

Handwritten Tamil Character Recognition Using SVM

Prof. Dr.J.Venkatesh¹, C. Sureshkumar²

¹ Assistant Professor – Systems and Production, Anna University Coimbatore, Tamilnadu, India.

venkijv@rediffmail.com

² Assistant Professor- CSE, J.K.K.M College of Technology, Tamilnadu, India.

ck_sureshkumar@yahoo.co.in

Abstract: *Hand written Tamil Character recognition refers to the process of conversion of handwritten Tamil character into Unicode Tamil character. 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. Each image glyph is subjected to feature extraction procedure, which extracts the features such as character height, character width, number of horizontal lines(long and short), number of vertical lines(long and short), horizontally oriented curves, the vertically oriented curves, number of circles, number of slope lines, image centroid and special dots. The extracted features considered for recognition are given to Support Vector Machine (SVM) 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. This approach can be extended to recognition and reproduction of hand written documents in South Indian languages.*

Keywords: Character recognition, Unicode, Support Vector Machines (SVM).

1. Introduction

Tamil is an ancient language with a rich literary tradition and Ancient India was popular in several fields such as medicine, astronomy and business. Ancient people recorded their knowledge in various fields in palm leaves. The handwritten text written in palm leaves decayed over a period of time. It is very difficult to preserve them in the same form. This paper proposes a new approach for converting handwritten Tamil script using unicode. The style of writing and the font were different compared to present day scripts. Lot of software tools is available only to read present day printed Tamil text with better recognition and accuracy.

1.1 Tamil Language

Tamil is a South Indian language spoken widely in Tamilnadu in India. Handwritten character recognition is a difficult problem due to the great variations of writing styles, different size and orientation angle of the characters. Among different branches of handwritten character

recognition it is easier to recognize English alphabets and numerals than Tamil characters. 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.

1.2 Vowels

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. The diphthongs are usually pronounced about 1.5 times as long as the short vowels, though most grammatical texts place them with the long vowels.

1.3 Consonants

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. Plosives are unvoiced if they occur word-initially or doubled. Elsewhere they are voiced, with a few becoming fricatives intervocalically. Nasals and approximants are always voiced. As commonplace in languages of India, Tamil is characterised by its use of more than one type of coronal consonants. Retroflex consonants include the retroflex approximant, which among the Dravidian languages is also found in Malayalam (example Kozhikode), disappeared from Kannada in pronunciation at around 1000 AD (the

dedicated letter is still found in Unicode), and was never present in Telugu. Dental and alveolar consonants also contrast with each other, a typically Dravidian trait not found in the neighboring Indo-Aryan languages.

1.4 Tamil Unicode

The Unicode Standard is the Universal Character encoding scheme for written characters and text. It defines the uniform way of encoding multilingual text that enables the exchange of text data internationally and creates the foundation of global software. The Tamil Unicode range is U+0B80 to U+0BFF [3].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 **ஆ** is 0BAE+0BC0. The Unicode is designed for various other Tamil characters.

2. Tamil character recognition functional block diagram

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.

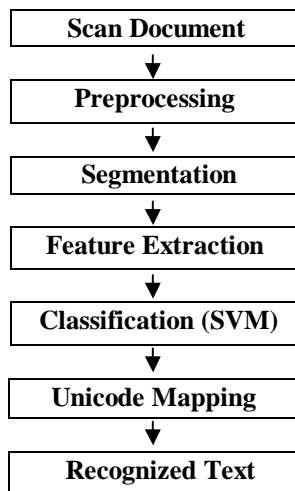


Figure 1. Handwritten Character Recognition System

2.1 Character Recognition Functions—Phase I

This phase includes the scanning, preprocessing, segmentation and feature extraction.

2.2 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. This is the first step in OCR. The

size of the input image is as specified by the user and can be of any length but is inherently restricted by the scope of the vision and by the scanner software length.

2.3 Preprocessing

This is the first step in the processing of scanned image. The scanned image is pre processed 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 2. Histograms for skewed and skew corrected images



Figure 3. Original Texts

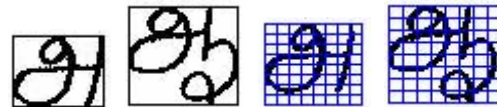


Figure 4. Character Segmentation

2.4 Segmentation

After pre-processing, the noise free image is passed to the segmentation phase, where the image is decomposed into individual characters. Fig.4 shows the image and various steps in segmentation.

Algorithm for segmentation:

- (1) The binarized image is checked for inter line spaces.
- (2) If inter line spaces are detected then the image is segmented into sets of paragraphs across the interline gap.
- (3)The lines in the paragraphs are scanned for horizontal space intersection with respect to the background. Histogram of the image is used to detect the width of the horizontal lines. Then the lines are scanned vertically for vertical space intersection.

Here histograms are used to detect the width of the words. Then the words are decomposed into characters using character width computation.

2.5 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. Character Recognition Functions Phase II

The second phase of the Character Recognition functions consists of classification and Unicode mapping and recognition strategies.

3.1 Classification

The various classification methods are as follows:

3.1.1 A typical rule based Classifier

The height of the character and the width of the character, various distance metrics are chosen as the candidates for classification when conflict occurs. Similarly, the classification rules are written for other characters. This method is a generic one since it extracts the shape of the characters and need not be trained. When a new glyph is given to this classifier block it extracts the features and compares the features as per the rules and then recognizes the character and labels it.

3.1.2 Support Vector Machine based classifier

The architecture chosen for classification is Support Vector machines, which in turn involves training and testing the use of Support Vector Machine (SVM) classifiers has gained immense popularity in recent years. SVMs have achieved excellent recognition results in various pattern recognition applications [1]. Also in off-line 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. They can implement flexible decision boundaries in high dimensional feature spaces. The implicit regularization of the classifier's complexity avoids over fitting and mostly this leads to good generalizations. Some more properties are commonly seen as reasons for the success of SVMs in real-world problems. [11] The optimality of the training result is guaranteed.

3.2 Classification using SVM

Support Vector Machine (SVM) is a classifier which performs classification tasks by constructing hyper planes in a multidimensional space [12]. It supports both classification and regression tasks. It is classified into two types namely 1. Classification SVM type-1(also known as C-SVM classification). 2. Classification SVM type-2 (also known as nu - SVM classification).



Figure 5. Classification using SVM

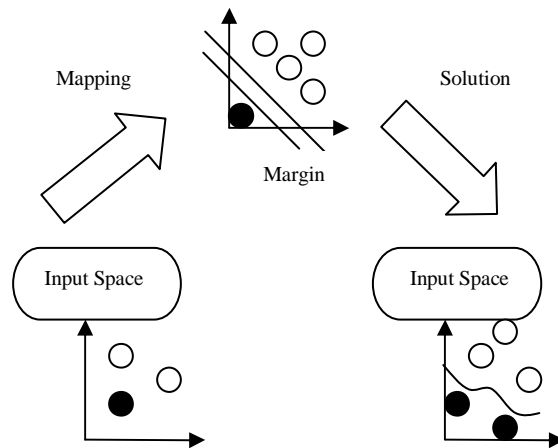


Figure 6. The SVM Classification Algorithm

3.2.1 Classification SVM Type-1

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

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

subject to the constraints:

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

Where C is the capacity constant, w is the vector of Coefficients, b a constant and ξ_i are parameters for handling nonseparable data (inputs). The index i label the N training cases [13]. Note that $y \in \pm 1$ represents the class labels and x_i 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. Thus, C should be chosen with care to avoid over fitting.

3.2.2 Classification SVM Type-2

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

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

subject to the constraints:

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

3.3 Kernel Function

s

There are number of kernels that can be used in Support Vector Machine models. These include 1. Linear, 2. Polynomial, 3. Radial basis function (RBF), 4. Sigmoid.

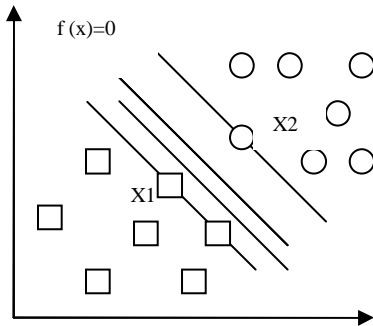


Figure 7. Sample SVM Classification

SVM consists of a learning module (svm_learn) and a classification module (svm_classify)

4. Unicode Mapping

The Unicode standard reflects the basic principle which emphasizes that each character code has a width of 16 bits. Unicode text is simple to parse and process and Unicode characters have well defined semantics [3] [7]. Hence Unicode is chosen as the encoding scheme for the current work. After classification the characters are recognized and a mapping table is created in which the unicodes for the corresponding characters are mapped.

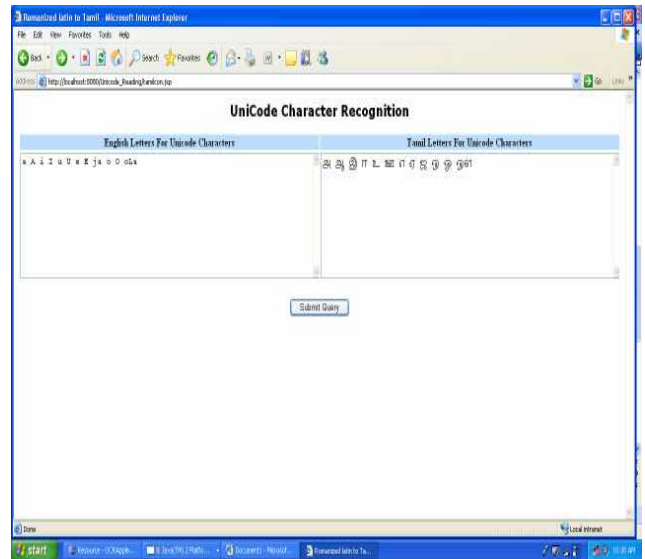


Figure 10. Sample Tamil Letters for Unicode Characters

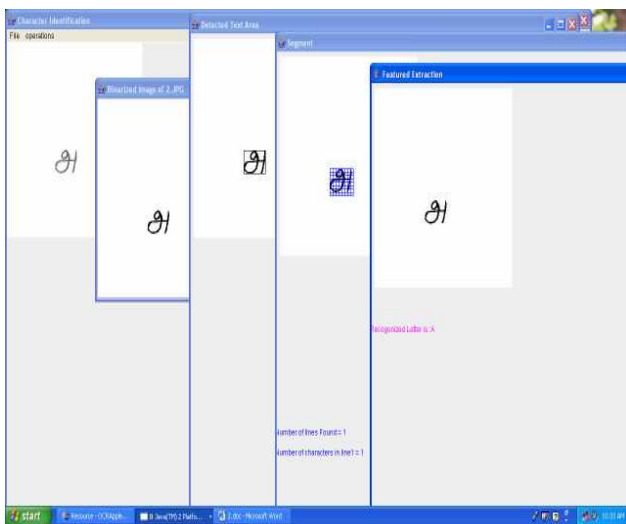


Figure 8. Sample identified character

5. Character recognition

The scanned image is passed through various blocks of functions and finally compared with the recognition details from the mapping table [6] from which corresponding unicodes are accessed and printed using standard Unicode fonts so that the Character Recognition is achieved.

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 Tamil Character 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. Character Recognition eliminates the difficulty by making the data available in handwritten format. 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.

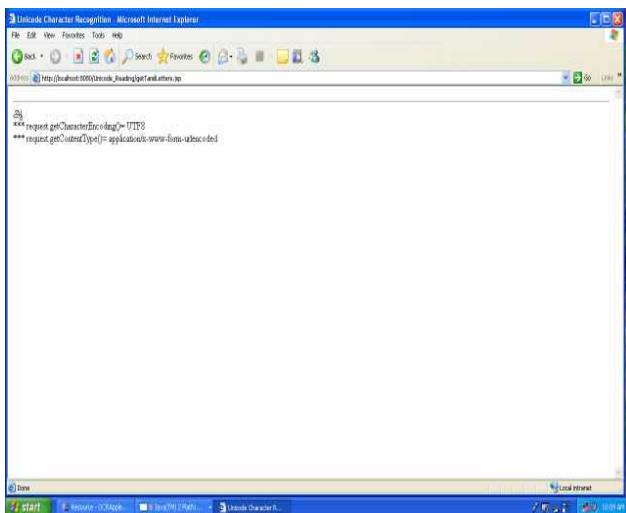


Figure 9. Sample Character Encoding

References

- [1] C. Papageorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision*, vol. 38, no. 1, pp. 15–33, 2000.
- [2] G. Guodong, S. Li, and C. Kapluk, "character recognition by support vector machines," in *Proc. IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 196–201.
- [3] K. Jonsson, J. Matas, J. Kittler, and Y. Li, "Learning Support Vectors for face verification and recognition," in *Proc. IEEE Int Conf on AutomaticFace and Gesture Recognition*, 2000.
- [4] B. Heisele, P. Ho, and T. Poggio, "Face recognition with support vector machines: global versus component-based approach," in *ICCV*, 2001, pp. 688–694.
- [5] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by Components," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 23, pp. 349–361, 2001.
- [6] 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, 1998, pp. 640–646.
- [7] O. Chapelle, P. Haffner, and V. Vapnik, "SVMs for histogram-based image classification," *IEEE Transactions on Neural Networks*, special issue on Support Vectors, 1999.
- [8] F. Jing, M. Li, H. Zhang, and B. Zhang, "Support Vector Machines for region-based image retrieval," in *Proc. IEEE International Conference on Multimedia and Expo*, 2003.
- [9] S. Belongie, C. Fowlkes, F. Chung, and J. Malik, "Spectral partitioning with indefinite kernels using the nyström extension," in *ECCV*, part III, Copenhagen, Denmark, may 2002,
- [10] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *Proceedings of the International Conference on Computer Vision*, vol. I, 2003, p. 257ff.
- [11] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Advances in Computational Mathematics*, vol. 13, pp. 1–50, 2000.
- [12] V. Vapnik, *The nature of statistical learning Theory*. John Wiley and sons, New York, 1995.
- [13] D. M. J. Tax and R. Duin, "Uniform object generation for optimizing one-class classifiers," *Journal of Machine Learning Research*, Special Issue on Kernel methods, no. 2, pp. 155–173, 2002.

Authors Profile



Prof. Dr. J. Venkatesh received a MBA degree in 1997 and a PhD in System Information in 2008, both from the University of Bharathiar, Tamilnadu, India. He is working as an Assistant Professor at Anna University Coimbatore, Tamilnadu, India, specialised in the field of Systems and Production. He published many papers on computer vision applied to automation, motion analysis, image matching, image classification and view-based object recognition and management oriented empirical and conceptual papers in leading journals and magazines. His present research focuses on statistical learning and its application to computer vision and image understanding and problem recognition.

Mr. C. Sureshkumar received a ME degree in Computer Science in 2006, from the University of Anna, Tamilnadu, India. He is a part-time PhD research scholar in the Department of Computer Science and Engineering, Anna University Coimbatore. His main interests in research are Hand written Tamil Character recognition refers to the process of conversion of handwritten Tamil character into printed Tamil character.