I J C T A, 8(5), 2015, pp. 1939-1945

Preprocessing of HP Data Set Telugu strokes in Online Handwritten Telugu Character Recognition

Srilakshmi Inuganti* and R. Rajeshwara Rao**

Abstract: Online Handwritten Character Recognition (OHCR) is the method of recognizing characters by a machine while the user writes, in which, the handheld devices record (x, y) coordinates of the track of the character. With the advent of handheld devices, there is a great attention towards OHCR of regional languages. Preprocessing is the main phase, in OHCR, As it increases the performance of next phases, by removing the inconsistency or the redundancy present in the data collected in real-world environment. In this paper, we depict the model of Preprocessing of Online Handwritten Telugu Strokes. The preprocessing stepswe address in our article are Normalization, Smoothing, Duplicate Point Removal, Interpolation and Resampling. The preprocessing algorithms are applied over HP labs hpl-telugu-iso-char-online-1.0 contains samples of the 166 character classes collected from different writers on ACECAD Digimemo (A4 sized) using an AcecadDigi memo DCT application. It consists nearly 270 samples of each of 166 Telugu "characters" written by native Telugu writers.

Keywords: Online Handwriting Character; Preprocessing; Telugu Strokes;

1. INTRODUCTION

With handheld devices reaching new heights of popularity every day and becoming almost indispensable in our busy lives, digital pens become a great alternative to keyboards, especially in case of PDAs, Hand Held PCs and high end mobile devices. A digital pen captures the handwriting of a user, converts handwritten information into digital data, enabling the data to be utilized in various applications. In this context Handwritten Character Recognition (HCR) is an immediate challenge in the area of pattern recognition. HCR can be classified into Online HCR and Offline HCR. OHCR is the task of identifying character written by a machine while the user writes, in which transducer required for capturing dynamic handwriting information. The dynamic information contains numbers, order, length, writing direction and speed of stroke and some devices record pressure information also (i.e. At pen tip). A stroke is the writing form pen-down to pen-up. Online data are associated with temporal information, so that accuracy is high in adverse to offline. Online data are highly interactive. Hence, errors can be debugged immediately with repeated tests. Online data offers reduction in memory and therefore space complexity. Even though many years of research in handwriting recognition [1,2], very less has been made towards Indian languages. OHCR is more realistic for Indian languages which have huge character set.

1.1. A Framework of OHCR

The block diagram OHCR illustrated in Figure1.

Figure 1: Steps in OHCR

* Computer Science and Engineering, GMR Institute of Technology, Rajam, India, *Email: srilakshmi.i@gmrit.org*

** Computer Science and Engineering, UCEV, JNTUK, *Email: raob4u@yahoo.com*

The details of each step are described in the following paragraphs.

1.2. Data Collection

Online handwriting recognition software incorporates the automatic conversion of the text simultaneously with the user's writing. These digitizers like PDAs use sensors to track movement of the input device like a stylus pen. When the pen tip makes contact with the screen, the sensors are activated. When the contact is broken, the sensors are automatically turned off. The acquisition interface outputs a sequence of (x, y)-coordinates representing the location of pen tip and binary value indicates pen up/pen down, the coordinates are recorded only period when the pen is in contact with the interface. This period is known as stroke.

1.3. Preprocessing

Before applying input to the system to get correct recognition result, data need to be pre-processed. As the data collected in real environment, it can be noisy and inconsistent. The important target of preprocessing is to eliminate the effect of noise, variation in writing size and style and repetition of points. It is carried out in five steps-Normalization, Smoothing, Duplicate Point Removal, Interpolation and Resampling. Even though preprocessing enhances recognition accuracy, excessive preprocessing is undesirable because it may result in loss of valuable information.

1.4. Feature Extraction

Feature extraction starts with measured data and builds features, which are informative and non-redundant. These features extracted should maximize inter-class similarity and minimize intra-class similarity. The accuracy of any recognizer depends on how well feature discriminate various classes.

1.5. Classification

The heart of any recognition model is Classification. Based on features extracted, compared with template stroke, decision rules are used to make some kind of decisions.

1.6. Post Processing

After analysis of the confusion matrix, confusing pairs are identified. Script specific features can be used to resolve ambiguities in confusing characters.

2. TELUGU SCRIPT

The Dravidian language Telugu is the official language in the states of Andhra Pradesh and Telangana.The government of India designated Telugu as the one of six classical languages of India. Telugu is the native

Table1
Combination of Telugu Characters

Character	Type
V	16
C	37
CV	592
CCV	21904
CCCV	810448
NUMERALS	10
Total	833007

language of 75 million people, according to 2011 census. There are 18 vowels and 36 consonants, of which 16 vowels and 37 consonants are in common usage. The syllables in Telugu script are vowels, consonants, and their combinations. The typical forms of syllables are V, CV, CCV and CCCV, thus have a generalized form of C*V. Thus the basic units of character of script O (10^2), these units forming O (10^4) number of composite characters. The total possible combinations are listed in the Table 1 below.

The complete symbol set containing a total of 108 symbols that covers the entire Telugu script is shown in Figure 2.

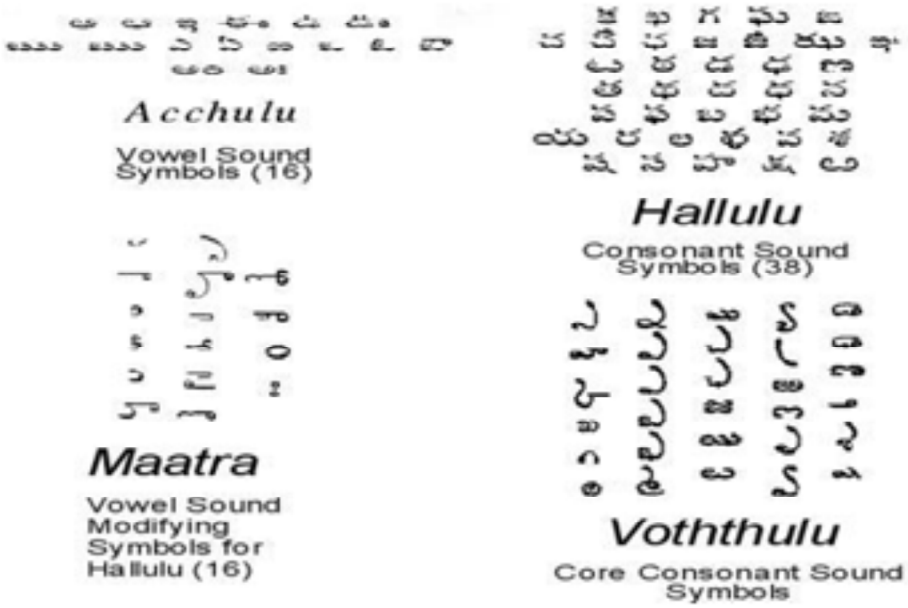

Figure 2: Characters in Telugu Script

3. DATASET USED

In a research area related to pattern recognition Benchmarking database is very important. In Telugu the data set available is Hp-Labs data in UNIPEN format. This dataset contains nearly 270 samples of each of 166 Telugu “characters” written by native Telugu writers [3]. The total training and testing samples are 44,613. In this Subset of approx 170 samples/char that may be used as the training set and Subset of approx 60 samples/char that may be used as the test set. The data are collected using Acecad Digimemo electronic clipboard devices using the Digimemo-DCT application.

4. PREPROCESSING TECHNIQUES OF TELUGU STROKES

Preprocessing is the main phase of online handwritten character recognition as it enhances the accuracy of the next stages. In the following sections we address the preprocessing techniquesNormalization, Smoothing, Duplicate Point Removal, Interpolation and Resampling. These techniques are applied over the HP dataset. The data set is in the form of Co-ordinates. These co-ordinates are input through writing pads by using stylus/pen. The strokes are recorded from. PEN_DOWN to. PEN_UP. The representation of one stroke stroke is shown in the Figure 3.

The variations in writing patterns of the same character are shown in Figure 4.

4.1. Normalization

Usually the recognition rate is high, if we normalize the character with respect to the width and height, along with a starting point.In this paper we normalize the size and starting position of the stroke. In our the

```
.VERSION 1.0
.HIERARCHY CHARACTER
.COORD X Y T
.SEGMENT CHARACTER 0-2 OK "4"
.H_LINE 2625 3375
.V_LINE 249 999
.X_DIM 749
.Y_DIM 750
.X_POINTS_PER_INCH 1000
.Y_POINTS_PER_INCH 1000
.POINTS_PER_SECOND 125
.COMMENT CALIB X: -36 Y: -76

.PEN_DOWN
523 2873 0
521 2879 0
520 2887 0
520 2895 0
520 2903 0
520 2912 0
520 2922 0
521 2930 0
522 2937 0
523 2943 0
524 2948 0
.PEN_UP
```

Figure 3: Representation of Stroke

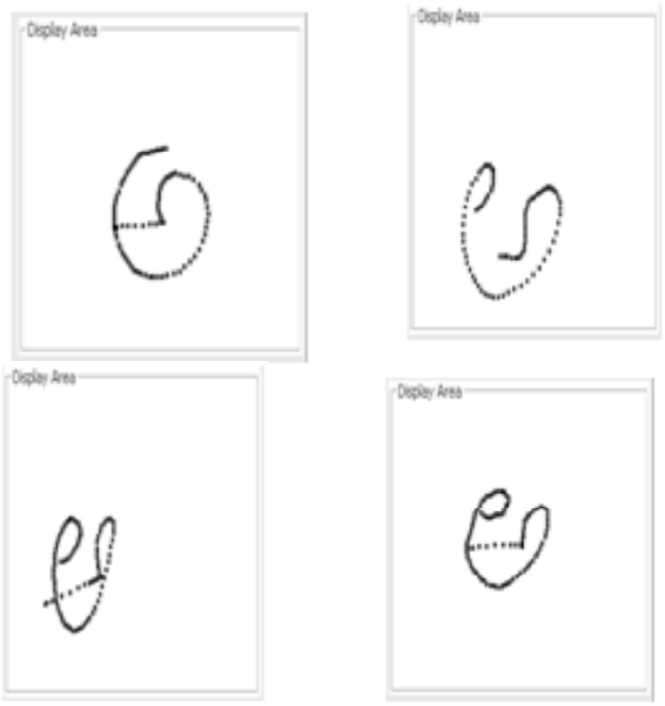

Figure 4: Character A written by 4 different users

window of size 260 × 250 pixels are considered for writing area. The size of the character varies from one user to another and time to time also. In size normalization the *x* and *y* coordinates are scaled both horizontally and vertically. The scale factors are calculated based on the ratio of height and width of the character with respect to height and width of the display window. In positional normalization every character is normalized with respect to starting position of the first stroke by using the translation of coordinates [4]. The algorithm for size normalization and centering of stroke is given below:

Algorithm:

In this algorithm the starting point is considered as (x_c, y_c) and set of pixels in which a Telugu stroke is represented as $\{(x_i, y_i)$: $xW_{min} \le x_i \le xW_{max}$, yW_{min}, $\le y_i \le yW_{max}$, i = 0, 1, ... n: No. of *pi* xels in the Telugu Character$\}$

Where $xW_{min} = min\{x_i\}$, $xW_{max} = max\{x_i\}$, $yW_{min} = min\ \{y_i\}$, $yW_{max} = max\ \{y_i\}$

Algorithm:

1. Set $xV_{min} = 0$, $xV_{max} = 260$, $yV_{min} = 0$, $yV_{max} = 250$
2. $S_x = (xV_{max} - xV_{min}), (xW_{max} - xW_{min})$

   $S_y = (yV_{max} - yV_{min}), (yW_{max} - yW_{min})$
3. $P_{xc} = x_c - x_0$

   $P_{yc} = y_c - y_0$
4. $x_i = (x_i - xW_{min}) * S_x \ \forall$ points i = 1, 2, … n

   $y_i = (y_i - yW_{min}) * S_y$
5. $x_i = (x_i + P_{xc}) \forall$ points i =1, 2, … n

   $y_i = (y_i + P_{yc})$

The above algorithm normalizes the stroke in size and starting position. The result is depicted in Figure 5.

(a) Input Character larger than display area **(b) Size normalized character** **(c) Position normalized character**

Figure 5: Normalized Character

4.2. Smoothing

Smoothing is performed to reduce the jitters in input obtained from the hardware or hand motion. 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[5].In linear smoothing a new coordinate (x_i, y_i) is calculated as follows:

$$x_i = (x_{i-1} + 2x_i + x_{i+1})/4$$
$$y_i = (y_{i-1} + 2y_i + y_{i+1})/4$$

The result is given in the Figure 6:

(a) Normalized Character **(b) Smoothed Character**

Figure 6: Smoothed Character

4.3. Removal of Repetition points

Sometimes input data contains duplicate points and does not contain any useful information for classification. If P_i and P_j are two consecutive points, then these points will be preserved if the following equation is satisfied :

$$x^2 + y^2 > d^2 \quad (1)$$

Where $x = x_i - y_i$ and $y = y_i - y_j$

We have set d equal to zero; The Equation 1 removes all consecutive repeated points.

4.4. Interpolation

Interpolation is the prerequisite for applying Re-sampling. Interpolation generates missing points, usually with the constraint that distance cannot be more than a certain threshold [6]. In this paper the missing points between P_i and P_{i+1} are calculated using the algorithm below:

Algorithm:

1. Initial the coordinates of two points $A(x_1, y_1)$ and $B(x_2, y_2)$ between which to calculate missing points.
2. [Calculate d_x and d_y]
   $d_x = (x_2 - x_1)$ and $d_y = (y_2 - y_1)$
3. [Calculate the length L]
   If $abs\ (x_2 - x_1) \geq abs\ (y_2 - y_1)$ then $L = abs\ (x_2, x_1)$
   Else $L = abs(y_2 - y_1)$
4. [Calculate the increment factor]
   $\Delta x = (x_2, x_1)/L$ and $\Delta y = (y_2 - y_1)/L$
   This step makes either Δx or Δy equal to 1, because L is either $|x_2 - x_1|$ or $|y_2 - y_1|$. Therefore a step increment in x or y direction is equal to 1.
5. [Obtain the new pixel between the points]
   Intialize I to 1
   while ($i \leq 1$)
   {
   New pixel is (Integer(x_{new}), Integer(y_{new}))
   $i = i + 1$
   }
   The result is illustrated in Figure 7.

(a) Smoothed Character **(b) Interpolated Character**

Figure 7: Interpolated Character

4.5. Re-Sampling

Re-Sampling is performed to normalize input character to a constant number of points which are at equal distances. In this paper each character is reampled to 64 intervals. The total length of the character is computed by adding the Euclidean distances between successive points. This is divided by the number of intervals required after re-sampling. The original points are replaced with a new set at this constant spacing using piece-wise linear interpolation.

When a character has multiple strokes, each stroke is resampled separately such that the total number of points using the technique below[7]. All training characters having the same number of strokes are

considered as a set. The number of points in each stroke is made proportional to the average length of strokes obtained from the corresponding set. The kth point from the series of N points is selected according to the following equation:

$$K = i * \left(\frac{N}{64} \right) \tag{2}$$

The value of *i* is successively taken as *i* = 0,1, 2, . . . 64, and right-hand side of Eq. (2) is rounded to the nearest integer value to get all the 64 successive selected points. The resampled character is shown in the Figure 8. The three strokes of character are resampled in the ratio of 37:13:12 intervals.The resampled character U with three strokes is shown in the figure 9.

(a) Interpolated Character **(b) Resampled Character** **(c) Interpolated Character** **(d) Resampled Character**

Figure 9: Resampled Character with multiple StrokesMultiple Strokes

5. CONCLUSION

We have demonstrated preprocessing techniques over online data of HP data set Telugu strokes. The preprocessing techniques used are Normalization, Smoothing, Duplicate Point Removal, Interpolation and Resampling. This implementation is initial step. There is lot of scope for future enhancement towards the implementation of more preprocessing techniques. This is also first step towards Online Hanbdwriiten Telugu Character Recognition using HP data set. In future work we will study the recognition accuracy over the preprocessed data using different recognition models.

References

[1] Plamondon, R., Srihari, S.N.: Online and Offline Handwriting Recognition: A Comprehensive Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 22(1) (2000).

[2] Bharath A. and SriganeshMadhvanath,"Onlin Handwriting Recognitio for Indic Scripts", HP Laboratories, India, HPL-2008-45, May 5, 2008.

[3] Sriganesh Madhvanath, Deepu Vijayasenan and ThanigaiMuruganKadiresanLipiTK: A Generic Toolkit for Online Handwriting Recognition.International Workshop on Frontiers in Handwriting Recognition (IWFHR-10), La Baule, France, Oct 2006.

[4] X. Li, D.-Y. Yeung, On-line handwritten alphanumeric character recognition using dominant points in strokes, Pattern Recognition 30 (1)(1997) 31}44.

[5] G. S. Reddy, P. Sharma, S. R. M. Prasanna, C. Mahanta, and L. N. Sharma. Combined online and offline assamese handwritten numeral recognizer. in Proc. 18th National Conference on Communications (NCC-2012), pages 1-4, 2011.

[6] A. Sharma, "Online Handwritten Gurmukhi Character Recognition", PhD. Thesis, Thapar University, 2009.

[7] B. Huang, Y.B. Zhang, M.T. Kechadi: Preprocessing Techniques for Online Handwriting Recognition.Intelligent Text Categorization and Clustering 2009: 25-45.