bmatrix}$$

(14)

where SSA_j is the threshold vector in a histogram of the image and is set as the member of SSA optimization. SSA_j^k is the k^{th} image threshold in j^{th} member and D is the total threshold utilized in an image.SSA is the population of m threshold and every threshold requires to be updated by the Otsu fitness function. It is expressed as:

Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$f_{Otsu} = \omega_0(\mu_0 - u_T)^2 + \omega_1(\mu_1 - u_T)^2 + \ldots + \omega_n(\mu_m - u_T)^2$$
(15)

The fitness function of m is the different threshold like $\{t_1, t_2, \ldots, t_m\}$ for the threshold of the image. For evaluating the population of the proposed model using the fitness function, every threshold is set based on the following expression:

$$f_{Otsu}(ASSA) = \begin{bmatrix} f_{Otsu}(ASSA_1^1 ASSA_1^2 \ldots\ldots\ldots ASSA_1^D) \\ f_{Otsu}(SSA_2^1 ASSA_2^2 \ldots\ldots\ldots ASSA_2^D) \\ \ldots\ldots \ldots\ldots \quad \ldots\ldots\ldots\ldots \\ f_{Otsu}(ASSA_n^1 ASSA_n^2 \ldots\ldots\ldots ASSA_n^D) \end{bmatrix} = \begin{bmatrix} f_{Otsu}(ASSA_1) \\ f_{Otsu}(ASSA_2) \\ \cdot \ \ldots\ldots \\ f_{Otsu}(ASSA_n) \end{bmatrix}$$
(16)

In this model, various random thresholds are generated initially, each of which salp and their range are set as the low and high light intensities in the images and it is expressed as:

$$ASSA_j^k = L + (H - L) \times r \quad (17)$$

where L and H are the low and high light intensities and r is the random number which ranges between 0 and 1. At every phase, ASSA is applied to every threshold vector for finding the optimal threshold in Otsu and improve the algorithm's exploration. The following process (Algorithm 1) is carried out for the optimal thresholding.

Algorithm 1: Optimal thresholding based on Otsu-ASSA
Input: The test image
Output: segmentation of characters
Initialize population, number of threshold, maximum iteration
Calculate the histogram of image
Code the threshold image in the order of salp $ASSA_j =<< ASSA_j^1, ASSA_j^2, ASSA_j^3, \ldots$
Every solution is generated in the high and low random intensity of light
for i=1
$SSA_j^k = L + (H - L) \times r$
end for
end for
The initial population of salp is: $ASSA = \begin{bmatrix} ASSA_1^1 & ASSA_1^2 \ldots\ldots\ldots\ldots ASSA_1^D \\ ASSA_2^1 & ASSA_2^2 \ldots\ldots\ldots\ldots ASSA_2^D \\ \ldots\ldots & \ldots\ldots \quad \ldots\ldots\ldots\ldots \\ ASSA_n^1 & ASSA_n^2 \ldots\ldots\ldots\ldots ASSA_n^D \end{bmatrix}$
Define every salp by the fitness function $f_{Otsu} = \omega_0(\mu_0 - u_T)^2 + \omega_1(\mu_1 - u_T)^2 + \ldots$
while $(i \le \max\_\ iter)$ **do**
The salp's position is updated by Equation (12)
else
The salp's position is updated by Equation (11)
end if
end for
$i = i + 1$
end while
Return the best solution for the thresholding

3.3 Feature Extraction And Classification

After segmenting the characters by the optimal thresholding, the DL model ResNet-CapsNet is used for automated feature extraction and character recognition. In the proposed DL model the average pooling in the ResNet model is replaced by the CapsNet. By this DL model, the relative position and feature direction are explained. Hence, the unnecessary features are eliminated and the deep features are extracted by the threshold based convolution.

ResNet: For the residual block, let the term and the potential mapping are given as $F(y)$ and $H(y)$. Then, the residual learning is $H(y) = F(y) + y$ and the identity mapping is obtained by $F(y) \rightarrow 0$.

$$F(y) = H(y) - y \quad (18)$$

where y is the residual block's input.

When the shortcuts are mapped by a similar dimension, the expression of the residual term is given as:

$$F = U_2 \sigma(U_1 y) \quad (19)$$

$$y = F(y, w_j) + y \quad (20)$$

When the shortcuts are mapped by the varying dimension, the expression of residual term is given as:

$$F = U_2 \sigma(U_1 y) \quad (21)$$

$$y = F(y, w_j) + U_s y \quad (22)$$

CapsNet: The DL model CapsNet has capsules and neurons. The capsule's output is vector and the neuron's output is scalar. This CapsNet model uses a dynamic routing for achieving better results. The conventional CNN model has the problem in the pooling layer; during the feature extraction, the CNN may lose its essential features and it decreases the accuracy. The CapsNet uses the dynamic routing to tackle this issue. When the actual and predicted values are close, the obtained values and DigitCaps capsule unit is utilized. When the actual and predicted values are not close, the obtained values and DigitCaps capsule unit are not promoted. The dynamic routing is computed by the p^{th} capsule layer output u_p, q^{th} prediction vector and W_{pq} is the weighting matrix.

$$\hat{u}_{q|p} = W_{pq} u_p \quad (23)$$

Then, the association between the neighbouring capsule layers c_{pq} is given as:

$$c_{pq} = \frac{\exp(b_{pq})}{\sum_s \exp(b_{ps})} \quad (24)$$

where b_{pq} is the probability values of p^{th} and q^{th} capsule layers. The dynamic routing process is given as:

$$S_q = \sum_p c_{pq} \times \hat{u}_{q|p} \quad (25)$$

where S_q is the capsule's q^{th} input vector and the capsule's q^{th} output vector is given as:

$$v_q = \frac{\| s_q \|^2}{1 + \| s_q \|^2} \frac{s_q}{\| s_q \|} \quad (26)$$

Finally, the loss is used in the CapsNet is the integration of margin and reconstruction losses. It given as:

$$loss_i = T_i \max(0, a^+ - \| v_i \|)^2 + \lambda(1 - T_i)\max(0, \| v_i \| - a^-)^2$$
(27)

where T_i is the classification function, a^+ is the false negative is penalized (upper bound), and a^- is the false positive is penalized (lower bound) and λ is the hyper-parameter used for adjusting the a^+ and a^-.

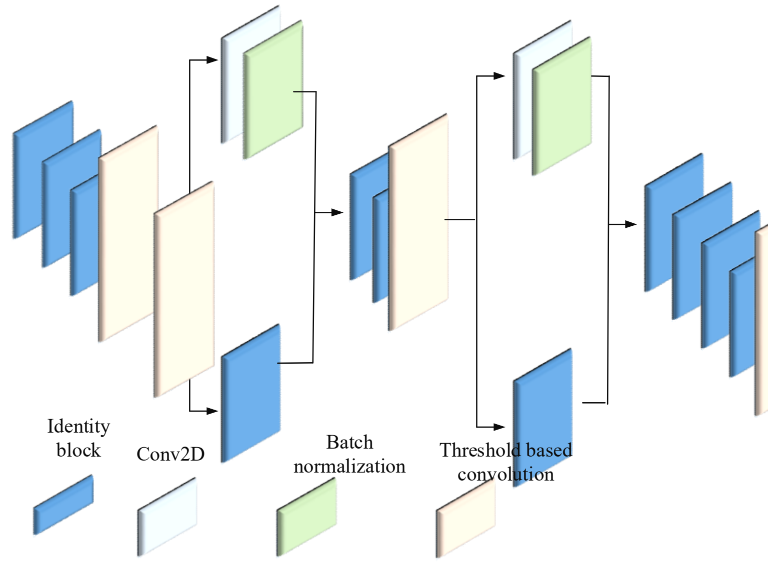


Figure 2: Structure Of Rescapsnet

ResCapsNet: This work considers ResNet34 and CapsNet for feature extraction and classifying the characters as shown in Figure 2. The ResNet34 model has the identity blocks of 3, 2, 4 and 2 and for extracting the essential features. Then, after the residual blocks the average pooling layers are placed and this layer is replaced by the CapsNet. The convolutional layer output is provided to the primary capsules and in this work there are 32 capsules are considered. Feature points include a lot of unnecessary features in addition to semantic information.

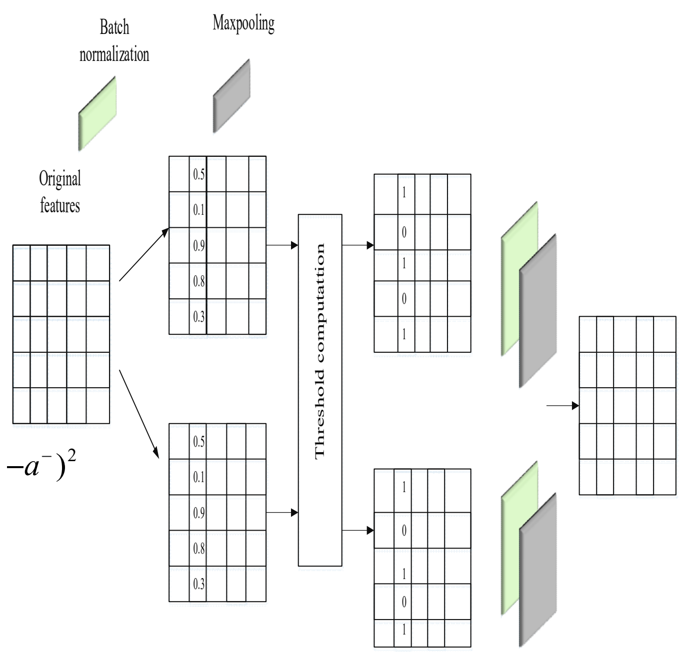


Figure 3: Structure Of Threshold Based Convolutional Operation

When there are a lot of unnecessary features, the model can have an impact on other feature points and ultimately degrade the classification performance. To tackle this, this work consider the convolutional operator after the threshold for reducing the unnecessary features and achieve better semantic features. The proposed threshold based convolutional operation is to utilize the sigmoid which keeps the output of feature map's features. Figure 3 shows the model of threshold based convolutional operation. For all feature points, the values of weights are produced. Moreover, this work considered two array sets with similar feature maps. When the values of the threshold is higher than the values of weights produced by the Sigmoid, then the feature points are converted into 0 and it reduces the unnecessary features in the ancient character recognition.

4. RESULTS AND DISCUSSION

This model is evaluated in Python with OpenCV for operations of image processing. The system configuration is 8GB RAM and Windows 7 Intel Pentium 4 CPU. The performance of the ResNet-CapsNet model is analyzed for various quality measures. Table 1 shows the hyper-parameters of ResNet-CapsNet.

Table 1: Hyper-Parameters Of Resnet-Capsnet

Hyper-parameters	**Value**
Size of batch	50
Activation Function	ReLU
Every capsule dimension	8
Primary and Digitcaps length	2
Learning rate	0.01
Optimizer	Adam

Some of the expressions used in this work are listed below:

Accuracy: It is the correctness of the model and it is expressed as:

$$A = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (28)$$

Sensitivity: The test can exactly determine the characters using the training characters. It is expressed as:

$$Se = \frac{T_p}{T_p + F_n} \quad (29)$$

Specificity: This measure is used for testing the proposed model without the training set and it is expressed as:

$$Sp = \frac{T_n}{T_n + F_n} \quad (30)$$

Precision: This measure is used for estimating the relation between positive and negative samples. It is expressed as:

$$P = \frac{T_p}{T_p + F_p} \quad (31)$$

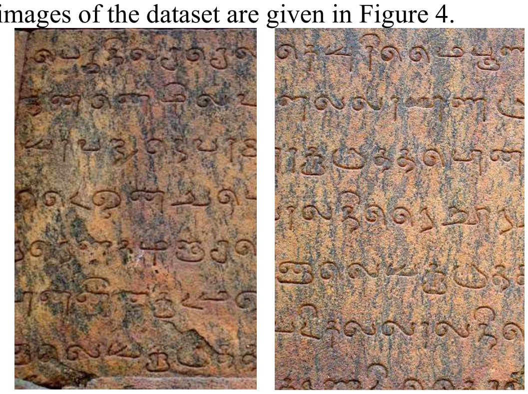
F-measure: In image classification, this measure is used for summarizing precision and specificity and it is expressed as:

$$F - measure = \frac{2T_p}{2T_p + F_p + F_n} \quad (32)$$

where T_p -true positive, F_p - False positive, T_n - true negative and F_n - false negative

4.1 Dataset Acquisition

There are no standard benchmarks that are unavailable for ancient Tamil scripts, the dataset is collected from Google and ancient Tamil script is obtained between the seventh and twelfth centuries. The dataset is divided into 70% for training and 30% for testing. Some of the sample images of the dataset are given in Figure 4.



Journal of Theoretical and Applied Information Technology
30th June 2025. Vol.103. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

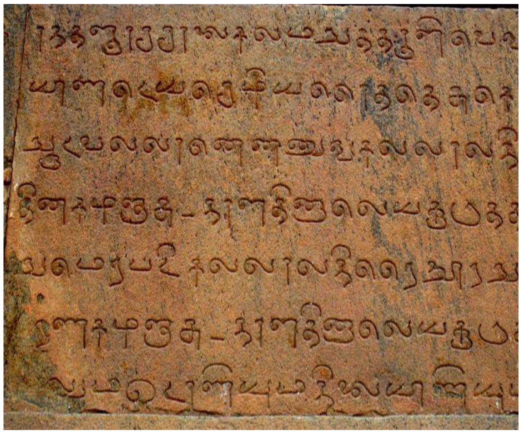
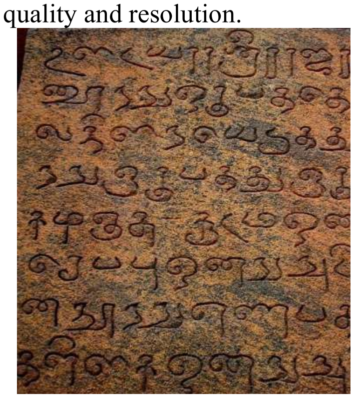


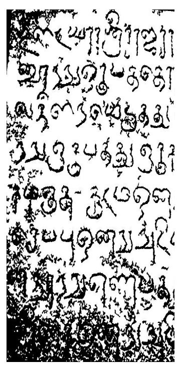
Figure 4: Sample Images Of The Dataset

4.2 Qualitative Analysis

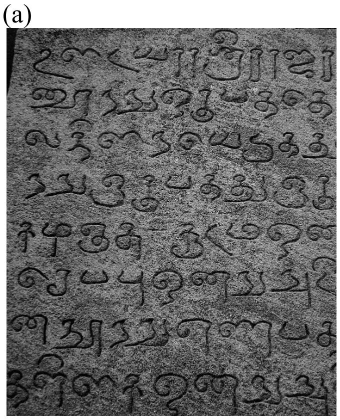
The following section presents the qualitative analysis of the proposed character recognition from the ancient Tamil inscriptions model. Further, the segmentation performance of the characters is also provided. The model is analyzed with various inscription images and identified the performance and accuracy of the model. The character's accuracy is based on the input image quality and resolution.



(a)



(b)



(c)

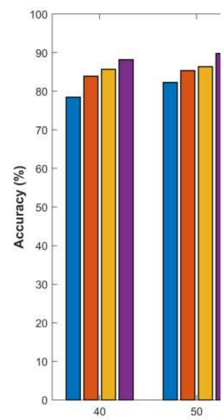
(d)

Figure 4: Qualitative Analysis (A) Input Image (B) Grayscale Image (C) Binary Image (D) Optimal Otsu Threshold Image

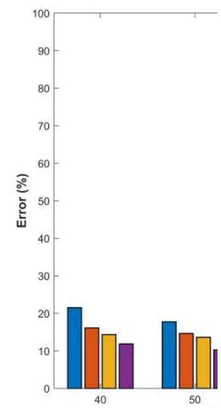
Figure 4 shows the qualitative analysis of the Input image, Grayscale image, Binary image and optimal Otsu threshold image. It is observed from the figure that the proposed model efficiently segmented the Tamil characters.

4.3 Quantitative Analysis

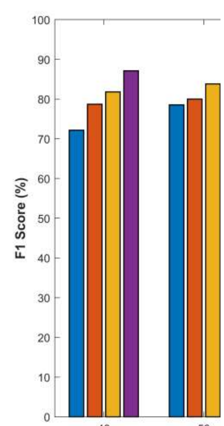
This section presents the quantitative analysis of the proposed character recognition from the ancient Tamil inscriptions model. The methods like CNN, ResNet, CapsNet are compared with the proposed ResNet-CapsNet model.



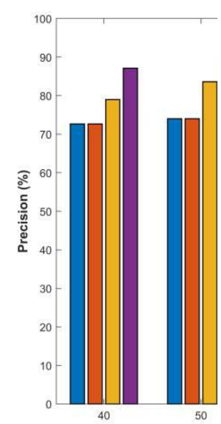
(a)



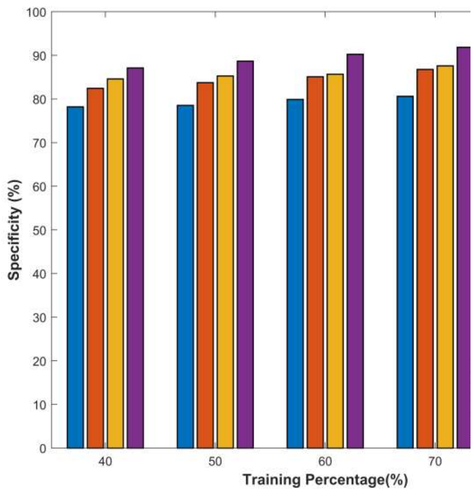
(b)



(c)



(d)



(e)

Figure 5: Performance Comparison Of Proposed And Existing Methods

Figure 5 shows the performance comparison of the various DL approaches. The performance is evaluated by varying the training percentage from 40% to 80%. In Figure 5 (a), the accuracy of the proposed model is 87%, 88%, 90%, 92% and 97% for the training percentages of 40%, 50%, 60%, 70% and 80% respectively. The error should be less for the proposed ResNet-CapsNet and it is seen from the Figure 5 (b) that the proposed ResNet-CapsNet achieved less error rate for all training percentages. Similarly, in Figure 5 (c) and 5 (d) that the F1-score and precision values achieved are 92.4% and 92.7% for the proposed ResNet-CapsNet model. Finally, it is seen that the specificity value achieved by the proposed 91.1% and it is 1.1%, 1.0%, and 0.8% better than the approaches like CNN, ResNet and CapsNet respectively. Moreover, the existing models achieved less accuracy than the proposed model. It is observed from the graph that when the training pentangle is increased, the performance is also increased. Hence, from the comparison, it is proved that the proposed model can be efficiently exploited in Tamil character recognition.

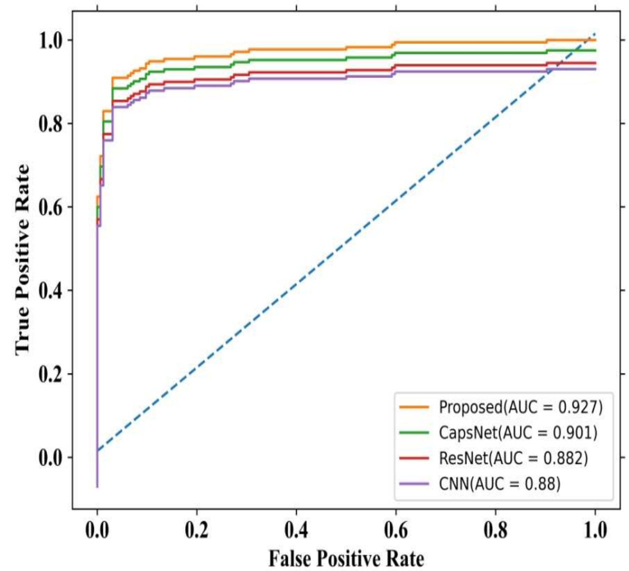
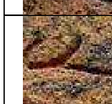


Figure 6: ROC Comparison

Figure 6 presents the ROC (receiver operating characteristic curve) of various approaches like CNN, ResNet, CapsNet are compared with the proposed ResNet-CapsNet model. The curve is marked between false positive rate and true positive rate. The AUC (area under the curve) values achieved by the CNN, ResNet, CapsNet are compared with the proposed ResNet-CapsNet model are 0.88, 0.882, 0.901 and 927.

Table 2: Performance Comparison Of The Letters Using The Resnet-Capsnet

Letters	Accuracy	Sensitivity	Specificity	Precision	F-measure
	98.4	93.7	95.4	96.9	97.8
	98.4	94.2	95.8	98.5	95.2
	98.7	94.5	95.2	94.2	96.7
	97.1	96.4	96.7	96.2	95.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

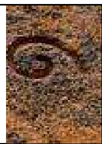
	95.3	97.3	97.4	97.3	96.6
	96.9	95.8	98.4	96.4	96.1
	96.6	94.6	96.9	95.4	97.7
	96.7	97.4	95.6	94.3	95.3
	98.8	97.6	95.4	93.2	94.2

Table 2 shows the performance comparison of various letters using the ResNet-CapsNet. Performance like Accuracy, Sensitivity, Specificity, Precision, and F-measure are considered for the comparison. The characters like ‘oo’, ‘ya’, ‘la’ and ‘thu’ achieved better accuracy of more than 98% respectively. Further, the precision performance is high for the letter ‘ya’ and the sensitivity performance is high for the letter ‘thu’.

5.COMPARATIVE ANALYSIS, CRITICAL EVALUATION, AND FUTURE RESEARCH DIRECTIONS

Numerous studies have explored ancient Tamil character recognition (TCR), particularly using image processing and deep learning techniques. Prior works have employed methods such as convolutional neural networks (CNN), bidirectional long short-term memory (Bi-LSTM), and artificial neural networks (ANN). For instance, Giridhar et al. used a 2D-CNN with Otsu thresholding and achieved an accuracy of 77.7%, while Subadivya et al. leveraged CNNs for Tamil-Brahmi script recognition with a reported accuracy of 94.6%. Athisayamani et al. developed a Bi-LSTM-based model for palm-leaf manuscript recognition, achieving high precision and accuracy. Although these approaches show promising results, they often fall short in handling the complex, noisy, and eroded nature of stone inscriptions.

The present study differs from earlier works in both motivation and methodology. The primary motivation was to address the challenges posed by irregular spacing, erosion, and inconsistent carving styles present in stone inscriptions. To this end, developed a hybrid deep learning framework that integrates an adaptive version of Otsu thresholding using the Salp Swarm Algorithm (ASSA) with a Residual Capsule Network (ResNet-CapsNet). This dual approach enhances segmentation and classification by combining the thresholding precision of ASSA with the spatial awareness and feature-preserving capabilities of CapsNet.

5.1 Study Design and Research Protocol

This study follows an experimental research design focused on developing and evaluating a novel deep learning-based character recognition model for ancient Tamil inscriptions. The pipeline consists of the following stages:

- Dataset Acquisition: Collection of Tamil inscription images from public historical archives and online sources.
- Preprocessing: Grayscale conversion, noise removal, and adaptive Otsu thresholding using ASSA.
- Segmentation: Character extraction using optimal threshold values derived via salp swarm optimization.
- Feature Extraction and Classification: Implementation of ResNet-CapsNet for feature learning and character classification.
- Evaluation Metrics: Accuracy, precision, specificity, F1-score, and AUC are computed to assess model performance.

All experiments were performed using Python, OpenCV, and deep learning frameworks under controlled settings, ensuring reproducibility and consistent evaluation.

5.2 Research Questions and Hypotheses

Research Questions:

1. Can an adaptive Otsu thresholding method based on metaheuristic optimization improve character segmentation accuracy in ancient Tamil inscriptions?
2. Does the integration of ResNet and CapsNet outperform standalone deep learning models in recognizing degraded ancient Tamil characters?

Hypotheses:

- *H1*: Adaptive thresholding using the Salp Swarm Algorithm significantly enhances segmentation precision in ancient character recognition.
- *H2*: A ResNet-CapsNet hybrid architecture achieves higher recognition accuracy and feature retention than conventional CNN and ResNet models.
-

5.3 Research Problems and Open Issues

Despite encouraging results, several challenges persist:

- Data Availability: A lack of standardized, annotated datasets for ancient Tamil inscriptions limits reproducibility and broader research applicability.
- Generalizability: The model is trained on a specific time frame (7th–12th centuries), and its adaptability to other script variations or time periods remains untested.
- Computational Efficiency: While effective, the combined ResNet-CapsNet and ASSA optimization pipeline may be resource-intensive, requiring simplification for practical deployment.
- Script Complexity: Tamil inscriptions often contain mixed scripts (Tamil-Grantha), complex ligatures, and damaged sections, which pose additional recognition difficulties.

Addressing these challenges opens future research directions for robust, scalable, and versatile models in epigraphical analysis.

6 FUTURE RESEARCH DIRECTIONS

Future research can proceed along several promising directions:

- **Dataset Expansion and Benchmarking**: Curating a standardized, annotated dataset of ancient Tamil scripts covering various centuries and inscription styles would enhance the reproducibility and comparability of future work.
- **Cross-Script Generalization**: Extending the model to support other Dravidian scripts, including Telugu, Malayalam, or Kannada, would validate its broader applicability.
-
- **Real-Time and Lightweight Models**: Developing optimized variants of the ResNet-CapsNet architecture suitable for mobile or edge devices could support field-based epigraphical studies.
- **Integration with NLP**: Incorporating natural language processing tools for automatic translation, word segmentation, and contextual analysis could help build end-to-end inscription understanding systems.
- **3D Surface and Multimodal Analysis**: Employing 3D imaging techniques or combining visual and tactile data could improve the recognition of highly degraded or eroded inscriptions.

7. CONCLUSION

This research advances the state-of-the-art in ancient Tamil character recognition by integrating adaptive thresholding with a robust deep learning framework. Despite some limitations, the model offers a solid foundation for further exploration and practical deployment in digital epigraphy and cultural heritage preservation.

REFERENCES

[1] Sharma, R. and Kaushik, B., 2020. Offline recognition of handwritten Indic scripts: a state-of-the-art survey and future perspectives. Computer Science Review, 38, p.100302.

[2] Sutramiani, N.P., Suciati, N. and Siahaan, D., 2021. MAT-AGCA: Multi Augmentation Technique on small dataset for Balinese character recognition using Convolutional Neural Network. ICT Express, 7(4), pp.521-529.

[3] Vijayalakshmi, R. and Gnanasekar, J.M., 2022, April. A Review on Character Recognition and Information Retrieval from Ancient Inscriptions. In 2022 8th International Conference on Smart Structures and Systems (ICSSS) (pp. 1-7). IEEE.

[4] Devi S, G., Vairavasundaram, S., Teekaraman, Y., Kuppusamy, R. and Radhakrishnan, A., 2022. A Deep Learning Approach for Recognizing the Cursive Tamil Characters in Palm Leaf Manuscripts. Computational Intelligence And Neuroscience, 2022.

[5] Subramani, K. and Subramaniam, M., 2021. Creation of original Tamil character dataset

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

through segregation of ancient palm leaf manuscripts in medicine. Expert Systems, 38(1), p.e12538.

[6] Janani, G., Vishalini, V. and Kumar, P.M., 2016, March. Recognition and analysis of Tamil inscriptions and mapping using image processing techniques. In 2016 Second International Conference on Science Technology Engineering and Management (ICONSTEM) (pp. 181-184). IEEE.

[7] Durga Devi, K. and Uma Maheswari, P., 2022. Recognition and Classification of Stone Inscription Character Using Artificial Neural Network. In Artificial Intelligence and Technologies (pp. 253-261). Springer, Singapore.

[8] Durga Devi, K. and Uma Maheswari, P., 2019. Digital acquisition and character extraction from stone inscription images using modified fuzzy entropy-based adaptive thresholding. Soft Computing, 23(8), pp.2611-2626.

[9] Devi, K.D. and Maheswari, P.U., 2021. Recognition and Classification of Stone Inscription Character Using Artificial Neural Network. Artificial Intelligence and Technologies: Select Proceedings of ICRTAC-AIT 2020, 806, p.253.

[10] Mahajan, H., Narayan, P., Reddi, S.I. and Sharmila Banu, K., 2021. A Systematic Approach to Handle Ancient Tamil Script Using Deep Learning Techniques. In Soft Computing and Signal Processing (pp. 57-68). Springer, Singapore.

[11] Manigandan, T.V.V.D.V.N.B., Vidhya, V., Dhanalakshmi, V. and Nirmala, B., 2017, August. Tamil character recognition from ancient epigraphical inscription using OCR and NLP. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 1008-1011). IEEE.

[12] Kavitha, B.R. and Srimathi, C., 2019. Benchmarking on offline Handwritten Tamil Character Recognition using convolutional neural networks. Journal of King Saud University-Computer and Information Sciences.

[13] Vinotheni, C., Lakshmana Pandian, S. and Lakshmi, G., 2021. Modified convolutional neural network of Tamil character recognition. In Advances in Distributed Computing and Machine Learning (pp. 469-480). Springer, Singapore.

[14] Sornam, M. and Vishnu Priya, C., 2017, October. Deep convolutional neural network for handwritten Tamil character recognition using principal component analysis. In International Conference on Next Generation Computing Technologies (pp. 778-787). Springer, Singapore.

[15] Rajakumar, S. and Bharathi, V.D.S., 2012. 7th century ancient tamil Character recognition from Temple wall inscriptions. Indian Journal of Computer Science and Engineering (IJCSE), 3(5).

[16] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M., 2017. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in engineering software, 114, pp.163-191.

[17] Chu, S.C., Tsai, P.W. and Pan, J.S., 2006, August. Cat swarm optimization. In Pacific Rim international conference on artificial intelligence (pp. 854-858). Springer, Berlin, Heidelberg.

[18] Giridhar, L., Dharani, A. and Guruviah, V., 2019. A novel approach to ocr using image recognition based classification for ancient tamil inscriptions in temples. arXiv preprint arXiv:1907.04917.

[19] Subadivya, S., Vigneswari, J., Yaminie, M. and Diviya, M., 2020. TAMILBRAHMI SCRIPT CHARACTER RECOGNITION SYSTEM USING DEEP LEARNING TECHNIQUE. International Journal of Computer Science and Mobile Computing, 9(6), pp.114-119.

[20] Athisayamani, S., Robert Singh, A. and Sivanesh Kumar, A., 2021. Recurrent neural network-based character recognition system for Tamil palm leaf manuscript using stroke zoning. In Inventive Communication and Computational Technologies (pp. 165-176). Springer, Singapore.

[21] Robert Singh, A., Athisayamani, S. and Alphonse, A.S., 2021. Enhanced speeded up robust feature with bag of grapheme (ESURF-BoG) for Tamil palm leaf character recognition. In Inventive Communication and Computational Technologies (pp. 27-39). Springer, Singapore.

[22] Suganya, T.S. and Murugavalli, S., 2020. A hybrid group search optimization: firefly algorithm-based big data framework for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ancient script recognition. Soft Computing, 24(14), pp.10933-10941.

[23] Kumar, A.N. and Geetha, G., 2019. Character recognition of ancient South Indian language with conversion of modern language and translation. Caribb. J. Sci, 53(20), pp.2019-2031.

[24] Karunarathne, K.G.N.D., Liyanage, K.V., Ruwanmini, D.A.S., Dias, G.K.A. and Nandasara, S.T., 2017. Recognizing ancient sinhala inscription characters using neural network technologies. Internationa Journal of Scientific Emgineering and Applied Sciences, 3(1), p.12.

[25] Giridharan, R., Vellingiriraj, E.K. and Balasubramanie, P., 2016, April. Identification of Tamil ancient characters and information retrieval from temple epigraphy using image zoning. In 2016 International conference on recent trends in information technology (ICRTIT) (pp. 1-7). IEEE.