An Ensemble Neural Network Model For Malayalam Character Recognition From Palm Leaf Manuscripts

DHANYA SUDARSAN

Cochin University of Science and Technology, Kochi, India, dhanyasudarsan@cusat.ac.in

DEEPA SANKAR

Cochin University of Science and Technology, Kochi, Kerala, India, deepasankar@cusat.ac.in

Palm leaf manuscripts (PLMs), crucial for ancient communication hold a wealth of information encompassing culture, art, literature, religion, and medicinal wisdom. Malayalam, Kerala's official language, significantly contributes to medical sciences, making palm scripts invaluable, especially in times of pandemics. This study introduces a ground-breaking model for automatic recognition of characters in Malayalam palm scripts. This is the first significant deep learning-based attempt, to our knowledge, to automate Malayalam character recognition in PLMs. The developed model is a fusion of fine-tuned Convolutional Neural Network (CNN) and Bi-directional Long Short-Term Memory (BiLSTM). Discriminative features were extracted from each character in the manuscript through multiple convolutional layers, and these feature vectors were then classified into their respective character classes using an ensemble deep learning model. The performance of the proposed method was evaluated using a self-generated dataset of old Malayalam PLMs from the period 1800 to 1908 AD. Overcoming challenges such as complex morphology, large character set, similar characters, and a unique writing style, the model achieved an impressive accuracy of 96.40%, outperforming state-of-the-art systems. Notably, the model obtained a negative predictive value (NPV) of 99.3%, positive predictive value (PPV) of 83.33%, sensitivity of 79.55%, specificity of 99.45% and F-Measure of 88.39%.Thus this advancement marks a significant milestone in automatic transcriptions providing a crucial tool for doctors and researchers.

Keywords: Convolutional neural network; Recurrent Neural Networks (RNN);Bi-directional Long Short-Term Memory ;Deep learning; Machine learning (ML) ; Malayalam character recognition; Palm leaf manuscripts

1 INTRODUCTION

Automatic recognition of handwritten characters poses a real challenge due to diverse writing styles, and its complexity increases when recognizing characters in Malayalam [1]. Malayalam, a Dravidian language, holds a unique place as one of the under-resourced languages. Until the 15th century, it was even considered a dialect of Tamil. The language has seen the use of various scripts, including the Vatteluttu script and the Pallava Grantha script, both of which are Tamil scripts. Malayalam is unique to Indian languages since it uses both Tamil (Dravidian language) and Sanskrit (Indo-European language) grammar, and it emphasises 'Chandrakkala' and 'Chillu', which are not written but expressed while speaking in Tamil and do not exist in Indo-European languages [2].

Malayalam character recognition process is highly challenging due to the complex shapes and similarities of Malayalam characters, leading to less accurate results [3].The task becomes extremely complex when we are trying to recognize letters written in Palm Leaf Manuscript (PLM). PLMs, crafted from dried out palm leaves, served as the most accepted writing material in olden times. Depending on the width of the leaves, one or two holes were bored through which a cord could be threaded to bind them into book form. In Figure 1, a sample of a palm leaf manuscript is displayed.

In contrast to typical handwritten documents, the writing instrument used for inscribing palm leaf manuscripts differs, and due to the rough surface of palm leaves, the characters' shapes in these manuscripts vary from those in standard handwritten documents. Moreover Palm leaves may be exposed to different forms of natural noise such as pollen, fungus, streaks, colour fading, pigment, textual specks, spots, black edges and lines. Punch holes in the surface of the leaf can be mistaken as a character. The presence of broken characters may occur due to factors such as fading of dyes used for polishing, infections, or insect bites.

ACM 2375-4699/2024/08-ART
http://dx.doi.org/10.1145/3686311

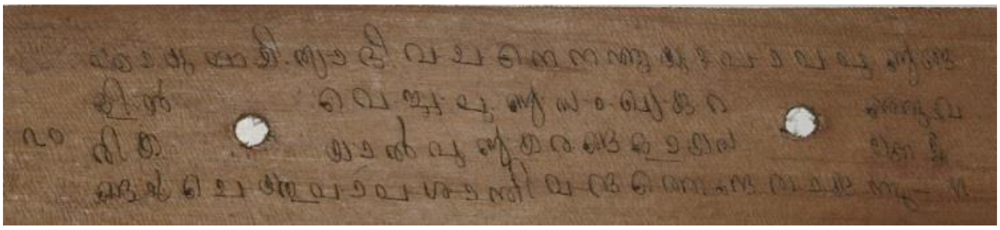


Fig.1. Palm leaf manuscript sample

PLMs are abundant sources of knowledge which communicate ancient art, culture, music, literature, and religion which disseminate Vedas, epics, mantras etc. Kerala is renowned for Ayurveda, a branch of the Atharva Veda. Ayurveda's core principles revolve around combating aging, promoting longevity, and preventing diseases. Kerala has deep-rooted connections with this ancient science, even before the formal introduction of classical Ayurveda in the early Christian era. Kerala's significant contribution includes Panchakarma therapies, which, unfortunately, faded in North India under the shadow of colonialism but were diligently preserved by the families in Kerala [5]. Kerala's hospitable climatic conditions have helped maintain the purity of these medicinal practices. Given that Palm leaf manuscripts served as the primary medium of communication in ancient times, and Malayalam being the official language of Kerala, many of the commentaries on these traditional medical practices were originally authored in Malayalam and inscribed on palm leaves.

While many research institutes and families in Kerala safeguard a treasure trove of knowledge, the majority of PLMs remain in their original state, making their preservation a daunting task. These delicate palm leaves are susceptible to damage from hot and humid climates, moisture, and insects. They require storage in dry, cool, well-ventilated spaces and are often housed in acid-free containers like folders, envelopes, or boxes to shield them from acidic materials. Furthermore, to prevent harm, PLMs necessitate handling by trained professionals who use special gloves or tools. However, this conservation and handling process is laborious, leading to an alarming rate of manuscript loss. The most effective way to preserve the valuable information within these manuscripts is through digitization. Since no automated system exists for deciphering palm scripts, the assistance of palm script readers for manual translation is required which is a laborious and challenging task. This process heavily relies on the availability and proficiency of script readers, and with very few experts in this field, it becomes expensive, impractical, and time-consuming. To address this difficulty, an automatic character recognition and translation system needs to be developed to digitize archived Malayalam palm leaf images.

Even though numerous major studies exist on the recognition of characters in ancient handwritten documents, covering both non-Indian and Indian scripts utilizing traditional [6, 7, 8, 9, 10] and Deep Learning (DL) Neural Network (NN) approaches [11, 12, 13, 14], they are not fully leveraged for Malayalam character recognition. So as the part of exploring the possibilities of DL, after evaluating the various popular DL models and its combinations the paper developed:

- A frame work for automating the entire process of Malayalam palm leaf manuscript digitization
- A novel ensemble neural network system which is a combination of CNN and Bi-LSTM for Malayalam character recognition from old Malayalam PLMs.

The effectiveness of the constructed model is assessed using the self-generated dataset of Malayalam palm script characters, and the outcomes are compared with those obtained using cutting-edge techniques. Moreover, even with the reduced number of layers, the novel character recognition model was able to obtain a high performance without compromising the accuracy which demonstrates the overall quality of the developed model. To our knowledge, this marks the first major attempt to use DL for automating the Malayalam character recognition from PLMs.

The remaining paper is arranged as follows: Section 2 explores the work in this area; section 3 presents the suggested architecture and methodologies section 4 describes the dataset and experimental setup. Results and discussions are included in the 5th section. Performance analysis, validation with the state-of-the art and error analysis are conducted in the 6th, 7th and 8th sections, respectively. The conclusion and future scope are discussed in the final section.

2 RELATED WORKS

While several studies have explored Malayalam character recognition, only a limited number have specifically addressed the challenges of recognizing characters within Malayalam palm scripts. This is due to the additional challenges such as background noise, special writing style in PLMs and unavailability of a standard dataset. The related work section is bifurcated into two segments: the initial part delves into character recognition from Malayalam PLMs, and the subsequent part emphasizes character recognition from handwritten Malayalam documents.

2.1 Character recognition from Malayalam PLMs

Sudarsan *et al*. developed a technique to digitize Malayalam characters in old palm scripts [15] which used contrast based adaptive binarization for background removal and CNN for character recognition. Even though the model gives an accuracy of 96.7%, the developers themselves identified that the model as such is not perfectly suitable for recognizing all characters in Malayalam. Sruthy S Kumar proposed a character recognition system from Malayalam pamphlets, where they integrated Ostu's for binarization, horizontal/vertical projection for segmentation and CNN for training. Accuracy obtained for recognizing 43 Malayalam characters was 95.30% [16]. Significant contributions in the field have been made by Sudarsan and Sankar, who pioneered a ground-breaking character segmentation and feature extraction technique for Malayalam character recognition from PLMs [17]. Their innovative approach combines Log-Gabor filter with uniform rotational invariant Local Binary Pattern (LBP), achieving remarkable success when integrated with a Support Vector Machine (SVM) classifier. Notably, their method attains an impressive accuracy of 95.57%, positioning it as the current state-of-the-art in the domain.

2.2 Character recognition from handwritten Malayalam documents

Sreeraj *et al.* introduced a method using time domain characteristics and dynamic representations for online Malayalam character recognition achieving 98.125% accuracy with K-Nearest Neighbour (KNN) classifier [18]. Chacko and Anto developed an online sequential algorithm for 44 Malayalam characters; achieving 96.83% accuracy with single hidden layer feed forward neural networks [19]. Primekumar *et al.* proposed a wavelet transform based feature extractor with Fuzzy ARTMAP classifier, achieving 81% accuracy [20]. Reference [21] evaluates SVM and Hidden Markov Model (HMM) for online Malayalam handwriting recognition, achieving 97.97% accuracy using discrete wavelet transformation. Sudarsan and Joseph used adaptive contrast-based binarization and CNN, achieving 96.46% accuracy [23]. Alex and Das utilized the Otsu algorithm and SVM for character recognition with a maximum accuracy of 89.2% [24]. Nair *et al.* employed CNN for Malayalam Optical Character Recognition (OCR) [25]. Kishna and Francis used HMM for cursive Malayalam font recognition [26]. Manjusha *et al.* described a scattering network using Singular Value Decomposition (SVD) for recognizing machine-printed and hand-printed Malayalam characters [27]. James *et al.* developed zoning-based techniques for Malayalam handwritten character recognition [28].

Handcrafted features are used in traditional machine learning models, which require strong prior knowledge about the dataset. This is indeed a difficult task that could lead to inter- and intra-operator deviations. Moreover, there exist a need to design a feature extractor which can merge both lower level and higher-level features. DL methods, using a hierarchical learning approach automatically generate powerful discriminative features. The literature survey reveals that DL-based character recognition methods have gained recognition due to their excellent performance, which eliminates the need for manual feature extraction and selection. This improvement is largely due to the use of architectures such as CNNs [11, 12], RNNs [13, 14], and Long Short-Term Memory (LSTM) [29, 30] networks. However, there exists a significant research gap in the utilization of DL methods for improving the accuracy of Malayalam character recognition from PLMs. So this paper develops a NN model for Malayalam character recognition from PLMs.

3 PROPOSED RESEARCH FRAMEWORK AND METHODOLOGY

The proposed research frame work consists of a pre-processing phase and a character recognition phase. Figure 2 demonstrates the overall research framework.

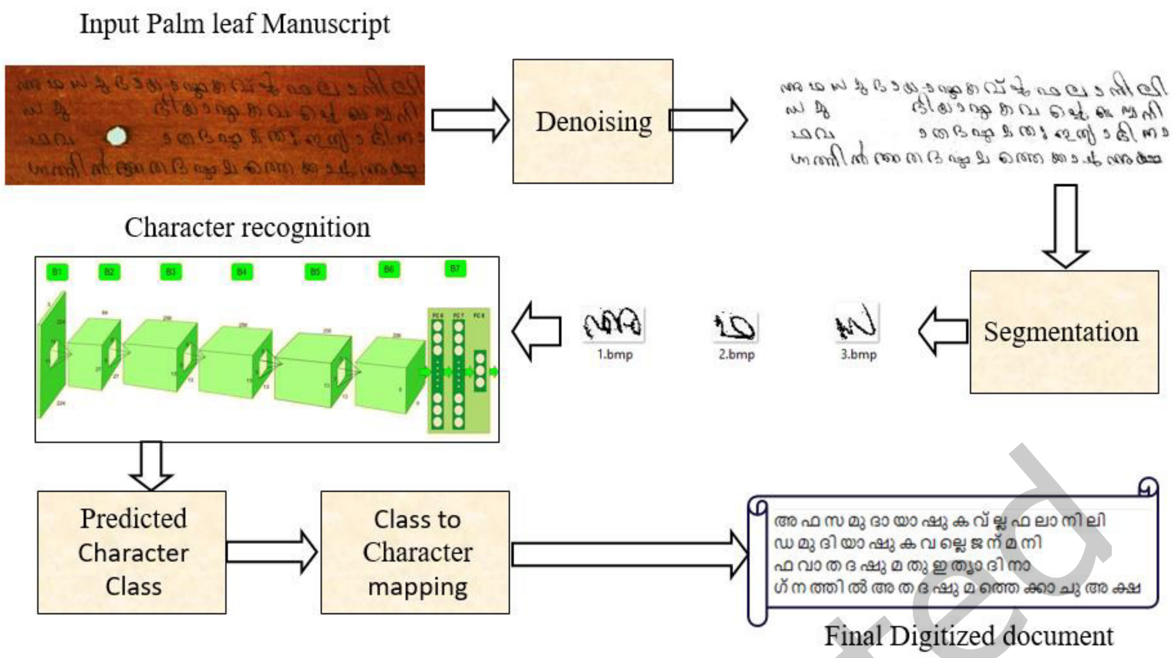


Fig.2. Research frame work for the character recognition from Malayalam palm leaf manuscripts

3.1 Pre-processing

In the pre-processing phase the palm leaf manuscript images are denoised using a complete denoising solution proposed in [31]. The denoising algorithm is a two-step process in which consists of background noise removal followed by the punch hole removal. The background noise removal algorithm is a perfect combination of a set of filters and morphological operations which can deal with manuscripts having different background colours, taken under different lighting conditions; different noise levels and stains. Initially a low pass filter is used to filter high-frequency noise components. In palm leaf manuscript there can be the presence of lots of high frequency components such as pollen, fungus, streaks, pigment ink sipping, markings, textual specks, spots, black edges and lines. Since the low pass filter only passes signals below its cut-off frequency while attenuating all signals above it, all the high frequency noise components will get filtered off at this stage. The chosen window size is 5×5 which was found to remove almost all high-frequency noise and resulted in the smoothing of edges. This is followed by fast non-local means algorithm for sharpening the character edges and eliminating Gaussian noise in the image based on window similarity. Then a morphological opening and closing functions are applied. Opening smoothes internal corners and eliminates slowly varying intensities; thereby enhancing the image and closing soften the contours. The morphological black hat enhances the pixels and Otsu thresholding eliminates the background. Finally an adaptive thresholding is used to remove the remaining noise present in the image and also performs a final sharpening, which separates the filled-up character boundaries that occurred after Otsu thresholding. This denoised image is given for punch removal. The output of the denoising algorithm is the punch hole removed denoised palm leaf manuscript. This denoising technique was able to give a PSNR value of 19.40 dB.

This denoised output is given for segmentation. We used the connected component based segmentation algorithm proposed in [32] for segmenting characters. All the challenges during segmentation such as presence of broken characters, presence of vowel and 'Chandrakkala' (Symbol signifying the end of a consonant sound) are properly dealt by this novel segmentation algorithm. The denoised input image can be smoothed using a 1 D median filter to remove any left out noise after initial noise removal process. The algorithm starts with line segmentation. A horizontal projection profile of the image is generated which calculates the total number of pixels along each row of an image. This maximum peak is used for locating the lines. After segmenting each line, characters need to be segmented. During this stage, two problems need to be addressed. Since we are dealing with palm-leaf manuscripts, during the denoising process, some of the foreground pixels may be wrongly classified as background pixels and get removed. Also due to the fading of dyes used for polishing some pixels may not be visible. All these cause the occurrence of broken characters in the palm-leaf manuscripts. The absence of intermediate pixels hinders the segmentation algorithm from recognizing subsequent pixels as part of a single connected component, leading to incorrect character segmentation. To deal with this problem, a dilation operation is performed initially, using a disk structuring element, which will add extra pixels and joins the broken characters. As a second level, to deal with 'Chandrakkala', dilate the resultant binary image with vertical line structuring element. Vertical line structuring element adds extra pixels vertically, which helps in connecting the 'Chandrakkala' that comes vertically above the character thereby considering the character and 'Chandrakkala' together as a single connected component. Now using connected component algorithm retrieve the coordinates of the smallest bounding box enclosing each connected component and apply it on the denoised input image. By this stage all the characters except the dependent vowel sounds gets properly enclosed in the bounding box. The bounding box enclosing the dependent vowel sounds will still contain the part of its adjacent character. To deal with this, inside the bounding box drawn on the input image, remove all the components touching the boundary. After this operation the unnecessary segment will get removed and the connected component which is supposed to be inside remains since it is not touching the boundary. Now get the first maximum peak value and find the bounding box with y-coordinate greater than the maximum peak value. Crop the character and save in a folder corresponding to its line number. Now check for the next maximum peak and repeat the procedure for next line. Continue this procedure till no more maximum peak value is found. The segmentation algorithm was able to produce an accuracy of 95.04%.

These segmented characters constitute the initial dataset for training newly developed the neural network model for character recognition from Malayalam palm leaf manuscripts described in next section. In order to overcome the segmentation errors, the segmented characters are manually checked before adding it to the training set. The developed character recognition system predicts the class of each character and the character corresponding to the class is printed via class to character mapping.

Traditional models were employed for noise removal and segmentation because a neural network model is deemed inappropriate in this context. The limited availability of PLMs, poses challenges, making it difficult to acquire sufficient samples for training a neural network model without risking overfitting.

3.2 Deep learning model developed for character recognition from Malayalam palm leaf manuscripts

In this section we developed a NN based system for recognising the segmented characters from Malayalam palm scripts. The novel model is a combination of CNN [33] and BiLSTM [34] as shown in Figure 3. Here, CNN layers are used for feature extraction on input data, combined with BiLSTM to support prediction.

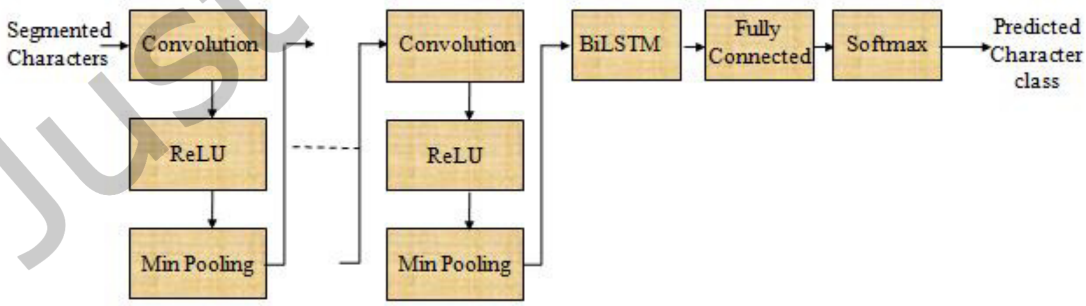


Fig.3. Architecture of the novel model developed for Malayalam character recognition from palm leaf manuscripts

3.2.1 Model evolution.

The model building process commenced with the creation of an initial CNN architecture tailored for Malayalam palm leaf character recognition. CNN was chosen due to its ability to automatically learn relevant features from input, capture a spatial hierarchy of features, and be invariant to spatial transformations. They are robust to noise/variations in the input data, such as changes in lighting, background, or occlusion. The foundational design incorporated two convolutional layers with rectified linear unit (ReLU) activation aiming to capture hierarchical features. Given Malayalam's complex morphology and high degree of similarity between characters, we chose ReLU as the activation function to prevent information squeeze.

Each convolutional layer was followed by Min Pooling. While Max pooling tends to amplify these outliers, min pooling has the opposite effect, making it more robust to outliers and is better at preserving small features and is effective at reducing the impact of background noise. It is most suitable for our application since we have a light background, and min pooling is effective in selecting dark pixels from a light background.

Striving for depth and abstraction, we appended a third convolutional layer. Recognizing the need of further feature extraction, we experimented with additional convolutional layers, observing the marginal improvements in model's discriminative capabilities.

After experimentation, we found that while the customized CNN model achieved satisfactory accuracy compared to other existing models, it struggled to predict Malayalam characters with a high degree of similarity. The similar characters that were not correctly predicted by the developed customized CNN model are presented in Table 1.

Table 1. Similar characters misclassified by the developed model CNN Model

Character	Actual class	Predicted class
	ഈ	ഇ
	വ	പ
	ഢ	ഡ
	ള്ള	ള
	ങ	ന്ദ

Therefore, we leveraged the capabilities of LSTM. LSTMs are adept at capturing intricate and evolving patterns present in the images, and they are also effective at minimizing information loss by operating directly on sequences, preserving spatial and structural details within the images [33]. LSTM memory can be seen as a gated cell, where each cell decides whether to store or delete information based on the importance assigned to the feature, which is decided through weights learned by the algorithm. The adaptable learning capability of LSTM is particularly valuable when certain aspects of the image become more significant during the evaluation process, which was extremely beneficial in differentiating similar Malayalam characters. After extracting features using CNN, we experimented with various LSTM architectures including classic LSTM, stacked LSTM and BiLSTM. Among these BiLSTM demonstrated the highest accuracy. By combining BiLSTM with the newly developed CNN architecture, our model was able to rectify the misclassification errors.

3.2.2 Feature extraction.

Nine learnable layers make up our model; five of them are convolutional, one bidirectional LSTM layer and the final three are FC layers, as illustrated in Figure 4.

In this work, the segmented characters are directly provided to the customized CNN, and the kernel is applied to the pixel intensities in the image. The first convolutional layer with a 4-pixel stride ensures quick processing. Reducing the number of convolutional layers reduces the parameters to be trained. The convolution operation in each convolution layer can be expressed as follows:

$$f_i^c = R(\sum_i^{N-1} K_{pq} * I_i^{c-1} + Bias_i) \qquad (1)$$

where, f_i^c is the i^{th} output of the c^{th} convolution layer, K_{pq} is the trainable filter of size $(p * q)$, I_i^{c-1} is the last feature maps or input image data, the symbol $*$ denotes discrete convolution operation, and R(.) is the Rectified linear unit (ReLU) activation function [33] expressed as follows:

$$f(x) = Max(0, x) \quad (2)$$

Where $f(x)$ represents the output of the ReLU activation function when it is applied to an input x, which is the weighted sum of inputs to a neuron before the ReLU activation is applied. ReLU computes the maximum of two values. If x is greater than or equal to 0, $f(x)$ is x itself and $f(x)$ is 0 otherwise.

Fig.4. Fine-tuned neural network model structure

The features extracted from CNN are fed to BiLSTM. The internal constitution of a BiLSTM cell is illustrated in Figure 5. It consists of three gating units : the forget gate, the input gate, and the output gate, which are needed to reset the content of the cell, decide when to send data to the cell, and read out the entries from the cell, respectively [34].

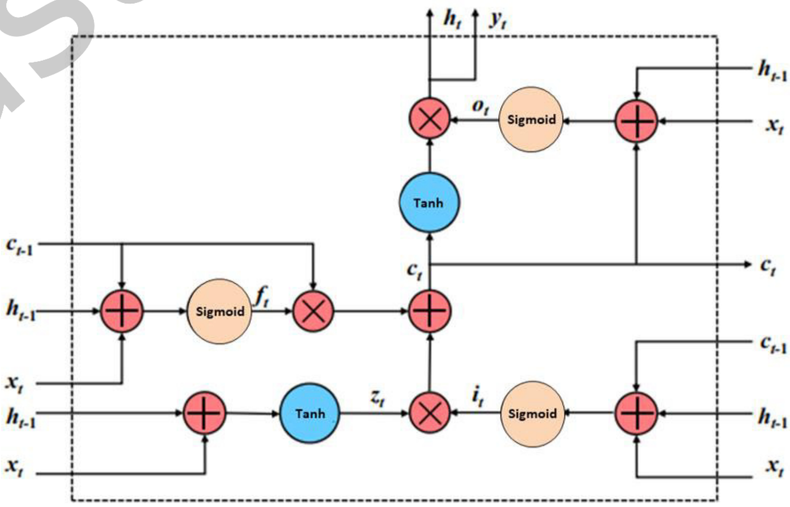


Fig 5. Memory cell of the LSTM [34]

For the first BiLSTM layer, data is transmitted forward in a chronological order and the values of the forget, input, and output gates at the t^{th} step are obtained through the following equations [34]:

$$\boldsymbol{f}_t^F = \sigma\left(\boldsymbol{U}_f^F \boldsymbol{x}_t + \boldsymbol{W}_f^F \boldsymbol{h}_{t-1} + \boldsymbol{T}_f^F \boldsymbol{c}_{t-1} + \boldsymbol{b}_f^F\right) \qquad (3)$$
$$\boldsymbol{i}_t^F = \sigma(\boldsymbol{U}_i^F \boldsymbol{x}_t + \boldsymbol{W}_i^F \boldsymbol{h}_{t-1} + \boldsymbol{T}_i^F \boldsymbol{c}_{t-1} + \boldsymbol{b}_i^F) \qquad (4)$$
$$\boldsymbol{o}_t^F = \sigma(\boldsymbol{U}_o^F \boldsymbol{x}_t + \boldsymbol{W}_o^F \boldsymbol{h}_{t-1} + \boldsymbol{T}_o^F \boldsymbol{c}_t + \boldsymbol{b}_o^F) \qquad (5)$$

Where $\mathbf{U}^{\mathrm{F}}$, $\mathbf{W}^{\mathrm{F}}$and $\mathbf{T}^{\mathrm{F}}$ are the weights matrices and $\mathbf{b}^{\mathrm{F}}$is the bias vector used for the positive data flow, and subscripts f, i, and t are the weight matrices and the bias vectors used in the forget, input, and output gates, respectively. σ is the sigmoid function which takes the value 1 or 0 representing whether the data is retained or discarded by the forget gate, respectively, while the reverse is true in the input and output gates. The value of the memory cell and the hidden layer's output at the t^{th} step is computed using equations 6 and 7.

$$\boldsymbol{c}_t^F = \boldsymbol{f}_t^F \boldsymbol{c}_{t-1}^F + \boldsymbol{i}_t^F \tan h(\boldsymbol{U}_c^F \boldsymbol{x}_t + \boldsymbol{W}_c^F \boldsymbol{h}_{t-1}^F + \boldsymbol{b}_c^F \qquad (6)$$
$$\boldsymbol{h}_t^F = \boldsymbol{o}_t^F \tan h(\boldsymbol{c}_t^F) \qquad (7)$$

Where $\tan h$ () is an activation function. In the second layer of BiLSTM, data is propagated backwards. Similarly, the output $\mathbf{h}^B$of the hidden layer for the reverse data flow can also be obtained using equations (6) and (7). The output of the BiLSTM at each time is obtained using the following relation:

$$\boldsymbol{h}_t = \boldsymbol{h}_t^F + \boldsymbol{h}_t^B \quad (8)$$

Schematic view of the BiLSTM used in this study is shown in Figure 6 [34]. The features from BiLSTM has been flattened and integrated into the Fully Connected (FC) layer of the tailored model and observed an incremental performance improvement.

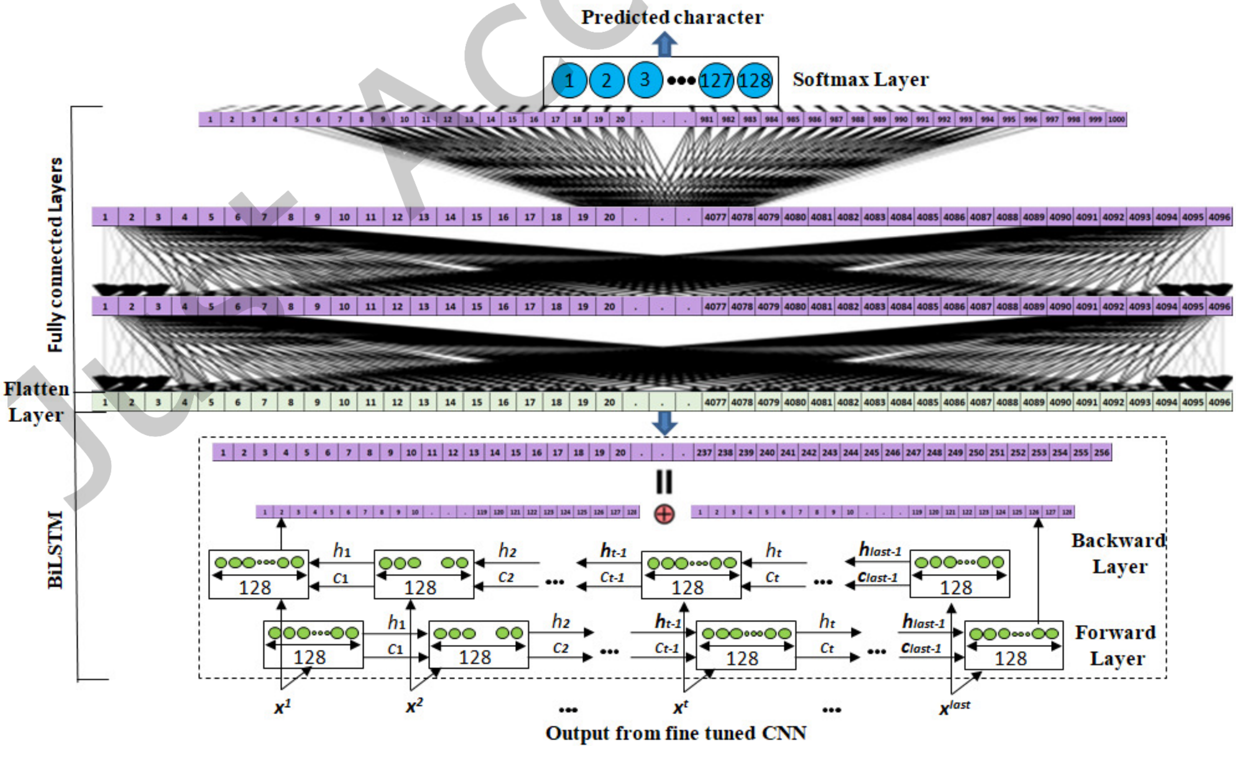


Fig.6. Schematic view of the BiLSTM used in this study [34]

The block wise architecture of the developed neural network model is illustrated in Figure 7. Block 1 (B1) and Block 2 (B2) each contain a convolutional layer, a ReLU activation function, and a normalization layer. Block 3 (B3), Block 4 (B4) and Block 5 (B5) each constituting one convolutional layer and ReLU. Block 6 (B6) includes one BiLSTM layer with drop out and a flattening layer and Block 7 (B7) constitutes three FC layers. The initial blocks B1 and B2 extract low-level features, while the blocks B4 and B5 extract high-level features.

A Local Response Normalization (LRN) [35] is performed at the end of first two blocks (Block 1 and Block 2) to promote lateral hampering which refers to the ability of a neuron to decrease the activity of its neighbours. The LRN operator performs a channel-wise sliding window normalization of each column of the input array so that locally maximum pixel values are used as excitation for the subsequent layers. The equation is as shown below:

$$qv^{i}_{(x,y)} = \frac{pv^{i}_{(x,y)}}{\left(k+\alpha \sum_{j=max(0,i-\frac{n}{2})}^{min(N-1,i+\frac{n}{2})} (a^{j}_{x,y})^2\right)^{\beta}} \qquad (9)$$

Here $pv^{i}_{(x,y)}$ is the activity of the neuron before applying kernel i at the position (x, y), while $qv^{i}_{(x,y)}$ is the activity of the neuron after applying kernel i at the position (x, y) where the sum runs over 'n' adjacent kernel maps at the same spatial position. N represents the kernels in each layer, α is the normalizing constant, βcontrols the power to which normalization term is raised. The length of the neighborhood, or the series of adjacent pixels to be taken into account during normalizing, is determined by the variable n. a represents the activation of a neuron, k is a constant added to the denominator to prevent division by zero and to control the amount of normalization applied. LRN is most suitable for Malayalam character recognition because it helps the network focus on the most important features of each character. In Malayalam script, characters are typically composed of several sub-components or strokes, which can be arranged in different ways. The stroke order and the position of the strokes relative to each other are important features that the network needs to learn to recognize characters accurately. LRN helps to enhance these features by normalizing the responses of neurons that are activated by similar features. For example, if two strokes in different characters are arranged in the same way, the neurons that respond to those strokes will be normalized together, making it easier for the network to distinguish between those characters. Additionally, LRN can help to reduce the impact of irrelevant features or noise in the input, improving the accuracy of the recognition process.

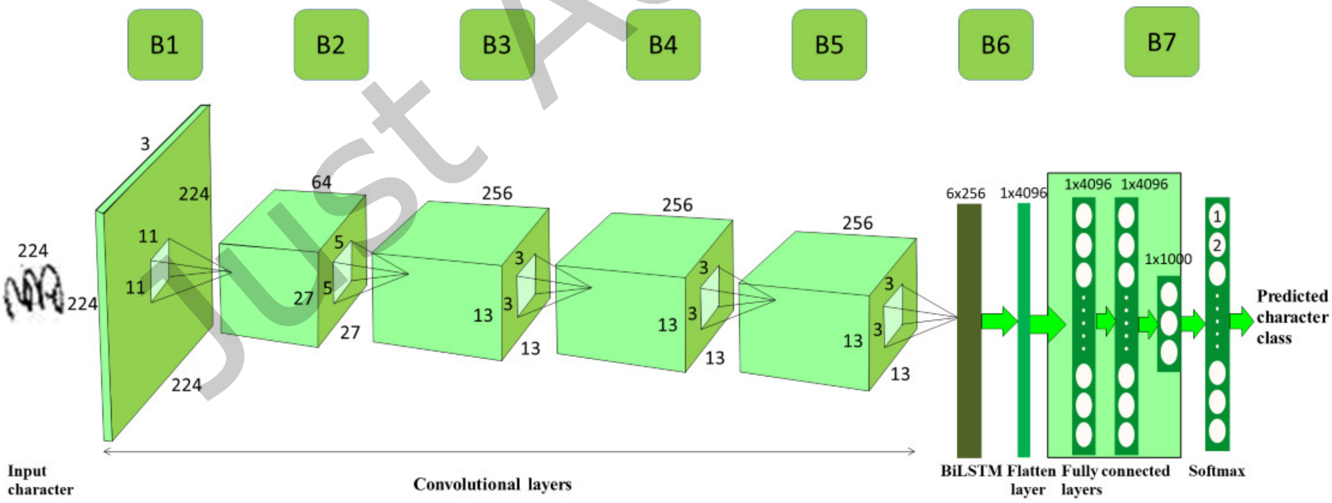


Fig.7. Block wise architecture of the developed neural network model

3.2.3 Classification and recognition of characters

Subsequently, these features from FC layers of the fine-tuned model were utilized and forwarded to the customized SoftMax layer for classification. Since it is a multiclass classification problem, we employed multi class cross entropy as the loss function. The error function of each data point can be calculated using the formula:

$$L\,(P,Q) \;=-\; \sum_{i=0}^{c} P_i log(Q_i)\,. \qquad (10)$$

Where $L(P, Q)$ represents cross-entropy loss between the true probability distribution P and the predicted probability distribution Q and c represent the number of classes.

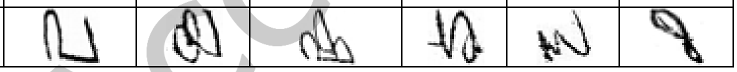
4 DATASET AND EXPERIMENTAL SET UP

This section includes details about the datasets, the experimental setup, and the model's hyperparameters.

4.1 Dataset description

As mentioned in section 1, unlike typical handwritten documents, PLMs are inscribed with a stylus on rough surfaces, resulting in different character shapes and loss of curves. Therefore, standard handwritten Malayalam character datasets are unsuitable for this research. Table 2 illustrates the differences in character shapes between normal handwritten Malayalam characters and those written in Malayalam PLMs.

Table 2. Character shape difference between the normal handwritten Malayalam characters and the characters written in Malayalam PLMs

Malayalam characters	അ	ഇ	ഉ	ഖ	ഗ	ഘ
Palm leaf characters						
Malayalam characters	ഠ	ഡ	ത	ദ	ധ	ന
Palm leaf characters						
Malayalam characters	വ	യ	ഷ	ശ	സ	ൃ
Palm leaf characters						

For this investigation, we collected 3500 ancient PLMs that are inscribed during 1800 to 1908 AD. The PLMs are manually collected from diverse sources, such as churches, manuscript libraries, and ancestral residences. Details of the palm leaf dataset are provided in Table 3.

Table 3. Details of palm leaf dataset

Source	Period	No: of leafs	Number of characters
Oriental Research Institute and Manuscript Library, Trivandrum	1800-1889 AD	788	23598
Kuruppampady St. Peter & Paul Catholic Church	1800-1908 AD	750	19229
Ernakulam Major Archbishop's Palace - Vechoor Church	1865-1866 AD	620	10052
Ancestral residencies(Majority were Palm leafs with medical imprints)	1800-1900 AD	1220	30019
Internet	1800-1860 AD	122	502
Total Manuscripts		**3500**	
Total characters in the dataset		**83,400**	

We considered 128 Malayalam language characters (fifteen vowels, thirty-six consonants, five chillu, three consonant signs, nine vowel signs, anuswaram, visargam, chandrakkala and fifty-seven conjunct consonants).For character recognition, we started with 100 segmented palm leaf character images for each character. Subsequently, data augmentation is employed to enhance the dataset. Given the significant similarity between certain Malayalam characters, caution must be exercised during data augmentation to prevent misinterpretation of the characters. The ranges are identified after careful investigation. Image is rotated in the angle 20 degree to 10 degree clock wise, shear in the angle 20 degree to 15 degree clock wise and 15 degree anti clock wise to 20 degree clock wise around its centre point. The scaling can vary between 0.8 along the vertical axis and 1.2 along the horizontal axis.

Table 4 shows some sample cases. In table 4, rotation and shear beyond the specified range will lead to interpret the character a) as character b) and for image c) scaling beyond the range will lead to misinterpretation of character as c) as character d) or as a punch hole. Manually labelled segmented images of each character are taken for experimentation. The character recognition dataset consists of 200 images of each character and 250 images of characters which showed misclassification errors (listed in table 1). The dataset consists of a total of 83,400 character images, with 70% allocated for training and model generation, and 30% for testing the model's accuracy. Additionally, 20% of the training images are used for validating the proposed NN architecture.

Table 4. Similar Malayalam Characters

a) ഡ	b) ന
c) o	d) O

Moreover, to accommodate various handwriting styles, characters with a high degree of variability were meticulously chosen to compose the dataset. In order to illustrate the diversity within our dataset, we present a selection of sample characters along with their respective Euclidean distances [36] in Table 5.

Table 5. Comparison of the Euclidean distance of each character with the reference character indicating the variability of the dataset

1.	Reference character	Sample characters from dataset			
	ഗ				
	Euclidean distance	4.1794	3.4050	1.9031	2.3761
2.	Reference character	Sample characters from dataset			
	ക				
	Euclidean distance	3.9664	1.9869	3.4879	2.5106
3.	Reference Character	Sample characters from dataset			
	മ				
	Euclidean distance	3.8742	2.3943	1.9872	3.4319
4.	Reference Character	Sample characters from dataset			
	ന				
	Euclidean distance	1.9013	3.3365	4.1679	2.4654

4.2 Experimental set up

The experiments were conducted using MATLAB®2019a, 64-bit (win64) and its deep learning toolbox. The developed models were trained and tested on an Intel Core i3 PC with CPU @ 1.70GHz, Windows 10 professional 64-bit with 6GB DDR3 SDRAM.

The hyperparameters of the newly developed model are summarized below. All of the aforementioned optimal values of different hyperparameters have been yielded using Bayesian optimization technique [37].

4.2.1 Hyperparameters of conventional neural network.

The segmented characters are resized to 224x224. Number of kernels and size of kernel for the convolution operations vary in each block. Table 6 presents the Block wise kernel, stride and input/output dimension details of fine-tuned architecture and Table 7 presents the optimal values for various hyperparameters of convolution layers in the fine-tuned NN model.

Table 6. Block wise kernel, stride and input/output dimension details of fine-tuned architecture

Block	Name	Type	Kernel	Stride	Input dimension	Output dimension
	Input	Image Input	N/A	N/A	224x224	NA
Block1	conv1	Convolution	11x11	4	224x224x3	54x54x64
	pool1	Min Pooling	2x2	2	54x54x64	27x27x64
Block2	conv2	Convolution	5x5	1	27x27x64	27x27x256
	pool2	Min Pooling	2x2	2	27x27x256	13x13x256
Block3	conv3	Convolution	3x3	1	13x13x256	13x13x256
Block4	conv4	Convolution	3x3	1	13x13x256	13x13x256
Block5	conv5	Convolution	3x3	1	13x13x256	13x13x256
	pool5	Min Pooling	2x2	2	13x13x256	6x6x256
Block6	Bilstm	BiLSTM	6x6	1	6x6x256	36x36x256
		Flatten	N/A	N/A	36x36x256	1x1x4096
Block7	Fc1	Fully Connected	6x6	1	1x1x4096	1x1x4096
	Fc2	Fully Connected	1x1	1	1x1x4096	1x1x4096
	Fc3	Fully Connected	1x1	1	1x1x4096	1x1x1000

Table 7. Optimal values for various hyperparameters of convolution layers in the fine-tuned neural network model

Block name	Kernel Size Search Space	Optimal value for Kernel Size	Number of Kernels search space	Optimal value for number of Kernel
B1	11x11, 5x5, 9x9	11x11	32-96	64
B2	9x9,5x5, 3x3	5x5	64-384	256
B3	9x9,5x5, 3x3	3x3	64-384	256
B4	9x9,5x5, 3x3	3x3	64-384	256
B5	9x9,5x5, 3x3	3x3	64-384	256

4.2.2 Hyperparameters of BiLSTM.

In the developed method, the reduced feature set of size 6 x 256 obtained from a CNN has been converted into temporal sequence of feature values for feeding it to BiLSTM layer. The values are fed in 36 different timestamps. The BiLSTM layer with 256 hidden units followed by a drop out layer with a value of 0.5 was found to give the maximum accuracy. All the 256 dimensional feature vectors obtained from t_0 to t_{36} have been combined to generate a feature matrix of dimension 36 x 256. The matrix has been flattened to generate a single feature vector of dimension 4096, which is then fed into the FC layers. Table 8 presents the optimal values for various hyperparameters of BiLSTM.

Table 8. Optimal values for various hyperparameters of BiLSTM

Hyperparameters of BiLSTM	Search Space	Optimal value
Number of BiLSTM layers	1-3	1
Memory blocks	256-384	256
Dropout	0.1-0.5	0.5

The network was trained with an initial learning rate of 0.01 and Stochastic Gradient Descent with Momentum (SGDM) 0.9. During training the correctness of the network is tracked by defining the validation data and validation frequency. Here validation is done after every 30 iterations. Every epoch involved rearrangement of the data. Throughout training, the accuracy is calculated using the validation data and the training data. The training of our novel model got terminated after 720 iterations with 30 epochs. From each character image, 1000 features are extracted by the model, which are then used by the softmax layer to determine the most probable class for each character. After a block wise fine tuning, it is observed that a learning rate of 0.01 ensures proper convergence. In LRN the optimum value of α, β, k, n and N was found to be 10^{-4}, 0.75,2, 5 and 3 respectively. The optimum value of momentum was found to be 0.9.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

The experimental results obtained by the novel model are presented in this section. To check out the visual impact of our finding, the activation map of the novel tailored model is presented in Figure 8. The output of block 1 to block 5 with respect to figure 6 are shown by giving the Malayalam character ‘ക ‘as input. In figure 8, the output of activation of a channel in each block of the developed model is represented by a square in the montage.

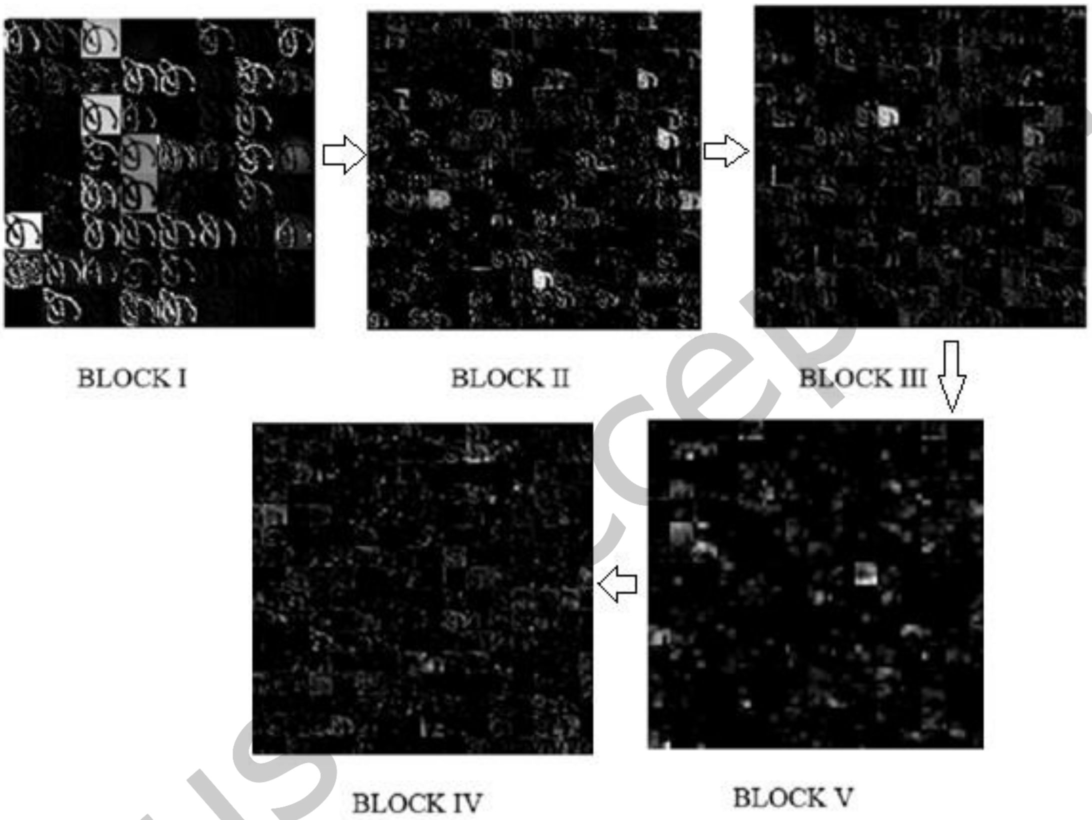


Fig.8. Visual outcome of the features obtained from each block of the newly developed model by giving the Malayalam Character ‘ക ‘

The performance metrics considered for evaluating the model are accuracy, error-rate, sensitivity and specificity computed using the formula below:

$$Accuracy = \frac{TP+TN}{(FN+FP+TP+TN)} \tag{11}$$

$$Error\ Rate = \frac{FP+FN}{(FN+FP+TP+TN)} \tag{12}$$

$$PPV = TP/TP + FP \tag{13}$$

$$NPV = TN/TN + FN \tag{14}$$

$$Sensitivity = \frac{TP}{(TP+FN)} \tag{15}$$

$$Specificity = \frac{TN}{(TN+FP)} \tag{16}$$

$$F\ Measure = \frac{(2*Precision*Recall)}{(Precision+Recall)} \quad (17)$$

True positives and true negatives are denoted by TP and TN and false positives and false negatives are denoted by FP and FN respectively.

In the experiment, the novel fine-tuned CNN model achieved an accuracy of 94.99% when used alone, with misclassification errors detailed in Table 1. So in order to improve the performance, the customized CNN was evaluated using various LSTM variants as discussed in section 3.2.1. The results are presented in Table 9.

Table 9. Accuracy of the proposed character recognition method against the various LSTM combinations

LSTM Variants	**Optimal value for the number of layers**	**Accuracy**
Classic LSTM	1	95.01%
Stacked LSTM	2	95.67%
BiLSTM	1	96.40%

As per the experiment conducted, the novel fine-tuned CNN model with BiLSTM was able to obtain the highest accuracy of 96.40% with an error rate of 3.86%, 99.3% NPV and 83.33% PPV. The acquired sensitivity, specificity and F-measure values are 79.55%, 99.45% and 88.39% respectively. The results are concluded in Table 10.

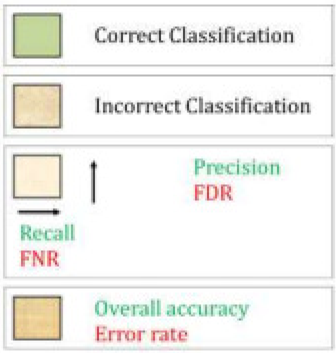
Table 10. Evaluation results obtained for the Novel model

Accuracy	**Error-Rate**	**NPV**	**PPV**	**Sensitivity**	**Specificity**	**F-measure**
96.40%	3.86%	99.3%	83.33%	79.55%	99.45%	88.39%

Confusion plot is another measure that can demonstrate how well the neural network has fit the data. The network uses the randomly divided data set - training set, validation set and test set in the ratio 70:15:15. In the test confusion matrix presented in Figure 10, the diagonal elements represent the count of correct predictions, while off-diagonal elements are those incorrectly labeled by the classifier. The greater diagonal values of the confusion matrix indicate higher prediction accuracy. Since we have a large number of character classes, the confusion matrix will be extensive. For facilitating visualization and analysis, we have categorized the characters into 5 classes: vowels and diacritics (class 1), consonants (class 2), chillus (class 3), common consonant ligatures (class 4), and consonant diacritics and other symbols (class 5). Figure 9 provides a description of the color code and other details regarding the confusion matrix of the developed model presented in Figure 10.

In Figure 10, the green squares on the diagonal represent correct classifications, while light brown squares indicate incorrect classifications. The values in the light brown squares should be low, suggesting few misclassifications if the network has learned to classify correctly. Precision and false discovery rates are displayed in the squares on the plot's far right. Recall and FNR are displayed in the squares at the bottom of the matrix. The overall accuracy is displayed in the dark brown cell at the lower right corner. From the confusion matrix in figure 10, we observed an overall accuracy of 96.40%. The precision for classes 1 to 5 was 97.9, 96.2, 100, 98.2, and 91.1, respectively, and the false rate for classes 1 to 5 was 2.1, 3.8, 0, 1.8, and 8.9, respectively. Recall values obtained for classes 1 to 5 were 98.9, 98.1, 100, 100, and 73.3, and FNR obtained for classes 1 to 5 were 1.1, 1.9, 0, 0, and 22.7, respectively.

The ROC and AUC of the novel model is shown in figure 11 and the class wise (as per Figure 9.c) ROC curves of fine- tuned model is depicted in Figure 12.



a) Colour code description of confusion matrix

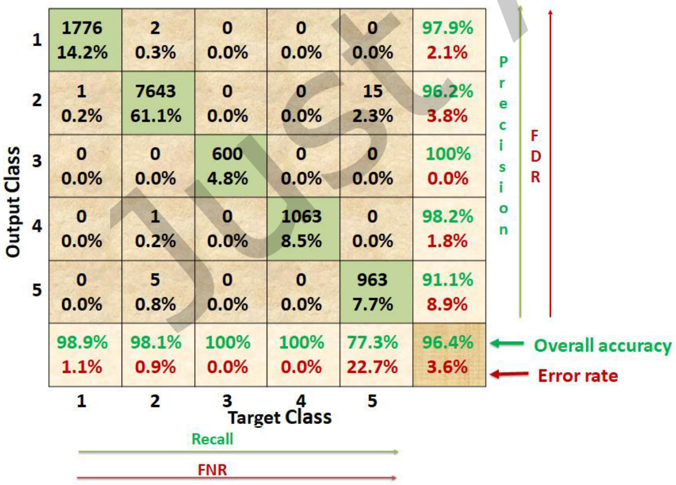
X Axis	Target Class
Y Axis	Output Class

b) X and Y axis description of confusion matrix

1	Class 1	Vowels and diacrities
2	Class 2	Consonants
3	Class 3	Chillus
4	Class 4	Common Consonants and ligatures
5	Class 5	Consonant diacrities and other symbols

c) Description of classes shown in confusion matrix

Fig. 9. Color code, axis and class description of confusion matrix shown in Fig. 10



X Axis	Target Class
Y Axis	Output Class

Fig.10. Confusion Matrix of the proposed fine-tuned model

The mini-batch loss and accuracy along with the validation loss and accuracy are obtained from the training progress graph. This training graph makes it easier to understand how the training is going. We can assess the correctness of the network, its rate of improvement, and whether it is beginning to over fit the training set of data. When the network's accuracy reaches a plateau and the accuracy is no longer increasing, the training is terminated. It returns the trained network together with its validation accuracy once training is over. The model obtained a validation accuracy of 96.8%

after 30 epochs. The convergence graph of accuracy and loss function of the novel model is shown in Figure 13 and 14 respectively.

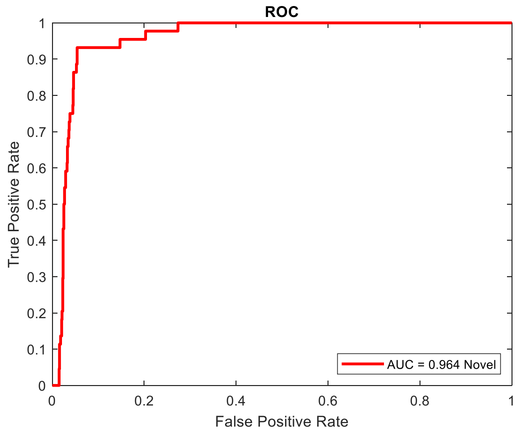


Fig.11. ROC and AUC of Proposed Neural Mode

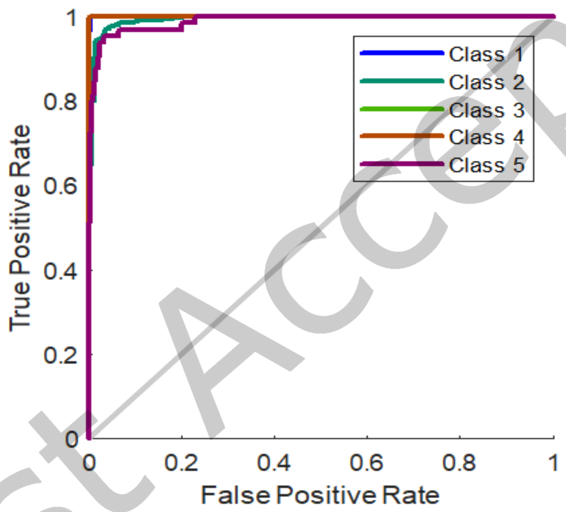


Fig.12. Category wise Receiver Operating Characteristic Plot of Proposed Neural Network Model

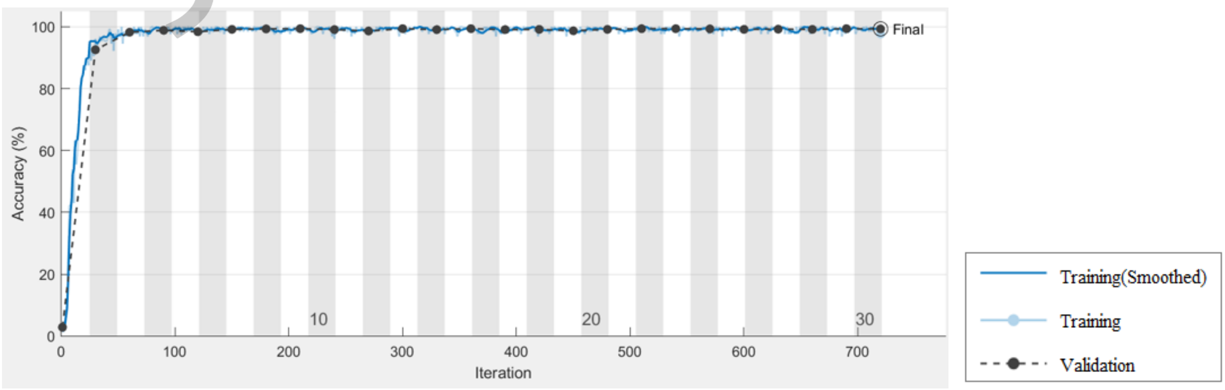


Fig.13. Convergence graph of accuracy of the proposed NN model

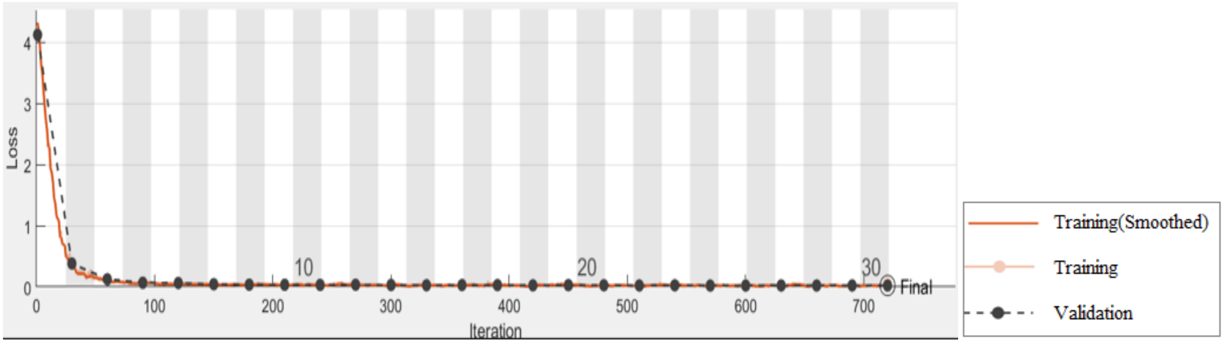


Fig.14. Convergence graph of loss function using proposed NN model

Accuracy of the proposed character recognition model during training, validation, and testing is presented in Table 11.

Table 11. Accuracy of the proposed character recognition model during training, validation, and testing

Phase	Accuracy
Training	98.2%
Validation	96.8%
Testing	96.4%

6 PERFORMANCE ANALYSIS

Analysis of the performance of the novel architecture involves comparing the results with established standard CNN architectures such as Xception, Inception, Googlenet, Alexnet, Resnet, LeNet, and VGG. The results of the developed model, juxtaposed with those of existing pre-trained models. The results are presented in Table 12. Additionally, the accuracy obtained by the proposed CNN before the inclusion of BiLSTM is also included in the table.

Table 12. Proposed model comparison with existing pre-trained models

Architecture	Accuracy	Error Rate	Sensitivity	Specificity	PPV	NPV	FM
Xception	0.8939	0.1061	0.6272	0.9690	0.7163	0.9811	0.7615
InceptionV3	0.8957	0.1043	0.7064	0.9581	0.6959	0.9805	0.8132
Googlenet	0.9015	0.0985	0.6591	0.9906	0.7073	0.9883	0.7915
Alexnet	0.9303	0.0697	0.7273	0.9890	0.6957	0.9906	0.8381
Resnet50	0.9326	0.0674	0.6364	0.9937	0.7778	0.9875	0.7758
LeNet	0.9379	0.0621	0.7045	0.9953	0.8378	0.9899	0.8250
VGG16	0.9317	0.0683	0.7045	0.9945	0.8158	0.9899	0.8247
Proposed CNN without BiLSTM	0.9499	0.0501	0.5455	0.9929	0.7273	0.9845	0.7041
Novel	0.9640	0.0386	0.7955	0.9945	0.8333	0.9930	0.8839

The comparison graph of the ROC curve obtained for the proposed model and the current existing models which were able to achieve accuracy more than 90% is shown in Figure 15.The respective AUC are also marked in figure. From the plot it is clear that compared to the all existing models, the AUC of the proposed model is more close to one.

Table 13 compares the number of layers, training time, and average recognition time per character for the novel and existing models. Our model achieves better accuracy in less time, with an average recognition time of 0.0417 seconds and a training time of 4 minutes 2 seconds.

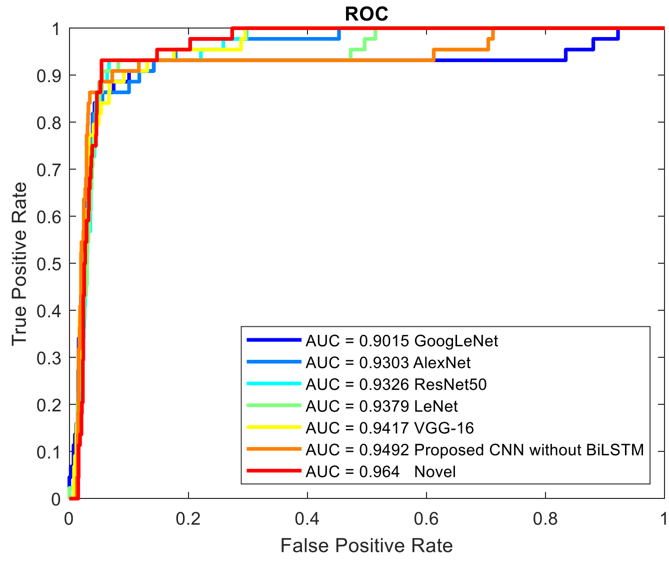


Fig.15. Comparison graph of ROC curve and AUC obtained for the proposed model and existing models

Table 13. Comparison of the number of layers and time taken for model training and character recognition

Architecture	Number of layers	Time taken for recognizing a single character(in secs)	Time taken for training the model(in minutes)
Xception	71	0.1998	15.61
InceptionV3	48	0.1591	12.58
Alexnet	8	0.0449	5.7
Googlenet	22	0.1148	11.39
Resnet50	50	0.1678	13.48
LeNet	101	0.2215	18.66
Resnet152	152	0.3109	27.9
VGG16	16	0.3981	29.57
CNN without BiLSTM	9	0.3976	7.06
Novel Neural Network Model	**10**	**0.0417**	**4.2**

7 VALIDATION WITH THE STATE-OF-THE-ART METHODS

Table 14 summarizes the limited research on automatic character recognition from Malayalam palm scripts, detailing the techniques used, datasets utilized, accuracies achieved, and the scope of the existing works as reported in the published paper. The results obtained by the existing algorithms on the dataset created for the present experiment are also included in the table. Compared to the scope of the existing methods the novel method was able to correctly recognize all 128 Malayalam characters written in PLMs.

Even though the writing instruments and styles differ in PLMs, the character recognition process is the same as in Malayalam handwritten documents. Due to a lack of major research on palmlet recognition, we tested existing handwritten recognition methods on a self-generated palm leaf dataset. Results are in Table 15.

Table 15 shows that existing methods for Malayalam handwritten character recognition are inadequate for palm scripts and our model was able to give remarkable accuracy when compared to all these existing models. The achieved results confirm our model's superior accuracy, lowest error rate, and high sensitivity and specificity compared to state-of-the-art methods.

Table 14. Summary of the state-of-the-art techniques

Method	Dataset	Feature	Classifier	Accuracy	Scope of the work	Accuracy obtained on the dataset created for the present experiment
Sudarsan, D., Sankar, D [17]	Self -generated Dataset with 41600 images	Log Gabor and uniform rotational invariant LBP	SVM	95.57%	Accuracy level not adequate	95.01%
Sudarsan D *et al.* [15]	Self -generated Dataset ,number of images not mentioned	Convolutional features	Neural Network	96.7%	Able to correctly recognize only very few Malayalam characters written in PLMs	66.7%
Sruthy S Kumar [16]	Self -generated Dataset with 21,500 images	Convolutional features	Neural Network	95.30%	Able to correctly recognize only 43 Malayalam characters written in PLMs	61.98%
Proposed NN Model	**Self -generated Dataset 83,400 with images**	**Hybrid DL based**	**CNN-BiLSTM**	**96.40%**	**Able to correctly recognize all 128 Malayalam characters written in PLMs**	**96.40%**

Table 15. Comparison with Malayalam Handwritten character recognition methods

Author	Features	Classifier	Accuracy
Sreeraj *et al*. [2010]	Time domain features, writing direction, curvature	KNN	88.19%
Chacko and Anto [2011]	Division point features	SLFN	86.14%
Premkumar *et al*.[2011]	Time domain features, Wavelet transform	SFAM	90.71%
Indhu *et al.*[2012]	Structural, directional	SFAM	89.16%
Primkumar *et al.* [2013]	Time domain features, Angular features	SVM	91.857%
Proposed	**CNN with BiLSTM**	**Softmax**	**96.40%**

Moreover, to evaluate the overall performance, the novel model developed was tested on the Kaggle Malayalam benchmark dataset of 17,388 handwritten characters, achieving an accuracy of 87.12%. This is reasonable as the model is tailored for Malayalam character recognition from PLMs.

8 ERROR ANALYSIS

The system's overall precision relies on the category of palm leaf image received, as the effectiveness of noise removal and segmentation is contingent on this factor. Misclassification errors of the developed model are presented in Table 16. These errors occur because the features extracted from the model gives equal class probabilities, leading to a tie and toss approach, resulting in 50% incorrect predictions.

Based on the features of these characters, it is even difficult for a human to properly distinguish between these characters. In such cases we need to depend on language models and other heuristics based algorithms to suggest corrections for misspelled words which can be considered as a future scope of this research.

Table 16. Misclassification errors

Character	Actual class	Predicted class
	ഭ	ദ
	ൺ	ൻ
	മ	ച
	ന	ത
	ച	പ
	ഖ	വ

9 CONCLUSIONS AND FUTURE SCOPE

In this research, we developed a framework for automating the digitization of Malayalam PLMs using an ensemble model combining fine-tuned CNN and BiLSTM. This innovative strategy can serve as an essential tool for doctors, researchers, historians, astrologers, and archaeologists worldwide. Amid the global challenges in controlling the spread of new viruses, the digitization of Malayalam PLMs, which document various Ayurvedic formulations, remains crucial. Our literature survey reveals a significant gap in this field, highlighting the need for further research. Therefore, our work will significantly contribute to preserving and transmitting this valuable knowledge accurately and consistently to future generations. The proposed fine-tuned model achieved a remarkable accuracy of 96.40%, surpassing all current best practices. The Malayalam script has evolved over centuries, and our research specifically focuses on PLMs from AD 1800 to 1908. This scope can be expanded in the future to include manuscripts from all periods. Additionally, our dataset currently excludes Malayalam numerals, digits, and fractions. Future research can incorporate these elements, enhancing the scope and applicability of our findings.

REFERENCES

[1] Prasanthkumar P V, Midhun T P, Archana Kurian. 2015. A survey on Script and Language identification for handwritten document images. IOSR Journal of Computer Engineering 17, 2 (April 2015), 105-109. DOI: 10.9790/0661-1725105109

[2] Lekha Kumari. 2021. The origin of Malayalam Language-The Linguistic theories, International journal of creative research thoughts 9, 12 (December 2021), 253-259.

[3] Anitha Mary M.O Chacko, Dhanya P.M. 2014. Handwritten character recognition in Malayalam scripts – a review. Int. Journal of Artificial Intelligence & Applications 5, 1 (January 2014), 79-89. DOI : 10.5121/ijaia.2014.5106

[4] D. Udaya Kumar, G.V.Sreekumar, U. A. Athvankar.2009.Traditional writing system in Southern India-Palm leaf manuscript, Design Thoughts (IDC, IIT Bombay), vol. 2 (July 2009), 2-7.

[5] Contributions of Kerala.2015. NHP CC DC publication. Retrieved from https://www.nhp.gov.in/contributions-of-kerala_mtl

[6] V. Katsouros, V. Papavassiliou, F. Simistira and B. Gatos.2016.Recognition of Greek Polytonic on Historical Degraded Texts Using HMMs. 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece,11-14 April 2016, 346-351, DOI: 10.1109/DAS.2016.60.

[7] Munish Kumar, Simpel Rani Jindal, M. K. Jindal, Gurpreet Singh Lehal. 2018. Improved recognition results of medieval handwritten gurmukhi manuscripts using boosting and bagging methodologies. Neural Process Lett 50,1(September 2018), 43–56, DOI:10.1007/s11063-018-9913-6

[8] Kumar, Munish & Jindal, M. & Narang, Sonika. 2019. Devanagari ancient documents recognition using statistical feature extraction techniques. Sadhana 44, 6, 32 (June 2019) 1–8. DOI:10.1007/s12046-019-1126-9

[9] Sonika Rani Narang ,M. K. Jindal, Shruti Ahuja, Munish Kumar .2020. On the recognition of devanagari ancient handwritten characters using sift and gabor features. Soft Comput 24,22,(May 2020), 17279–17289, DOI: 10.1007/s00500-020-05018-z

[10] Suganya T, Murugavalli S.2020. A hybrid group search optimization: firefly algorithm-based big data framework for ancient script recognition. Soft Comput 24, 14 (May 2020)10933–10941.DOI: 10.1007/s00500-019-04596-x

[11] Avadesh M, Goyal N.2018. Optical character recognition for sanskrit using convolution neural networks. 13th IAPR international workshop on document analysis systems. Austria, Vienna 24-27 April 2018 447–452. DOI: 10.1109/DAS.2018.50.

[12] Narang, S.R., Kumar, M. & Jindal, M.K.2021. Deepnetdevanagari: a deep learning model for devanagari ancient character recognition. Multimed Tools Appl 80, 13

(March 2021) 20671–20686. DOI:10.1007/s11042-021-10775-6

[13] Silvia Cascianelli, Marcella Cornia,Lorenzo Baraldi, Rita Cucchiara.2022. Boosting modern and historical handwritten text recognition with deformable convolutions. Int J Doc Anal Recogn 25,(August 2021) 1–11 DOI:10.1007/s10032-022-00401-y

[14] K. Markou, L. Tsochatzidis, K. Zagoris, A. Papazoglou, X. Karagiannis, S. Symeonidis & I. Pratikakis .2021.A convolutional recurrent neural network for the handwritten text recognition of historical greek manuscripts. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science LNIP, volume 12667, Springer, Cham 10-15 January 2021, 249–262 DOI:10.1007/978-3-030-68787-8_18

[15] Dhanya Sudarsan; Parvathy Vijayakumar; Sharon Biju; Soniya Sanu; Sreelakshmi K. Shivadas. 2018. Digitalization of Malayalam Palmleaf Manuscripts Based on Contrast-Based Adaptive Binarization and Convolutional Neural Networks, IEEE Int. Conf. Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22-24 March 2018 1-4 DOI: 10.1109/WiSPNET.2018.8538588.

[16] Sruthy S Kumar.2022. Character Recognition of Malayalam Palm leaf manuscripts Using Convolutional Neural Networks, Int.Journal of Scientific Research in Engineering and Management (IJSREM)

[17] Sudarsan, D., Sankar, D. 2024. Development of an effective character segmentation and efficient feature extraction technique for malayalam character recognition from Palm leaf manuscripts. Sādhanā 48, 156 (August 2023) DOI:10.1007/s12046-023-02181-5

[18] Sreeraj M, Sumam Mary Idicula.2010. K-NN based On-Line Handwritten Character recognition system, First Int.Conference on Integrated Intelligent Computing, First International Conference on Integrated Intelligent Computing, Bangalore, India, 5-7 August 2010, 171-176 DOI: 10.1109/ICIIC.2010.58.

[19] Binu P. Chacko, Babu Anto P .2011. Online Sequential Extreme Learning Machine Based Handwritten Character Recognition, Proceeding of the 2011 IEEE Students' Technology Symposium 14-16 January 2011 142-147, DOI: 10.1109/TECHSYM.2011.5783843

[20] Primekumar K.P,Sumam Mary Idiculla.2010. On-line Malayalam Handwritten Character Recognition using Wavelet transform and SFAM, 3rd Int. Conference on Electronics Computer Technology(ICECT), Kanyakumari, India, 8-10 April 2011 49-53, DOI:10.1109/ICECTECH.2011.5941558.

[21] Primekumar K.P,Sumam Mary Idiculla.2013. On-line Malayalam Handwritten Character Recognition using HMM and SVM, Int. Conference on Signal Processing, Image Processing and Pattern Recognition(ICSIPR) Coimbatore, India 7-8 February. 2013 322-326, DOI: 10.1109/ICSIPR.2013.6497991

[22] P. V. Raveena, A. James and C. Saravanan.2017. Extended zone based handwritten Malayalam character recognition using structural features, 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), Coimbatore, India, 2017, 1-5, February 2017 22-24 DOI: 10.1109/ICECCT.2017.8117898.

[23] Dhanya Sudarsan,Shelbi Joseph.2018. A Novel Approach for Handwriting Recognition in Malayalam Manuscripts using Contour Detection and Convolutional Neural Nets , IEEE Int. Conf. Advances in Computing, Communications and Informatics (ICACCI) Bangalore, India , 19-22 September 2018 1818-1824 DOI: 10.1109/ICACCI.2018.8554592.

[24] Meenu Alexa, Smija Das.2016. An approach towards Malayalam handwriting recognition using dissimilar classifiers, Elsevier Global Colloquium on Recent Advancements and Effectual Researches in Engineering, Science and Technology 25, (September 2016)224-231 DOI: 10.1016/j.protcy.2016.08.101

[25] Pranav P. Nair; Ajay James; C. Saravanan.2017. Malayalam handwritten character recognition using convolutional neural network , International Conference on Inventive Communication and Computational Technologies (ICICCT) , Coimbatore, India 10-11 March 2017 278-281 DOI: 10.1109/ICICCT.2017.7975203

[26] N. P. T. Kishna and S. Francis.2017.Intelligent tool for Malayalam cursive handwritten character recognition using artificial neural network and Hidden Markov Model, International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23-24 Nov. 2017, 595-598 DOI: 10.1109/ICICI.2017.8365201.

[27] Manjusha, K., Anand Kumar, M. & Soman, K.P.2017. Reduced Scattering Representation for Malayalam Character Recognition. Arab J Sci Eng 43, (November 2017) 4315–4326 DOI:10.1007/s13369-017-2945-9

[28] A James, PV Raveena, C Saravanan .2018. Handwritten Malayalam Character Recognition using Regional Zoning and Structural Features. Int. Journal of Engineering & Technology.7,4 (2018) 4629-36 DOI:10.14419/ijet.v7i4.12551

[29] F. Simistira, A. Ul-Hassan, V. Papavassiliou, B. Gatos, V. Katsouros and M. Liwicki.2015.Recognition of historical Greek polytonic scripts using LSTM networks, 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23-26 August 2015 766-770 DOI: 10.1109/ICDAR.2015.7333865.

[30] M. R. Yousefi, M. R. Soheili, T. M. Breuel, E. Kabir and D. Stricker.2015. Binarization-free OCR for historical documents using LSTM networks, 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23-26 August 2015, 1121-1125 DOI:10.1109/ICDAR.2015.7333935.

[31] Dhanya Sudarsan, Deepa Sankar.2022. A Novel Complete Denoising Solution for Old Malayalam Palm leaf manuscripts. Pattern Recognit. Image Anal. 32 (March 2022) 187–204 DOI:10.1134/S1054661822010096

[32] Dhanya Sudarsan, Deepa Sankar.2024.Enhancing Malayalam Palm Leaf Character Segmentation: An Improved Simplified Approach. SN Computer science ,5,577 (May 2024) DOI:10.1007/s42979-024-02848-8

[33] Amar Jindal,Rajib Ghosh.2023. A hybrid deep learning model to recognize handwritten characters in ancient documents in Devanagari and Maithili scripts. Multimedia Tools and Applications 83 (June 2023) 8389–8412 DOI: 10.1007/s11042-023-15826-8

[34] An-Feng Hu,Sen-Lin Xie,Tang Li, Zhi-Rong Xiao, Yuan Chen, Yi-Yang Chen .2023.Soil parameter inversion modeling using deep learning algorithms and its application to settlement prediction: a comparative study Acta Geotechnica 18 (June 2023) 5597–5618 DOI: 10.1007/s11440-023-01935-z

[35] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton.2012. ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems, Communications of the ACM, Volume 60, Issue 6 (May 2017) 84 – 90 DOI:10.1145/3065386

[36] Mathworks.2011. Matlab Retrieved from https://in.mathworks.com/matlabcentral/answers/24399-how-to-find-euclidean-distance-for-an-image. Accessed 26 June 2020

[37] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Md. Mostofa Ali Patwary, Prabhat, Ryan P. Adams.2015. Scalable bayesian optimization using deep neural networks. 32nd International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37 (July 2015)2171–2180 DOI: 10.48550/arXiv.1502.05700