ISSN 2047-3338

DataSet Generation and Feature Extraction for Telugu Hand-Written Recognition

Aparna Varalakshmi[1], Atul Negi[2] and Sai Krishna[3]
[1]Bharat Institutions, Hyderabad, Andhra Pradesh, India
[2]Hyderabad Central University, Hyderabad, Andhra Pradesh, India
[3]QIS College of Engineering, Ongole, Prakasam District, Andhra Pradesh, India
aparna.qatester@gmail.com, atul.negi@gmail.com, tvsai.kris@gmail.com

***Abstract*– In this paper we propose feature extraction for Telugu handwritten recognition based on the candidate search and elimination technique. The initial candidates for recognition are found by applying by zoning method on input glyphs. We propose cavities as a structural approach suited specifically for Telugu script, where cavity vectors are used to prune the candidates by zoning. It gives the 100% features and cavity features of the input dataset.**

***Index Terms*– DataSet Generation, Feature Extraction, OCR for Telugu Hand-Written Text and Telugu Hand-Written Text**

I. INTRODUCTION

TELGU is one of the prominent scripts in India and Asia, with more than 846 million speakers. While it is seen that OCR technology is in a mature stage of development for English and other Roman/Latin scripts, the progress of OCR in Asian and particularly Indian scripts is in a relatively nascent stage. One of the reasons is the complexity of the orthography, especially in *Telugu*. While potentially 10000 syllables are frequently used in the language, the orthographic units are composed by combinations of 36 consonants and 16 vowels. A practical OCR system for *Telugu* script was proposed and developed by Negi et al [3], where the complexity of *Telugu* script and methods for its reduction were proposed. Their approach consists of identification and recognition of connected components. In this paper we propose an improved and robust recognition strategy which first uses the pixel distributions of the script and later exploits the structural information of *Telugu* orthography. In this paper we do not discuss layout related issues for the isolation of *Telugu* text regions, which is taken up elsewhere.

II. PROPOSED APPROACH

Following the approach of Negi [1], we focus on recognizing from the order of 983 distinct glyphs, which are extracted as connected components from the input image. We attempted a technique which is not greatly affected by the size of the training set. However, this would imply a system based on a candidate search and elimination technique. Our system consists of the stages as shown in Fig. 1.

A. Input Page Description

Input page is shown as follows in Fig. 2.

i) Forms are made with proper of letters on pages. It is made so that automatic extraction is possible.
ii) Each row containing 10 characters except last page each page containing 90 characters. Totally 983 Telugu characters are to be filled with single handwriting.
iii) Each page containing a circle at the top of the page which denoting the page number of the one set of handwriting.
iv) Each page containing the horizontal and vertical block at bottom right corner for representing the right orientation of the scanned image.
v) Every vertical and horizontal line extends each column or row for identifying the starting column or row position in the form of pixel.
vi) Each block is having the width of 50 pixels and length of 60 pixels.

B. Preprocessing

Preprocessing will enhance the image clarity by cleaning-up the image and increasing the threshold value. The preprocessed image will give input to the next phase.

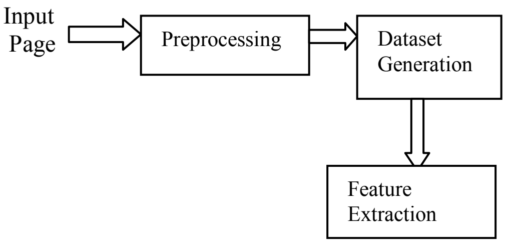


Fig. 1. Phases of the OCR system

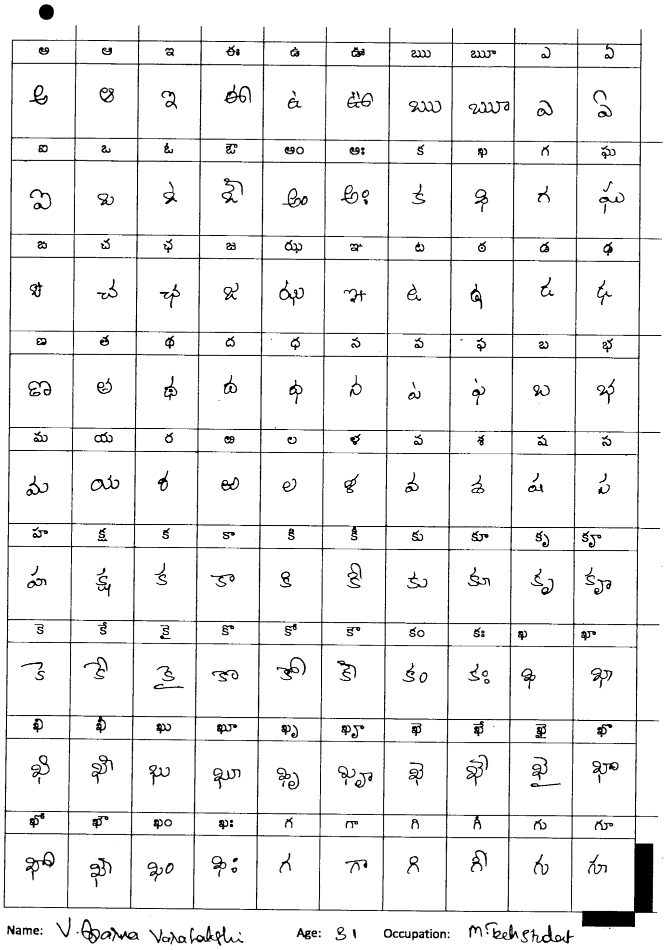

Name: V. Aparna Varalakshmi
Age: 31
Occupation: M.Tech Student

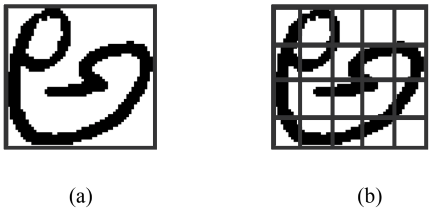
Fig. 2. Gray level image scanned with 300 dpi

C. Dataset Generation

i) It creates the 983 folders for 983 characters
ii) It takes the input of the image and check for orientation of the page. If the page is not in right orientation rotate 180 degrees
iii) Check for the circle position of the image for identifying what is the actual character starting number
iv) Find the first row and column positions of the page
v) Find the box coordinates
vi) Check for the bounding box character coordinates
vii) Crop the character and put it on the respective folder
viii) Repeat (v) to (vii) until last box coordinates finding out

D. Feature Extraction

1) Candidate Search (zoning): For a candidate search we use the measure of density of pixel distribution in different zones of the input glyph as a feature vector. First the input glyph is broken into zones by super-imposing a grid and then the percentage of the number of foreground pixels is calculated as in Fig. 2. This produces a 16 (5x4) dimensional feature vector represented as *(i1,i2,...,i20)* which corresponds to the grids from top left to the bottom right, in that order. A codebook of this feature vector is pre-computed from the training set. The feature vector of the input glyph is computed and searched in the codebook to obtain *k* (5 in our case) nearest neighbors *(n)*. The distance measure is Euclidean Distance between the feature vectors.



0.279221	0.383117	0.000000	0.000000
0.500000	0.344156	0.402597	0.493506
0.474026	0.246753	0.571429	0.350649
0.370130	0.162338	0.103896	0.402597
0.422078	0.435065	0.448052	0.038961

(c)

Fig. 3. (a) Input Glyph (b) Grid Superimposed (c) Corresponding grid weights as percentages

For finding the grid weights we have taken the formula as:

*Grid weight = (black count in zone/Total number of pixels) * 100*

2) Cavity Based Structural Analysis: Many *Telugu* characters have cavities (holes) in them. Cavities are used as structural features in our recognition. The existence and position of these cavities is a structurally distinguishing feature. We use cavities since they provide discrimination between glyphs which are could be very confusing for recognition. For example we show the glyphs in the Fig. 4.

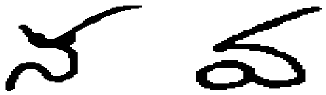


Fig. 4. Glyphs which are very confusable without Cavity Analysis

Fig. 5. The letter written by two different personalities

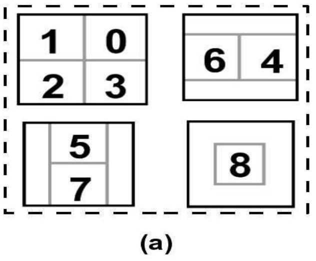


Fig. 6. (a) Overlapping quadrants [1] (b) Contour image and its cavity vector

The cavities of the same character have changed depending upon the handwritings of two different personalities which is shown in Fig. 5.

Cavities are detected by generating a contour of the glyph and performing connected component detection on the contour image, since cavities get disconnected from outer boundary in a contour image. The bounding box of the cavity contour should cover an area between threshold low (5%) and threshold high (25%) of the total glyph area, else it is discarded either as being too low or too large. To determine the position of the cavity we divide the total glyph area into 9 overlapping quadrants as shown in Fig. 6.

They are numbered sequentially from 0 to 8. The existence of a cavity in these nine quadrants is shown by a boolean value (since more than one cavity in a quadrant is not possible for the *Telugu* orthography) generating a 9 bit vector. Since contour generation, and connected component analysis are low complexity algorithms, the overall complexity of this stage remains low.

III. CONCLUSION AND FURTHER WORK

The feature generation phase will give the features of the ideal character and various sampled characters. We have taken the 100 different sets of characters .Each sample consisting of 983 Telugu characters. After finding the feature we require to match the unknown sample to the existing letter features then recognize the character. The entire OCR required implementing the Matching procedure for recognizing the character. We can follow the various different procedures for generating the features like wavelet transformation.

REFERENCES

[1] Atul Negi, Chandra Kanth Chereddi. Candidate Search and elimination approach for telugu OCR, TENCON Conference, 2003.

[2] Atul Negi, Chakravarthy Bhagavati, and B. Krishna. An OCR System for Telugu. In Proceedings 6th ICDAR, Seattle, USA, 2001.

[3] Atul Negi, Nikhil Shanker, and Chandra Kanth Chereddi Localization, Extraction and Recognition of text in Telugu Document Images. In Proceedings of the 7th ICDAR, Edinburgh, Scotland, 2003.