Eighth Century Tamil Consonants Recognition From Stone Inscriptions

S.RajaKumar
Research Scholar, Department of ECE
Sathiyabama University
Chennai, India
rajkumarmeae135@yahoo.co.in

Dr.V.Subbiah Bharathi
Principal
DMI College of Engineering
Chennai, India
yughasurya@yahoo.co.in

***Abstract*— Ancient Tamil character recognition is an important research and application area on pattern recognition theory, which plays an important role on realizing automation of inputting character at all cases. In order to improve the rate of character recognition and decrease the time of recognition training, referencing to immune biological principle, an ancient Tamil character recognition algorithm based on artificial immune is proposed. The simulation results shown that the method has faster speed and higher accuracy than the traditional ancient Tamil character recognition based on neural network. The algorithm steals the merit of self-adaptive learning, and immune memory in the biology immune system, which can also be applied to abnormity detection and pattern recognition. Ancient Tamil character recognition includes complex per-processes such as; segmenting, smoothing filtering , normalizing and other processing. Different algorithms bring many differences in speed and accuracy of recognition.**

Keywords-component; Tamil stone inscriptions; Gober filter; Support vector

I. INTRODUCTION

The ancient Tamil character recognition system includes three stages : image preprocessing, feature extractor, and classifier. The process of ancient Tamil character recognition involves extraction of some defined characteristics called features to classify an unknown character into one of the known classes. Pre-processing is primarily used to reduce variations of handwritten characters. A feature extractor is essential for efficient data representation and extracting meaningful features for later processing. A classifier assigns the characters to one of the several classes. There exist a whole lot of tasks to complete before the actual character recognition operation is commenced. These preceding tasks make certain the scanned document is in a suitable form so as to ensure the input for the subsequent recognition operation is intact. The process of refining the scanned input image includes several steps. The pre-processing stage comprises three steps: Binarization, Noise Removal, Skew Correction. Ancient Tamil Font Recognition is one of the Challenging tasks in Optical Character Recognition. Most of the existing methods for character recognition make use of local typographical features and connected component analysis. In this paper, Ancient Tamil character recognition is done based on global texture analysis. The main objective of this proposal is to employ support vector machines (SVM) in identifying various fonts in Tamil. The feature vectors are extracted by making use of Gabor filters and the proposed SVM is trained using these features. The method is found to give superior performance over neural networks by avoiding local minima points. The SVM model is formulated tested and the results are presented in this paper. It is observed that this method is content independent and the SVM classifier shows an average accuracy of 94%.

II. PREPROCESSING

A. *Binarization*

Extraction of foreground from the background is called as thresholding. Typically two peaks comprise the histogram gray-scale values of a document image: a high peak analogous to the white background and a smaller peak corresponding to the foreground. Fixing the threshold value is determining the one optimal value between the peaks of gray scale values . Each value of the threshold is tried and the one that maximizes the criterion is chosen from the two classes regarded as the foreground and background points.

B. *Block diagram*

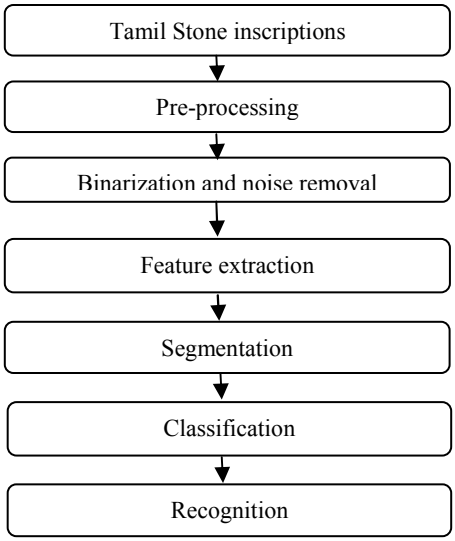

Fig 1 Process flow diagram

C. Noise Removal

The presence of noise can cost the efficiency of the character recognition system. Noise may be due the poor quality of the input image, but whatever is the cause of its presence it should be removed before further processing. We have used Gabor filtering for the removal of the noise from the image.

Gabor Filter:

Then the part of the image needed is filtered with the help of Gabor filters since it is a better approximation to the receptive field profile of simple cells in visual cortex. It is defined by the following equation.

$$g_{\gamma,\eta,\varphi,\lambda} = \exp\left(\frac{x'^2+\gamma 2y'^2}{2\sigma^2}\right).\cos\left(\frac{2\pi x'}{\lambda}+\varphi\right) \quad (1)$$

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta - y\cos\theta \quad (2)$$

The Gabor filters can be better used by varying the parameters like λ, γ, φ and θ. In the above equations, x and y represent image coordinates; s is the standard deviation of Gaussian function which is usually set to 0.56 λ; λ is the wave length of cosine equation; γ characterizes the shape of Gaussian, circular shape for γ=1 and elliptic for γ<1 and θ represents the channel orientation and takes values in interval (0, 360). Since it is symmetric, θ varies from zero to 180. The response of this filter is nothing but the convolution given by the following equation

$$\iint I(\varepsilon,\eta)\,g(x-\varepsilon, y-\eta)\,d\varepsilon d\eta \quad (3)$$

The value of θ and σ must be taken under some considerations to make the choice of filter to be optimum.

D. Conversion to image files

The text files to be trained cannot be directly used for global feature extraction. Hence, they are converted to image files, which are further normalized and converted to binary image with the help of Otsu's algorithm. The following steps can achieve it

1. Compute the histogram and probabilities of each intensity level and set up initial class probabilities $\omega_i(0)$) and class means ($\mu_i(0)$).
2. Step through all possible thresholds maximum intensity 't'.
3. Update ω_i and μ_i, compute s=σ*σ and desired threshold corresponds to the maximum's'.
4. Then the intensity of each pixel is compared with the threshold calculated.
5. If the intensity is greater than the threshold it is considered as white, else it is considered as black.

III. Feature Extraction

In this stage, each pre-processed sample is transformed into a sequence of feature vectors.

A. Time-domain features.

The time-domain features are largely adapted and are described below.

1) Normalized x-y coordinates: The *x and y coordinates* from the normalized sample constitute the first 2 features.

2) Normalized first derivatives: The normalized first derivatives $\hat{x}'$ and $\hat{y}'$ are calculated .

$$x'_i = \frac{\sum_{i=1}^{2} i.(x_{i+1}-x_{i-1})}{2.\sum_{i=1}^{2} i^2} \qquad y'_i = \frac{\sum_{i=1}^{2} i(y_{i+1}-y_{i-1})}{2.\sum_{i=1}^{2} i^2}$$

$$\hat{x}'_i = \frac{x'_i}{\sqrt{x'^2_i+y'^2_i}}; \qquad \hat{y}'_i = \frac{y'_i}{\sqrt{x'^2_i+y'^2_i}};$$

3) Normalized second derivatives: The second derivatives are computed by replacing x and y with $\hat{x}_2$ and $\hat{y}_2$ in the first part of formulae and normalized similarly.

4) Curvature: Curvature at a point on a plane curve is defined as the inverse of the radius of the osculating circle. It is calculated as

$$k_t = \frac{\hat{x}'.\hat{y}'' - \hat{x}''.\hat{y}'}{(\hat{x}'^2+\hat{y}'^2)^{3/2}}$$

5) Aspect: Aspect at a point characterizes the ratio of the height to the width of the bounding box containing points in the neighborhood. It is computed as in NPen++ [8]. It is given by

$$A(t) = \frac{2\times\Delta y(t)}{\Delta x(t)+\Delta y(t)} - 1$$

where $\Delta x(t)$ and $\Delta y(t)$ are the width and the height of the bounding box containing the points in the neighborhood of the point under consideration. In all our experiments, we have used a neighborhood of length 2 i.e. two points to the left and two points to the right of the point along with the point itself.

6) Curliness: Curliness at a point gives the deviation of the neighborhood points from the line joining the first and last points in the neighborhood. It is given by

$$C(t) = \frac{L}{\max(\Delta x, \Delta y)} - 2$$

where L is the sum of all the line segments along the trajectory in the neighborhood of the point [27].

7) Lineness: It is the average squared distance between every point in the neighborhood and the line joining the first and last points of the neighborhood .

B. Frequency-domain features.

To determine the frequency domain features, the character sequence is viewed as a complex function: : $f :\rightarrow (x_t + iy_t)$ where t denotes time and x_t and y_t are the coordinates of the point at time t. The frequency domain features were computed along the stroke using a sliding Hamming window. At each point, the window is centred and Discrete Cosine Transform (DCT) is evaluated on the windowed sequence. The real and imaginary parts of the lowest 4 coefficients excluding DC coefficient were added to the feature vector. The number of coefficients to be considered was determined empirically .

IV. SEGMENTATION

Segmentation is a process of distinguishing lines, words, and even characters of a hand written or machine printed document, a crucial step as it extracts the meaningful regions for analysis. There exist many sophisticated approaches for segmenting the region of interest. For handwritten document, this is quiet difficult. The details of line, word and character segmentation are discussed as follows.

A. Line Segmentation.

Obviously the ascenders and descenders frequently intersect up and down of the adjacent lines, while the lines of text might itself flutter up and down. Each word of the line resides on the
imaginary line that people use to assume while writing and a method has been formulated based on this notion. The local minima points are calibrated from each component to approximate this imaginary baseline. To guesstimate and categorize the minima of all components and to recognize different handwritten lines clustering techniques are deployed .

B. Word and Character Segmentation.

The process of word segmentation succeeds the line separation task. Most of the word segmentation issues usually concentrate on discerning the gaps between the characters to distinguish the words from one another other. This process of discriminating words emerged from the notion that the spaces between words are usually larger than the spaces between the characters. There are not many approaches to word segmentation issues dealt in the literature. In spite of all these perceived conceptions, exemptions are quiet common due to flourishes in writing styles with leading and trailing ligatures. Alternative methods not depending on the one-dimensional distance between components, incorporates cues that humans use. Meticulous examination of the variation of spacing between the adjacent characters as a function of the corresponding characters themselves helps reveal the writing style of the author, in terms of spacing. The segmentation scheme comprises the notion of expecting greater spaces between characters with leading and trailing ligatures.

Recognizing the words themselves in textual lines can itself help lead to isolation of words. Segmentation of words in to its constituent characters is touted by most recognition methods. Features like ligatures and concavity are used for determining the segmentation points. The algorithm exploits the caps between character segments and heights of character segments too.

V. LEARNING PROCESS AND CLASSIFICATION

The features are extracted and are then fed to the trained SVM for classification. The developed SVM is evaluated and a 10- fold cross validation is done to verify the results. The accuracy of the SVM model is computed. The SVM model can be understood through the following steps:

1. For each font, four images are chosen corresponding to each style and the
2. The image is resized to the dimension 300 x 300 and is subdivided into nine non-overlapping blocks.
3. The features are extracted for each block using the above algorithm.
4. Being a Multi class problem, this is solved using the one-against-all algorithm.
5. The SVM Model is verified by 10-fold cross validation and the influence of number of training instances over accuracy is analyzed.

VI. EXPERMENTAL RESULTS

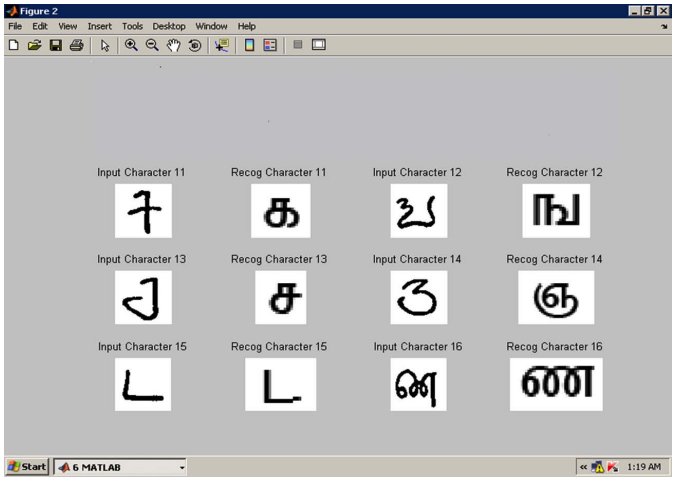

Fig 2 Output-1

Fig 3 Output-2

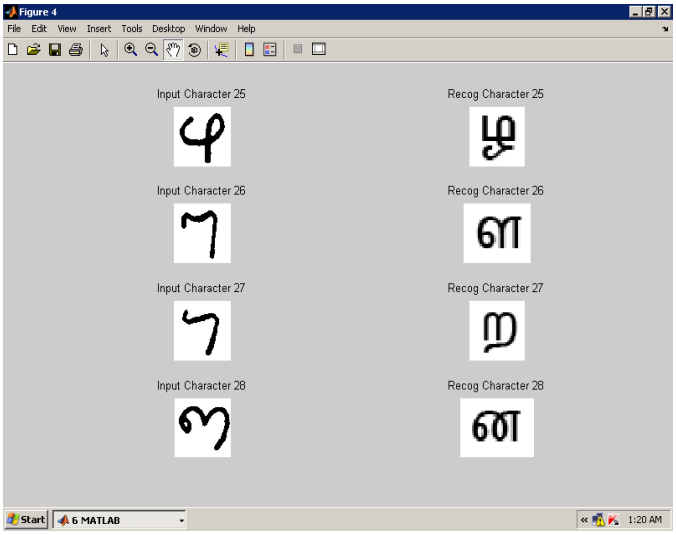

Fig 4 Output-3

Output 1, 2, 3 shows the validation of Tamil stone inscriptions through SUM model classification. The proposed algorithm helps in Tamil character recognition which gives improved accuracy of recognition.

REFERENCES

[1] Xiaoqing Ding, Tao Wu. "Character Independent Font Recognition on a Single Chinese Character". IEEE Transactions on pattern analysis and machine intelligence, Vol.29, N0.2, February 2007.

[2] Seethalakshmi et.al, "Optical Character Recognition for printed Tamil text using Unicode", Journal of Zhejiang University SCIENCE, Vol. 6A No. 11, 2005.

[3] R.Ramanathan, N.Valliappan, S. Pon Mathavan, M.Gayathri, R.Priya, K.P.Soman "Generalised and Channel Independent SVM based Robust Decoders for Wireless Applications" to be published in the Proceedings of International Conference on Advances in Recent Technologies in Communicaion and Computing – ARTCom 2009, India, 27-28 Oct 2009

[4] R.Ramanathan, Arun.S.Nair, V.Vidhyasagar, N.Sriram, K.P.Soman "A Support Vector Machines Approach for Efficient Facial Expression Recognition" to be published in the Proceedings of International Conference on Advances in Recent Technologies in Communicaion and Computing – ARTCom 2009, India, 27-28 Oct 2009

[5] R.Ramanathan, P.A.Rohini, G.Dharshana., K.P.Soman"Investigation and Development of Methods to Solve Multi-ClassClassification Problems " to be published in the Proceedings of International Conference on Advances in Recent Technologies in Communicaion and Computing – ARTCom 2009, India, 27-28 Oct 2009