Augmentation of Local, Global Feature Analysis for online Character Recognition System for Telugu Language using Feed Forward Neural Networks (FFNN)

Goda Srinivasarao, Rajeswara Rao Ramisetty

Abstract: ***In this paper, we propose ANN based online handwritten character recognition for Telugu Language. In literature review, it is observed that Size of the database and preprocessing approaches plays prominent role in the recognition performance. Preprocessing techniques like normalization, interpolation, Uniformization, Smoothing, Slant Correction and resampling techniques are performed for better recognition performance. Local features like(x,y)co-ordinates,*** $(\Delta x, \Delta y)$ $(\Delta^2 x, \Delta^2 y)$ ***and the global features like tan(Φ) are considered as features for ANN modeling and Classification of 52 Telugu vowels and consonants. Recognition performance is evaluated by augmentation the local, global features and and tan (Φ) Features. Theperformance is evaluated in terms of precision, recall and F-measure. Significant Improvement is reported by augmentation andby adopting preprocessing techniques. The database used for the study is HP-online Telugu database.***

Index Terms — ANN, HP-database, local features, global features

I. INTRODUCTION

Though languages like English can be or given as an input to the computers to execute as commands or process the data. It is not the same for quite a few languages like Telugu, Chinese, Hindi and other Indian or Japanese languages. Because these languages involve lot of stroke variations from writer to writer. But, rather than giving input via keyboard or voice, it is advisable to give it via handwritten samples (like parchments of paper or electronic pens).

For instance, entering data into the database from the hand-filled Railway-reservation applications is a tedious task and can be automated. Moreover, properly trained systems will be capable of recognizing the hand-written text better than that of the human. And this handwriting recognition is plays a crucial role in the human computer interaction model. Efforts have already been made to build system in both online and offline fields for achieving various aims, like recognizing numeric characters, language recognitions like Assamese [2], Thai [5], and Arabic [4].

Unlike English, the basic characters in Telugu script consist of 16 vowels and 36 consonants. The characters in telugu script are a combination of these basic characters and their modifiers which gives rise to about 18,000 unique characters. All these unique characters in Telugu can be represented as a combination of a manageable set of 235 strokes. Also the character strokes, other the first stroke taken as main stroke, can be divided, based on the position of the stroke, into three - top stroke, bottom stroke and baseline stroke. As a preliminary attempt, we use character based recognition for on-line handwriting recognition of Telugu which is a very popular south Indian language, in which much research has not yet done.

Telugu language found in the South Indian states of Andhra Pradesh and Telangana as well as several other neighboring states. Subset Telugu symbols given in the Figure 2. In Telugu script, many of the characters resemble one another in structure. The framework for online handwritten character recognition is depicted in Figure 2.Further, many users write two or more characters in a similar way which can be difficult to classify correctly. In Telugu some of the confusing pairs are there. An SVM based stroke recognition method used in [1] for Telugu characters. Based on proximity analysis, the recognized strokes are mapped onto characters using information of stroke combinations for the script. Each stroke is represented as preprocessed (x, y) coordinates. The data sample size 37817 was collected from 92 users using the SuperpenTM, a product of UC Logic. The observed recognition accuracy is 83%. Importance of annotation of online handwritten data illustrated in [1]. Modular approach for recognition of strokes proposed in [2]. Based on the relative position of strokes in the character, the strokes are categorized into baseline, bottom, top strokes. The recognition model SVM was used for each category separately. The recognition accuracy is high for each stage, when compared to combined classifier. Elastic matching technique, DTW used in [3]. The features used are local features: x-y features, Tangent Angle (TA) and Shape Context (SC) features, Generalized Shape Context (GSC) feature and the fourth set containing (x, y) coordinates, normalized first and second derivatives and curvature features.

Revised Manuscript Received on July 05, 2019.
Goda Srinivasarao, Research Scholor, Department of Computer Science & Engineering, JNTUA-Anantapuramu, A.P, India
Rajeswara Rao Ramisetty, Professor, Department of Computer Science & Engineering, JNTUK-Unviersity College of Engineering-Vizianagaram

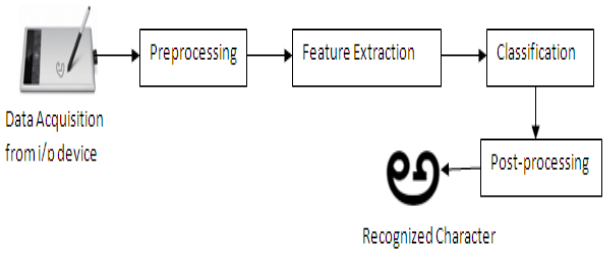


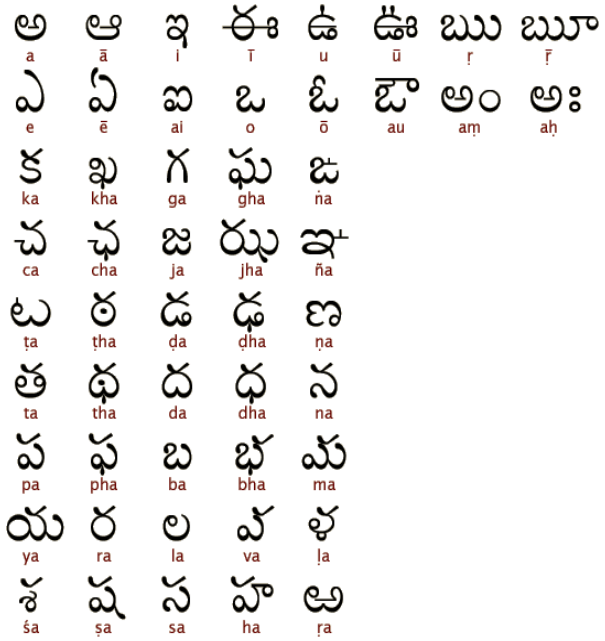
Figure 1. Steps involved for Character Recognition

II. FEATURES USED FOR CHARACTER RECOGNITION

A. Preprocessing

The input data has to be cleansed before feeding it to the neural network [6]. This is because, though the data is extracted from a HP Labs Data Collection Tool, still the data isn't suitable for the input feeding. Hence a series of preprocessing steps are performed. The prime motive of this preprocessing is to eliminate noise and bring uniformity in the data. This preprocessing is carried using series of steps which are: Normalization, Interpolation, Uniformization, Smoothing, Slant Correction and Resampling .

Figure:2 Telugu characters (Vowel& Consonants) characters



B. Normalization and Centering

Thenormalization [7] is an essential step which needs to be done so as to increase the recognition rate of the model. Every person has his own size of representing or drawing the character. When observed cumulatively they appear to be out of scale. With the use of normalization and centering the size of all the characters presented by all the user were brought to a common size and are scaled to a common size window size of 300X300.

$$X_i^N = \frac{(x_i - x_m)}{(X_M - x_m)} W + x_o \quad (1)$$

$$Y_i^N = \frac{(y_i - y_m)}{(Y_M - y_m)} H + y_o \quad (2)$$

The normalized x and y coordinates (X_i^N, Y_i^N) as calculated using the above formula, where x_m and X_M are minimum and maximum values of x. Similarly, for y. W and H are the max length of the new character which are taken as 300 as in Figure 3 and 4.

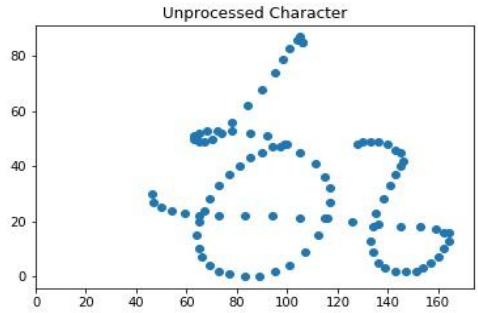


Figure: 3 Out of Scale Character (Before Normalization and Centering

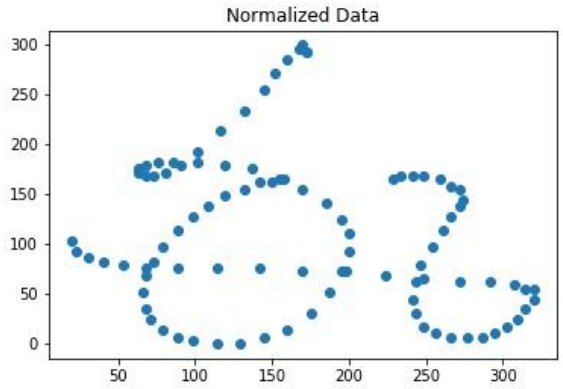


Figure 4. Normalized Character with fixed set of boundaries

C. Interpolation

As most of the points turn out to be discrete and due to variation in writing speeds of the users the sequence of points captured by the DCT tools might not be suitable for the training task. Hence we used two different types of interpolation techniques which are Bezier Interpolation and B-Spline interpolation[8]. As we can see in the above image the distance between 2 consecutive Cartesian points representing the character are not equally spaced.Henceforth, stroke-wise Interpolation. Bezier interpolation considers a window size of 4 points. By the use of Bezier and B-spline interpolation equations, we obtained a series points forming a proper curve as in figures 5 and 6 below.

$$P_{new_x} = (1-i)^3 P_{ix} + 3(1-i)^2 i P_{ix} + 3(i)^2(1-i) P_{(i+2)x} + i^3 P_{(i+2)x} \quad (3)$$

$$P_{new_y} = (1-i)^3 P_{iy} + 3(1-i)^2 i P_{iy} + 3(i)^2(1-i) P_{(i+2)y} + i^3 P_{(i+2)y} \quad (4)$$

$$P_{new_x} = \frac{(1-i)^3}{6} \cdot P_{(i-1)x} + \frac{(3i^3 - 6i^2 + 4)}{6} \cdot P_{ix} + \frac{(-3i^3 - 3i^2 + 3u + 1)}{6} \cdot P_{(i+1)x} + \frac{u^3}{6} \cdot P_{(i+2)x} \quad (5)$$

$$P_{new_y} = \frac{(1-i)^3}{6} \cdot P_{(i-1)y} + \frac{(3i^3 - 6i^2 + 4)}{6} \cdot P_{iy} + \frac{(-3i^3 - 3i^2 + 3u + 1)}{6} \cdot P_{(i+1)y} + \frac{u^3}{6} \cdot P_{(i+2)y}$$

(6)

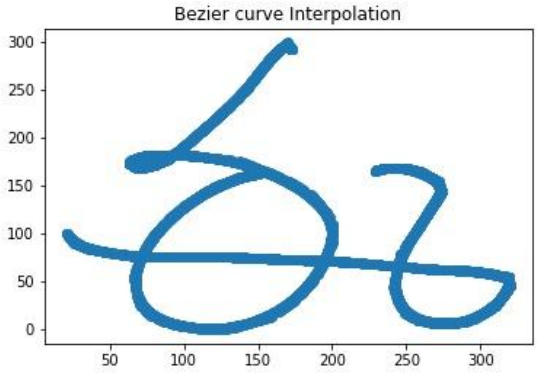


Figure 5: Bezier Interpolated character for the normalized character.

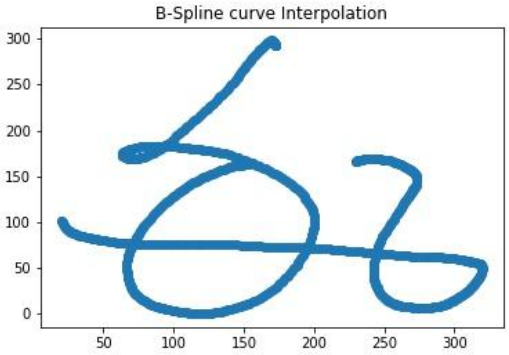


Figure 6. B-Spline Interpolated Character

D. Uniformization

After the interpolation we have observed many points were too close to each other, which indeed would not contribute to the recognition task as the distance metric between those two points was negligible. Hence, we chose a distance threshold to choose the points to be in our list forming the character and eliminated the rest by using equation 7. While doing so we observed that Bezier interpolation create on an average of 8000 points while B-Spline created a 1200 points using their respective formulae[8]. But both gave the same curves when plotted as seen in the above figures 7.

Foreach stroke in the gesture:

for point in list:

$distance = Eucledian_{Distance(P_{ref}+P_i)}$ (7)

If $distance \geq distance_{threshold}$:

Yield Pi

$$P_{ref} = P_i$$

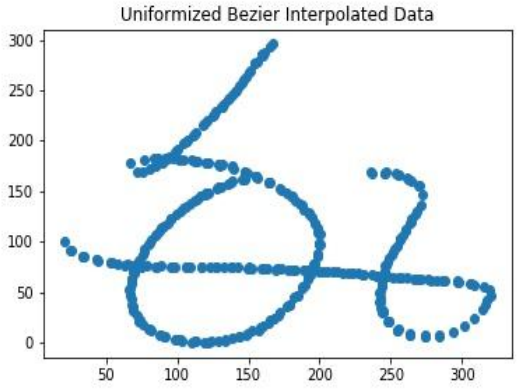


Figure 7. Bezier Uniformized character

E. Smoothing

In order to reduce the jitters in input obtained from the hardware or hand motion we performed the smoothing of the charcter. In this paper a linear smoothing approach is adopted. The end points are preserved by taking special care. Each pattern is smooth both in horizontal and vertical directions separately using equation 8 and 9(Li and Yeung, 1997) and result illustrated in Figure 8,9 and 10.

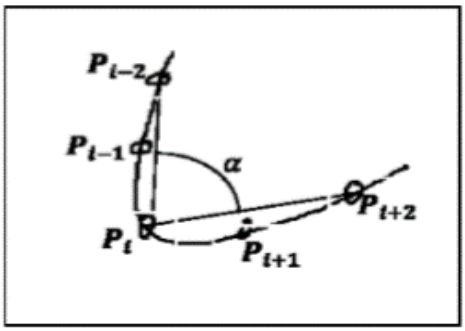


Figure 8.Iterate for all points in stroke:

Calculate angle between $P_{(i-2)}$,P_i, $P_{(i-2)}$ as a

$$P_{ix} = \frac{P_{(i-2)x} + P_{(i-1)x} + \alpha * P_{ix} + P_{(i+2)x}}{4+\alpha} \quad (8)$$

$$P_{iy} = \frac{P_{(i-2)y} + P_{(i-1)y} + \alpha * P_{iy} + P_{(i+2)y}}{4+\alpha} \quad (9)$$

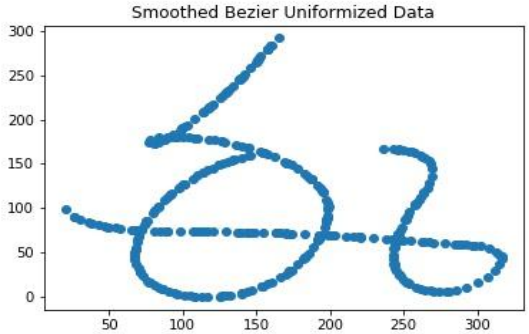


Figure 9. Bezier smoothed character

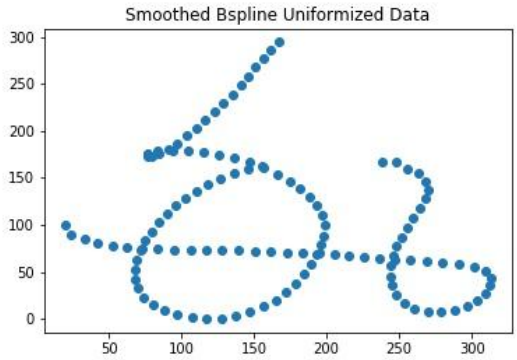


Figure 10. B-spline smoothed character

F. Slant Correction

Slant correction is an important preprocessing step to be followed to obtain better accuracy while recognition [10]. We used 16 directional chain coding method for this task. Though this step did not receive great accuracy change for right-handed writer, but it showed great results in the case of left-handed writers. Calculate histogram of the direction based on 8 directional set for each stroke in the gesture using below equation

$$a = \frac{(2 * n_1 + 2 * n_2 + n_3) - (n_5 + 2 * n_6 + 2 * n_7)}{(n_1 + 2 * n_2 + 2 * n_3) + (2 * n_5 + 2 * n_6 + n_7)}$$

$$A = \tan^{-1}(a)$$

if (A>45 and A<90) or (A>90 and A<135) then

$$P_{ix} = P_{ix} + P_{iy} * \tan(A)$$

G. Resampling

After all the above mentioned preprocessing steps still data is not feed-ready to the learning model. We observed some character being presented in the set of 150 points while other around 200. But so as to feed them to a model. All character must be brought to common sized points. We selected 70 points from the character which based on the proportion of each stroke. We selected every Kth point from the stroke where k is ($K = i*(N/70)$) as in Figure 11 and 12.

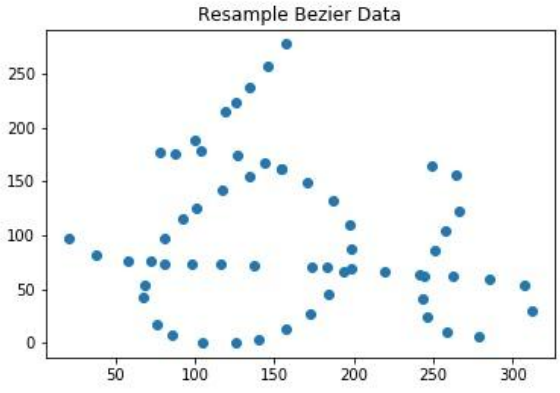


Figure 11. Resample Bezier Data

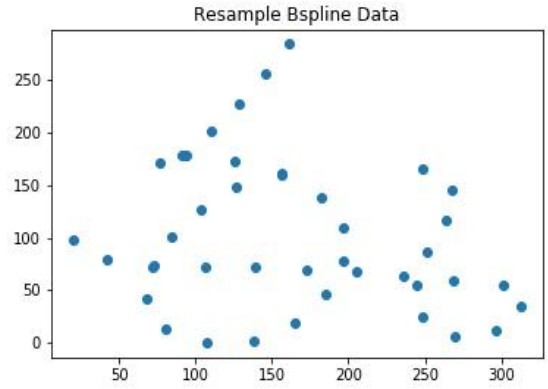


Figure 12. Resample Bspline Data

H. Feature Extraction

In the module the features of handwritten characters are analyzed for training and recognition [11]. In our approach we have consider few feature which are time-domain feature while others being the discrete ones.

(i).Local Features

Preprocess x-y coordinates: The normalized and preprocessed x-y coordinates make up the first set of features which are used for the recognition task.

First Order Derivatives: The first order derivatives are considered as one of the feature for the recognition task by using equations 10 and 11. Particularly this was chosen because it keeps track of the change in curvature with the progression of time. A two point window was considered for the calculation of this features at each point and the derivatives of x and y are calculated independent of each other.

$$x'(j) = \frac{\sum_{i=1}^{2} i(x_{j+i} + x_{j-i})}{2 * \sum_{i=1}^{2} i^2} \quad (10)$$

$$y'(j) = \frac{\sum_{i=1}^{2} i(y_{j+i} + y_{j-i})}{2 * \sum_{i=1}^{2} i^2} \quad (11)$$

Second Order Derivative: The second order derivative was considered to examine the change in trajectory of the current point using equations 12 and 13. Here window size of two is considered.

$$x''(j) = \frac{\sum_{i=1}^{2} i(x'_{j+i} + x'_{j-i})}{2 * \sum_{i=1}^{2} i^2} \quad (12)$$

$$y'(j) = \frac{\sum_{i=1}^{2} i(y'_{j+i} + y'_{j-i})}{2 * \sum_{i=1}^{2} i^2} \quad (13)$$

Aspect Ratio: Aspect at a point characterizes the ratio of the height to the width of the bounding box containing points in the neighborhood. It is given by below equation

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

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

ii) Global Features

Tangent Angle between End Points: This feature calculates the tangent value between the end points of stroke by using equation 14. We chose this feature as it can helps the system in discerning the characters as some have a closed loop while others open, resulting in varied tangent angles.

$$\tan(\theta) = \frac{\Delta y}{\Delta x} \quad (14)$$

Where Δy is the difference between y of the end-point similarly x.

Euclidian Distance between End Points: This feature calculates the Euclidian distance between the ends of stoke equation 15. This features goes hand-in-hands with the previous feature where closed loops and open looped characters can be discerned.

$$E_{distacne} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (15)$$

Where (x_2, y_2) and (x_1, y_1) are ends of the stroke.

III. ALGORITHM FOR IMPLEMENTING CLASSIFIER

1. *Collect the Data from HP Labs Data Collection Tool*
2. *Preprocessing of Collected Data (the collected data not suitable for direct processing*
3. *i) Normalization and Centering- To scale down all Character to window size of 300X300*

ii) Interpolation- Bezier and B-spline interpolation to obtain a series points forming a proper curve

iii) Uniformization- To obtain points which are at equal distances using Euclidian distance parameter

iv) Smoothing- to reduce the jitters in input obtained from the hardware or hand motion, linear smoothing approach is adopted

v) Slant correction

vi) Resampling- To normalize input character to a constant number of points

3. Feature Extraction using Local and Global Features

4. i)Local Features-First order derivative ,Second order derivative and Aspect ratio

ii) Global Features- Tangent Angle between End Points and Euclidian Distance between End Points

5. Extracted features are given as input to Feed Forward Neural Networks

IV. EXPERIMENTATION AND RESULT ANALYSIS

i) Experimental Setup:

ANN is used in the classification stage and the total numbers of characters used for classification are 52. Figure 13, shows the architecture used for train and test purpose. Two hidden layers are used for training and testing .Tensor flow is used on Windows 10 with I7 HQ processor clock speed of 3.5Ghz, 12 GB of RAM and 2GB Nvidia GTx 950M GPU. The features are extracted and trained with MLP by considering four hidden layers.

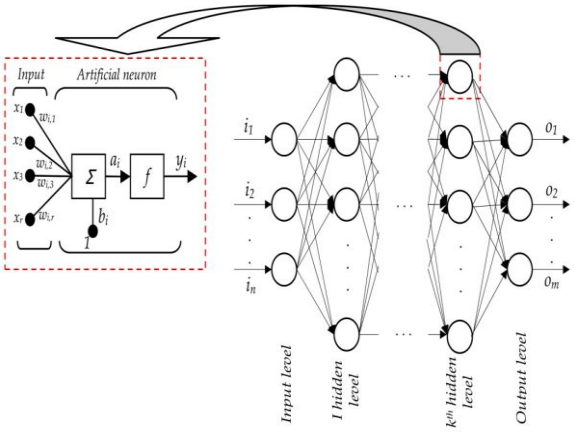


Figure13: Multilayer Neural Network for Telugu Character Training.

ii) **Result Analysis:**

In model proposed by Amit Arora and Anoop M. Namboodiri [15], the strokes are treated as sequence emanating from the underlying sequence of aksharas in the word. This structure is common among most Indian languages. Once the aksharas are recognized, one can decompose it into the underlying basic characters to generate a Unicode representation, which is defined at that level.The recognition accuracy in this Hybrid approach is 93.10. In online handwritten Telugu character recognition using SVM,a two stage method is adopted [16].One for CV classifier and another for C2 type classifier to recognize a character. The overall recognition accuracy is 96.69%. In the method proposed by Raju Dara and Urmila Panduga [17], the features are extracted using 2-D FFT and Support Vector Machine, and the recognition accuracy is 71% even though the method uses a single stage for feature extraction.In our model Telugu character (Vowel + Consonants), each of 450 samples are considered for modeling and tested with 70 samples of each TeluguCharacter. As shown in the table 2, performance evaluation matrices like Precision, Recall and F-Measure are calculated by using equation 16,17 & 18. From Table 1, the minimum and maximum precession values are 0.99 and 1. Similarly for recall the minimum value is 0.98 and maximum value is 1. Further F-Measure is calculated. From the Table 1 it is clearly evident that by augmenting Local Features, Global Features and tan (Φ) Features, the error rate is very minimal.Finally 98% performance is obtained for 52 Telugu characters. We have compared this with existing methods and illustrated in Table 2 and Figure 14.

Precision= $\frac{TP}{TP + FP}$ (16)

Recall= $\frac{TP}{TP + FN}$ (17)

F= $2 * \frac{precision * \text{Re}\,call}{\text{Pr}\,ecision + \text{Re}\,call}$ (18)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

SNO	VOWELS & CONSONANTS	PRECISION	RECALL	F1-SUARE
0	అ	0.99	1.00	1.00
1	ఆ	1.00	0.99	1.00
2	ఇ	1.00	1.00	1.00
3	ఈ	1.00	1.00	1.00
4	ఉ	1.00	1.00	1.00
5	ఊ	1.00	1.00	1.00
6	ఋ	0.99	0.98	0.99
7	ౠ	0.99	1.00	0.99
8	ఎ	1.00	1.00	1.00
9	ఏ	1.00	1.00	1.00
10	ఐ	1.00	1.00	1.00
11	ఒ	0.99	1.00	1.00
12	ఓ	1.00	1.00	1.00
13	ఔ	1.00	1.00	1.00
14	అం	1.00	1.00	1.00
15	అః	1.00	1.00	1.00
16	క	1.00	1.00	1.00
17	ఖ	1.00	1.00	1.00
18	గ	1.00	1.00	1.00
19	ఘ	1.00	1.00	1.00
20	ఙ	1.00	1.00	1.00
21	చ	1.00	1.00	1.00
22	ఛ	0.99	1.00	0.99
23	జ	1.00	1.00	1.00
24	ఝ	1.00	1.00	1.00
25	ఞ	1.00	0.99	1.00
26	ట	1.00	1.00	1.00
27	ఠ	1.00	1.00	1.00
28	డ	0.99	0.99	0.99
29	ఢ	1.00	0.99	1.00
30	ణ	1.00	1.00	1.00
31	త	1.00	1.00	1.00
32	థ	0.99	0.97	0.98
33	ద	1.00	1.00	1.00
34	ధ	0.98	0.99	0.98
35	న	1.00	1.00	1.00
36	ప	0.99	1.00	1.00
37	ఫ	1.00	1.00	1.00
38	బ	1.00	1.00	1.00
39	భ	1.00	0.99	0.99
40	మ	1.00	0.99	1.00
41	య	0.99	0.99	0.99
42	ర	1.00	1.00	1.00
43	ల	1.00	1.00	1.00
44	వ	1.00	1.00	1.00
44	వ	1.00	1.00	1.00
45	శ	1.00	1.00	1.00
46	ష	1.00	1.00	1.00
47	స	1.00	1.00	1.00
48	హ	1.00	1.00	1.00
49	ళ	1.00	1.00	1.00
50	క్ష	1.00	1.00	1.00
51	ఱ	1.00	1.00	1.00

Table 1: Performance of character Recognition for Telugu Vowels & Consonants

Method	Method	Accuracy
A Hybrid Model for Recognition of Online Handwriting in Indian Scripts[15]	Method 1	93.10
Online Handwritten Character Recognition for Telugu Language Using Support Vector Machines[16]	Method 2	96.69
Telugu Handwritten Isolated Characters Recognition using Two Dimensional Fast Fourier Transform and Support Vector Machine[17]	Method 3	71.00
Feed Forward Neural Networks	Proposed Method	98.00

Table 2: Performance of Telugu Character Recognition

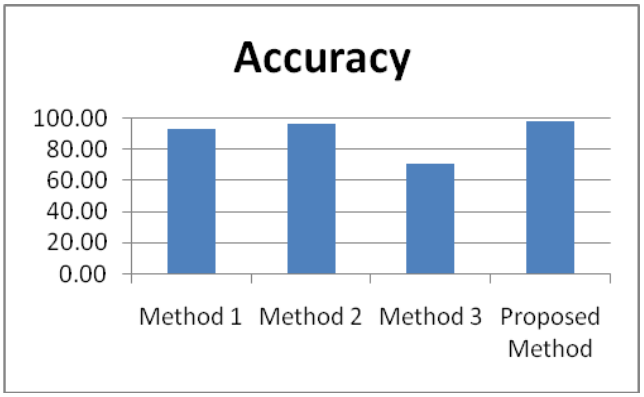


FIGURE 14: COMPARISON OF PERFORMANCE

V. CONCLUSION

The idea of agumentation is implemented in our study for Telugu characters (vowels and consonants). From the results it is observed that the proposed method of augmenting the local, global, and tan (Φ) Features have given good recognition performance by minimizing the error rate for Telugu character. This work can be further be improved by extending the methodology on other classification techniques and by considering other local and global features.

ACKNOWLEDGMENT

I would like to thank HP Labs for providing dataset to carry out our work.

REFERENCES

1. H. Choudhury, S. Mandal, S. Devnath, S. R. M. Prasanna and S. Sundaram, "Combining HMM and SVM based stroke classifiers for online Assamese handwritten character recognition," *2015 Annual IEEE India Conference (INDICON)*, New Delhi, 2015, pp. 1-6.
2. Reddy, G. S., Sarma, B., Naik, R. K., Prasanna, S. R. M., & Mahanta, C. (2012, December). Assamese online handwritten digit recognition system using hidden markov models. In *Proceeding of the workshop on Document Analysis and Recognition* (pp. 108-113). ACM.
3. Uchida, Seiichi & Sakoe, Hiroaki. (2005). A survey of elastic matching techniques for handwritten character recognition. IEICE Trans Inf Syst E88-D(8):1781-1790. IEICE - Transactions on Information and Systems. E88-D. 1781-1790. 10.1093/ietisy/e88-d.8.1781.
4. AbdelRaouf, Ashraf & Higgins, Colin & Khalil, Mahmoud. (2008). A Database for Arabic Printed Character Recognition. 5112. 567-578. 10.1007/978-3-540-69812-8_56.
5. Pornpanomchai, C., Batanov, D. N., & Dimmitt, N. (2001). Recognizing Thai handwritten characters and words for human–computer interaction. *International Journal of Human-Computer Studies*, *55*(3), 259-279.
6. Chaudhuri, A., Mandaviya, K., Badelia, P., & Ghosh, S. K. (2017). Optical Character Recognition Systems. In *Optical Character Recognition Systems for Different Languages with Soft Computing* (pp. 9-41). Springer, Cham.
7. Liu, Cheng-Lin & Marukawa, K. (2004). Normalization ensemble for handwritten character recognition. Proceedings - International Workshop on Frontiers in Handwriting Recognition, IWFHR. 69- 74. 10.1109/IWFHR.2004.76.
8. Zamani, Mehdi. (2018). An Investigation of Bspline and Bezier Methods for Interpolation of Data.
9. Shwetha, Dimple & Lokesh, Ramya. (2014). Comparison of Smoothing Techniques and Recognition Methods for Online Kannada Character Recognition System. 2014 International Conference on Advances in Engineering and Technology Research, ICAETR 2014. 10.1109/ICAETR.2014.7012888.
10. Mohamed, A., Yusof, R., Mutalib, S., & Rahman, S. A. (2009). Online slant signature algorithm analysis. *WSEAS Transactions on Computers*, *8*(5), 864-873.
11. Mori, M., Uchida, S., & Sakano, H. (2014). Global feature for online character recognition. *Pattern Recognition Letters*, *35*, 142-148.
12. A. M. Namboodiri and A. K. Jain, "Online handwritten script recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 124-130, Jan. 2004
13. Ramakrishnan, A. G., & Urala, K. B. (2013, August). Global and local features for recognition of online handwritten numerals and Tamil characters. In *Proceedings of the 4th international workshop on multilingual OCR* (p. 16). ACM.
14. Gupta, N., Gupta, M., & Agrawal, R. (2012). Preprocessing of Gurmukhi Strokes in Online Handwriting Recognition. *International Proceedings of Computer Science and Information Technology*, *56*(163), 2012.
15. Amit Arora and Anoop M. Namboodiri," A Hybrid Model for Recognition of Online Handwriting in Indian Scripts"`
16. K. Vijay Kumar, R.Rajeshwara Rao, "Online Handwritten Character Recognition for Telugu Language Using Support Vector Machines", International Journal of Engineering and Advanced Technology, Vol.3, Issue.2, pp. 189-192, 2013.
17. Raju Dara, Urmila Panduga," Telugu Handwritten Isolated Characters Recognition using Two Dimensional Fast Fourier Transform and Support Vector Machine" International Journal of Computer Applications (0975 – 8887) Volume 116 – No. 5, April 2015