

Handwritten Tamil Character Recognition Using RCS Algorithm

C.Sureshkumar
Department of Information Technology,
J.K.K.Nataraja College of Engineering,
Namakkal, Tamilnadu, India.

Dr.T.Ravichandran
Department of Computer Science & Engineering,
Hindustan Institute of Technology,
Coimbatore, Tamilnadu, India

ABSTRACT

Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size and orientation angle of the characters. The scanned image is segmented into paragraphs using spatial space detection technique, paragraphs into lines using vertical histogram, lines into words using horizontal histogram, and words into character image glyphs using horizontal histogram. The extracted features considered for recognition are given to Support Vector Machine, Self Organizing Map, RCS, Fuzzy Neural Network and Radial Basis Network. Where the characters are classified using supervised learning algorithm. These classes are mapped onto Unicode for recognition. Then the text is reconstructed using Unicode fonts. This character recognition finds applications in document analysis where the handwritten document can be converted to editable printed document. Structure analysis suggested that the proposed system of RCS with back propagation network is given higher recognition rate.

Keywords

Support Vector, Fuzzy, RCS, Self organizing map, Radial basis function.

1. INTRODUCTION

Hand written Tamil Character recognition refers to the process of conversion of handwritten Tamil character into Unicode Tamil character. Among different branches of handwritten character recognition it is easier to recognize English alphabets and numerals than Tamil characters. Many researchers have also applied the excellent generalization capabilities offered by ANNs to the recognition of characters. Many studies have used fourier descriptors and Back Propagation Networks for classification tasks. Fourier descriptors were used in to recognize handwritten numerals. Neural Network approaches were used to classify tools. There have been only a few attempts in the past to address the recognition of printed or handwritten Tamil Characters. However, less attention had been given to Indian language recognition. Some efforts have been reported in the literature for Tamil scripts. In this work, we propose a recognitionsystem for handwritten Tamil characters. Tamil is a South Indian language spoken widely in TamilNadu in India. Tamil has the longest unbroken literary tradition amongst the Dravidian languages. Tamil is inherited from Brahmi script. The earliest available text is the Tolkaappiyam, a work describing the language of the classical period. There are several other famous works in Tamil like Kambar Ramayana and Silapathigaram but few supports in Tamil which speaks about the greatness of the language. For example, Thirukural is translated into other languages due to its richness in content. It is a collection of two sentence poems efficiently conveying things in a hidden language called Slaydai in

Tamil. Tamil has 12 vowels and 18 consonants. These are combined with each other to yield 216 composite characters and 1 special character (aayutha ezhuthu) counting to a total of (12+18+216+1) 247 characters. Tamil vowels are called uyireluttu (uyir – life, eluttu – letter). The vowels are classified into short (kuril) and long (five of each type) and two diphthongs, /ai/ and /auk/, and three "shortened" (kuril) vowels. The long (nedil) vowels are about twice as long as the short vowels. Tamil consonants are known as meyyeluttu (mey - body, eluttu - letters). The consonants are classified into three categories with six in each category: vallinam - hard, mellinam - soft or Nasal, and itayinam - medium. Unlike most Indian languages, Tamil does not distinguish aspirated and unaspirated consonants. In addition, the voicing of plosives is governed by strict rules in centamil. As commonplace in languages of India, Tamil is characterised by its use of more than one type of coronal consonants. The Unicode Standard is the Universal Character encoding scheme for written characters and text. The Tamil Unicode range is U+0B80 to U+0BFF. The Unicode characters are comprised of 2 bytes in nature. For example, the Unicode for the character அ is 0B85; the Unicode for the character ட is 0BAE+0BC0. The Unicode is designed for various other Tamil characters.

2. TAMIL CHARACTER RECOGNITION

The schematic block diagram of handwritten Tamil Character Recognition system consists of various stages as shown in figure. They are Scanning phase, Preprocessing, Segmentation, Feature Extraction, Classification, Unicode mapping and recognition and output verification.

2.1. Scanning

A properly printed document is chosen for scanning. It is placed over the scanner. A scanner software is invoked which scans the document. The document is sent to a program that saves it in preferably TIF, JPG or GIF format, so that the image of the document can be obtained when needed.

2.2. Preprocessing

This is the first step in the processing of scanned image. The scanned image is preprocessed for noise removal. The resultant image is checked for skewing. There are possibilities of image getting skewed with either left or right orientation. Here the image is first brightened and binarized. The function for skew detection checks for an angle of orientation between ± 15 degrees and if detected then a simple image rotation is carried out till the lines match with the true horizontal axis, which produces a skew corrected image.



Figure 1. Histograms for skewed and skew corrected images, Original Texts



Figure 2. Character Segmentation

Knowing the skew of a document is necessary for many document analysis tasks. Calculating projection profiles, for example, requires knowledge of the skew angle of the image to a high precision in order to obtain an accurate result. In practical situations, the exact skew angle of a document is rarely known, as scanning errors, different page layouts, or even deliberate skewing of text can result in misalignment. In order to correct this, it is necessary to accurately determine the skew angle of a document image or of a specific region of the image, and, for this purpose, a number of techniques have been presented in the literature. Figure 1 shows the histograms for skewed and skew corrected images and original character. Postal found that the maximum valued position in the Fourier spectrum of a document image corresponds to the angle of skew. However, this finding was limited to those documents that contained only a single line spacing, thus the peak was strongly localized around a single point. When variant line spacing's are introduced, a series of Fourier spectrum maxima are created in a line that extends from the origin. Also evident is a subdominant line that lies at 90 degrees to the dominant line. This is due to character and word spacing's and the strength of such a line varies with changes in language and script type. Scholkopf, Simard expand on this method, breaking the document image into a number of small blocks, and calculating the dominant direction of each such block by finding the Fourier spectrum maxima. These maximum values are then combined over all such blocks and a histogram formed. After smoothing, the maximum value of this histogram is chosen as the approximate skew angle. The exact skew angle is then calculated by taking the average of all values within a specified range of this approximate. There is some evidence that this technique is invariant to document layout and will still function even in the presence of images and other noise. The task of smoothing is to remove unnecessary noise present in the image. Spatial filters could be used. To reduce the effect of noise, the image is smoothed using a Gaussian filter. A Gaussian is an ideal filter in the sense that it reduces the magnitude of high spatial frequencies in an image proportional to their frequencies. That is, it reduces magnitude of higher frequencies more. Thresholding is a nonlinear operation that converts a gray scale image into a binary image where the two levels are assigned to pixels that are below or above the specified threshold value. The task of thresholding is to extract the foreground from the background. Global methods apply one threshold to the entire image

while local thresholding methods apply different threshold values to different regions of the image. Skeletonization is the process of peeling off a pattern as any pixels as possible without affecting the general shape of the pattern. In other words, after pixels have been peeled off, the pattern should still be recognized. The skeleton hence obtained must be as thin as possible, connected and centered. When these are satisfied the algorithm must stop. A number of thinning algorithms have been proposed and are being used. Here Hilditch's algorithm is used for skeletonization.

2.3. Segmentation

After pre processing, the noise free image is passed to the segmentation phase, where the image is decomposed [2] into individual characters. Figure 2 shows the image and various steps in segmentation.

2.4. Feature extraction

The next phase to segmentation is feature extraction where individual image glyph is considered and extracted for features. Each character glyph is defined by the following attributes: (1) Height of the character. (2) Width of the character. (3) Numbers of horizontal lines present short and long. (4) Numbers of vertical lines present short and long. (5) Numbers of circles present. (6) Numbers of horizontally oriented arcs. (7) Numbers of vertically oriented arcs. (8) Centroid of the image. (9) Position of the various features. (10) Pixels in the various regions.

3. NEURAL NETWORK APPROACHES

The architecture chosen for classification is Support Vector machines, which in turn involves training and testing the use of Support Vector Machine (SVM) classifiers [1]. SVMs have achieved excellent recognition results in various pattern recognition applications. Also in handwritten character recognition they have been shown to be comparable or even superior to the standard techniques like Bayesian classifiers or multilayer perceptrons. SVMs are discriminative classifiers based on vapnik's structural risk minimization principle. Support Vector Machine (SVM) is a classifier which performs classification tasks by constructing hyper planes in a multidimensional space.

3.1. Classification SVM Type-1

For this type of SVM, training involves the minimization of the error function:

$$\frac{1}{2} w^T w + c \sum_{i=1}^N \xi_i \quad (1)$$

subject to the constraints:

$$y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i \text{ and } \xi_i \geq 0, i = 1, \dots, N \quad (2)$$

Where C is the capacity constant, w is the vector of Coefficients, b a constant and ξ_i are parameters for handling non separable data (inputs). The index i label the N training cases [6, 9]. Note that y=1 represents the class labels and xi is the independent variables. The kernel ϕ is used to transform data from the input (independent) to the feature space. It should be noted that the larger the C, the more the error is penalized.

3.2. Classification SVM Type-2

In contrast to Classification SVM Type 1, the Classification SVM Type 2 model minimizes the error function:

$$\frac{1}{2} w^T w - \nu \rho + \frac{1}{N} \sum_{i=1}^N \xi_i \quad (3)$$

subject to the constraints:

$$y_i (w^T \phi(x_i) + b) \geq \rho - \xi_i \text{ and } \xi_i \geq 0, i = 1, \dots, N; \rho \geq 0 \quad (4)$$

A self organizing map (SOM) is a type of artificial neural network that is trained using unsupervised learning to produce a low dimensional (typically two dimensional), discretized representation of the input space of the training samples, called a map. Self organizing maps are different than other artificial neural networks in the sense that they use a neighborhood function to preserve the topological properties of the input space.

This makes SOM useful for visualizing low dimensional views of high dimensional data, akin to multidimensional scaling. SOMs operate in two modes: training and mapping. Training builds the map using input examples. It is a competitive process, also called vector quantization [7]. Mapping automatically classifies a new input vector.

The self organizing map consists of components called nodes or neurons. Associated with each node is a weight vector of the same dimension as the input data vectors and a position in the map space. The usual arrangement of nodes is a regular spacing in a hexagonal or rectangular grid. The self organizing map describes a mapping from a higher dimensional input space to a lower dimensional map space.

3.3. Algorithm for Kohonon's SOM

- (1) Assume output nodes are connected in an array,
- (2) Assume that the network is fully connected all nodes in input layer are connected to all nodes in output layer.
- (3) Use the competitive learning algorithm.

$$|w_i - x| \leq |w_k - x| \quad \forall k \quad (5)$$

$$w_k(\text{new}) = w_k(\text{old}) + \mu \chi(i, k)(x - w_k) \quad (6)$$

Randomly choose an input vector x , Determine the "winning" output node i , where w_i is the weight vector connecting the inputs to output node.

A new neural classification algorithm and Radial- Basis-Function Networks are known to be capable of universal approximation and the output of a RBF network can be related to Bayesian properties. One of the most interesting properties of RBF networks is that they provide intrinsically a very reliable rejection of "completely unknown" patterns at variance from MLP. Furthermore, as the synaptic vectors of the input layer store locations in the problem space, it is possible to provide incremental training by creating a new hidden unit whose input synaptic weight vector will store the new training pattern. The specifics of RBF are firstly that a search tree is associated to a hierarchy of hidden units in order to increase the evaluation speed and secondly we developed several constructive algorithms for building the network and tree.

3.4. RBF Character Recognition

In our handwritten recognition system the input signal is the pen tip position and 1-bit quantized pressure on the writing surface. Segmentation is performed by building a string of "candidate characters" from the acquired string of strokes [16]. For each stroke of the original data we determine if this stroke does belong to an existing candidate character regarding several criteria such as: overlap, distance and diacriticity. Finally the regularity of the character spacing can also be used in a second pass. In case of text recognition, we found that punctuation needs a dedicated processing due to the fact that the shape of a punctuation mark is usually much less important than its position. It may be decided that the segmentation was wrong and that back tracking on the segmentation with changed decision thresholds is needed. Here, tested two encoding and two classification methods. As the aim of the writer is the written shape and not the writing gesture it is very natural to build an image of what was written and use this image as the input of a classifier.

Both the neural networks and fuzzy systems have some things in common. They can be used for solving a problem (e.g. pattern recognition, regression or density estimation) if there does not exist any mathematical model of the given problem. They solely do have certain disadvantages and advantages which almost completely disappear by combining both concepts. Neural networks can only come into play if the problem is expressed by a sufficient amount of observed examples [12]. These observations are used to train the black box. On the one hand no prior knowledge about the problem needs to be given. However, it is not straightforward to extract comprehensible rules from the neural network's structure. On the contrary, a fuzzy system demands linguistic rules instead of learning examples as prior knowledge. Furthermore the input and output variables have to be described linguistically. If the knowledge is incomplete, wrong or contradictory, then the fuzzy system must be tuned. Since there is not any formal approach for it, the tuning is performed in a heuristic way. This is usually very time consuming and error prone.

3.5. Hybrid Fuzzy Neural Network

Hybrid Neuro fuzzy systems are homogeneous and usually resemble neural networks. Here, the fuzzy system is interpreted as special kind of neural network. The advantage of such hybrid NFS is its architecture since both fuzzy system and neural network do not have to communicate any more with each other. They are one fully fused entity [14]. These systems can learn online and offline. The rule base of a fuzzy system is interpreted as a neural network. Thus the optimization of these functions in terms of generalizing the data is very important for fuzzy systems. Neural networks can be used to solve this problem.

4. PROPOSED NEW APPROACH

Once a boundary image is obtained then Fourier descriptors are found. This involves finding the discrete Fourier coefficients $a[k]$ and $b[k]$ for $0 \leq k \leq L-1$, where L

Is the total number of boundary points found, by applying equations (7) and (8)

$$a[k] = 1/L \sum_{m=1}^L x[m] e^{-jk(2\pi/L)m} \quad (7)$$

$$b[k] = 1/L \sum_{m=1}^L y[m] e^{jk(2\pi/L)m} \quad (8)$$

Where $x[m]$ and $y[m]$ are the x and y co-ordinates respectively of the m th boundary point. In order to derive a set of Fourier descriptors that have the invariant property with respect to rotation and shift, the following operations are defined [3,4]. For each n compute a set of invariant descriptors $r(n)$.

$$r(n) = \left[|a(n)|^2 + |b(n)|^2 \right]^{1/2} \quad (9)$$

It is easy to show that $r(n)$ is invariant to rotation or shift. A further refinement in the derivation of the descriptors is realized if dependence of $r(n)$ on the size of the character is eliminated by computing a new set of descriptors $s(n)$ as

$$s(n) = r(n)/r(1) \quad (10)$$

The Fourier coefficients $a(n)$, $b(n)$ and the invariant descriptors $s(n)$, $n = 1, 2, \dots, (L-1)$ were derived for all of the character specimens [5].

4.1 RCS Algorithm

The major steps of the algorithm are as follows:

1. Initialize all W_{ij} s to small random values with W_{ij} being the value of the connection weight between unit j and unit i in the layer below.
2. Present the 16-dimensional input vector y_0 , input vector consists of eight fourier descriptors and eight border transition values. Specify the desired outputs. If the net is used as a classifier then all desired outputs are typically set to zero except for that corresponding to the class the input is from.
3. Calculate the outputs y_j of all the nodes using the present value of W , where W_{ij} is the value of connection weight between unit j and the unit4 in the layer below:

$$y_i = \frac{1}{1 + \exp(-\sum_j y_j w_{ij})} \quad (11)$$

This particular nonlinear function is called a function sigmoid

4. Adjust weights by :

$$W_{ij}(n+1) = W_{ij}(n) + \alpha \delta_j y_i + \xi (W_{ij}(n) - W_{ij}(n-1))$$

where $0 < \xi < 1$

(12)

where $(n+1)$, (n) and $(n-1)$ index next, present and previous, respectively. The parameter α is a learning rate similar to step size in gradient search algorithms, between 0 and 1 which determines the effect of past weight changes on the current direction of movement in weight space. δ_j is an error term for node j . If node j is an output node, d_j and y_j stand for, respectively, the desired and actual value of a node, then

$$\delta_j = (d_j - y_j) y_j (1 - y_j) \quad (13)$$

If node j is an internal hidden node, then :

$$\delta_j = y_j (1 - y_j) \sum_k \delta_k w_{kj} \quad (14)$$

Where k is over all nodes in the layer above node j .

5. Present another input and go back to step (2). All the training inputs are presented cyclically until weights stabilize (converge).

4.2. Structure Analysis of RCS

The recognition performance of the RCS will highly depend on the structure of the network and training algorithm. In the proposed system, RCS has been selected to train the network [8]. It has been shown that the algorithm has much better learning rate. Table 1 shows the comparison of various approach classification. The number of nodes in input, hidden and output layers will determine the network structure.

Type of classifier	Fonts	Error	Efficiency
SVM	1	0.001	91%
SVM	more	0.2	88%
SOM	1	0.02	91%
SOM	more	0.1	88%
FNN	1	0.06	90%
FNN	more	0.2	87%
RBN	1	0.04	88%
RBN	more	0.1	90%
RCS	1	0	97%
RCS	more	0.0001	95%

Table 1 Comparison of classifiers

4.3. Number of Hidden Layer Nodes

The number of hidden nodes will heavily influence the network performance. Insufficient hidden nodes will cause under fitting where the network cannot recognize the numeral because there are not enough adjustable parameter to model or to map the input output relationship. The minimum number of epochs taken to recognize a character and recognition efficiency of training as well as test character set as the number of hidden nodes is varied. In the proposed system the training set recognition rate is achieved and in the test set the recognized speed for each character is 0.1sec and accuracy is 97%. The training set produced much higher recognition rate than the test set. Structure analysis suggested that RCS is given higher recognition rate. Hence Unicode is chosen as the encoding scheme for the current work. The scanned image is passed through various blocks of functions and finally compared with the recognition details from the mapping table from which corresponding unicodes are accessed and printed using standard Unicode fonts so that the Character Recognition is achieved.

5. EXPERIMENTAL RESULTS

The invariant Fourier descriptors feature is independent of position, size, and orientation. With the combination of RCS and back propagation network, a high accuracy recognition system is realized. The training set consists of the writing samples of 25 users selected at random from the 40, and the test set, of the remaining 15 users. A portion of the training data was also used to test the system. In the training set, a recognition rate of 100% was achieved and in the test set the recognized speed for each character is 0.1sec and accuracy is 97%. Understandably, the training set produced much higher recognition rate than the test set. Structure analysis suggested that RCS with 5 hidden nodes has lower number of epochs as well as higher recognition rate.

6. CONCLUSION

Character Recognition is aimed at recognizing handwritten Tamil document. The input document is read preprocessed, feature extracted and recognized and the recognized text is displayed in a picture box. The TamilCharacter Recognition is implemented using a Java Neural Network. A complete tool bar is also provided for training, recognizing and editing options. Tamil is an ancient language. Maintaining and getting the contents from and to the books is very difficult. In a way Character Recognition provides a paperless environment. Character Recognition provides knowledge exchange by easier means. If a knowledge base of rich Tamil contents is created, it can be accessed by people of varying categories with ease and comfort.

7. ACKNOWLEDGEMENT

The researchers would like to thank S. Yasodha for his assistance in the data collection and manuscript preparation of this article. Ms.S. Yasodha is currently a student in the department of Computer science at the Anna University of Technology, Chennai.

8. REFERENCES

- [1] B. Heisele, P. Ho, and T. Poggio, "Character Recognition with Support Vector Machines: Global Versus Component Based Approach," in ICCV, 2006, vol. 02, no. 1. pp. 688–694.
- [2] Julie Delon, Agnès Desolneux, "A Nonparametric Approach for Histogram Segmentation," IEEE Trans. On image processing., vol. 16, no. 1, pp. 235-241. 2007
- [3] B. Sachine, P. Manoj, M.Ramya, "Character Segmentations," in Advances in Neural Inf. Proc. Systems, vol. 10. MIT Press, 2005, vol.01, no 02 pp. 610–616.
- [4] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for Histogram-based Image Classification," IEEE Transactions on Neural Networks, special issue on Support Vectors, vol 05, no 01, pp. 245-252, 2007.
- [5] Simone Marinai, Marco Gori, "Artificial Neural Networks for Document Analysis and Recognition "IEEE Transactions on pattern analysis and machine intelligence, vol.27, no.1, Jan 2005, pp. 652-659.
- [6] M. Anu, N. Viji, and M. Suresh, "Segmentation Using Neural Network," IEEE Trans. Patt. Anal. Mach. Intell., vol. 23, pp. 349–361, 2006.
- [7] B. Scholkopf, P. Simard, A. Smola, and V. Vapnik, "Prior Knowledge in Support Vector Kernels," in Advances in Neural Inf. Proc. Systems, vol. 10. MIT Press, 2007, pp. 640–646.
- [8] Olivier Chapelle, Patrick Haffner, "SOM for Histogram-based Image Classification," IEEE Transactions on Neural Networks, 2005. Vol 14 no 02, pp. 214-230.
- [9] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral Partitioning with Indefinite Kernels Using the Nystrom Extention," in ECCV, part III, Copenhagen, Denmark, 2006, vol 12 no 03, pp. 123-132
- [10] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization Networks and Support Vector Machines," Advances in Computational Mathematics, vol. 13, pp. 1–11, 2005.
- [11] P. Bartlett and J. Shawe Taylor, "Generalization performance of support vector machines and other pattern classifiers," in Advances in Kernel Methods Support Vector Learning. 2008, MIT Press Cambridge, USA, 2002, vol 11 no 02, pp. 245-252.
- [12] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector machines: an application to face detection," in IEEE CVPR'07, Puerto Rico, vol 05 no 01, pp. 354-360, 2007.
- [13] V. Johari and M. Razavi, "Fuzzy Recognition of Persian Handwritten Digits," in Proc. 1st Iranian Conf. on Machine Vision and Image Processing, Birjand, vol 05 no 03, 2006, pp. 144-151.
- [14] P. K. Simpson, "Fuzzy Min-Max Neural Networks- Part1 Classification," IEEE Trans. Neural Network., vol. 3, no. 5, pp. 776-786, 2002.
- [15] H. R. Boveiri, "Scanned Persian Printed Text Characters Recognition Using Fuzzy-Neural Networks," IEEE Transaction on Image Processing, vol 14, no 06, pp. 541-552, 2009.
- [16] D. Deng, K. P. Chan, and Y. Yu, "Handwritten Chinese character recognition using spatial Gabor filters and self-organizing feature maps", Proc. IEEE Inter. Confer. On Image Processing, vol. 3, pp. 940-944, 2004.