Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 11; April-June, 2015 pp. 85-90
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Evaluation of Neural Based Feature Extraction Methods for Printed Telugu OCR System

M Swamy Das[1] and Ram Mohan Rao Kovvur[2]

[1]Dept. of CSE, CBIT, Hyderabad, India
[2]Dept. of CSE, Vasavi College of Engg. Hyderabad, India
E-mail: [1]msdas@cbit.ac.in, [2]krmrao@staff.vce.ac.in

Abstract—*The Telugu is one of the oldest and most popular languages of India, especially in South India. The reported works on development of optical character recognition (OCR) systems for Telugu script is little. Moreover, Telugu is a complex script in which the characters are made up of one or more connected components resulting in a huge number of possible combinations, running into hundreds of thousands. In any OCR system, feature extraction is one of the most important phases. There are several methods that are suitable for different language scripts. These methods are broadly classified into template base, structural, statistical, neural network based and SVM based. In this paper we describe various feature extraction methods and evaluate by applying to Telugu script. In this process we have identified diagonal based, geometrical based and distance metric based feature extraction methods and also proposed a Pixel based feature extraction method. All these methods are implemented and evaluated with 364 Telugu characters using multilayer neural network as a classifier. The recognition accuracies of geometrical, diagonal, pixelmap and distance metric based feature extraction methods are 98.6%, 100%, 98.31% and 99.32% respectively. From the experiment it is understood that diagonal based method most suitable for Telugu script than other feature extraction methods.*

1. INTRODUCTION

There are millions of old and historic document that are handwritten or machine printed. In order to make them available on the net, we need to digitize. OCR is a tool or software program that automatically translate the machine printed or handwritten documents into machine editable text document. Once it is converted then it can be stored in UNICODE format. There are several applications of OCR systems. These include as tool for blind people, processing bank cheques, sorting postal mails, compression, index creation, language translation etc [1-6].

Several commercial OCR systems with 100% recognition accuracy are available but most they are suitable for Latin based scripts. From the Indian context not much of the work on OCR is reported. Telugu is one of the most popular South Indian scripting languages. It is a complex script because of its structure and the size of the character set. Fig. 1 shows the Telugu script alphabet which comprises of vowels, consonants, maatras (vowel modifiers) and vaththus(consonant modifiers). With these components we can find several characters in Telugu script with very complex structure. So it is an area for research. The accuracy of any OCR system depends on the feature extraction method. In this work various feature extraction methods are identified and evaluated by using the Telugu script alphabet [7-10].

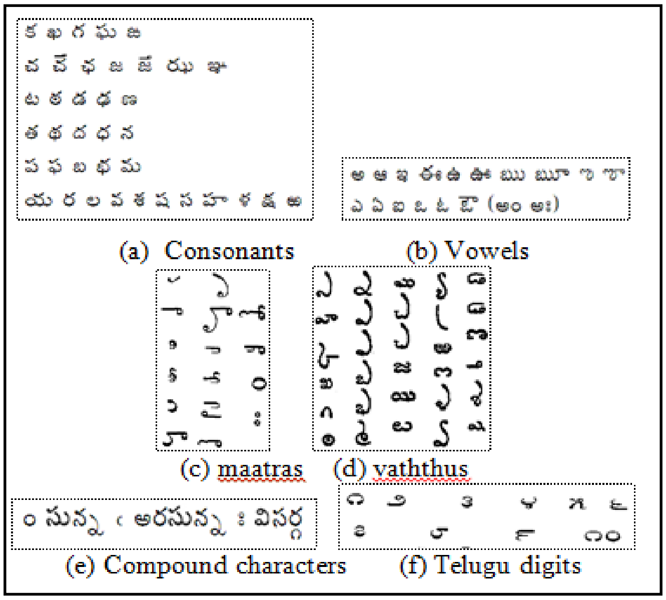


Fig. 1: Alphabet of Telugu Script

This paper is organized into six sections. Section two describes the basic operation of any OCR system, in section three the feature extraction methods including the proposed pixelmap methods are discussed, section four deals with the neural network design, section five with the implementation details, result analysis and finally section six gives the conclusion.

2. PROCESSING STEPS OF A TYPICAL OCR SYSTEM

A typical OCR system basically contains of image acquisition, preprocessing, segmentation, feature extraction & Selection and recognition components as shown in Fig. 2. Character images can be acquired from a scanner or any other electronic source. It will be stored any image format and that image may be a color, gray scale, but the actual processing take place on binary images. The preprocessing module translates the color/grayscale image into a binary image. It also detects and corrects the noise. Noise may be caused by the quality of the paper, print etc. and the skew may be due human interference in the image acquisition. There are several methods to detect and correct the noise as well as skew. One popular method may be the Otsu approach [11].

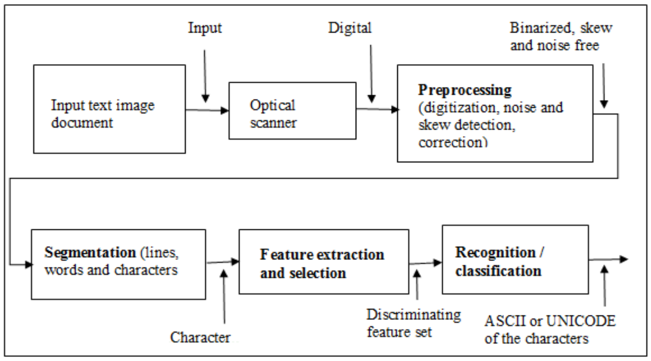

Input
Digital
Binarized, skew and noise free
Input text image document
Optical scanner
Preprocessing (digitization, noise and skew detection, correction)
Segmentation (lines, words and characters
Feature extraction and selection
Recognition / classification
Character
Discriminating feature set
ASCII or UNICODE of the characters

Fig. 2: Block diagram of a typical OCR System

Next module is the segmentation which takes noise and skew free binary image. This module divides the binary image into lines and words followed by the characters. For segmentation we have several methods that are based on connected components, projection profile based etc. [12-14]. The output from the segmentation module is a stream of character images which will be given as input

The feature extraction module will extract features that uniquely discriminates one character from the other characters. These feature extraction methods include template based, structural, statistical. We can find several feature extraction methods that are suitable for different scripts. The section three describes about the feature extraction methods that can be used with Telugu characters. The extracted feature set may contain redundant or irrelevant features. These features may be further reduced by using the corresponding feature selection algorithms like PCA (Principal Component Analysis), CFS (Correlation based Feature Selection)

The recognition module will recognize the characters by using the features extracted by the feature extraction method. There are several recognition/classification approaches including template, neural networks, decision tree based, Bayesian based, Support Vector Machines SVM etc. In this work we have used the neural network with multilayer perceptrons whose topology is described in section 4.

3. ENRICHED NEURAL NETWORK BASED FEATURE EXTRACTION ALGORITHMS

From the literature we have identified some of the feature extraction algorithms that are proposed by different people for different scripts and modified them to adopt for the Telugu script. Some of these methods are the structural or geometric, Diagonal based and the distance metric based feature extraction. The geometric base feature extraction approach has been propose by Dinesh Dileep [15-16] for the recognition of hand written characters of English, the diagonal based approach has been propose by J Pradeep [17] for the recognition of handwritten numerals and the distance metric approach was proposed by Rajasekharan [18] for the recognition of Kannada characters.

In this work we have modified those approaches and adopted for the recognition of Telugu characters and also proposed a simple pixel based approach named PixelMap.

3.1 Geometric Feature extraction Approach

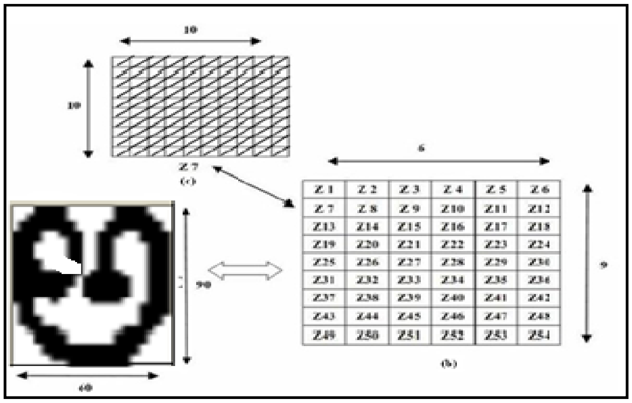
This method basically uses the geometric/ structural features such as lines. In this method the given binary image is divided into 9 equal sized zones. From each zone different line segments are extracted by traversing the zones. After the extraction of line segments they have to be classified into any one of the following four line types: vertical, horizontal, left diagonal and right diagonal lines. The number of each line type and their normalized values resulting eight values treated as features. It also extracts the area of each region resulting nine features. It also includes the global features like Euler number, eccentricity and extent. We have also added three more features of aspect ratio, convex and conclave curves that are very common in Telugu characters.

```
Algorithm: Geometric (Image, Feature_Vector)
Input    : Noise free character image of size MxN
Output: Feature vector of size 86
Begin
    1.  Divide the image into 9 (3x3)equal zones
    2.  For each zone determine
            a.  No. of horizontal lines, vertical lines,
                Right diagonal lines, Left diagonal lines(
                4 values)
            b.  Normalized Length of all the above
                values (4 values)
            c.  Normalized area of the skeleton
    3.  Compute the global parameters of Euler number,
        eccentricity, extent, convex and concave curve
        orientations.
    4.  Return extracted features i.e. Feature_Vector
End;
```

Fig. 3: Geometric based feature extraction algorithm

The Euler number is defined as the difference between the number of objects and the holes in the image. Eccentricity is defined as the smallest ellipse that fits the skeleton of the image. The extent is defined as the portion of the pixels in the bounding box that is also in the region.

The complete algorithm is given in Fig. 3. The normalized length of a line is calculated by dividing the total no. of pixels in each line type with the total pixels in that zone. Hence the number of features in the output vector would be 9x9+5=86.



Z 1	Z 2	Z 3	Z 4	Z 5	Z 6
Z 7	Z 8	Z 9	Z10	Z11	Z12
Z13	Z14	Z15	Z16	Z17	Z18
Z19	Z20	Z21	Z22	Z23	Z24
Z25	Z26	Z27	Z28	Z29	Z30
Z31	Z32	Z33	Z34	Z35	Z36
Z37	Z38	Z39	Z40	Z41	Z42
Z43	Z44	Z45	Z46	Z47	Z48
Z49	Z50	Z51	Z52	Z53	Z54

Fig. 4: Logical representation of Diagonal approach

```
Algorithm: Diagonal (Image, Feature_Vector)
Input   : Noise free character image of size MxN
Output: Extracted feature vector of size 69 (Feature_Vector).
Begin
    1. Divide the input image into 54 zones of each zone being 10x10
    2. For each zone
        a. If a zone is empty, set the feature value of that zone is 0
        b. Starting from the center diagonal in the input image matrix, sum up all the pixels moving upward and then move downward. (i.e. summing the pixels diagonal wise)
        c. Compute the average by dividing the pixel sum with the number of diagonals. (No. of diagonals are 19), which is the diagonal feature.
        d. Add the diagonal feature value in the Feature_Vector
    3. Sum all the diagonal features row wise and column wise
    4. Add these column and row features to the Feature_Vector
    5. Return the feature set Feature_Vector.
                    End;
```

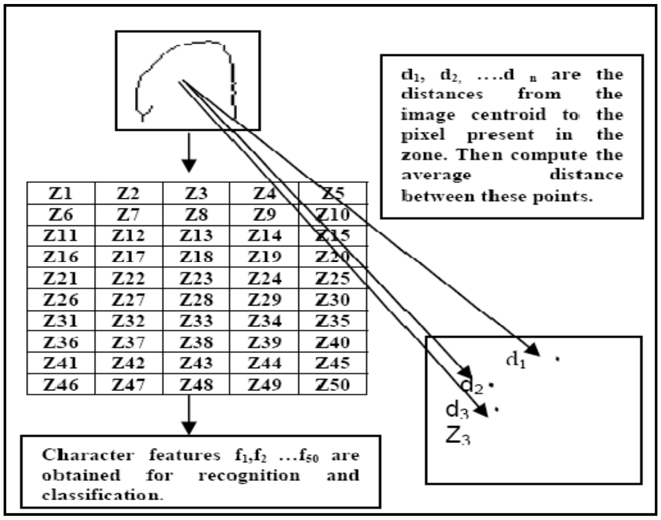
Fig. 5: Diagonal Feature extraction algorithm

3.2 Diagonal Approach

In this method given image is divided in to *m* (here it is 6 rows by 9 columns= 54) zones. The features are extracted from each zone pixels by counting the black pixels along diagonals. If a zone has n diagonal lines and the black pixels present long each diagonal line is summed to get a single sub-feature resulting n sub-features. These *n* sub-features values are averaged to form a single feature value. This procedure is sequentially repeated for the all the zones. If a zone is empty then the feature values corresponding to these zones are zero. Finally, the number of features is extracted for each character is equivalent to the number of zones. In addition, some more features are obtained by averaging the feature values of zones row wise and column wise. Finally, the number of features is extracted for each character is equivalent to the number of zones plus the number of zones vertically and horizontally (ie. 54+9+6). Fig. 4gives the logical representation and the complete procedure of diagonal approach. The Fig. 6 shows the complete algorithm:

3.3 Distance Metric Approach

In this approach first image centroid is computed. Then the image is divided into 50 equal zones and the average distance from the centroid of the character image to each pixel in the zone is calculated. This procedure is repeated to each zone and if zone is empty its values is set to 0. The logical representation of distance metric approach is explained by the Fig. 4.2.



Z1	Z2	Z3	Z4	Z5
Z6	Z7	Z8	Z9	Z10
Z11	Z12	Z13	Z14	Z15
Z16	Z17	Z18	Z19	Z20
Z21	Z22	Z23	Z24	Z25
Z26	Z27	Z28	Z29	Z30
Z31	Z32	Z33	Z34	Z35
Z36	Z37	Z38	Z39	Z40
Z41	Z42	Z43	Z44	Z45
Z46	Z47	Z48	Z49	Z50

d_1, d_2, ….d_n are the distances from the image centroid to the pixel present in the zone. Then compute the average distance between these points.

Character features f_1,f_2 …f_{50} are obtained for recognition and classification.

Fig. 6: Logical representation of Distance metric approach.

The formula for computing the zone feature value is:

Zone feature value= ∑(distance from all the black pixels to the image centroid)/distance between zone centroid and image centroid.

Hence the number of features for the given input character would be 50.

```
Algorithm: Distance_Metric(Image, Feature_Vector)
Input   : Noise free character image of size MxN (Image)
Output: Set of 50 features extracted from the input image (Feature_Vector).
Begin
    1.  Compute the centroid of the input image.
    2.  Divide the input image in to N zones. Here N is taken as 50.
    3.  For each zone
            a.  Compute the distance between the image centroid to each pixel present in that zone.
            b.  Repeat step 3 for all the pixels  present in the zone
            c.  Compute average distance between these points.
            d.  Compute the zone feature by dividing the average distance by the distance between the zone centroid to the image centroid.
            e.  Add this feature to the Feature_Vector.
    4.  Return the feature set Feature_Vector.
End;
```

Fig. 7: Distance metric feature extraction algorithm

3.4 Proposed PixelMap Approach

This approach tries to reduce the computational complexity by selecting random pixels. In this approach the feature vector is generated by resizing the image into an appropriate size. The given input character image is first resized to a uniform size of 30x20 and then preprocessed. After pre--processing then it is further reduced to a feature matrix of size 15x10 using random pixel selection procedure.

```
Algorithm 4.4: PixelMap(Image, Feature_Vector)
Input    : Noise free character image of size MxN (Image).
Output :Set of features extracted from the input image (Feature_Vector).
Begin
    1.  First resize the image to a uniform size.  For example  30x20
    2.  Preprocess the image (binarization, removal of noise and skew, thinning etc).
    3.  Now further reduce the image (Ex. m=15, n=10 i.e. 15x10) to reduce the computation time.
    4.  Extract features since the neural net needs one dimensional input vector. Initialize k=1;
                for each row i= 1 to m
                for each column j=1 to n
                Feature_Vector (k) = Image (i,j);
                k=k+1;
                end;
end;
    5.  Return the feature set Feature_Vector
End;
```

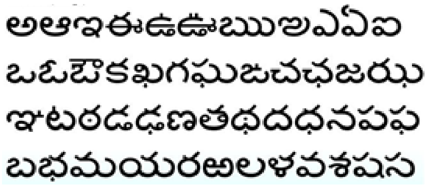
Fig. 8: PixeMap feature extraction algorithm

So there would be 150 pixels in the feature matrix. Since the neural network needs a 1-D vector, this 15x10 matrix is converted to a linear feature vector of size 150. The complete algorithm is given in Fig. 8.

4. NEURAL NETWORK DESIGN

In this work a MLPN network is used as a recognizer and a with back-propagation algorithm as the training algorithm. In this process a simple dataset with Telugu based characters shown in Fig. 9 is created and then trained. The network parameters like number of hidden layers, number neurons in the hidden layer, weights, bias values, no. of epochs are determined by trial and error basis [19-21]. We got good results with one hidden layer. The initial values set for different parameters are:

- Training algorithm: Gradient descent with momentum training and adaptive learning
- Perform function: Mean Square Error
- Training goal achieved: 0.000001(10^{-6})
- Max. Training epochs: 1000000
- Training momentum constant: 0.9



అఆఇఈఉఊఋఌఎఏఐ
ఒఓఔకఖగఘచఛజఝ
ఞటఠడఢణతథదధనపఫ
బభమయరఱలళవశషస

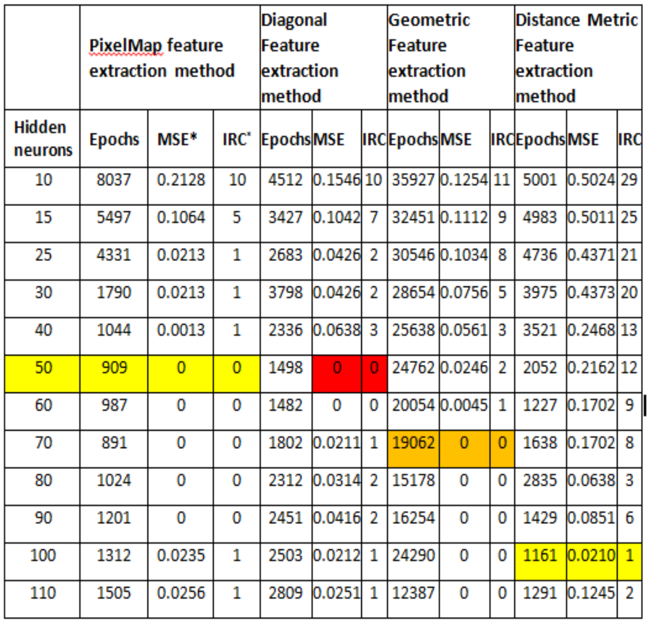
Fig. 9: Sample Telugu character set

The size of input layer depends on the feature extraction approach since different approaches extract different number of features. In the experimentation we have considered a Telugu training character set of size 47 as shown in Fig. 9, so the output layer size is 47.

5. IMPLEMENTATION AND RESULTS ANALYSIS

All the algorithms are implemented using MATLAB7.2. Since there ae no standard dataset for Telugu script, A sample dataset shown in Fig. 9 consisting of 47 base characters is created using a multilingual text editor. Then it is converted into an image format and trained using the network topology given in section 4. The number of hidden layers, number of neurons in the hidden layer and the epochs are determined from the experimentation by varying the parameters. The results are tabulated are tabulated in table 1. The trained network is tested with the same Telugu character set by using each feature extraction method. These results after training are plotted. The number of hidden neurons have been varied and by setting the number of epochs to a large value (1000000).

Table 1: Recognition accuracies of different methods



	PixelMap feature extraction method			Diagonal Feature extraction method			Geometric Feature extraction method			Distance Metric Feature extraction method		
Hidden neurons	Epochs	MSE*	IRC*	Epochs	MSE	IRC	Epochs	MSE	IRC	Epochs	MSE	IRC
10	8037	0.2128	10	4512	0.1546	10	35927	0.1254	11	5001	0.5024	29
15	5497	0.1064	5	3427	0.1042	7	32451	0.1112	9	4983	0.5011	25
25	4331	0.0213	1	2683	0.0426	2	30546	0.1034	8	4736	0.4371	21
30	1790	0.0213	1	3798	0.0426	2	28654	0.0756	5	3975	0.4373	20
40	1044	0.0013	1	2336	0.0638	3	25638	0.0561	3	3521	0.2468	13
50	909	0	0	1498	0	0	24762	0.0246	2	2052	0.2162	12
60	987	0	0	1482	0	0	20054	0.0045	1	1227	0.1702	9
70	891	0	0	1802	0.0211	1	19062	0	0	1638	0.1702	8
80	1024	0	0	2312	0.0314	2	15178	0	0	2835	0.0638	3
90	1201	0	0	2451	0.0416	2	16254	0	0	1429	0.0851	6
100	1312	0.0235	1	2503	0.0212	1	24290	0	0	1161	0.0210	1
110	1505	0.0256	1	2809	0.0251	1	12387	0	0	1291	0.1245	2

*MSE: Mean Square Error
*IRC: Incorrectly Recognized Characters

Table 1 shows the variation of error against the number of hidden neurons for various feature extraction methods. Form that for the PixelMap feature extraction method it can be observed that,

1. At 10 and 15 hidden neurons there is relatively high error.
2. The error remains constant through the neurons 25 to 35. At neurons 40 through 90 it is constant. At neuron 70 epochs are less and we can assume that it is converged.

For the Diagonal based feature extraction method it can be observed that

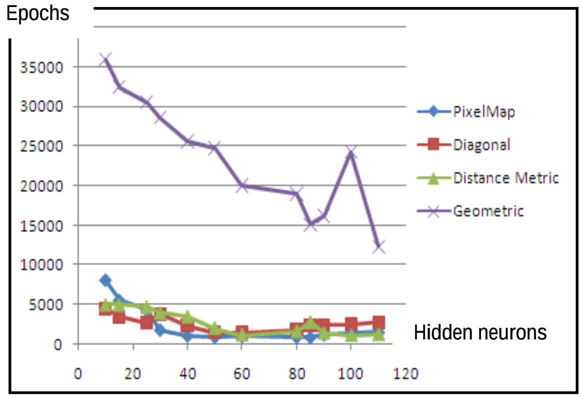
1. The error remains constant till 40 hidden neurons, starting from 10 neurons.
2. Anamolous results are observed at 40 neurons.
3. The value however converges at 60 neurons, with minimum number of epochs hence making it the best case.

The variation of error with respect to number of hidden neurons by the Geometric based feature extraction approach, it can be observed that

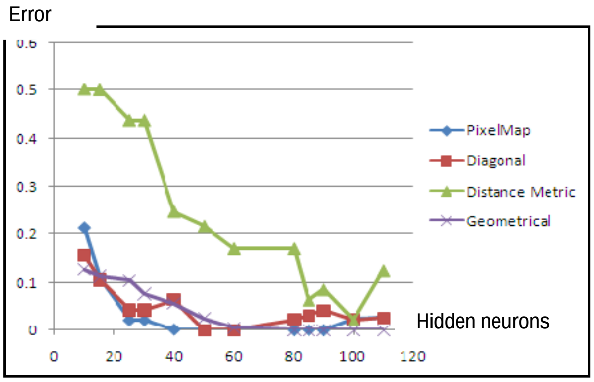
1. The best case achieved here is at 60 neurons where error rate is minimum and the number of epochs are very less at 80
2. The case with 90 neurons is also better with less error.

For the distance metric based feature extraction method in Table 1, the mean square error is very high from neurons 10 to 30 and minimum at neurons 100. The error has never come to 0 with this approach.

Among all these methods the PixelMap is best one because with less number of hidden neurons and minimum number of epochs it converges. Second best one is the geometric approach where it requires more number of hidden neurons and epochs with less good recognition accuracy. Graph1 shows the variation of epochs with respect to hidden neurons and Graph 2 shows the variation of error against number of hidden neurons. From the graphs 1 and 2 one can observe that the geometric approach takes more number of epochs but the error rate minimum.



Graph 1: Hidden neurons Vs. Epochs



Graph 2: Hidden neurons Vs Error

6. CONCLUSIONS AND FUTURE SCOPE

In this work we have identified and evaluated various feature extraction methods. The geometric, diagonal and distance metric feature extraction methods are modified and applied for the feature extraction of printed Telugu base characters. A simple new method Pixelmap was also proposed. A Multi-Layer Neural network with the supervised backpropagation algorithm is used to train and test those features for the recognition of the Telugu base characters. The recognition accuracy of the methods varies from 98-100%. However, there are some limitations like the dataset size is of 47 characters, fixed font and font sizes. It can be extended to include more number of characters including compound characters with different fonts and font sizes. The features can also be used with other classifiers like decision trees, SVM, Bayesian classifiers.

7. ACKNOWLEDGEMENTS

This work was supported in part by CBIT from TEQIP phase-II of AICTE.

REFERENCES

[1] U Pal, B.B Chauduri, "Indian script character recognition: a survey", Computer Vision and Pattern Recognition, Vol. 37, Issue 9, 2004, pp. 1887-1899

[2] Vikas, J Dongre, Vijay H Mankar "A Review of Research on Devanagri Character Recognition", International Journal of Computer Applications Volume 12-No.2, Nov 2010 , pp. 8-15

[3] Vikas J Dongre et al. "A Review of Research on Devanagari Character Recognition", IJCA (0975-8887), Vol.12, No.2, Nov-2010, pp. 8-15

[4] Rangachar Kasturi, Lawrence O'Gorman and Venu Govindaraju "Document Image Analysis: A primer", Sadhana, Volume. 27, Part 1, February, pp. 3-22.

[5] Prof. Snatanu Chaudhury et. al. "OCR technical report for the project Development of Robust Document Analysis and Recognition System for Printed Indian Scripts" sponsored by ministry of Communication and Information Technology, July 2008

[6] R Jagadeesh Kannan R Prabhakar " A Comparative Study of Optical Character Recognition for Tamil Script", European Journal of Scientific Research, ISSN 1450-216X, Vol. 35 No. 4 (2009), pp. 570-582

[7] B. Anuradha Srinivas Arun Agarwal and C. Raghavendra Rao, "An Overview of OCR Research in Indian Scripts", IJCSES, Vol.2, No.2, 2008, pp. 137-148.

[8] C Vasantha Lakshmi et al. "Optical Character Recognition of Basic Symbols in Printed Telugu Text", IE(I)Journal-CP, 2003, Vol 84, pp. 66-71.

[9] Atul Negi, K Narayan Murthy, Chakravarthy Bhagavati, "Issues of Document Engineering in Indian Scripts and Telugu as a case Study" , UGC R&D, RCILTS

[10] Chakravarthy Bhagvati , Negi et al. "An OCR system for Telugu", International Conference on Data Acquisition and Recognition (ICDAR), IEEE, 2001, pp. 1110-1114.

[11] N Otsu "A threshold selection method from gray-level Histograms",IEEE Transactions on Systems Man and Cybernetics, Vol. SMC-9, NO. 1, JANUARY 1979, pp. 62-67

[12] M. Swamy Das , et. al., "Segmentation of overlapping text lines, characters in printed Telugu text document images", IJEST, Vol. 2(11), 6606-6610, 2010.

[13] M.K. Jindal et al. "Segmentatation of Horizontally Overlapping Lines in Printed Indian Scripts", IJCIR, Vol. 3, No.4, 2007, pp. 277-286

[14] B.M. Sagar, G. Shoba, DR. P. Ramakanth Kumar "Character Segmentation algorithms for Kannada optical character Recognition", Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, 2008, pp.339-342.

[15] Dinesh Dileep "A Feature Extraction Technique Based On Character Geometry for Character Recognition", 2009

[16] http://www.mathworks.in/help/images/regionprops.html

[17] J Pradeep et. al. "Diagonal based Feature Extraction for Handwritten Alphabets Recognition System using Neural Network", IJCSIT, Vol 3., No.1, Feb 2011, pp 27-38.

[18] Rajasekaradhya, S V Ranjan "Neural Network based Handwritten Numeral Recognition of Kannada and Telugu Scripts", TENCON, IEEE 2008, pp.1-5.

[19] http://www.learnartificialneuralnetworks.com.

[20] Bogdan M. Wilamowski "Neural Network Architectures and Learning", 07803-7852-0/03, ICIT-IEEE, 2003, PP. TU1 –TU12

[21] Demuth, H., M. Beale, MATLAB Neural Network Toolbox v. 4.0.4,