

# Repossession and recognition system: transliteration of antique Tamil Brahmi typescript

S. Brindha\* and S. Bhuvaneshwari

Department of Computer Science and Engineering, Easwari Engineering College, Chennai 600 089, India

Tamil is among the ancient languages in the world with a rich literature. Recognition of antique Tamil scripts is difficult and different from the present form of the language. The character recognition of Brahmi script poses a big challenge even today. In this paper, a new technique for extracting the features is proposed, and converting the ancient Tamil script into the present form. Initially, the system is implemented by performing the pre-processing steps. Then the characters are individually separated using the segmentation process. The processed image undergoes a new feature extraction technique, where the system applies a chi-square test to check whether all the zoning feature values of the image are independent or dependent. The characters are recognized from the extracted features using neural networks. NNTool is employed to train the featured image and the data are compared with the database to recognize the Brahmi characters. The feature extraction technique along with the neural network achieved recognition rate accuracy of 91.3% and error rate of 8.7% using the confusion matrix. Our experiment has been simulated using MATLAB.

**Keywords:** Ancient script, chi-square test, confusion matrix, feature extraction technique, neural networks.

THE Brahmi characters are ancient southern form of the Tamil script commonly used during the Asoka dynasty from 269 to 239 BC. The Tamil script used during ancient fictional works like *Tholkappiyam*, *Kambaramayanam* and *Thirukkural* is diverse. The Tamil alphabet consists of 12 vowels, 18 consonants, a combination of vowels and consonant numbering 216 and one Ayutha letter. Around 77 million people around the world speak Tamil, with 68 million living in Tami Nadu, India. It is one of the official languages in India, Sri Lanka and Singapore. Ancient Tamil scripts are getting degraded, and not many technologies are available to prevent their degradation. They are stolen or lost over time, so the existing documents have to be processed and digitized for preservation. Historical documents usually pose much squalor,

due to weather conditions, preservation and handling methods. The Brahmi script is based on geometrical forms, so it is difficult to recognize handwritten characters for any language. Tamil characters contain small circles or loops, which are difficult to acknowledge. The proposed system uses machine learning algorithms for recognition and conversion of the ancient characters with the help of image processing techniques.

Image processing is a method to extract some useful information from the image. The digital image is saved on the computer in the form of a JPEG file. Character recognition and digitization are done only for offline characters. The image processing techniques involved are preprocessing, segmentation, feature extraction, recognition and character conversion. The present study introduces a new feature extraction process for a higher recognition rate accuracy.

## Allied works

Selvakumar and Ganesh<sup>1</sup> introduced the actualized canny edge detection algorithm for examining and also removing portions from a corrupted image. Initially, the image is preprocessed, and then canny edge detection algorithm with an artificial neural network is used to focus content edge pixels. Mahalakshmi and Sharavanan<sup>2</sup> outlined the recognition and translation of ancient Tamil scripts to the present form using the process of segmentation and contourlet transform. The main scope of the present study is to segment the ancient inscriptions and translate them into alphabets used at present by the implementation of segmentation algorithms like PSO (particle swarm optimization), DPSO (Darwinian PSO), FO-DPSO (fractional-order DPSO), and image enhancement module. MATLAB is used for the segmentation process, whereas the M-file is called in LABVIEW to have the recognition and translation work done.

Rajakumar and Subbiah Bharathi<sup>3</sup> used contourlet based method to recognize ancient Tamil scripts. The 2D wavelet and contourlet-based methods were compared and higher accuracy produced. The neural network has been traditionally used to refer to a network or circuit of biological neurons. The neural network requires data for

\*For correspondence. (e-mail: brindhasaikiran@gmail.com)

accurate training. Vellingiriraj and Balasubramanie<sup>4</sup> proposed a simple method for converting ancient Tamil handwritten scripts into text format. The objective of this paper is to convert the palm manuscript image into digitized text format using a Boolean matrix and BFS graph concepts. Janani *et al.*<sup>5</sup> focused to enhance the image quality by removing the noise using Wiener and median filters. Then morphological operations like dilation and erosion are used, followed by pixel segmentation for segmenting the character. Finally, the correlation matching techniques are connected and matched with the present Tamil language, where correlation matching is done based on pixel size.

Subramani and Murugavalli<sup>6</sup> proposed a method for retrieving images from degraded palm leaves. Since the letters are degraded, the image resolution is enhanced using Otsu method and mean shift algorithm. To improve image quality binarization is done, for which image post-processing trimmed median filter process is used. Banumathi and Nasira<sup>7</sup> used SOM technique and artificial neural network for recognizing ancient Tamil characters. Jagadeesh *et al.*<sup>8</sup> proposed a combination of the time-domain and frequency-domain feature. The input images are scanned and then made to undergo preprocessing stages (binarization, noise removal, skew correction). The preprocessed image is segmented into lines, words, characters and later the segmented image features are extracted along with HMM (hidden Markova model) algorithm applied for training and recognizing the characters.

Kavitha *et al.*<sup>9</sup> suggested two main approaches for increasing the classification accuracy rate, namely, skewness-based approach (SA) for Indus document classification and nearest neighbour-based approach (NNA) for classifying English from South Indian scripts. SA reveals that skewness between the components in the Indus document image is higher than that between the components in English and South Indian documents. NNA recognizes the presence or absence of modifiers which are general in South Indian document images and are not present in English document images.

Subashini and Kodikara<sup>10</sup> reported that all preprocessed images are differentiated by a set of local SIFT feature vectors. The codebook for each character is created with the help of the *K*-means clustering algorithm, which is an optimization algorithm. However, this algorithm takes a long time to converge. It constructs an initial codebook using the Linde Buzo and Gray (LBG) algorithm, so that the union time for *K*-means is reduced considerably. Target character is recognized into one of 20 categories by *K*-nearest neighbour classification. The character level has been achieved in these experiments using 6000 training and 2000 testing images of 20 selected characters.

Pirilo and Impedovo<sup>11</sup> used zoning techniques for recognizing the images to overcome the problem of

membership function initially. The static and dynamic zoning methods, and the Voronoi-based zoning topology have been used for optimal zoning. Then the adaptive zone-based function is generated. A genetic algorithm is used to determine the optimization process. Finally, benchmark databases are generated for numerals and alphabets. Boufenar *et al.*<sup>12</sup> proposed an artificial immune system which is a supervised learning technique to overcome the complex classification problems. This technique is mostly used for isolated handwritten Arabic characters. The original database is developed with 5600 characters and is known as the IFN/ENIT benchmark. Then the new database is cropped and resized to  $128 \times 128$ . The optimization of parameters with the cross-validation grid-search method increases the classification accuracy to 93.25%. A tool called Scikit is used for predictive data analysis and mining.

Rajakumar and Subbiah Bharathi<sup>13</sup> discussed about the non-novel uniform slant correction. Initially the slant correction is applied to the text and recognition rate is compared with the other slant correction techniques, and the impact of slant correction for recognition is studied. In normalization, the character is normalized with related to skew slant and baseline. Then, the vertical angle is corrected and the writing is transformed to normal position, and height is adjusted to assume the width and height of the image. Offline recognition hidden Markova method is used along with the Viterbi algorithm. Bhuvanewari and Subbiah Bharathi<sup>14</sup> developed an acquisition system for digitizing ancient stone writing. It produces the silhouette for the input image which enhances the image quality for further processing. The photometric stereo system duo image gives superior accuracy, similar to the usual photographic image.

Medhi and Kalita<sup>15</sup> proposed a model called feed-forward neural network to recognize Assamese characters. Lines and characters are segmented with the help of a segmentation algorithm followed by the zoning feature extraction. Future focus is to extend the diagonal, slice zoning feature and also to implement fuzzy or neuro-fuzzy to increase the recognition rate. Vellingiriraj *et al.*<sup>16</sup> proposed an algorithm called zoning for extracting the feature of ancient Tamil characters. The image is decomposed into  $64 \times 64$  matrices that help recognize the characters with the help of Unicode. Then the ancient script is converted into the present form. Kale *et al.*<sup>17</sup> used the Zernike moment feature to recognize Devanagari compound character and also suggested SVM and *k*-NN based classifications. Twenty thousand character images are preprocessed, normalized and decomposed into  $30 \times 30$  pixel zones. The Zernike moments are executed for every zone of the images.

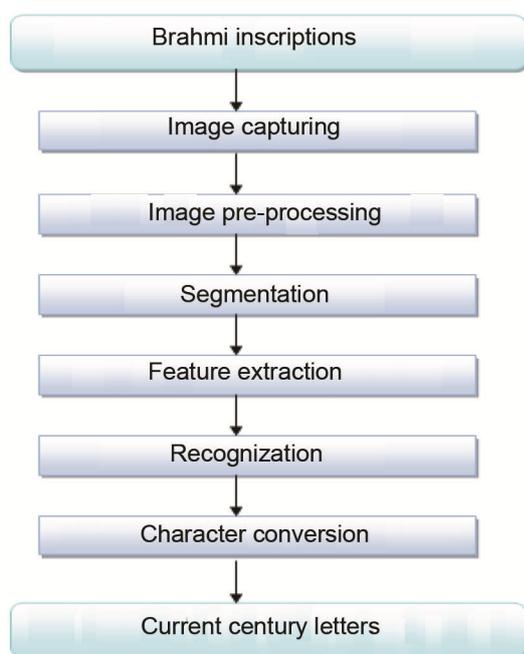
On the basis of literature survey, the recognition rate is the key for the proposed work. So, a combination of Zernike and zoning with neural network is proposed for improving the overall accuracy of the system.

**Proposed methodology**

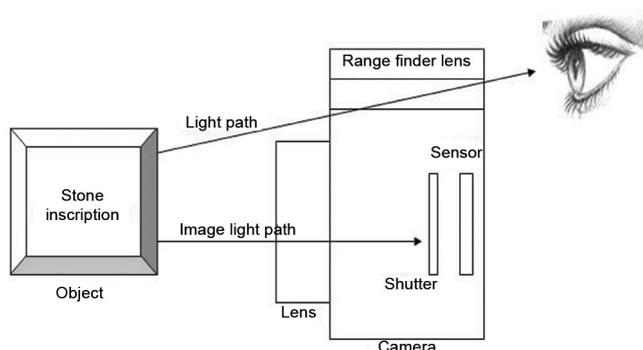
The proposed model explains the complete process of recognizing and converting the Brahmi Tamil characters into the present form of Tamil characters<sup>18</sup>. Figure 1 shows the required framework, viz. image capturing and image pre-processing, segmentation, feature extraction, character recognition and conversion.

*Image capturing*

Image capturing is the process of taking a digitized image from a camera using a vision sensor. Figure 2 explains the process of capturing the image. Initially, the Brahmi lexis images are captured with high definition cameras from various places and stored in the form of JPEG images.



**Figure 1.** Proposed system architecture.



**Figure 2.** Image capturing.

*Image pre-processing*

Image preprocessing is a method where the image abstraction is increased from a lower to a higher level using some basic preprocessing techniques. The abstraction of an image can be increased by removing noise and small unclear pixels of the image. Figure 3 shows the preprocessing method used in the proposed model.

*Image resizing:* The original image with noise of  $32 \times 32$ -sized images is converted to a  $64 \times 64$ -sized image to increase the image pixel density<sup>16</sup>. The original image is less dense for preprocessing and so the images are resized and used for the grey scale image process.

*Grey-scale image:* The second step is to change the resized image into a grey scale image<sup>19</sup>. Where the grey scale image colour lies between black and white. In the proposed system, the value of each pixel is represented as a single frequency and the intensity of grey scale is stored as an 8-bit integer. Figure 4 a shows a grey-scale image.

*Image smoothening:* Image smoothening is used to reduce the noise in the input image. The smoothening process is done using a low-pass filter called Gaussian filter, which is used for noise removal. Figure 4 b shows the result of image smoothening.

The Gaussian filter is used to smoothen the image by increasing the sigma value. Based on the process described below the system undergoes filtering function.

Step 1: Define the Gaussian kernel function using the formula:

$$\frac{1}{2\pi\sigma^2} \exp(-(x_1^2 + x_2^2)/2\sigma^2),$$

where  $\sigma$  is the standard deviation and  $(x_1, x_2)$  is the kernel size.

Step 2: Gaussian kernel function value is estimated using the above formula.

Step 3: Convolve the kernel and local region of the image.

Step 4: Add the selected part of the kernel values in the vector. The added value is called as Gaussian filter value.

Step 5: Repeat steps 1 to 5 for all regions of the image.



**Figure 3.** Image preprocessing.

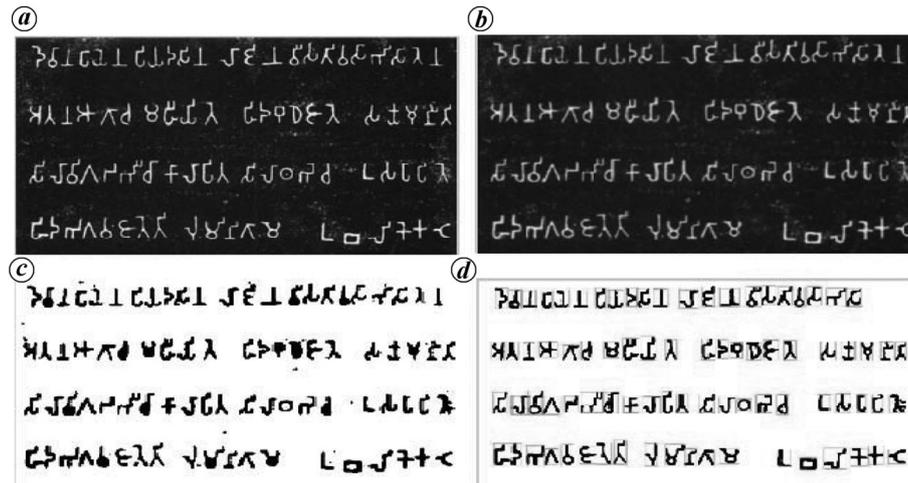


Figure 4. Noise-removing process: (a) grey-scale image, (b) smoothed image, (c) binary image and (d) binary image after segmentation.

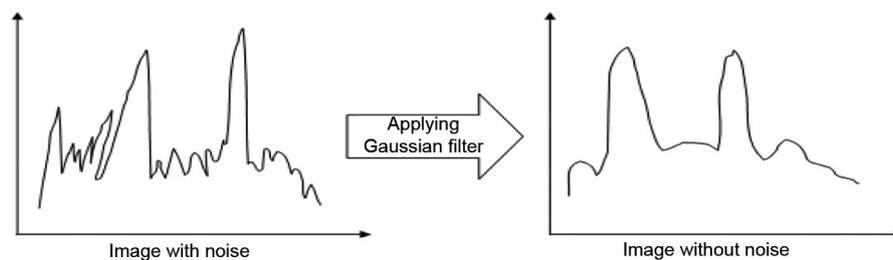


Figure 5. Gaussian filtering.

Step 6: Finally, the values are sorted and then applied to the image for noise removal.

The grey-scale image with noise is taken as the input and by applying Gaussian filter the noise from the image is removed. This image is taken as the output (Figure 5). In this way, the Gaussian smoothing filter is used to remove noise from the inscription data.

**Image binarization:** The smoothed image is converted into binary image, an effective way of converting the grey-scaled image into a binary image. Here the smoothed gray-scale image with value 0 represents black colour and value 1 represents white colour. In the proposed system Otsu image thresholding algorithm is used, which includes histogram and multi-level process that exhibit relatively high performance. Figure 4c shows the threshold values and variance of grey scale smoothed image.

#### Image segmentation

The binarized image is difficult to process, so the larger image is segmented for feature extraction to recognize the characters individually<sup>20</sup>. The image uses line and charac-

ter segmentation. The line segmentation extracts lines from the documents horizontally and character segmentation extracts individual characters from the line segmented image (Figure 4d). The bounding box analysis is made to test the adjacency relationship between the components of black pixel and the extracted component objects.

Figure 6 shows the process of segmentation where the binary image is given as the input and then that image is decomposed into an individual character which is labelled into an object. The labelled object is the output of the segmentation and input for edge detection.

#### Edge detection

In the proposed system the segmented image is dilated using image dilation to enlarge the boundaries of foreground pixels and the finite holes in the image are filled using an image-filling method. Then, the edges are detected using canny edge detector<sup>21</sup>. The following steps were followed for the edge detection.

Step 1: Input images of the filtered binary image taken as  $p(x_1, x_2)$ .

Step 2: Determine the gradients  $G_x$  and  $G_y$  of the image by marking its edges  $G = \sqrt{G_x^2 + G_y^2}$  in  $x$  and  $y$  directions.

Step 3: Suppressing local maxima should be marked as edges  $\theta$ .

Step 4: Perform a double threshold function to determine the potential edges using differential operations

$$\frac{\partial p(x_1, x_2)}{\partial x_1} \text{ and } \frac{\partial p(x_1, x_2)}{\partial x_2}.$$

Step 5: Final edges  $d(x_1, x_2)$  are evaluated by suppressing all the edges that are tracked by the hysteresis threshold.

*Algorithm*

Alg CannyED ( $E_i$  Region of image,  $n$  Size, F Filter, ED Edge Detection).

Divide the image into four regions as  $E_i$  where  $i = 0, 1, 2 \dots \infty$ .

```

For each region do,
    if  $n(E_i) < \text{base case}$ 
        then filter  $E_i$  and perform ED
    else call ED( $E_i$ )
    end if
end for
sum all the region
display joined image
end alg CannyED.
    
```

The above algorithm defines the function of edge detection for Brahmi character recognition. The edge detector detects the image, and the whole image is divided into four regions and for every region, the base case function is provided. Finally summing up all the regions, the edge detected image is displayed.

*Dataset creation*

A new dataset is developed for analysis called ‘Brahmi-CharSet’. Figure 7 shows the dataset collected from

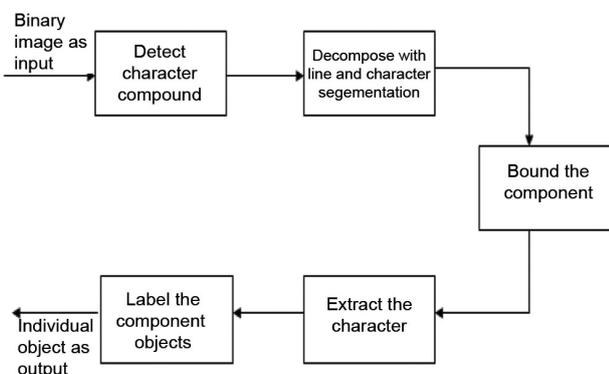


Figure 6. Stages of the segmentation process.

various handwritten writers for training the network. Figure 7a represents the character dataset ‘ $\text{అ}$ ’ and Figure 7b represents the character dataset ‘ $\text{ప}$ ’. Similarly, datasets for all other characters are also developed in the same manner from different handwritten writers.

*Feature extraction*

Feature extraction<sup>22,23</sup> extracts the detailed information from the input image. Ancient Tamil characters cannot be recognized accurately due to slant, skew, broken edge and loop obtained in the inscription data. So, it is essential to extract the character features<sup>24</sup>. To overcome these problems, this study implements a new feature extraction method called Zernike moment and zoning features along with the chi-square test for goodness of fit. Zernike moment describes the function of the unit circle and analyses the shape of the object. The zoning feature method is used to extract geometric features of an object. The chi-square test checks whether the feature vector values are independent are not.

*Zernike moment feature:* Zernike moment functions for mapping the digital images onto a set of 2D complex Zernike polynomials. Due to the linear or circular shape of the object, features could not be recognized accurately. So, the Zernike feature was computed in this study to identify ancient Tamil characters more accurately. The formula for deriving Zernike moment is as follows<sup>25</sup>; <https://www.slideshare.net/SandeepKumar622/zernike-moments>

$$V_{nm}(\rho, \theta) = R_{nm}(\rho)e^{jm\theta}.$$

The above equation is used to define a set of complex polynomials  $V_{nm}(\rho, \theta)$ , where  $j = \sqrt{-1}$ ,  $\theta = 1/x \tan y$ ,  $\theta$  represents the angle between  $\rho$  and the  $x$ -axis in a clockwise direction,  $\rho$  represents the length of vector from the origin to the pixel  $(x, y)$ .  $x$  and  $y$  are the pixel values

$$R_{nm}(\rho) = \begin{cases} \sum_{i=0}^{(n-m)/2} \frac{(-1)^i (n-m)!}{i! \left[ \frac{1}{2}(n+m)-i \right]!} \rho^{n-2i} & \text{for } n-m \text{ even,} \\ \left[ \frac{1}{2}(n-m)-i \right]! & \\ 0 & \text{for } n-m \text{ odd.} \end{cases}$$

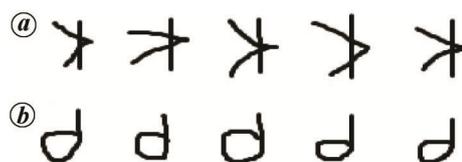


Figure 7. Dataset sample input.

$R_{nm}(\rho)$  denotes the radial polynomial notation, where  $n \geq 0$ ,  $m \leq n$  and  $n - m = \text{even}$ .  $n$  and  $m$  are the orders of the polynomials.

$$Z_{nm} = \frac{m+1}{\pi} \iint_{x,y} f(x,y) [V_{nm}(x,y)]^* dx dy.$$

The above statistics is used to compute the order of Zernike moments with repetition. The above formula derives the order number with value of  $n$  and  $m$

$$\hat{f}(r, \theta) = \sum_{m=0}^{N_{\max}} \sum_n Z_{nm} V_{nm}(r, \theta).$$

Function  $\hat{f}(r, \theta)$  is evaluated with double summation with respect to  $n$  and  $m$  values, and helps reconstruct the image. In the proposed system ancient character  $\text{க}$  is mapped over the unit disk to compute the set of complex polynomial values. The centroid of the image with the radius is taken. Then radial polynomial values and Zernike moments of the particular character are identified using the above formula. Radial polynomial  $R_{nm}$  is used to compute the real and imaginary parts of the Zernike moment and takes the desired values in the form of moments. The Zernike polynomials take  $x^2 + y^2 \leq 1$  as their computation domain.  $Z_{nm}$  and it is used to compute the Zernike moments order for the function  $f(r, \theta)$ .

Figure 8 shows the Zernike moments based on order  $n = 0, 1, 2, 3, 4$  and with repetition  $m = 0, 1, 2, 3, 4$ . The total number of moments to extract the feature is 15.

The amplitude and phase values of each ancient Tamil character are computed and elapsed time is evaluated in

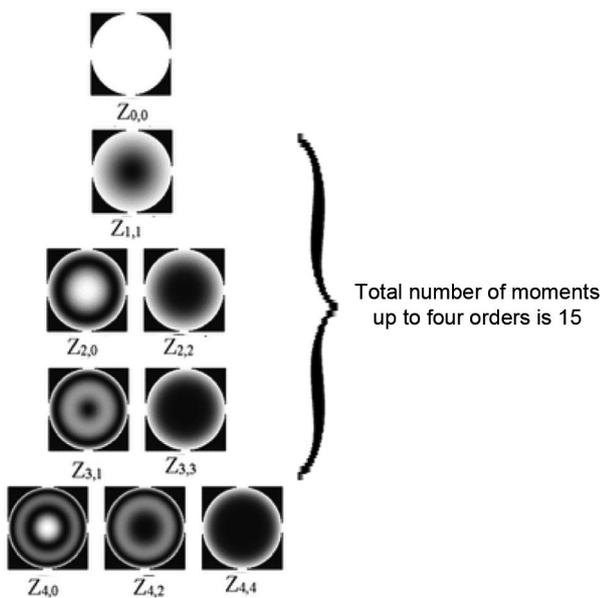


Figure 8.  $Z_{nm}$  polynomial order degree.

the recognition system (Table 1). The Zernike feature is able to identify Tamil characters more appropriately. The moment values of Zernike feature are complex and in the form  $x + yi$ , where  $x$  represents real values and  $y$  the imaginary values. Table 2 shows a few image examples based on Zernike moments of different orders.

**Zoning feature:** The input image is divided into  $N \times M$  zones. From each zone the features are extracted to form the vector values<sup>11</sup>. The main aim of zoning is to obtain the local characteristics of the image instead of extracting global characteristics. In this method, the resized binary image is given to zone based method where the image is divided into four zones (Figure 9).

From each zone, a set of four features is extracted. The features are vertical line section (VLS), right diagonal line section (RDS), horizontal line section (HLS) and left diagonal line section (LDS). So, a total of 16 features set are extracted from the input image.

The features are defined as follows: 1 – vertical line section, 2 – right diagonal section, 3 – horizontal line section, 4 – left diagonal section. Figure 10 shows the line-classified zoning image. This process is repeated sequentially for all 16 zones which are stored in the form of feature vector, followed by line classification. Finally, we obtain 16 features from zone-based technique and store them in a feature vector.

**Chi-square test for feature extraction:** To extract the Zoning feature, the system applies the chi-square test to check whether all the feature values of the image are independent or related. The test utilizes contingency table to analyse the data. Chi test is expressed in two ways: hypothesis  $H_0$ : variable 1 is independent to variable 2 and hypothesis  $H_1$ : variable 1 is not independent of variable 2. The test enables assumed theoretical distribution fit to the observed data. The main objective of the chi-square is to test the independence of the attributes and the homogeneity. (<https://libguides.library.kent.edu/SPSS/ChiSquare>). The test statistic for the chi-square test of independence is denoted as  $\chi^2$ , and calculated as

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}},$$

Table 1. Zernike moment, amplitude and phase values with elapsed time

Brahmi characters	$Z_{nm}$	Amplitude	Phase value	Elapsed time (sec)
க	$Z_{4,4}$	0.051692	178.6702	0.31879
	$Z_{4,4}$	0.017998	83.9701	0.064381
	$Z_{4,4}$	0.040157	-116.5482	0.027552
	$Z_{4,4}$	0.074466	1.8013	0.014147
	$Z_{4,4}$	0.010846	-143.8048	0.071782

**Table 2.** Moment value table for order  $n$  and  $m$  from 0 to 4

Brahmi characters					
K			D		
$n$	$m$	Moment values	$n$	$m$	Moment values
0	0	0.081437	0	0	0.084977
1	1	0.038159 + 0.0006871i	1	1	0.031502 + 0.03595i
2	0	-0.48143	2	0	-0.016915
2	2	0.025862 + 0.0019326i	2	2	0.015374 - 0.0025869i
3	1	-0.011903 + 0.01395i	3	1	-0.001020 + 0.034095i
3	3	0.028905 + 0.018359i	3	3	-0.00075664 - 0.001454i
4	0	0.067778	4	0	0.16319
4	2	-0.0073574 + 0.026213i	4	2	-0.067157 - 0.008042i
4	4	-0.051678 + 0.0011997i	4	4	-0.017948 + 0.035923i

**Table 3.** Chi-square test table

Category	$O_i$	$E_i$	$(O_i - E_i)^2/E_i$
1	15	12	0.75
2	6	12	3
3	14	12	0.33
4	21	12	6.75
5	18	12	3
6	7	12	2.08
7	10	12	0.33
8	19	12	4.08
9	9	12	0.75
10	4	12	5.33
11	8	12	1.33
12	20	12	5.33
13	5	12	4.08
14	13	12	0.08
169			$\chi^2 = 37.22$

where  $E_{ij}$  is the observed frequency in the  $i$ th row and  $j$ th column of the contingency table and  $O_{ij}$  is the expected frequency in the  $i$ th row and  $j$ th column of the contingency table.

$$E_{ij} = (\text{row } i \times \text{total} \times \text{col } j \times \text{total}) / \text{cumulative total.}$$

The calculated  $\chi^2$  value is compared with the critical value from the  $\chi^2$  table with degrees of freedom  $d.o.f = (N - 1)$ , where  $N$  is the total number of values in the table. If the calculated  $\chi^2$  value is greater than the tabulated  $\chi^2_{\alpha}$  value, then reject the null hypothesis. If the calculated  $\chi^2$  value is greater than the tabulated  $\chi^2_{\alpha}$  value, then accept the null hypothesis. Thus, if hypothesis is rejected, the ancient Tamil characters feature is extracted.

*Goodness-of-fit test illustration:* The chi-square test is used to determine the relationship between the independent feature and dependent feature. In feature extraction, the system aims to select the high dependent features for the neural network training process. The hypotheses  $H_0$  and  $H_1$  of the test are defined as follows

$H_0$ : The feature vector values of the zoning method are independent.

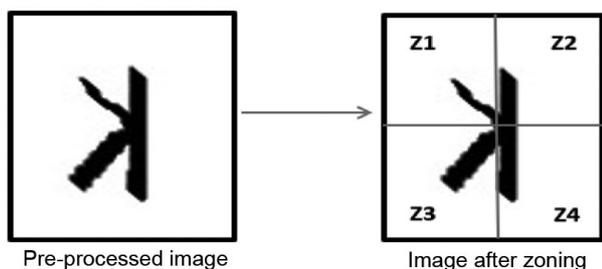
$H_1$ : The feature vector values of the zoning method are not independent.

From Table 3, the calculated chi-square test value is  $\chi^2 = 37.22$ .

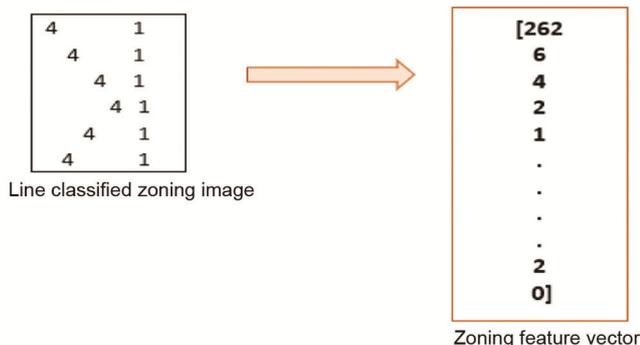
The number of observed frequency  $N = 14$  and the degree of freedom  $= (N - 1) = (14 - 1) = 13$ . Significance value for 95% confidence is  $\alpha = 0.05$ . The tabulated value is determined as  $\chi^2_{\alpha} = (\alpha, \text{degree of freedom}) = (5, 13) = 22.362$ . (<http://kisi.deu.edu.tr/joshua.cowley/Chi-square-table.pdf>). So, the calculated value is higher than the tabulated value ( $37.22 > 22.362$ ). Hence,  $H_1$  is accepted and the feature vector values are determined to be dependent, and we proceed with the recognition process.

*Neural network for character recognition*

Character recognition recognizes text characters in electronic files<sup>26,27</sup>. This system uses a machine learning tool



**Figure 9.** Zoning feature extraction.



**Figure 10.** Extracted zoning feature vector.

called a neural network with a gradient descent method for recognizing the ancient characters. The neural networks for a particular image are classically organized in layers. These layers are represented as a number of interconnected nodes that contains an activation function. Patterns are reachable via the input layer, which exchanges the words to the hidden layers, then it is connected to an output layer. The data division, training and derivatives were performed in the recognition process. By this process, the data are trained in the proposed system model of neural networks.

Figure 11 shows a multi-layered arrangement of an interconnected neural network representation<sup>17</sup>. It consists of an input layer, an output layer and a hidden layer with different functions. Each connecting input line ( $x_1, \dots, x_n$ ) has an associated weight ( $w_1, \dots, w_n$ ). The neural networks are trained by modifying these input weights, so that the considered outputs may be approximated by the preferred values. The output from a given neuron is considered by applying a transfer function to a weighted summation  $\Sigma$  of its input to give an output<sup>18</sup>. The activation function  $\phi$  is associated with the output line ( $o_1, \dots, o_n$ ) of the function for providing a preferred value, where the neural networks train the data based on the following formula

$$f(x) = \begin{cases} 1, & \text{when } x \geq \theta, \\ 0, & \text{when } x < \theta. \end{cases}$$

It is called as limiter function, where the  $f(x)$  value depends upon the threshold value  $\theta$ ,  $f(x) = 1/1 + e^{-x}$  is used to calculate sigmoid function  $f(x)$ . Neural networks are used to achieve better performance and accuracy. In this system the training of the data is done using the back propagation algorithm and recognition of data is done using the gradient descent algorithm.

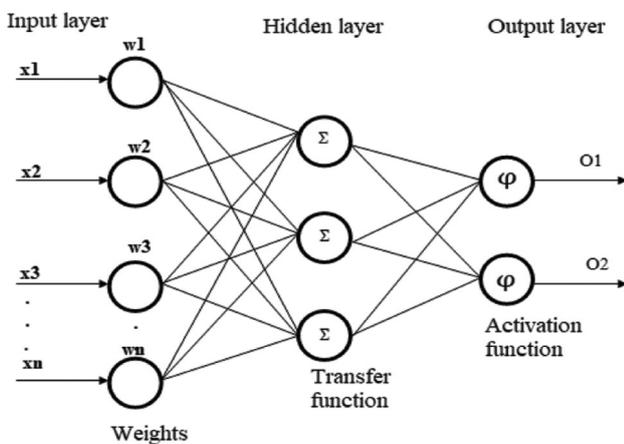


Figure 11. Representation of multi-layered neural networks.

**Back propagation:** This method calculates the gradient of the error function with respect to the neural network's weight<sup>28</sup>. The training of data is done using NNTool with the help of back propagation method. Initially, the image is loaded in MATLAB and then the training process is selected to analyse the data. During the analysis, the network divides the data into three subsets as training subset, validation subset and test subset. The propagation steps followed in the process are as follows:

- Step 1. Initialize all weights and threshold values of the network to a random value within a given range.
- Step 2. Feed-forward: Present the input vector as  $x = x_1, x_2, \dots, x_{n0}$  and the target output as  $y = y_1, y_2, \dots, y_{nm}$ .
- Step 3. Pass the input values to the first layer and for every layer find the output using  $o = \sum x_{ij} y_{ji} w_k$ .
- Step 4. Back propagation error: Calculate error value in backward order  $\delta e = y_{ji}(1 - y_{ji})(o_{ij} - y_{ji})$ .
- Step 5. Update weight: Update the weight propagating the error associated with the output neurons.
- Step 6. Iterate this process till the  $n$  number.
- Step 7. End the function when iteration exits.

The main purpose of using this technique is because it has a simple implementation. It can also be used for any network and provides smooth effects of weight correction terms.

**Gradient descent:** The gradient descent method is a type of neural network which is used to recognize the data. It is generally used for prediction purposes. In this system, the input image is recognized by gradient technique which is used to solve problems to find the optimized solutions. Every iteration is inexpensive and it does not require a second derivative to solve the problems. The optimal solution of the data is derived from the performance of gradient descent. The process of gradient descent is as follows:

- Step 1. Initialize the training epoch value.
- Step 2. Initialize the weight and bias value with random value. Then calculate the sum of squared error (SSE) using the formula

$$\frac{1}{2} \text{Sum}(\text{actual data} - \text{predicated data}).$$

- Step 3. Adjust the weighted value with gradient to reach optimal value where SSE decreases.
- Step 4. Use new weighted value for prediction and determine out the new SSE.
- Step 5. Repeat steps 2 and 3 until the error is reduced.

The epoch value is assigned as 80; the validation process takes place until it reaches the maximum limit. Figure 12



Figure 12. Character recognition.

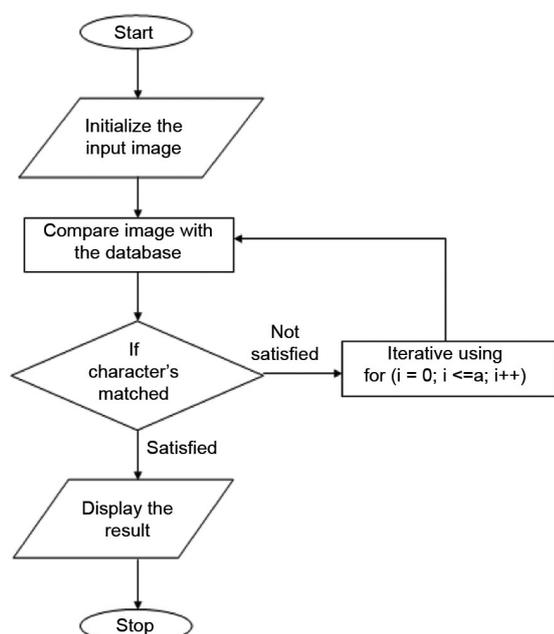


Figure 13. Flow chart of character conversion.

represents the character recognition of Brahmi characters using NNTool.

### Character conversion

The recognized data are used for the character identification module<sup>12</sup>. Character conversion is a process of converting the data from one form to another. Here, the input details should be about ancient Tamil scripts and the output should be the present Tamil characters (refer Table RR system). Figure 13 shows a flow chart of the function of the character conversion process. The character is initialized for comparison, where the recognized image is taken as the input. Then the image is compared with the database. If the characters match, the system

displays the result. Else the process is repeated  $n$  number of times using loop condition. After the result is displayed the function gets terminated and moves to the new recognition function. It extracts a clear image of every given Tamil character. An extracted image from the network was harmonized with the conversion function. The converted antique Tamil character was displayed in a web browser with .html extension.

### Experiment results

#### Confusion matrix

The ancient Tamil character recognition system uses the confusion matrix to analyse the overall accuracy. Generally, the confusion matrix is used to measure the performance of the model based on the values of  $N \times N$  matrix. Here,  $N$  is the number of predicted classes that compare with the actual class values by machine learning tools. This matrix helps in predicting the accuracy of the model and error rate<sup>29</sup>. In Figure 14a, rows represent actual classes and columns represent the predicted class. The row, 'Total' represents the sum of correct and incorrect observation of actual class, whereas the column, 'Total' represents the sum of correct and incorrect observation of predicted class. The colour scale in Figure 14b represents the intensity of values based on priority low to high. The high colour intensity represents the highest value 25 of the matrix in dark shade, and the lower colour intensity represents the lowest value 0 of the matrix in lighter shade. In this study, 25 samples of each ancient Tamil character image are prepared for the training process. Each character pattern at the input layer is represented as '1', which has only one neuron of the output layer that is the highest confidence. Value 0 is given to all other remaining neurons. For all the characters the output is in  $26 \times 1$  column matrix in which 1 has appeared in a single place and all the remaining entries are 0.

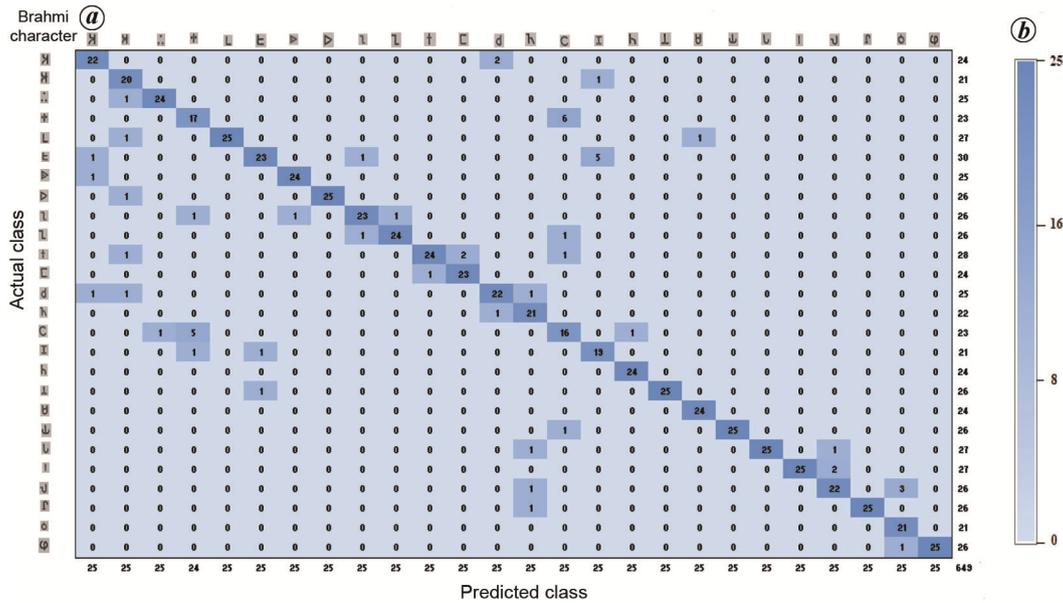


Figure 14. a, Confusion matrix with total sum of correctly and incorrectly classified values. b, Colour scale bar.

For instance, the character ‘𑀓’ results in (1, 0, 0, ..., 0), character ‘𑀔’ results in (0, 1, 0, ..., 0) and character ‘𑀕’ results in (0, 0, 1, ..., 0) and so on. In this way, all the individual ancient characters are represented. As there are 26 characters in a sample, the output of a sample presented at the input is a 26 × 26 matrix. In the proposed system, 25 samples are trained using NN with 26 character which is equal to 650 times (25 samples × 26 characters = 650 times). In Figure 14 a, the character ‘𑀓’ is recognized correctly 22 times out of 25, and incorrectly recognized as a character ‘𑀔’ two times. The character ‘𑀔’ is recognized correctly 17 times and recognized incorrectly as ‘𑀕’ six times. Likewise, all the correct and incorrect data are entered in the matrix that will help calculate the accuracy, and error rate of the system. To evaluate accuracy the system needs true positive (TP), true negative (TN), false positive (FP) and false negative (FN) values.

TP indicates correct character recognition with positive prediction. TN indicates correct character recognition with negative prediction. FP indicates incorrect character recognition with positive prediction. FN indicates incorrect character recognition with negative prediction.

*Confusion matrix illustration*

In Figure 14 a, there are 26 class samples from which TP, TN, FN and FP values are identified. The values of TP from {TP<sub>1</sub>, TP<sub>2</sub>, TP<sub>3</sub>, TP<sub>4</sub> ... TP<sub>23</sub>, TP<sub>24</sub>, TP<sub>25</sub> and TP<sub>26</sub>} are {22, 20, 24, 17 ... 22, 25, 21 and 25} respectively, which represent the main diagonal values of the confusion matrix. The values of FN for each class (output class) are calculated by adding all errors in the column of

that class<sup>29</sup>. FN<sub>1</sub> = 2, FN<sub>2</sub> = 1, FN<sub>3</sub> = 1, FN<sub>4</sub> = 6, FN<sub>5</sub> = 1 + 1 = 2. Similarly FN values of all other class labels till FN<sub>26</sub> are calculated.

The values of FP for each predicted class are calculated by adding all errors in the row of that class. For example, FP<sub>1</sub> = 1 + 1 + 1 = 3, FP<sub>2</sub> = 1 + 1 + 1 + 1 + 1 = 5, FP<sub>3</sub> = 1, FP<sub>4</sub> = 1 + 5 + 1 = 7, FP<sub>5</sub> = 0. Similarly, FP values of all other class labels till FP<sub>26</sub> are calculated. The value of TN for class label 1 (TN<sub>1</sub>) can be calculated by adding TP<sub>1</sub> + FP<sub>1</sub> + FN<sub>1</sub> and subtracting with total number of correctly and incorrectly classified values (TP + TN + FN + FP) = 649 based on the formula given below.

$$TN = (TP + TN + FP + FN) - (TP + FP + FN).$$

For example, TN<sub>1</sub> = 649 – 22 + 2 + 3, TN<sub>1</sub> = 622. Likewise, the TN values are calculated for the other remaining samples. Using TP, TN, FP and FN of each class, the accuracy and error rate can be determined. The total number of correctly classified values (TP + TN) = 593 and the total number of incorrectly classified values (FP + FN) = 56

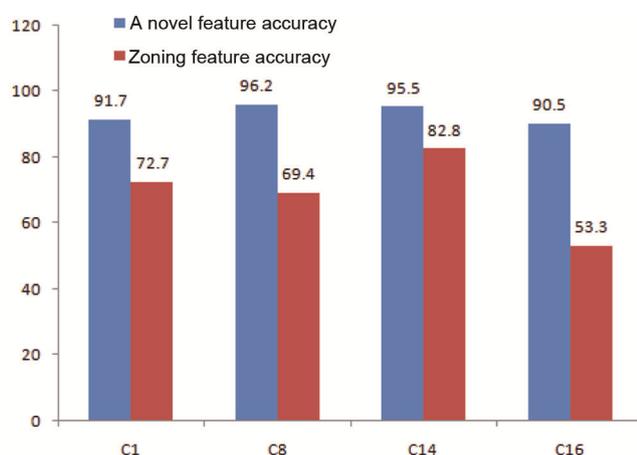
$$\text{Overall accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

$$\text{Error rate} = \frac{FP + FN}{TP + TN + FP + FN}.$$

The overall accuracy of the model and error rate are determined to be 91.3% and 8.7% respectively, based on the values obtained from the confusion matrix and the formula given above.

**Table 4.** RR system results

Class label	Segmented image	Edge detection	Image dilation	Image filling	Recognized character	Converted character	A novel feature accuracy	Zoning feature accuracy
C1						அ	91.7%	72.7%
C8						ஏ	96.2%	69.4%
C14						ஃ	95.5%	82.8%
C16						ஊ	90.5%	53.3%



**Figure 15.** Comparison of recognition rate accuracy.

*Comparison of recognition rate result*

A graphical user interface (GUI) model is developed for the repossession and recognition system (RR system). Table 4 shows the class labels c1, c8, c14 and c16 of each character with results of each phase of image processing like segmented image, edge detection, image dilation, filling, character recognition and character conversion along with the accuracy it obtains. The accuracy obtained using this feature extraction is compared with that of the zoning feature method with a neural network to highlight the performance of the algorithm. The accuracy percentage is evaluated based on the confusion matrix and its statistics. Figure 15 compares the accuracy between novel feature and zoning feature from Table 4. Based on the accuracy it can be concluded that the feature extraction gives good results compared to the zoning feature alone. So, it is suitable for recognition of Brahmi characters.

**Conclusion and future work**

The input image undergoes preprocessing, the features are extracted and the Brahmi characters are recognized. Here the recognized characters are converted into present Tamil letters and output appears in an html page. The image features are extracted using novel feature extractions, Zernike moment and zoning feature. A chi-square test is performed to determine whether the values of the vectors are independent or dependent. The 91.3% accuracy rate is achieved using the confusion matrix. Recognition of Tamil characters using neural networks takes less time and displays higher precision.

There are still some problems in the recognition. This method can be further extended based on larger datasets, on-line character recognition and improvization of accuracy by applying advanced algorithms.

- Selvakumar, P. and Ganesh, S. H., Tamil character recognition using canny edge detection algorithm. In *Computing and Communication Technologies*, 2017, pp. 250–254.
- Mahalakshmi, M. and Sharavanan Malathi, Ancient Tamil script and recognition and translation using LabVIEW. In *International Conference on Communication and Signal Processing*, 2013, pp. 1021–1026.
- Rajakumar, S. and Subbiah Bharathi, V., Century identification and recognition of ancient Tamil character recognition. *Int. J. Comput. Appl.*, 2011, **26**(4), 32–35.
- Vellingiriraj, E. K. and Balasubramanie, P., Recognition of Ancient Tamil handwritten characters in palm manuscripts using genetic algorithm. *Int. J. Sci. Eng. Technol.*, 2016, **2**(5), 342–346.
- Janani, G., Vishalini, V. and Mohan Kumar, P., Recognition and analysis of Tamil inscriptions and mapping using image processing techniques. *IEEE Trans. Sci. Technol. Eng. Manage.*, 2016.
- Subramani Kavitha and Murugavalli, S., A novel binarization method for degraded Tamil palm leaf images. In *IEEE Eighth International Conference on Advanced Computing*, 2016.

7. Banumathi, P. and Nasira, G. M., Handwritten Tamil character recognition using artificial neural networks. *IEEE Trans.*, 2011.
8. Jagadeesh Kannan, R., Prabhakar, R. and Suresh, R. M., Off-line cursive handwritten Tamil character recognition. *IEEE Trans. Int. Conf. Security Technol.*, 2008.
9. Kavitha, A. S., Shivakumara, P. and Hemantha Kumar, G., Skewness and nearest neighbour based approach for historical document classification. *IEEE Trans. Commun. Syst. Network Technol.*, 2013.
10. Subashini and Kodikara, N. D., A novel SIFT-based codebook generation for handwritten Tamil character recognition. *IEEE Trans. Ind. Inf. Syst.*, 2011.
11. Pirlo, G. and Impedovo, D., Adaptive membership functions for handwritten character recognition by Voronoi-based image zoning. *IEEE Trans. Image Process.*, 2012, **21**(9).
12. Boufenar, C., Batouche, M. and Schoenauer, M., An artificial immune system for offline isolated handwritten Arabic character recognition. *Springer Trans. Evol. Syst.*, 2016.
13. Rajakumar, S. and Subbiah Bharathi, V., Ancient Tamil script recognition from stone inscriptions using slant removal method. *IEEE Trans. Elect., Electron. Biomed. Eng.*, 2012.
14. Bhuvaneshwari, G. and Subbiah Bharathi, V., An efficient method for digital imaging of ancient stone inscriptions. *Curr. Sci.*, 2016, 245–250.
15. Medhi, Kalyanbrat and Sanjib Kr Kalita, Assamese character recognition using zoning feature. In *Advances in Electronics, Communication and Computing*, Springer, Singapore, 2018, pp. 371–380.
16. Vellingiriraj, E. K., Balamurugan, M. and Balasubramanie, P., Information extraction and text mining of ancient vattezhuthu characters in historical documents using image zoning. *IEEE Trans. Int. Conf. Asian Language Process.*, 2016.
17. Kale, K. V., Deshmukh, P. D., Chavan, S. V., Kazi, M. M. and Rode, Y. S., Zernike moment feature extraction for handwritten Devanagari compound character recognition. In *IEEE Science and Information Conference*, 2013, pp. 459–466.
18. Abandah, G. A., Jamour, F. T. and Qaralleh, E. A., Recognizing handwritten Arabic words using grapheme segmentation and recurrent neural networks. *Int. J. Doc. Anal. Recogn.*, 2014, **17**(3), 275–291.
19. Plamondon, R. and Srihari, S. N., Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **22**(1), 63–84.
20. Sumathi, C. P. and Karpagavalli, S., Techniques and methodologies for recognition of Tamil typewritten and handwritten characters: a survey. *Int. J. Comput. Sci. Eng. Surv.*, 2012, **3**(6), 23–35.
21. Indra Gandhi, R. and Iyakutti, K., An attempt to recognize handwritten Tamil character using Kohonen SOM. *Int. J. Adv. Net. Appl.*, 2009, **1**(3), 188–192.
22. [http://en.wikipedia.org/wiki/brami\\_script](http://en.wikipedia.org/wiki/brami_script)
23. Suganya, T. S. and Murugavalli, S., A survey on character recognitions in Tamil scripts using OCR, 2012.
24. JeyaGreeba, M. V. and Bhuvaneshwari, G., Recognition of ancient Tamil characters in stone inscription using improved feature extraction. *Int. J. Recent Develop. Eng. Technol.*, 2014, **2**(3), 38–41.
25. Zernike, F., *Physica*, 1934, **1**, 689.
26. Wang, Qiu-Feng Yin, Fei and Liu, Cheng-Lin, Handwritten Chinese text recognition by integrating multiple contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**(8), 1469–1481.
27. Pornpanomchai, C., Wongsawangtham, V., Jeungudomporn, S. and Chatsumpun, N., Thai handwritten character recognition by genetic algorithm (THCRGA). *IACSIT J. Eng. Technol.*, 2011, **3**(2), 148–153.
28. Jose, T. M. and Wahi, A., Recognition of Tamil handwritten characters using Daubechies wavelet transforms and feed-forward backpropagation network. *Int. J. Comput. Appl.*, 2013, **64**(8), 0975–8887.
29. Tharwat, A., Classification assessment methods. *Applied Computing and Informatics*, 2018.

Received 15 February 2018; revised accepted 2 November 2020

doi: 10.18520/cs/v120/i4/654-665