IMPROVEMENT OF TELUGU OCR BY SEGMENTATION OF TOUCHING CHARACTERS

J. Bharathi[1], P. Chandrasekhar Reddy[2]

[1] *Associate Professor, Department of ECE, DCET, Hyderabad, Telangana, India*
[2] *Professor, Department of ECE, JNTUCE, Hyderabad, Telangana, India*

Abstract

The reported success rates for Telugu OCRs are 84-87% for fonts sizes from 12-20 and 95.4-98.5% for sizes from 15 to 35. Some of the issues mentioned in the literature are noise and confusion characters. Studies by the authors have indicated that the touching characters constitute about 1% - 2% of the total characters in printed books of normal size fonts (14 pts). The editable output of OCR System has additional errors due to incorrect code selection emphasizing the need to identify the touching characters. Identification of touching characters is a challenge as the touching may occur at different places due to orthography and rules of grammar. A complete strategy of identification, segmentation and recognition system is proposed along with syllable models for segmentation. Effect of normalization methods at preprocessing stage for improving the identification of touching characters and recognition rates of normal characters is studied. A new algorithm is proposed for segmenting the touching conjunct consonants. The use of augmented database shows clear improvement in the recognition rates. The touching characters are identified and segmented successfully with 83% success rate, thus improving the overall performance of OCR System for Telugu.

Key Words: *Telugu OCR, Touching characters, Syllable model Non Linear Normalization, Hausdorff distance, Augmented Database*

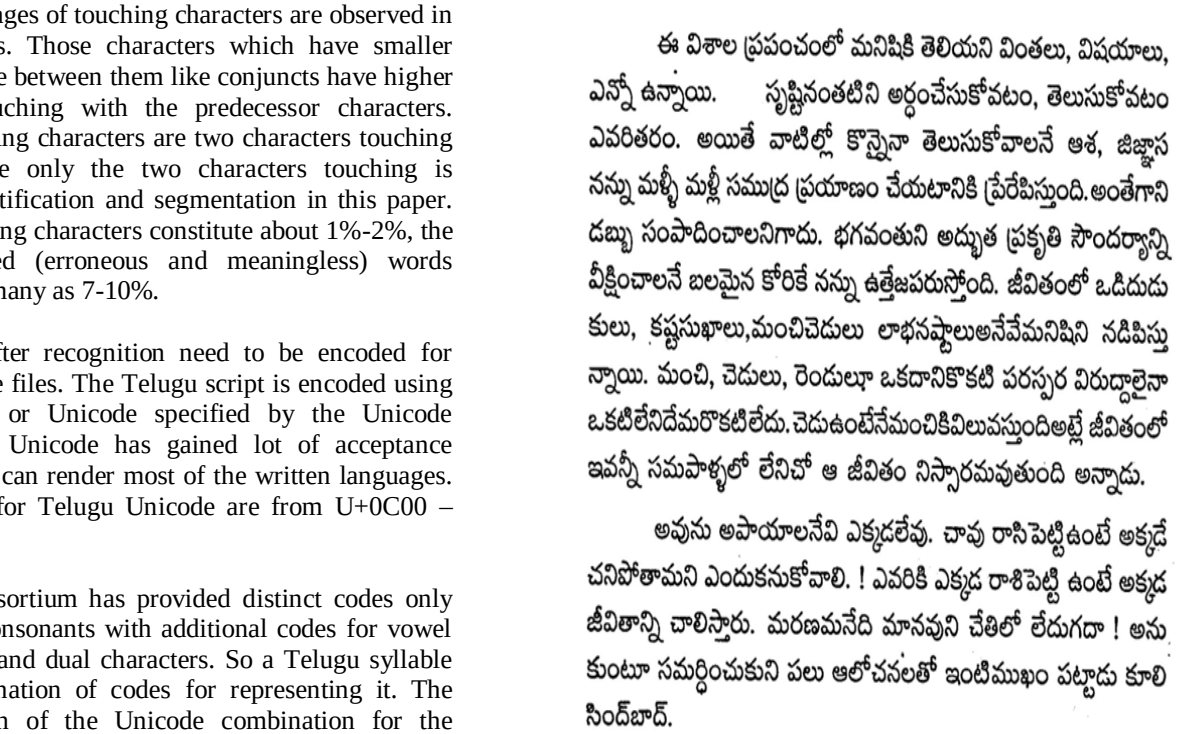
----------***----------

1. INTRODUCTION

The errors in each OCR module add up to further errors and reduce the success rate. Improper binarization, low grade paper, small font characters, ink smear results in touching characters and cause errors in line segmentation and classification errors. Size of font also plays considerable role; high percentages of touching characters are observed in small sized prints. Those characters which have smaller separation distance between them like conjuncts have higher possibility of touching with the predecessor characters. Majority of touching characters are two characters touching each other, hence only the two characters touching is proposed for identification and segmentation in this paper. Though the touching characters constitute about 1%-2%, the incorrectly formed (erroneous and meaningless) words account for in as many as 7-10%.

The characters after recognition need to be encoded for producing editable files. The Telugu script is encoded using either the ISCII or Unicode specified by the Unicode Consortium. The Unicode has gained lot of acceptance among users as it can render most of the written languages. The code points for Telugu Unicode are from U+0C00 – U+0C7F [1].

The Unicode consortium has provided distinct codes only for vowels and consonants with additional codes for vowel modifiers, halant and dual characters. So a Telugu syllable requires a combination of codes for representing it. The incorrect selection of the Unicode combination for the touching character and the next character code may lead to further error and the word formed out of this erroneous recognition at times cannot be printed properly. Unicode of a combined character depends on the primary character and secondary form of consonant modifiers. The incorrectly recognized primary character as a conjunct consonant prevents the printing of second character properly with a preceding dotted circle.



ఈ విశాల ప్రపంచంలో మనిషికి తెలియని వింతలు, విషయాలు, ఎన్నో ఉన్నాయి. సృష్టినంతటిని అర్థంచేసుకోవటం, తెలుసుకోవటం ఎవరితరం. అయితే వాటిల్లో కొన్నైనా తెలుసుకోవాలనే ఆశ, జిజ్ఞాస నన్ను మళ్ళీ మళ్లీ సముద్ర ప్రయాణం చేయటానికి ప్రేరేపిస్తుంది.అంతేగాని డబ్బు సంపాదించాలనిగాదు. భగవంతుని అద్భుత ప్రకృతి సౌందర్యాన్ని వీక్షించాలనే బలమైన కోరికే నన్ను ఉత్తేజపరుస్తోంది. జీవితంలో ఒడిదుడుకులు, కష్టసుఖాలు,మంచిచెడులు లాభనష్టాలుఅనేవేమనిషిని నడిపిస్తున్నాయి. మంచి, చెడులు, రెండులూ ఒకదానికొకటి పరస్పర విరుద్ధాలైనా ఒకటిలేనిదేమరొకటిలేదు.చెడుఉంటేనేమంచికివిలువస్తుందిఅట్లే జీవితంలో ఇవన్నీ సమపాళ్ళలో లేనిచో ఆ జీవితం నిస్సారమవుతుంది అన్నాడు.

అవును అపాయాలనేవి ఎక్కడలేవు. చావు రాసిపెట్టిఉంటే అక్కడే చనిపోతామని ఎందుకనుకోవాలి. ! ఎవరికి ఎక్కడ రాశిపెట్టి ఉంటే అక్కడ జీవితాన్ని చాలిస్తారు. మరణమనేది మానవుని చేతిలో లేదుగదా ! అనుకుంటూ సమర్థించుకుని పలు ఆలోచనలతో ఇంటిముఖం పట్టాడు కూలి సింద్‌బాద్.

Fig -1: Scanned page in Telugu, an Indian Language

ఈ విశాల ప్రపంచంలో మనిపికి తెలియని వింతలు, విషయాలు,
ఎనీఉనియి. సృష్టినంతటిని ఆర్థంచేసుకోవటం, తెలుసుకోవటం
ఎవరితరం. ఆయితే వాటిల్లో కొన్నినా తెలుసుకోవాలనే ఆశ, జిజ్ఞాస
నన్ను మన్ మళ్లీ సముద్ర ప్రయాణం చేయటానికి ప్రేరేపిస్తుంది.అంతేగాని
దశ సంపాదించాలనిగాదు. భగవంతుని ఆశ్చత ప్రకృతి సొందఇనీ
వీకక్షించాలనే బలమైన కోరికే నన్ను ఉత్తేజపరుస్తోంది. జీవితంలో ఒడిదుడు
కులు, కష్టసుఖాలు,మంచిచెడులు లాభనపులుఆనేవేమనిషిని నడిపిస్తు

శియి. మంచి, చెడులు, రెండులూ ఒకదానికొకటీ పరస్పర విరుద్దాలైనా
ఒకటీలేనిదేమరొకటిలేదు. చెడుఉంటేనేమంచికీవిలువస్తుందిఆట్లే జీవితంలో
ఇవపు సమపాళ్ళలో లేనిచో ఆ జీవితం నినీరమవుతుంది అళడు.
అవును అపాయాలనేవి ఎక్కదలేవు. చావు రాసిపెట్టిఉంటే అక్కదే
చనిపోతామని ఎందుకనుకోవాలి. ఎవరికి ఎక్కడ రాశిపెట్టి ఉంటే ఆకళో
జీవితావు చాలిస్తారు మరణమనేది మానవుని చేతిలో లేదుగదా . ఆను
కుంటూ సమర్థించుకుని పలు అలోచనలతో ఇంటీముఖం పట్టాడు కూలి సి ద్బా ద్ .

Fig -2(a): OCR output before segmentation in Telugu

ఈ విశాల ప్రపంచంలో మనిపికి తెలియని వింతలు, విషయాలు,
ఎన్నో ఉన్నాయి. సృష్టినంతటిని ఆర్థంచేసుకోవటం, తెలుసుకోవటం
ఎవరితరం. ఆయితే వాటిల్లో కొనెయ్యినా తెలుసుకోవాలనే ఆశ, జిజ్ఞాస
నన్ను మళ్ళీ మళ్లీ సముద్ర ప్రయాణం చేయటానికి ప్రేరేపిస్తుంది.అంతేగాని
దబ్బు సంపాదించాలనిగాదు. భగవంతుని ఆద్భుత ప్రకృతి సొందర్యాన్ని
వీకక్షించాలనే బలమైన కోరికే నన్ను ఉత్తేజపరుస్తోంది. జీవితంలో ఒడిదుడు
కులు, కష్టసుఖాలు,మంచిచెడులు లాభనపాటలుఆనేవేమనిషిని నడిపిస్తు

న్నాయి. మంచి, చెడులు, రెండులూ ఒకదానికొకటీ పరస్పర విరుద్దాలైనా
ఒకటీలేనిదేమరొకటిలేదు. చెడుఉంటేనేమంచికీవిలువస్తుందిఆట్లే జీవితంలో
ఇవన్నీ సమపాళ్ళలో లేనిచో ఆ జీవితం నిస్సారమవుతుంది అన్నాడు.
అవును అపాయాలనేవి ఎక్కదలేవు. చావు రాసిపెట్టిఉంటే అక్కదే
చనిపోతామని ఎందుకనుకోవాలి. ఎవరికి ఎక్కడ రాశిపెట్టి ఉంటే ఆకజ్క
జీవితాన్ని చాలిస్తారు మరణమనేది మానవుని చేతిలో లేదుగదా . ఆను
కుంటూ సమర్థించుకుని పలు అలోచనలతో ఇంటీముఖం పట్టాడు కూలి సి ద్బా ద్ .

Fig -2(b): OCR output after segmentation in Telugu

2. LITERATURE SURVEY

C. Vasantha Lashmi, Ritu Jain and C. Patvardhan [2], have proposed an OCR for Telugu and reported success rates of 98.5% for font sizes between 15 and 18. Negi et al. [3] reported success rates of 92% for their 'Dhrishti' OCR system. Some of the issues reported are noise, confusion characters. The touching characters are not considered or are not present due to larger font sizes used for the studies.
Pavan Kumar and Negi have proposed algorithms for recognition of broken characters for improving the OCR accuracy [4].

Garain et al and Veena Bansal have proposed algorithms for segmentation of Devanagari script in [5,6]. Touching characters in degraded documents in Gurumukhi are studied by Jindal and Manish Kumar [7,8].

The touching characters and their segmentation attracted scarce attention for Telugu script. L. P. Reddy and others [9] have considered an optimum threshold which is an averaging function of the syllable object widths to be classified for identifying the touching character and proposed a split profile algorithm for segmenting them. This is useful for characters touching in middle zone only.

Bharathi and Chandrasekhar Reddy have identified four major types of frequent touching characters [10,11,12] due to their canonical structure and also proposed algorithms to segment them, thus increasing the success rates. The recognition of the touching characters from a regular character is really a challenging task. In this paper Normalization of characters is proposed as suitable technique to identify the touching characters.

A syllable model is proposed in this paper which facilitates the proposal of segmentation algorithms for touching characters.

3. STRATEGIES FOR IMPROVING THE OCR

3.1 Syllable Model

The Telugu script is syllabic in nature. Each character i.e. vowel, consonant or their modifier may consist of one or more connected components or glyphs.

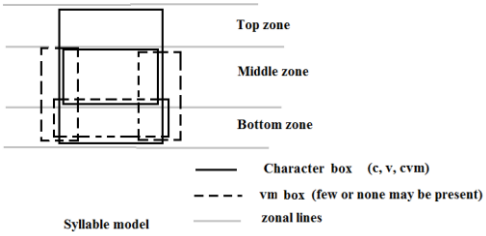


Fig -3: Proposed model of Telugu Syllable

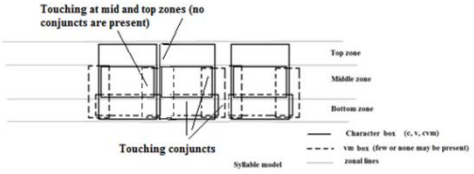


Fig -4: Syllable model with touching positions

The Syllable follows the C(C(V)) form where C is the pure consonant and V is the vowel. The conjunct consonants may not be more than five.
Combined characters in Telugu are written with the first character in full form and the second character below or next to the first character (Fig.3). The second character is known as vattu or secondary form of consonant. The shape for most of the constants differs from the primary consonant shape.

The above proposed model has primary form of character (either vowel or consonant) followed by none or more consonant modifiers. Only one form of consonant modifier is written on left side (a variant of 'ra' vattu). Touching

usually occurs between the primary character and conjuncts within the syllable. The touching also occurs between syllables when there are no conjuncts between them (Fig. 4). The above model facilitates the understanding of touching behavior clearly and suitable algorithms can be developed for segmenting the touching characters. Examples touching characters with segmentation locations are shown in Fig. 5.

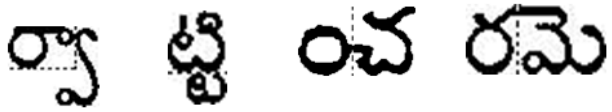


Fig -5: Example touching characters and their segmentation in Telugu

Bharathi and Chandrasekhar Reddy [10,11,12] have identified and segmented four types of touching characters, conjunct consonants type I, type II, characters touching in middle zone and touching in top zone. They have proposed an overlapping bounding box algorithm for segmenting the type I conjuncts. A more efficient algorithm is presented in this paper based on the concavity at touching point in bottom side profile.

3.2 Normalization

Pre-processing is a major step before classification stage. Many of the features extracted for the character recognition systems depend on size of the image. The fixed size feature vectors can be easily compared. The size normalization of an image transforms it to a pre-defined size. A smaller normalized size leaves out finer details where as larger size may not represent accurate details due to artifacts. The size should be so chosen to judiciously balance these two issues.

3.2.1 Linear Normalization

Linear transformation is extensively adopted for the character recognition problems. The linear normalization transforms a location (x,y) into (x`,y`) given by

$$x` = a_1x + a_2y + a_3 \quad (1)$$

$$y` = a_4x + a_5y + a_6 \quad (2)$$

where a_1, a_2, … a_6 are constants.

In Aspect Ratio Adaptive Normalization (ARAN) [13,14] the aspect ratio of the input image R1 is equal to the aspect ratio of the normalized image R2. The larger dimension fills the full side of normalized image and the shorter side is centered in the image.

$R_1 = \min(W_1,H_1)/\max(W_1,H_1)$

Scaling factor $\alpha = L/R_1$

Assuming that W_1 is the longer side, shorter side $H_2 = \alpha H_1$

Where W_1,H_1 = width and height of image before normalization

W_2,H_2 = width and height of image after normalization

L x L = size of the normalization plane

R_1 = Aspect ratio of input image

The height is centered leaving a space at top and bottom of the image

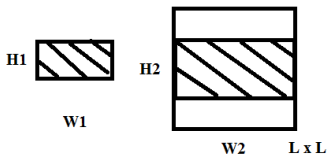


Fig -6: Normalization with fixed aspect ratio

3.2.2 Non-Linear Normalization

The font variations and the large set of classes make the identification of the class of each character challenging.
Shape variation or distortion is an important issue to be solved besides variations like position, size, rotation, inclination. These variations need to be corrected or compensated by some form of transformation for better results.

Many normalization techniques are presented in the literature [14,15,16,17,18]. Negi et al. [19] used non-linear normalization method for recognizing the Telugu characters.

Non-linear normalization (NLN) reduces the within class shape variation [14] and achieves significant improvements in the recognition accuracy.

Two methods of NLN are proposed by Yamashita et al. [16] and Yamada. In both the methods a characteristic feature is considered and the cumulative feature projection is normalized. The feature densities are equalized by projecting them on a horizontal or vertical axis and re-sampling the feature projections.

Dot density feature: The method is proposed by Yamashita et al [16]. In this method dot densities are obtained by projecting the black pixels on to horizontal and vertical axis.
Consider the initial binary image *P(i,j), i = 1,2, .. I; j = 1,2, … J.* The transformed normalized image is denoted by *Q(m,n), m = 1,2, .. M; n = 1,2, … N*. The feature projection functions are $fp_x(i)$ and $fp_y(j)$ projected respectively on horizontal and vertical axes. The feature density equalization calculates the new position *(m,n)* of the initial position *(i,j).*

Feature projection

$$fp_x(i) = \sum_{j=1}^{J} P(i,j) + \alpha_x \quad (3)$$

$$fp_y(j) = \sum_{i=1}^{I} P(i,j) + \alpha_y \quad (4)$$

where α_x and α_y are constants and considered zero in the present calculations.

Feature density equalization

The new position *(m,n)* are given by

$$m = \sum_{k=1}^{i} fp_x(k) \frac{M}{\sum_{k=1}^{I} fp_x(k)} \quad (5)$$

$$n = \sum_{l=1}^{j} fp_y(l) \frac{N}{\sum_{l=1}^{J} fp_y(l)} \quad (6)$$

Line density by crossing lines (Zero Crossings) method: Yamada et al. developed this method. In this method the zero crossing of the black pixels are considered as line density.

Feature projection

$$fp_x(i) = \sum_{j=1}^{J} (\sim P(i, j-1)).P(i,j) + \alpha_x \quad (7)$$

$$fp_y(j) = \sum_{i=1}^{I} (\sim P(i-1, j)).P(i,j) + \alpha_y \quad (8)$$

Where *P(i,0) = 0, P(0,i) = 0, ~P(i,j)* is a logical inversion of the image.

Feature density equalization

The feature density equalization is achieved by the equations 5 and 6 in which fp_x and fp_y are replaced by the zero crossing projections on horizontal and vertical axes respectively.
There may be holes in the normalized image if the image is enlarged. These holes are filled using the smearing technique.

Moment normalization: The moment normalization is a linear global method [17]. It aligns the centroid of input image to the geometric center and scales the image with the second order one dimensional moments. The moments are calculated using equations below

$$\mu_{20} = \sum_x (x - x_c)^2 f_x(x) \quad (10)$$

$$\mu_{02} = \sum_x (y - y_c)^2 f_y(y) \quad (11)$$

Where

x_c, y_c are coordinates of centroid

$f_x(x), f_y(y)$ are projections

μ_{20} *and* μ_{02} are one dimensional moments

The centroid of the image is shifted to the center of the normalized plane ($W_2/2, H_2/2$). The width and height of the image plane is re-determined according to the moments calculated above.

$$\delta_x = \alpha\sqrt{\mu_{20}} \quad (12)$$

$$\delta_y = \alpha\sqrt{\mu_{02}} \quad (13)$$

The image boundaries are reset to $[x_c-\delta_x/2, x_{c+}\delta_x/2]$ and $[y_c-\delta_y/2, y_{c+}\delta_y/2]$. The coordinates are mapped according to

$$x' = (x - x_c)\frac{W_2}{\delta_x} + x'_c \quad (14)$$

$$y' = (y - y_c)\frac{H_2}{\delta_y} + y'_c \quad (15)$$

This moment normalization method is applied at preprocessing stage and the recognition rates are evaluated.

3.3 Modified Algorithm for Segmentation of Touching Conjunct Consonants

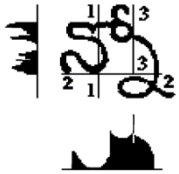


Fig -7: Segmentation of touching conjunct consonant (Type-I) in Telugu

The touching conjuncts constitute a major portion of touching characters. In Samyuktaksharas or combined syllables, the primary form of a character is followed by a secondary form of a consonant or vattu. The shape of a secondary form of a consonant is completely different except in case of eight characters. The primary character is written in middle zone, the secondary character is written next to the primary, and occupies bottom and middle zone. This is categorized as Type-1 and is characterized by the overlap of the bounding box of each character. Because of the overlap we cannot use a vertical segmenting section for separating them.

Bharathi and Chandrasekhar Reddy [10] have proposed an algorithm for segmenting this type of touching, however the algorithm cannot segment correctly for the touching character shown above where the extending arm at right on top is corrugated instead of a straight line. An improved algorithm based on the concavity in the bottom side profile at horizontal segmentation line is described below.

As the conjunct modifier is written next to the first character, we have space at bottom of the first character. A partial horizontal projection profile of the image calculated at distance of 0.2W-0.3W from left edge will have the pixel count zero below the character (Fig. 7). The bottom line (2-2) of the first character is inferred from the above partial horizontal profile. The segmentation line (3-3) is passing through the touching point. This point is easily identified while traversing from right to left. The shape of conjunct consonant above the line (2-2) is curved towards left followed by the concavity of the right most part of the first character. In the reverse traversal along line 1-1 from right side the left edge of the conjunct should be identified. A bottom side profile of at line 2-2 is calculated. The profile moves up when traversed towards left, then moves down after the correct segmentation location (touching point) along concavity. The segmentation line is along 3-3 then move left along 2-2.

Algorithm

1. Read the image
$$I \xleftarrow{read} imagefile$$
2. Convert to gray scale image
$$I_g \xleftarrow{convert} I$$
3. Binarize the image
$$I_b \xleftarrow{binarize} I_g$$
4. Construct $P\_HPP_{0.3W}$ partial horizontal projection profile at 0.3W from left.
$$P\_HPP_{0.3W} = \sum_{j}^{0.3W} I_b(i,j) \ for \ \forall i$$
5. Find the starting position of the character, ZWH
ZWH = h if $P\_HPP_{0.3w}(h\text{-}1) = 0$ for all h < ZWH
$P\_HPP_{0.3w}(h) \sim= 0$
where *ZWH* is the zero width height of the profile from bottom measured from bottom of image
6. Construct bottom side profile B_SP_{ZWH} of the top part.
$$B\_SP_{ZWH} = \sum_{i=ZWH}^{k-1} I_b(i,j), \qquad until \ I_k = 0$$
7. Identify the touching point TP
$B\_SP_{ZWH}(TP) >= B\_SP_{ZWH}(TP\text{-}1)$
$B\_SP_{ZWH}(TP) > B\_SP_{ZWH}(TP+1)$
when traversing from right
8. Segmentation line is along *3-TP-3* and *2-2 at ZWH*

3.4 Baseline OCR

The identification of touching characters and subsequently successfully segmenting them improves the overall performance of the OCR system. A baseline OCR system is developed for testing the segmentation algorithms.

The distance between two character images is computed and the image matching with the shortest distance is selected as recognized character. In the present paper the recognition task is performed using template matching.

Negi et. al [3] have considered the Fringe distance in the Telugu OCR 'Drishti' developed for Linux systems with success rate of 92%. We have used the fringe distance metric for the recognition module to test the performance of the algorithms presented.

3.4.1Template Matching

Template matching in Pattern Recognition is one of the methods of identifying the class of an image for an unlabeled image. Template matching of image involves defining measures of similarity and comparing the measure of similarity of an image to those of a large number of template images to satisfy a specified criterion. The correct match is the one which has least dissimilarity except for small differences in pixel positions and intensities.

The objective is to assign a label to an object **X** from one of classes **Z** or populations which are specified by templates { **Y***i* } for i = 1,2,… **Z**. A metric M(**X**, **Y***i*) is defined between object **X** and all classes { **Y*i*** }, then a label of class *k* may assigned to **X** if M*k*(**X**, **Y***k*) < M*i*(**X**, Y*i*) for all $i \neq k$ [20].

$$X \xleftarrow{\text{assign label}} \{Y_i\} \ \forall \ i = 1 \ldots Z$$
$$M_k < M_i \ \forall \ i \neq k$$

A comparatively efficient method allows a large set of templates to be tested for better result. The property considered for matching and the distance metric are important in Template matching.

R.L. Brown[21] has proposed a method based on Fringe distance measure which is faster when compared to the Gaussian blurring and at the same time gives comparable results to it. Pre computed values can be efficiently used in Fringe Distance method to speed up the process.

Fig -8: Image A and B of the same character in Telugu

3.4.1 Distance Metric

Fringe distance: Consider two images A and B (Fig. 8). They are similar if each pixel in A (p_A) is at least close to nearest pixel in B (p_B) and vice versa. Denote the distance between p_A and p_B as 'd_{AB}', calculated after overlaying image A over image B (Fig. 9). The sum of these distances 'd_{AB}' and 'd_{BA}', is defined as Fringe Distance between the two images. The distance reflects small distortions and shape variation in both the images. Brown has proposed 'Fringe Map' to rapidly calculate this distance (Fig. 10).

$Fringe\ distance = \sum_A d^i_{AB} + \sum_B d^i_{BA}$
d^i_{AB} = nearest distance between two pixels in A and B.

Fig -9: Superimposed images of A and B in Telugu. Image A is shown in gray for comparison

Fig -10: Fringe map of character image

Hausdorff distance: The Hausdorff distance between two sets of points A and B is defined as

$$H(A,B) = \max(h(A,B), h(B,A))$$

Where $h(A,B) = \underset{a \in A}{max} \ \underset{b \in B}{min}(||a-b||)$

$$A = \{a_1, \dots a_p\}, \qquad B = \{b_1, \dots b_q\}$$

The h(A,B) is the directed Hausdorff distance from A to B. Each point in A is considered and the maximum distance among all points of A to any point in B is the h(A,B). The maximum between the two directed distances from A to B and B to A is the Hausdorff distance. It measures the mismatch between the two sets of points.

Partial Hausdorff distance: Huttenlocher et al. [22] defined the partial Hausdorff distance as

$$d(A,B) = {}^{x}K^{th}_{a \in A}\, d(a,B)$$

where ${}^{x}K^{th}_{a\in A}$ represents the k^{th} rank distance such that $k/N_a = x$ %. The ${}^{50}K^{th}_{a\in A}$ is the median of the distance. M-P. Dubuisson and A.K.Jain [23] have defined 6 directed distances and 4 undirected Hausdorff distances. They have shown that

$$d(A,B) = \frac{1}{N_a}\sum_{a\in A} d(a,B) \qquad (15)$$

$$f = \max\big(d(A,B), d(B,A)\big) \qquad (16)$$

Has more tolerance to noise and has desirable properties like robustness to outliers, monotonous increase as the rest of edge points increase.

Similar to the Fringe map a Hausdorff map (Fig. 11) is pre-calculated and stored. This leads considerable time saving while calculating the Hausdorff distance.

The baseline OCR is used for identifying the touching characters and the performance with Fringe distance and the Hausdorff distance.

Fig -11: Hausdorff map for a character

4. METHODOLOGY

The page is scanned at a predefined resolution. The image is rotated at small increments in both the clockwise and counter clockwise directions. The horizontal projection profile of the black pixel count is calculated and the variance is found. The skew can be eliminated by rotation of the image to give minimum variance of horizontal projection profile.
The zonal disposition of the Telugu character intrudes into the space separating the lines and may sometimes touch the strokes of the top and bottom lines. A segmenting algorithm proposed by Bharathi et al. [24] is used to segment the lines.

The words are then segmented using the vertical projection profile of black pixels. The words are separated by wider spaces than the character separation spaces. After word separation, characters are segmented based on the white space between characters. The character may have zero or several connected components. While segmenting the combined characters, the components are written to the disk file starting from the top most component in sequence [4]. This helps in identification of the component and recognizing the combined character sequentially. The sequence is important as the modification of basic character is to be applied the consonant modifier in Unicode mapping. The improper sequence may result in erroneous result. These are separated using the connected component analysis and are written to a disk file. Word separators and line separators are inserted while writing the connected components to the disk which helps in displaying the content properly. The characters are normalized with Non-linear normalization at recognition stage.

4.1 Threshold Selection

The histogram of all the minimum distances to prototype characters approximates positively skewed normal distribution (Fig. 12). The outliers are towards right side of the mean of the distribution. Small sized characters like full stop, comma and components of multi component consonants (talakattu, gudi etc.,) , characters having smaller heights (some conjunct consonants like la, tha vattu) and touching characters have high fringe distances. It is found that threshold value having (μ+1.5*σ) distance can discriminate the touching conjuncts with fairly good success rates.

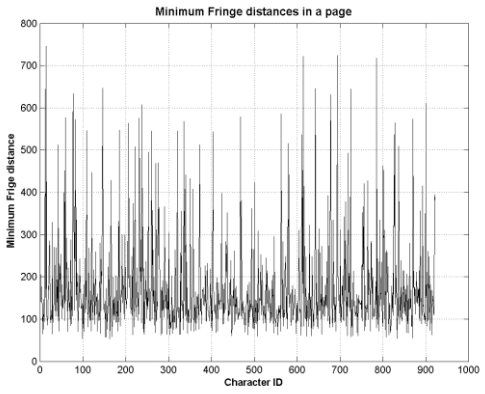


Fig -12(a): Distances in a page

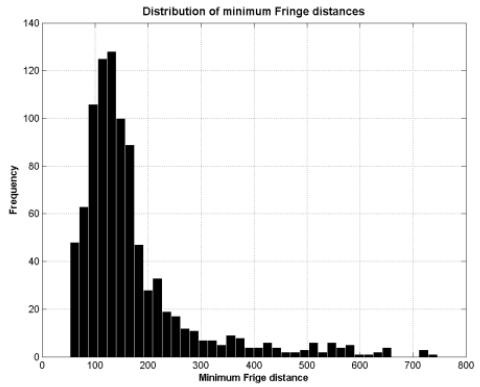


Fig -12(a): Histogram of distances in a page

The characters having less than the threshold value are considered as correct and are written to the output file. The mid zone touching and top zone touching characters have less fringe distance when compared to touching conjuncts and may fall below the proposed threshold value. So the characteristic feature of having white space area in the bottom side profile is used to consider as probable candidates for segmenting even though they have less fringe distance. The characters having more than threshold value are considered as touching. However these include characters having high pixel densities and secondary form of consonants of smaller size. These are filtered out and the rest are the candidates for segmentation.

Each candidate image is segmented with the algorithms for touching conjunct consonants of type-1 and type-2, mid zone and top zone touching characters. The resulting individual characters are labeled and the segmentation yielding low fringe distance is written to the output file.

All the algorithms described in [10,11,12] are applied to the touching character and subsequently segmented. The resulting segmented characters for each algorithm are again identified with the recognition module and distances are calculated. The resulting total distance should be minimum, if the segmentation is correct. Hence the total distances using each segmentation algorithm is computed and the least among them is considered as correct segmentation.

4.2 Displaying the final output

The output in Unicode is written to a file. The file can be opened by a Unicode compliant browser. We propose to utilize the web browsers to render the syllable properly.

4.3 Augmented Database

The database should have images of all variants of prototypes. However large set of images require more time. Fringe distance map and Hausdorff distance maps of the images in the database are pre-calculated to reduce computation time.

Some Fonts have more connected components due to the variation of styles. Some characters have large variations and some have strokes touching within the same characters. These are all included in the database.

5. RESULTS

5.1 Performance of Baseline OCR

Pages from different books are scanned at 300 dpi and some pages are taken from books downloaded from Digital Library of India (DLI). A threshold value of (μ+1.5*σ) is found. If the minimum nearest distance is more than the threshold value, the characters are shown in red color. The some of the confusing characters are not correctly recognized even though the fringe distance is less than the threshold value. If the touching characters are shown in red they can be considered as correctly identified.

5.1.2 Normalization

The Non-linear normalization is found to be reducing the in-class variation of characters, thus better at recognition rates of regular characters. Moment normalization of characters also improves recognition, but the distance for touching characters to the model characters is not very much discriminative as the shifting of center of gravity requires reduction of size of characters to accommodate all the high and wider character classes.

5.1.3 Distance metric

The results for random 16 pages are shown in Table 2, Fig. 13. The results show an average success rate of 95.89% for Hausdorff distance and 96.12% for the Fringe distance in the presence of touching characters. The computational time is more for both the methods due to large number of calculations. However Hausdorff distance calculations require more time due to floating point.

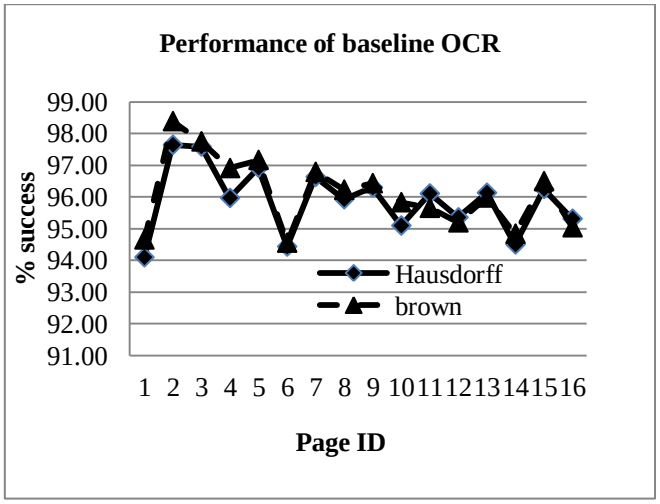


Fig -13: Performance of OCR in the presence of touching characters

5.2 Performance of Modified Algorithm for Segmenting Touching Conjuncts (Type-I)

The success rate achieved is 93.70 % for Type-1 touching conjuncts (Table 1).

Table -1: Segmentation of Touching characters (Type-I)

Total documents	188
Total characters	187,084
Total touching characters	3,149
Conjunct consonants (Type-1)	1,776
% of conjunct consonants	56.40 %
Correctly segmented	1,664
% of success	93.70 %

Table -2: Performance of OCR in the presence of touching characters

Page ID	Cha-rs	Wor-ds	Hausdorff distance		Fringe distance	
			errors	% success	errors	% success
1	711	142	20	94.09	16	94.66
2	804	165	15	97.64	9	98.38
3	621	109	9	97.58	8	97.75
4	743	146	17	95.96	10	96.90
5	810	168	14	96.91	12	97.16
6	808	135	18	94.43	17	94.55
7	653	146	12	96.63	11	96.78
8	662	144	14	95.92	12	96.22
9	756	173	14	96.30	13	96.43
10	815	160	20	95.09	14	95.83
11	667	142	16	96.10	19	95.65
12	645	148	20	95.35	21	95.19
13	671	147	15	96.13	16	95.98
14	854	162	19	94.50	18	94.84
15	799	175	16	96.25	15	96.49
16	767	165	19	95.31	22	95.04

5.3 Identification of touching characters

Pages: 16
Total characters: 11556
Total touching characters: 226
Correctly recognized: 181

The results of random 16 pages for the identification and segmentation of touching characters are presented in Tables 3 and 4. and graphically in Fig 14. Success rate of 83% is achieved in segmenting the touching characters.

Table -3: Segmentation of touching characters (Hausdorff distance)

Page ID	Cha-rs	Wo-rds	Tou-ching	Seg-me-nted	% char success	% word success
1	711	142	22	15	68.18	10.56
2	804	165	4	4	100.00	2.42
3	621	109	6	6	100.00	5.50
4	743	146	13	13	100.00	8.90
5	810	168	11	9	81.82	5.36
6	808	135	27	19	70.37	14.07
7	653	146	10	10	100.00	6.85
8	662	144	13	12	92.31	8.33
9	756	173	14	11	78.57	6.36
10	815	160	20	16	80.00	10.00
11	667	142	10	8	80.00	5.63
12	645	148	10	7	70.00	4.73
13	671	147	11	8	72.73	5.44
14	854	162	28	18	64.29	11.11
15	799	175	14	12	85.71	6.86
16	767	165	17	12.5	73.53	7.58

Table -4: Segmentation of touching characters (Fringe distance)

Page ID	Cha-rs	Wo-rds	Tou-ching	Seg-me-nted	% char success	% word success
1	711	142	22	15	68.18	10.56
2	804	165	4	4	100.00	2.42
3	621	109	6	6	100.00	5.50
4	743	146	13	13	100.00	8.90
5	810	168	11	9	81.82	5.36
6	808	135	27	19	70.37	14.07
7	653	146	10	10	100.00	6.85
8	662	144	13	12	92.31	8.33
9	756	173	14	13	92.86	7.51
10	815	160	20	19	95.00	11.88
11	667	142	10	8	80.00	5.63
12	645	148	10	7	70.00	4.73
13	671	147	11	7.5	68.18	5.10
14	854	162	26	19.5	75.00	12.04
15	799	175	13	8	61.54	4.57
16	767	165	16	12.5	78.13	7.58

The small sized characters like comma, full stop have large fringe distance, even though they are correctly recognized. They are easily eliminated from the set of identified touching characters with the 'high pixel density' criteria.

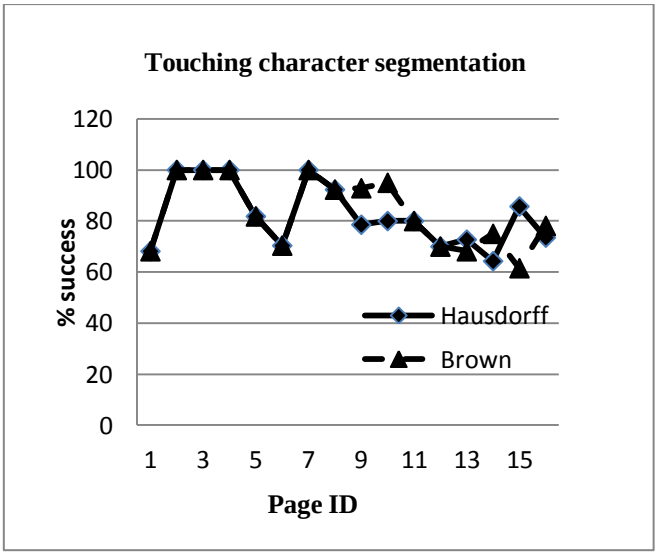


Fig -14: Touching character segmentation with Brown and Hausdorff distances

6. CONCLUSIONS

The Non Linear Normalization (NLN) at preprocessing stage is showing better performance. It is found to be reducing the in-class variation thereby giving reduced fringe distance.

The Fringe distance and Hausdorff distances have nearly equal performance; however Fringe distance calculations have lower computation time.

The threshold value of the distance can be arrived at with an initial run on the page. The value at 1.5 times the standard deviation from the mean identified the touching characters successfully with average segmentation rate of 83%. This improves the OCR performance by at least 1.5-1.8% at character level and 7.5% at word level.

Multiple strategies like non linear normalization at preprocessing stage, segmentation of touching characters, having an augmented database yields good results. This reduces the errors at each stage and improves the OCR success rate.

7. FUTURE STUDY

Only touching of two characters is considered in this paper. However multiple touching characters, touching of two secondary form of consonants needs further study.

REFERENCES

[1].Telugu Unicode Point table www.unicode.org/charts/PDF/ U0C00.PDF
[2]. C. Vasantha Lashmi, Ritu Jain and C. Patvardhan, "OCR of Printed Telugu Text with High Recognition Accuracies", Proceedings of ICVGIP, 2006, pp 786-795.
[3]. Atul Negi, Chakravarthy Bhagavati and B. Krishna, "An OCR System for Telugu", ICDAR, IEEE, 2001, pp 1110-1114.
[4]. P. Pavan Kumar, Chakravarthy Bhagvati, Atul Negi, Arun Agarwal, B. L. Deekshatulu, 2011. "Towards improving the Accuracy of Telugu OCR Systems", ICDAR, 2011, pp. 910-914.
[5]. Veena Bansal, Sinha, R. M. K., 2002, "Segmentation of touching and fused Devanagari Characters,", Pattern Recongnition, Vol. 35, No. 4, pp. 875–893.
[6]. Utpal Garain and Bidyut B. Chaudhuri, "Segmentation of touching characters in printed Devnagari and Bangla scripts using fuzzy multifactorial analysis", IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, Vol. 32, No. 4, November 2002, pp 449-459.
[7]. Jindal, M. K., Sharma, R. K., Lehal, G. S.,. "A study of different kinds of degradations in Printed Gurmukhi Script", , International Conference on Computing: Theory and Applications, ICCTA, 2007, pp. 538-544.
[8]. Manish Kumar, "Degraded Text Recognition of Gurmukhi Script", Ph.d Thesis, Dept of Computer Science and Engineering, Thapar University, India, March, 2008
[9]. Pratap Reddy, L., Ranga Babu, T., Venkata Rao, N., Raveendra Babu, B., 2010. "Touching Syllable Segmentation using Split Profile Algorithm", IJCSI, Vol. 7, Issue 3, No. 9, Nov 2010, pp. 17–26.
[10]. Bharathi. J, Chandrasekhar Reddy. P, "Segmentation of Telugu touching conjunct consonant using overlapping bounding boxes" in International Journal on Computer Science and Engineering (IJCSE), Vol. 5, No. 06, Jun 2013, pp 538-546.
[11]. Bharathi. J, Chandrasekhar Reddy. P, "Segmentation of touching conjunct consonants in Telugu using minimum area bounding boxes" in International Journal on Soft Computing and Engineering (IJSCE), Vol. 3, Issue. 03, July 2013, pp 260-264.
[12]. Bharathi. J, Chandrasekhar Reddy. P, "Classification and Segmentation of Telugu Touching Characters" in CiiT International Journal of Digital Image Processing, Vol. 5, Issue. 08, August 2013, pp ????.
[13]. C-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, 2000. "Aspect Ratio adaptive Normalization for Handwritten Character Recognition", Advances in Multimodal Interfaces, ICMI, Oct 2010, pp. 418-428.
[14]. Cheng-Lin Liu, Hiroshi Sako and Hiromichi Fugisawa, "Handwritten Chinese Character Recognition: Alternatives to Nonlinear Normalization", ICDAR 2003, pp 524-528.
[15]. Seong-Whan Lee and Jeong-Seon Park, "Non Linear Shape Normalization methods for the recognition of Large-Set Handwritten Characters", Pattern Recongnition, Vol 27, No. 7, 1994, pp 895-902.
[16]. Y. Yamashita, K. Higuchi, Y. Yamada and Y. Haga, "Classification of Printed Kanji Charactersby the Structured segment matching Method", Pattern Recongnition Letters, 1 7, 1983, pp 475-479.
[17]. R. G. Casey, "Moment Normalization of Hand Printed", IBM J. Res. Dev. 14, 1970, , pp 548-557.
[18]. C-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques", Pattern Recognition, 37(2), 2004, pp 265-279.
[19]. Atul Negi and C. K. Chereddi, "Candidate search and elimination approach for Telugu OCR", TENCON, IEEE, 2003, Vol 2,pp 745-748.
[20]. J. D. Tubbs, "A Note on Binary Template Matching", Pattern Recognition, Vol. 23, No. 4, 1989, pp 359-365.
[21]. R. L. Brown, "The Fringe Distance Measure: An Easily Calculated Image Distance Measure with Recognition Results Comparable to Gaussian Blurring", IEEE Transactions on Systems, Man and Cybernatics, January, 1994, Vol. 24, No. 1, pp 111-115.
[22]. D. P. Huttenlocher, G. A. Klanderman and W. J. Rucklidge, 1993 "Comparing Images using the Hausdorff distance", IEEE, Trans. PAMI, Vol 15, pp 850-863.
[23]. M.P Dubuisson and A.K.Jain, 1994, "A Modified Hausdorff Distance for Object Matching", Proc Intl. Conf. on Pattern Recognition, Jerusalem, pp566-568.
[24]. Bharathi. J, Chandrasekhar Reddy. P, 2013,"Segmentation of Text lines using sub-image Profile for Machine Printed Telugu Script", IJCET, Vol 4, Issue 6, pp 181-191.

BIOGRAPHIES

J. Bharathi received her M.Tech in DSCE from JNTUCE, Hyderabad, India. She is currently working as Associate Professor in DCET, Hyderabad, India. Her research interests include Image Processing, Speech and Signal Processing.

Dr. P. Chandrasekhar Reddy received his Ph.D from JNTU, Hyderabad, India. He is Professor in ECE Dept at JNTU, Hyderabad, India. He is an author of numerous technical papers in the Fields of High Speed Networking, Wireless Networks, Mobile and Wireless Communications and High Speed Communications and Protocols.