ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 8(S7), 110–116, April 2015*
DOI: 10.17485/ijst/2015/v8iS7/67780

Offline Tamil Handwritten Character Recognition Using Sub Line Direction and Bounding Box Techniques

S. M. Shyni[1*], M. Antony Robert Raj[2] and S. Abirami[2]

[1]Department of Electrical and Electronics Engineering, Sathyabama University, Chennai, India; shynima@gmail.com
[2]Department of Information Science and Technology, Anna University, Chennai, India; antorobert@gmail.com, abirami_mr@yahoo.com

Abstract

Character recognition plays an important role in the field of pattern recognition. Offline character recognition methodology mainly focuses on recognizing the characters irrespective of the difficulties that may arise due to the variations in writing style. This writing style becomes more complex when the characters are in curvy structure. The proposed recognition methodology was applied on one of the complex structures of south Indian language 'Tamil'. The novelty behind this process lies on the selection and extraction of the feature sets. Zoning and Chain Code procedures are employed here to select the features and Sub Line Direction and Bounding box algorithms are used for extracting the features. In order to achieve a better recognition rate, a learning algorithm, Support Vector Machine (SVM) has been implemented. These concepts are experimented on 30 Tamil character sets (Vowels and Consonants) and achieved an accuracy rate of 88%.

Keywords: Chain Code, OCR, SVM, Zoning

1. Introduction

Popularizing the language and reaping the benefits of available documents (ancient or present) of language is needed for present generation. Optical Character Recognition (OCR), interprets the scanned document image into machine understandable language. OCR system has two major phases: they are Offline and Online recognitions. Offline recognition is the process of identifying the characters from printed[4] or handwritten document images whereas online recognition recognizes the characters from the pen tip movement while writing characters. Recognition becomes quite difficult when one comes across complex structure of characters. The complexity will increase based on the nature of the language structure and writing style of various individuals. The writing style varies based on the different individual's age, mood and writing pressure.

Comparing with online recognition, researches on offline recognition is more complicated. Especially structures of Indian languages are highly hard to recognize. Tamil is one of the South Indian languages which has more character sets (247 characters) with 12 vowels, 18 consonants, 216 combinational characters, and one special character. The ancient structures of the Tamil characters are cursive ('Vatteluthu'). Much hand written ancient documents are available in Tamil with life giving word treasures. Hence choosing Tamil character recognition for the research is challengeable.

In Tamil offline handwritten OCR System, researchers have contributed various works to attain good recognition results. But still accuracy is yet to be improved. Tamil OCR framework consists of various stages such as Preprocessing, Segmentation, Feature Selection, Feature Extraction and Classification. Better features lead to better result. To obtain better results, keen focus is required on the phases such as features selection and extraction. Therefore, this research concentrates on a better feature selection and extraction to improve recognition. The coming section-2 describes the literature of Tamil OCR.

*Author for correspondence

Section-3 discusses about our novel selection, extraction mechanisms. Section-4 deals with the system results and finally section-5 concludes the work with future direction.

2. Literature Survey

Initially, Neural network concept was implemented by Jun Cao et. al,[3] to predict the exact handwritten numerals. Directional Chain Code and histogram based procedures were implemented in order to achieve high results[3].

Zoning and Projection based feature extraction techniques were implemented by Rajashekararadhya et al[6] work. They have implemented a neural network for achieving good results in their recognition system. Zoning[10,12,13] based centroid features were considered in another works of Rajashekararadhya et al[6], where SVM and neural network concepts were implemented for recognizing Tamil Numeral character features to achieve better recognition rate.

Shanthi and Duraisamy[8,14] experimented SVM based recognition system, where zone based pixel density was calculated as feature to get considerable result.

Later, Local Histogram of the Freeman's Chain Codes was computed by Bhattacharya et al[15]. Here, for each zone of the character, Multi-Layer Perception Algorithm was experimented to achieve accuracy. Two stages are followed in this recognition procedure, Unsupervised Clustering Method to group handwritten character classes, and Supervised algorithm for classifying the characters.

Pal et al[17] tried out directional features for recognition purpose. Here, the features are collected from the directional information. Here, the images are placed in bounding box, and gets segmented into blocks. These blocks were down sampled by the Gaussian filters to extract the required features. Two sets of features were employed for high speed recognition and high accuracy. They took 64 and 400 Dimensional feature sets with a modified quadratic classifier to classify the results.

3. Sub Line Direction and Bounding Box Based Handwritten OCR System

Selecting and extracting good features are essential to achieve good result in OCR System. Directions as features were extracted by Chain Code and shapes as a features were extracted by Bounding Box procedures. The following architecture (Figure 1) shows the structure of multiple processing systems of OCR.

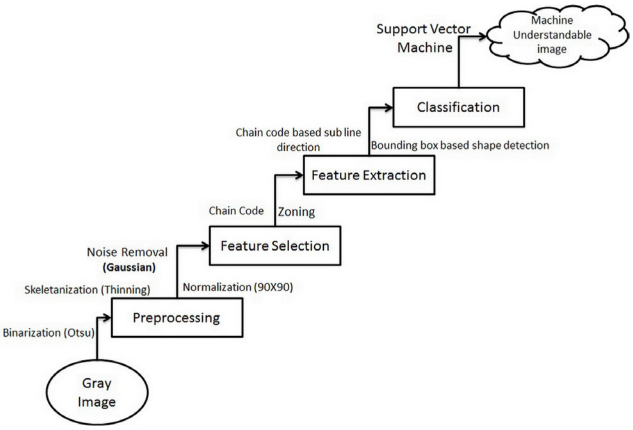

Figure 1. Structure of Tamil OCR system.

3.1 Pre-Processing

A clear and noise less image after preprocessing is required to recognize the characters accuracy, they are Bianrization, Skeletonization, Normalization and Noise removal.

Binarization is the basic operation to simplify the recognition work. Binarization is a process used to separate foreground and background portion of the given image. Otsu's Thresholding[1,5] procedure has been implemented, where histogram based threshold values are calculated for dividing the classes of a given image. Based on this, Threshold values the image was converted from gray scale image to binary images to a binary image (0-background, 1-foreground).

Binarized image may have noise disturbances, hence unwanted pixel portions may be present in the character image. In order to reduce those unwanted pixels, Gaussian filter has been employed here.

Skeletonization is the key part of pre-processing, where next to this we consider the structure of character features for recognition work. Skeletonization process helps us to get a single thin line structure of a foreground image without affecting the general shape. Thinning algorithm[6] has been used here to get the single structure. Finally the image has been brought back to the standard size to get the proper features. Normalization process attains the standard shape of the image, which is about (90X90).

3.2 Feature Selection and Extraction

3.2.1 Feature Selection

Later, zoning[3,2,6,10,12,13] has been applied over the resized image (90X90), and this zone was further sub divided into nine equal parts in which every zone has the dimension of 10X10. The segregated character parts are obtained in each sub zones. Eight Directional Chain Code[3] procedures have been applied on each sub zone to select the features.

A Row wise traversal over the character image has been performed to find a black pixel. If it is found, Chain Code Algorithm has been applied on that black pixel to find the black pixel with one neighbor or pixel with three neighbors called as junction point. This is recursively done from the first until all black pixels are visited. The Chain Code travels in anticlockwise direction[3].

Algorithm 1

```
Step 1:  [INITIALIZE - IMAGE SIZE – 90X90 → ROW COUMN]
Step 2:  DIVIDE [ROW COLUMN] → 9 (10X10)
Step 3:  [INITIALIZE i=1 to N, IMAGE(10X10) [Xi=0]
Step 4:  FOR EACH SUB IMAGES
Step 5:  SIZE → [R1, C1]
Step 6:  [FOR EACH (r to R1)
Step 7:  If POSITION(r, C1) equals to 1
Step 8:  X1 ← POSITION
Step 9:  FIND - REVERSLY (ONE OR TWO of 8-ADJACENT (r, C1) OF POSITION equals to 1)
Step 10: DO (ADJACENT (r, C1) →POSITION)
Step 11: REPEAT FROM STEP 7
Step 12: FIND – (TWO AND ABOVE of 8-ADJACENT (r, C1) OF POSITION equals to 1)
Step 13: REPEAT FROM STEP 5
Step 14: FIND – (ONE of 8-ADJUCENT (r, C1) OF X1 equals to 1)
Step 15: RETURN X1
Step 16: REPEAT FROM STEP 5
```

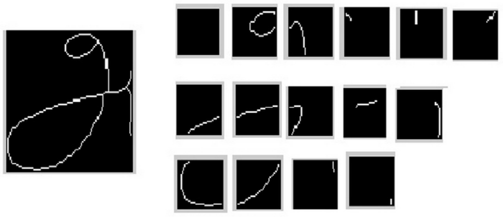

Figure 2. Seleceted features by chain code – sample.

[Repeat the step for images until all features are collected]

If black pixels are available but if the chain code fails to find any pixel with one neighbor, then it might be in a circular shape. Here, the traversal would be started by the Chain Code from any one of the black pixel with two neighbors until it comes to an end with the same pixel.

Here, new images (Xi, where i=1 … n) has been constructed using the visited pixels and hence used for features extraction. Figure 2 shows the sample features selected from the character 'அ'.

3.2.2 Feature Extraction

This is the most important stage of Tamil OCR system. Relevant information from the selected image has been extracted for classification. Different shapes of the character parts have been selected from feature selection algorithm. They might be curves or points or linear shapes. But most of them are open curve shaped.

In feature extraction, two techniques are proposed here. One is chain code based sub line direction and the other one is bounding box based shape detection. Here, directions are extracted by sub line direction and bounding box procedures from selected portion of character image.

3.2.2.1 Chain Code based Sub Line Direction

In this procedure, each row of the each selected feature image (Xi) from top left point has been traversed to find the two end points (A1(ra1 ca1), A2(ra2, ca2)) with one neighbor as shown in Figure 2. Then using those points mid-point (A3) of the curve has been computed using equation 1.

$$A3 = \sqrt{(ra2 - ra1)^2 + (ca2 - ca1)^2} \qquad (1)$$

Later, second and third sub mid points (A4) (A6) has been selected as shown in the Figure 2. Similarly A5, A7 are also obtained. Later, a traversal has been made from the third sub mid-point A6 to A7 towards the search of a first black pixel resulting in (R1,R2). A line has been drawn from R1 to R2, and it was considered as D (Figure 3).

8-Directional Chain Code Algorithm has been applied over D to find the direction. This provides the approximate direction of the particular curve.

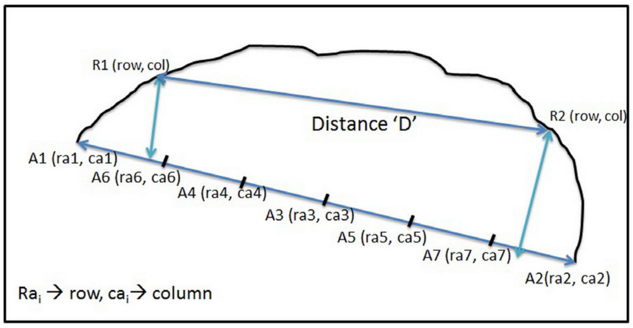

Figure 3. Sub line direction.

Based on the direction the directional vectors are gathered and taken as features.

- Vertical (top to bottom or bottom to top) direction has been taken as '1'
- Horizontal (left to right or right to left) direction has been taken as '2'
- Diagonal (left bottom to right top or right top to left bottom) direction has been taken as '3'
- Diagonal (left top to right bottom or right bottom to left top) direction has been taken as '4'
- If it was a closed curve (no end found among black pixels) the direction has been considered as '5'
- If it is a dot (only one black pixel) then the direction has been considered as '6'

The Figure 4 shows the collected sub line direction features from the character 'அ'.

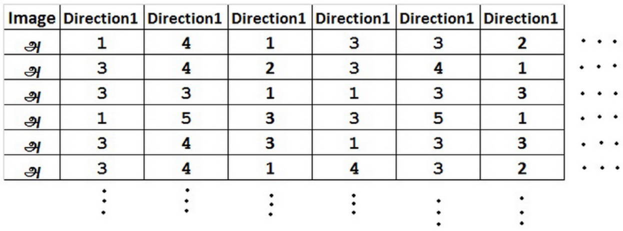

Image	Direction1	Direction1	Direction1	Direction1	Direction1	Direction1	
அ	1	4	1	3	3	2	. . .
அ	3	4	2	3	4	1	. . .
அ	3	3	1	1	3	3	. . .
அ	1	5	3	3	5	1	. . .
அ	3	4	3	1	3	3	. . .
அ	3	4	1	4	3	2	. . .
	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 4. Extracted features by chain code 'அ'.

3.2.2.2 Bounding Box based Shape Detection

The bounding boxes are drawn over the character portion of the selected feature images (Xi). Then the following procedure has been applied to find the approximate shape of the feature.

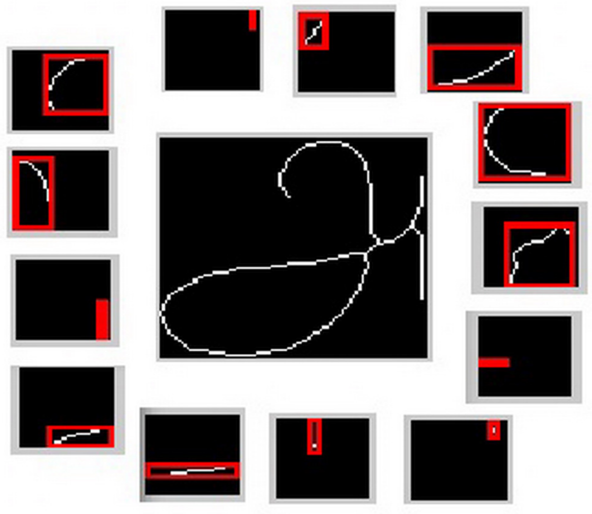

Figure 5. Bounding box – feature fets

Algorithm 2

```
[DRAW BOUNDING BOX ON CHRACTER PORTION]
[INITIALIZE [A1, B1] ] ← HEIGHT AND WIDTH OF BOUNDING BOX]
[INITIALIZE VARIABLE X1, R1, C1, P1 AND P2 AS ZERO]
 FOR EACH IMAGES_BOUNDING BOX PORTIONS
     FIND[A1 EQUALS TO B1]
       X1←1
     FIND [A1 ABOVE 2 AND B1 TWO AND BELLOW]
       X1 ← 2
     FIND [A1 AND B1 ABOVE 2]
       X1 ← 3
     FOR EACH SUB IMAGES
       SIZE → [R1, C1]
       [FOR EACH (r to 1) //ROW
         [FOR EACH (c to C1)
           IF POSITION(r, c) equals to 1
           P1 ← POSITION
       [FOR EACH (c to 1) //COLUMN
         [FOR EACH (r to R1)
           IF POSITION(c, r) equals to 1
           P2 ← POSITION
           CHECK(P1 AND P2 equals to 1)
           X1 ← 4
RETURN X1
[CONTINUE ALL BORDER AND FIND 5 to 11]
```

The height and width of the bounding box has found.

- If the height and width of the bounding box were the same, then it might be a dot, which has been considered as ‘1’.
- If the height >2 and width <=2 then it might be a vertical line or vertically diagonal line, which has been considered as ‘2’.
- If the height <=2 and width >2 then it might be a horizontal line or horizontally diagonal line, which has been considered as ‘3’.
- If the height >2 and width >2 then traverse the border of the bounding box are checked to find any black pixels. If the black pixels are found on left and top borders, then the shape was a curve from top to left or left to top direction, and has been considered as ‘4’.
- If the black pixels were found on left and bottom borders, then the shape was a curve on bottom to left or left to bottom direction, and it has been considered as ‘5’.
- If the black pixels were found on the right and top borders, then the shape was a curve on top to right or right to top direction, and has been considered as ‘6’.
- If the black pixels were found on right and bottom borders, then the shape was a curve on bottom to right or right to bottom direction, and has been considered as ‘7’.
- If the black pixels were found on left, right and top borders, then the shape was a curve on top direction, and has been considered as ‘8’.
- If the black pixels were found on left, right and bottom borders, then the shape was a curve on bottom direction, and has been considered as ‘9’.
- If the black pixels were found on top, bottom and right borders, then the shape was a curve on right direction, and has been considered as ‘10’.
- If the black pixels were found on top, bottom and left borders, then the shape was a curve on left direction, and has been considered as ‘11’

These features are considered for a better classification procedure to predict the exact character. The Figure 5 shows the features has been extracted by bounding box based feature extraction techniques from the character ‘அ’.

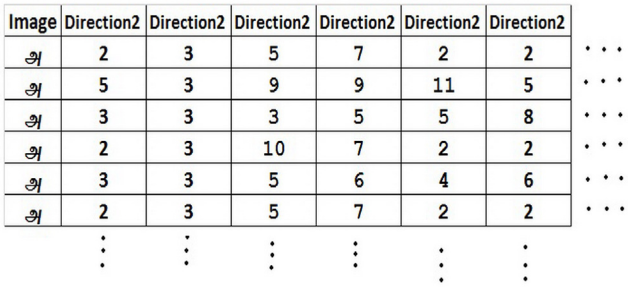

Image	Direction2	Direction2	Direction2	Direction2	Direction2	Direction2	
அ	2	3	5	7	2	2	. . .
அ	5	3	9	9	11	5	. . .
அ	3	3	3	5	5	8	. . .
அ	2	3	10	7	2	2	. . .
அ	3	3	5	6	4	6	. . .
அ	2	3	5	7	2	2	. . .
	⋮	⋮	⋮	⋮	⋮	⋮	

Figure 6. Shape features from the character ‘அ’.

3.3 Classification

These features have been collected and given as input into the Support Vector Machine (SVM)[8,9,15,16] which is a statistical classifier. Training samples are gathered from the features and used as support vectors in SVM. These support vectors are used for predicting the character classes.

The kernel function formula used for hyper plane is shown is equation (2)

$$(W.\ X_i) + B = 0\ldots\ldots\ldots \quad (2)$$

30 output classes are used for prediction. Multiclass-SVM was implemented in the experiment and threshold values were calculated from various training samples (X, Y). And it was used for taking a decision in SVM. Equation for multi class SVM is shown in equation (3)

$$Y_i\ (W^T.\ (X_i) + B) > a1 < Y_i\ (W^T.\ (X_i) + B) > a2 < Yi\ (W^T.\ (X_i) + B) <\ldots\ \ldots\ldots\ldots \quad (3)$$

where, Yi=1 or -1 and the a1, a2 … aN are the threshold values. W is the Weight factor calculated from Legrangian Theorem and B is the bias value.

4. Experimental Results

6000 samples of 30 Tamil characters are collected from HP India lab data set as well as from different hand-written documents. 3600 samples (120 samples * 30 characters) are gathered for training purpose in SVM, rest of them (80 samples * 30 characters) are used for testing purpose. The features of each data samples stored in Microsoft excel are fed in SVM algorithm which was written in Matlab. 88% accuracy rate was obtained from testing samples.

Table 1 and Figure 4 shows the representation of result and accuracy rate achieved.

Table 1. Data samples and accuracy achieved

S. No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Vowels & Consonents	அ	ஆ	இ	ஈ	உ	ஊ	எ	ஐ	ஏ	ஒ	ஓ	ஔ	க	ங	ச
Accuracy Achieved	87	86	90	90	88.2	90.2	88.5	85	87.5	86.1	86.8	92.8	90.3	91.1	86.9

S. No	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Consonents	ஞ	ட	ண	த	ந	ப	ம	ய	ர	ல	வ	ழ	ள	ற	ன
Accuracy Achieved	87.1	93.3	84	86.2	88.2	91.9	90.4	86	89.3	86.2	88.8	85.2	86.5	87.8	88.4

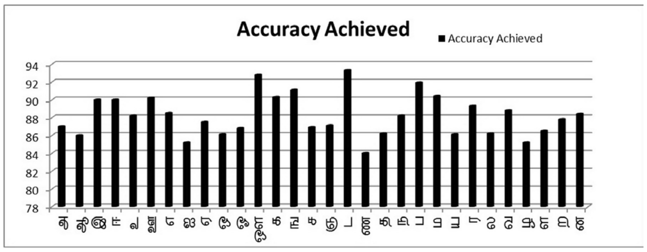

Figure 7. Accuracy rate for each chracter set.

5. Conclusion and Future Work

In this paper, zoning and chain code procedure has been experimented for selecting the features from various character samples. Bounding box based sh1ape detection and Chain code based sub line direction techniques are implemented to extract suitable features. 88% recognition rate has been achieved using SVM. This algorithm may be highly suitable for other characters also except similar shaped characters in Tamil.

In future, this work may be extended to all 247 character in Tamil language.

6. References

1. Antony Robert Raj M, Abirami S. A survey on Tamil handwritten character recognition using ocr techniques. The Second International Conference on Computer Science, Engineering and Applications (CCSEA). 2012; 5:115–27.
2. Antony Robert Raj M, Abirami S. Analysis of statistical feature extraction approaches used in tamil handwritten ocr. 12th Tamil Internet Conference – INFITT. 2014; p.144–50.
3. Antony Robert Raj M, Abirami S. Offline Tamil handwritten character recognition using chain code and zone based features. 13th Tamil Internet Conference- INFITT. 2014; p. 28–34.
4. Abirami S, Manjula D. Feature string based intelligent information retrieval from tamil document images. Int J Comput Appl Tech. Special Issue on Computer Applications in Knowledge Based Systems. Inter-science Publishers; 2009; 35(2/3/4):150–64.
5. Cao J, Ahmadi M, Shridhar M. Recognition of handwritten numerals with mutable feature and multistage classifier. Elsevier; Pattern Recogn. 1995; 28(2):153–60.
6. Rajashekararadhya SV, Vanaja Ranjan P, Manhunath Aradhya VN. Isolated handwritten Kannada and Tamil numeral recognition: a novel approach. First IEEE International Conference on Emerging Trends in engineering and Technology; 2008; p. 1192–95.
7. Liu CL, Nakashima K, Sako H, Fujisawa H. Handwritten digit recognition: benchmarking of state-of-art techniques. Elsevier, Pattern Recogn. 2003; 36:2271–85.
8. Shanthi N, Duraiswami K. A Novel SVM-based handwritten Tamil character recognition system. Springer; Pattern Analysis and Application. 2010; 13(2):173–80.
-9. Ramanathan R, Ponmathavan S, Thaneshwaran L, Arun S Nair, Valliappan N. Tamil font recognition using gabor and support vector machines. International Conference on Advances in Computing, Control and Telecommunication Technologies. 2009; p. 613–15.
10. Rajashekararadhya SV, Vanaja Ranjan P. Zone-based hybrid feature extraction algorithm for handwritten numeral recognition of two popular indian script. World Congress on Nature and Biologically Inspired Computing. 2009; p. 526–30.
11. Sigappi AN, Palanivel S, Ramalingam V. Handwritten document retrieval system for tamil language. Int J Comput Appl. 2011; 31(4):42–7. ISSN: 0975-8887,
12. Rajashekararadhya SV, Vanaja Ranjan P. Neural network based handwritten numeral recognition of Kannada and Telugu script. IEEE TENCON Conference; 2008; p. 1–5.

13. Rajashekararadhya SV, Vanaja Ranjan P. efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south indian scripts. Int J of Theoretical and Applied Information Technology. 2008; 1171–81.
14. Shanthi N, Duraiswami K. Performance comparison of different image size for recognizing unconstrained handwritten Tamil character using SVM. J Comput Sci. 2007; 3(9):760–4.
15. Bhattacharya U, Ghosh SK, Parui SK. A Two stage recognition scheme for handwritten Tamil characters. Ninth International Conference on Document Analysis and Recognition; 2007; 1:511–15.
16. Chanda S, Pal S, Pal U. Word-wise Sinhala Tamil and English script identification using Gaussian Kernel SVM. 2008; IEEE.
17. Pal U, Wakabayashi T, Kimura F. Handwritten numeral recognition of six popular scripts. Ninth International conference on Document Analysis and Recognition, ICDAR; 2007; 2:749–53.