# Chapter - 3

## Theoretical Analysis

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Name of the Sub-Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>AIM AND OBJECTIVES</td>
<td>43</td>
</tr>
<tr>
<td>3.1.1</td>
<td>BACKGROUND</td>
<td>43</td>
</tr>
<tr>
<td>3.1.2</td>
<td>AIM AND OBJECTIVES OF THE RESEARCH</td>
<td>45</td>
</tr>
<tr>
<td>3.1.3</td>
<td>PROBLEM STATEMENT</td>
<td>46</td>
</tr>
<tr>
<td>3.1.3.1</td>
<td>BASELINE OCR</td>
<td>46</td>
</tr>
<tr>
<td>3.1.4</td>
<td>SCOPE OF STUDY</td>
<td>47</td>
</tr>
<tr>
<td>3.1.5</td>
<td>ASSUMPTIONS</td>
<td>47</td>
</tr>
<tr>
<td>3.2</td>
<td>CHARACTERISTICS OF TELUGU SCRIPT</td>
<td>47</td>
</tr>
<tr>
<td>3.2.1</td>
<td>STRUCTURAL CHARACTERISTICS</td>
<td>51</td>
</tr>
<tr>
<td>3.2.2</td>
<td>GENERAL OBSERVATIONS ON TOUCHING CHARACTERS</td>
<td>53</td>
</tr>
<tr>
<td>3.3</td>
<td>SYLLABLE MODEL</td>
<td>54</td>
</tr>
<tr>
<td>3.4</td>
<td>LINE SEGMENTATION</td>
<td>56</td>
</tr>
<tr>
<td>3.4.1</td>
<td>UNDER SEGMENTATION</td>
<td>57</td>
</tr>
<tr>
<td>3.4.2</td>
<td>OVER SEGMENTATION</td>
<td>57</td>
</tr>
<tr>
<td>3.4.3</td>
<td>PROPOSED METHOD</td>
<td>58</td>
</tr>
<tr>
<td>3.4.4</td>
<td>AMBIGUOUS CASES IN MANUAL TYPESET DOCUMENTS</td>
<td>64</td>
</tr>
<tr>
<td>3.5</td>
<td>SEGMENTATION OF TOUCHING CHARACTERS</td>
<td>68</td>
</tr>
<tr>
<td>3.5.1</td>
<td>CLASSIFICATION</td>
<td>68</td>
</tr>
<tr>
<td>3.5.2</td>
<td>CONSONANT CONJUNCTS: TYPE-I</td>
<td>70</td>
</tr>
<tr>
<td>3.5.2.1</td>
<td>OVERLAPPING BOUNDING BOX APPROACH</td>
<td>71</td>
</tr>
<tr>
<td>3.5.2.2</td>
<td>LOCATING BOUNDING BOXES</td>
<td>71</td>
</tr>
<tr>
<td>3.5.2.3</td>
<td>FORWARD SEARCH: (METHOD-I)</td>
<td>73</td>
</tr>
<tr>
<td>3.5.2.4</td>
<td>AVERAGE STROKE WIDTH</td>
<td>75</td>
</tr>
<tr>
<td>3.5.2.5</td>
<td>BACKWARD SEARCH: CONCAVITY AT TOUCHING POINT (METHOD-II)</td>
<td>75</td>
</tr>
<tr>
<td>3.5.3</td>
<td>TOUCHING CONSONANT CONJUNCTS: TYPE-II</td>
<td>77</td>
</tr>
<tr>
<td>3.5.3.1</td>
<td>MINIMUM AREA OF THE BOUNDING BOXES</td>
<td>77</td>
</tr>
<tr>
<td>3.5.3.2</td>
<td>PEAK VALUES IN SIDE PROFILES</td>
<td>82</td>
</tr>
<tr>
<td>3.5.4</td>
<td>TOUCHING CHARACTERS IN TOP ZONE</td>
<td>84</td>
</tr>
<tr>
<td>3.5.5</td>
<td>TOUCHING CHARACTERS IN MIDDLE ZONE</td>
<td>87</td>
</tr>
</tbody>
</table>
3. THEORETICAL ANALYSIS

3.1 AIM AND OBJECTIVES

3.1.1 BACKGROUND

Imperfections and errors which creep in at each stage add to the OCR misrecognitions. Noise in the image, skew of the document, imperfect binarization, imperfect machine print and scanning resolution lower the recognition rates. The main causes of the touching are nature of the script, improper binarization, small fonts, low scanning resolution, ink smear, poor paper quality and poor print quality.

Threshold plays an important role in the binarization of a document. The optimal threshold aims at maximizing the distance between the foreground and background pixels. At higher threshold values, the characters touching with the nearby characters resulting in touching characters; At lower threshold values, the thin parts of the character strokes may break resulting in broken characters. Sample binarized images of a paragraph are shown in Fig 3.1, one with Otsu method and the other with threshold more than Otsu method. The first part has three touching characters and the latter has eight touching characters. That is improper selection of binarization threshold often results in touching characters and broken characters.

Without saying, the segmentation is one of the important stages. The flaws in the image acquisition stage and preprocessing stage have a large bearing on the segmentation.
Fig. 3.1: Binarized images. Top) with Otsu threshold. Bottom) with greater than Otsu threshold.
The recently launched Digital Library of India (DLI) is an abundant source of Telugu binarized books. Most of the available documents are already scanned and binarized images. These books with smaller fonts, improperly binarized images of old and historic books, books with print imperfections, manually typeset old books pose challenges in handling broken characters, segmentation of lines and touching characters. It is found in the experiments conducted that 1%-2% of the characters are touching characters.

It is thus, imperative to properly segment the touching characters for higher accuracy. The present available Telugu OCR System DRISHTI developed at DCIS, University of Hyderabad works under Linux operating system. It is reported that it can recognize from 12 to 20 pt font characters with success rates of 84-87%. This is yet to address the problem of touching characters. Su Liang stated “It is not difficult to design a character recognition system that recognizes well-formed and well-spaced printed characters” [60]. George Nagy opined that “touching characters are responsible for the majority of errors in the automatic reading of both machine printed and hand printed text” [78].

3.1.2 Aim and Objectives of the Research

The aim of the research is to improve the accuracy of Telugu OCR System.

The objectives of the proposed study are outlined as follows:

1. To identify the problem areas of the Telugu OCR System.
2. To develop strategies and algorithms for improving the accuracy of the OCR system.

3.1.3 Problem statement

It is found that most of the errors in OCR Systems are caused by the touching characters. Solving the segmentation problem goes a long way to improve OCR recognition rates at character level and word level. The final editable output will be error free.

- We propose to develop algorithms suitable for segmentation of text lines of Telugu script, to improve the correct rendering of characters and words while displaying the output.
- The touching characters lead to erroneous words and incomplete sentences in addition to erroneous character recognition. We propose to study the touching characters based on their structural features and script grammar rules and propose algorithms to segment them for improving the overall accuracy of any Telugu OCR System.
- Addition of character variants to the image database to improve the accuracy of the OCR.

3.1.3.1 Baseline OCR

We have collected several Telugu documents to study the segmentation of lines, syllable model, touching characters and frequency studies of touching characters. A baseline OCR to evaluate the performance of strategies and algorithms is developed.
3.1.4 Scope of Study

In the proposed study we analyzed the touching of two characters only which constitute about 86.31% of touching characters. However multiple touching characters can be segmented by suitably extending the proposed algorithms. Broken character recognition is not attempted in this research.

3.1.5 Assumptions

1. Documents are scanned at 300 dpi and 600 dpi resolutions and binarized using Otsu method.

2. Some documents are downloaded from DLI (Digital Library of India) which are already binarized.

3. Single column fine printed text documents are considered for the analysis.

4. Noise free documents are considered.

3.2 Characteristics of Telugu Script

India is a multi lingual nation. Several languages are spoken in India and about 150 of them are documented. The Constitution of India specifies 22 commonly used languages in the Eighth schedule. Each language has its own script. Many others languages are spoken by the people in their daily life. Several languages have a common script and some spoken languages have no script at all. Most of the scripts in India are derived from the Brahmi script. Dravidian, Indo-Aryan, Astro-Asiatic and Tibeto-Burman are the major language families. Telugu, Tamil, Kannada and Malayalam belong to the Dravidian family of languages. Telugu is predominantly spoken in
South Indian states of Andhra Pradesh, Telangana, and in several parts of Karnataka, Tamil Nadu and Chattisgarh by a population of about 110 millions.

An organized sequence of sounds is a syllable. The spoken unit of the language known as *Akshara* has direct correspondence with the written syllable. As there is association between the spoken language and the written syllables no spelling rules are required for the script.

The script has 18 vowels, 38 consonants (Fig 3.2). Out of them two each in vowels (lu and luu) and consonants (tsa and dza) are obsolete and are not in use. A ‘halant’ is also part of the character set but do not exist independently.

![Fig. 3.2(a) Vowels.](image)

![Fig. 3.2(b) Consonants.](image)
The base consonants which end with \textit{halant} are combined with the first vowel (/a/) to form the consonants. The base consonants have no vowel sound. They are seldom used in the modern language, but find place only at few word endings and when words of other languages are written in Telugu as they are spoken.

\begin{center}
\begin{tabular}{|l|l|l|l|}
\hline
\text{త + య} & \text{త + యయ} & \text{త + యయయ} \\
\text{త + వ} & \text{త + వయ} & \text{త + వయయ} \\
\text{త + వ్ల} & \text{త + వ్లయ} & \text{త + వ్లయయ} \\
\text{చ + వ} & \text{చ + వయ} & \text{చ + వయయ} \\
\hline
\end{tabular}
\end{center}

Fig. 3.3 Formation of consonants, vowel modified consonants and conjunct consonants.

The consonants and vowels are combined to form consonant-vowel forms (Fig 3.3). A base consonant followed by vowel sound is a vowel modified consonant. They have distinct shapes. The inclined top part of consonants known as \textit{talakattu} in most of the cases is replaced with the vowel modifiers (\textit{gunintalu}). The vowel modifiers or \textit{matras} are attached at different locations i.e. at right or on top or written separately at bottom or on the right forming separate shapes (Fig 3.4). Each vowel has the corresponding vowel modifier except the first vowel /a/ which is combined to form the consonants. The \textit{matras} have no independent existence.

\begin{center}
\begin{tabular}{|l|l|}
\hline
చ + వ + యయ + యయయ + యయయ + యయయ + యయయ + యయయ + యయయ \\
\hline
\end{tabular}
\end{center}

Fig. 3.4 Vowel \textit{matras} combined with consonants (\textit{Guninthalu}).
In addition the language has secondary form of consonants or conjuncts. They are also known as dependant consonants (*ottulu*). They are written separately. Some conjuncts have similar shape as the primary consonant (Fig 3.5 (b)) and some have a completely different shape (Fig 3.5 (a)).

![Fig. 3.5 Conjunct formatives.](image)

A consonant cluster or a combined character is formed by a base character followed by secondary form of consonants. Even though the number of secondary form of consonants is not specified in the script rules, in practice it is common to find only one or two. Hill [69] has given an instance which has five secondary form of consonants (*karthsanyam*) (Fig 3.6).

![Fig. 3.6 Syllable with many conjuncts.](image)

The base character is either a consonant or vowel modified consonant. The base character is the dead consonant i.e. it has lost the vowel part of the consonant and vowel part is attached to the last
secondary form of consonant. This is in contrast to Devanagari script where the base character is written in half form.

The complete character set then consists of vowels, dual characters, consonants, vowel modified consonants, secondary form of consonants. The theoretical combinations run into thousands; in practice few hundred are used.

The word may begin with a vowel; however a vowel cannot appear elsewhere. Both the consonants and the vowel modified consonants can also appear at the beginning of a word. The dual character *sunna* is always attached to the preceding characters and cannot exist independently. The dual characters *arasunna* and *visarga* are not in common use, they can be seen only in poetic compositions or *sanskritized* words. No separate capital letters exist for this script.

### 3.2.1 Structural Characteristics

The study of structural characteristics helps in the identification and extraction of feature set.

The script is written on a base line from left to right linearly one syllable after the other. It is zonal based as all other Indian scripts. It has three zones viz., top zone, middle zone and bottom zone (Fig 3.7). The shape of a character may spread into top zone or bottom zone or lie in middle zone or combinations of the above. A syllable can be a consonant, vowel, vowel modified consonant. A dual character succeeding the syllable is part of the syllable.
A syllable may have non-linear padding. The secondary forms of consonants forming a syllable are written in bottom and middle zones or in bottom zones. A form of *ra otthu* is written on the left side of the syllable.

Telugu script has no head line or *Sirorekha* like Devanagari. The orthography of Telugu script is circular and consists of curved segments concatenated non-linearly. Small circles are also part of some characters. The vowel *matras* corresponding to ’aad’ and ’au’ have horizontal strokes one in middle zone and the other in top zone. Several consonants have strokes resembling tick marks inclined at nearly 45° in top zone; no vertical stroke exists in the script.

Few characters have two components. One vowel and six consonants have two components. Two consonants have a dot in the middle of the character. Small drop like strokes are added to the base of the characters for some consonants. These strokes though they are attached to the character, appear as separate components in some fonts. The *visarga* a dual character has two smaller circular shapes (Fig 3.2).

The vowel *matras* are fused after removing some part of the consonants and some vowel *matras* are fused to the right of the
character. Thus they have different shapes while much of the consonant shape is retained.

About half of the secondary forms of consonants are written in the bottom zone and others in bottom and middle zone. The later forms overlap the first or base character in the syllable formation. The shapes of some of them are completely different and 6 of them have similar shapes and written below the base consonant with small size (Fig 3.8) [69].

![Fig. 3.8 Conjunct formatives.](image)

(a) Different from primary form. (b) Similar to primary form.

3.2.2 General Observations on Touching Characters

Touching characters occur due to improper binarization, smear and low grade paper. Two or more glyphs touch each other in the touching characters. The Telugu script has large character set. Theoretically large number of combinations is possible for the touching characters. The touching pairs are not unique and hence very large data set is needed for training. Recognition of touching characters as a single pattern is prohibitive. Correct segmentation reduces number of templates for matching.

Statistical analysis showed that about 85% of touching characters have two components only, however multiple touching characters are also observed. Degraded documents and documents
binarized with greater threshold than the optimal values may have more multiple touching characters. Words having more than two touching characters are very rare.

Characters touch with the neighboring characters at different zones, viz. top, middle and bottom zones. Analysis of large number of touching characters lead us to propose a syllable model to facilitate the understanding of touching and segmenting them correctly.

3.3 SYLLABLE MODEL

The zonal spread of the Telugu characters is better shown in the character model proposed (Fig 3.9). The characters may occupy middle zone, middle and top zone, middle and bottom zone and all the three zones i.e. middle, top and bottom zones.

Fig. 3.9 Proposed model of Telugu Syllable.
The syllable formation is given by $C(C(C(V)))$ where $V$ represents a vowel and $C$ represents a consonant, and the * indicates that the number of consonants can be zero or more. The first $C$ is written in full form with modification of vowel *matra* applied to it (inner most $C$), and then the consonants that follow are written with the secondary form of consonants or dependent consonants. Characters in a syllable are non-linearly padded i.e. they are not only written one after the other from left to right, but are also padded from top to bottom. A syllable has a core character, followed on the right by conjuncts. Some conjuncts are located in the bottom zone, some in the middle and bottom zones. These are shown in dotted line. One conjunct which seems to be an exception (a variant form of *ra otthu*) is written to the left of the characters.

The conjuncts in the bottom zone do not overlap with the base character, however the bounding box of conjuncts in the middle and bottom zones written on right of the characters overlap with the base or core character. They are very closely kerned and orthography of the characters allows touching to a large extent with conjuncts which occupy the middle and bottom zones (Fig 3.10). The figure shows the locations of the possible touching. Touching also happens at the junction of the base character and top of the conjunct in the bottom zone. If the base characters do not have conjuncts, there is possibility of touching in the middle and top zones with the next base character of the syllable. The touching locations are shown in the model.
We need to devise algorithms for segmentation of the touching characters, as per the touching locations presented in the model once it is known that the character image is a touching character. Few touching characters are shown in Fig 3.11 with segmentation locations.

3.4 LINE SEGMENTATION

Text line segmentation is one of the important steps in OCR process. The skew removal is an initial step before line segmentation and is removed using maximum variance of the profile.
Most of the scripts are written on horizontal base line. Some the characters extend beyond the base line into upper areas and bottom areas. The text line segmentation based on summation of pixels along the lines often fails in languages having zonal disposition in script writing.

3.4.1 **UNDER SEGMENTATION**

Each character or glyph is extracted from the binarized image by segmentation of lines followed by words, then characters and finally by connected component analysis. In the projection profile or pixel projection method, the foreground pixels along a scan line are added and profile of pixel count is plotted for all lines in the image. It is commonly used for segmenting the lines from the image document.

\[
hpp_i = \sum_{j=1}^{N} I_{b}(i,j) \quad 3.4.1
\]

The lines are segmented in between the interline space indicated by a valley with zero value. The projection of conjuncts in a line and vowel modifiers of the line below may sometimes leave no valley in the horizontal profile. In this case, the segmentation through the valley points yields fewer text lines. This is an under segmentation problem (See Fig 3.13). Improper binarization, manual typesetting, small skew which could not be removed at pre-processing stage also cause faulty segmentation of text lines.

3.4.2 **OVER SEGMENTATION**

All the Indian scripts are zonal based. The Telugu character set has large set of characters with ascenders and descenders. Some of
the conjuncts in Telugu are smaller characters written in the bottom zone with a space in between the base character and the conjunct. These characters when segmented may appear as a separate line with incorrect results. This is an over segmentation problem (See Fig 3.13).

3.4.3 PROPOSED METHOD

A method to properly segment the lines is described below. It is observed that in many documents the conjuncts and vowel modifiers of two consecutive lines are not one over the above in most of the occurrences leaving a clear space between the two lines. This observation is exploited and a segmentation path in the inter line white space between the lines is traced to correctly segment the text lines.

**Average height of the line:** Segmenting large number of image documents, it is observed that the line height measured as distance between the midpoints of two consecutive interline white spaces is about 40\(^{th}\) percentile of all the block heights obtained in segmentation. This is called the average height of the line. The segmentation is correct when all the segmented blocks have a height less than 1.8 times the average height of the line. The factor 1.8 is proposed based on the statistical observations and also to accommodate the usual larger line space before and after the paragraphs; otherwise the image is segmented using the proposed algorithm.

**Segmentation path:** The segmentation of lines in effect is dividing the interline space between the lines. The dividing path can be half way
between two consecutive lines. It is observed that the path starting slightly downwards i.e., at 70% of interline space from the bottom of top line gives better results.

In this case the path will move around conjuncts of top line even if they are overlapping into the top zone of bottom line. Location of next point along the segmentation path is estimated to be along the same line one pixel ahead. After calculating distance between characters in top and bottom lines, the path is initially set to 70% of the white space from top line (Fig 3.14).
Fig. 3.13 Page after segmentation with HPP.
Initial Estimation: \( s_k(i) = s_k(i-1) \) 3.4.2

Corrected Final Value: \( s_k(i) = p_1 + 0.7(p_2 - p_1) \) 3.4.3

\( T_k \leq s_k(i) \leq B_k \) 3.4.4

Increment \( i \), go to initial estimation step

Where \( s_k \) = segmentation path of \( k^{th} \) non-zero valley

\( i = i^{th} \) column in the image

\( T_k \) = Top bounding line of \( k^{th} \) non-zero valley

\( B_k \) = Bottom bounding line of \( k^{th} \) non-zero valley

\( p_1 \) = first encountered pixel in upward direction or \( T_k \)

\( p_2 \) = first encountered pixel in downward direction or \( B_k \)
The distance to the characters at top and bottom cannot be found if we come across a word space in top or bottom line. So the bounding lines of interline space (top and bottom lines of consecutive lines) may be used to find the distance between lines (Fig 3.15).

**Bounding lines of line space:** The horizontal projection profile has fewer valleys (of zero value) than the number of line spaces (See Fig 3.12). The top and bottom lines of interline space intersects the pixel profile at steep slopes of the profile. A line of suitable threshold value is passed through the profile and the intersecting points are projected back on to the image to obtain bounding lines of the white line space. A large threshold increases the white space and smaller value results in bounding lines between the conjuncts and vowel modifiers. A value 15% of the peak value of the profile is found to give approximate bounding lines. More than usual number of conjuncts in a line gives wrong line space. The bounding lines which produce smaller width of
space are filtered out after clustering the line spaces into two groups and leaving smaller value.

In the proposed algorithm, only the approximate location of the bounding is required. The horizontal projection profile has fewer valleys (of zero value) than the number of line spaces (See Fig 3.12). The points on the steep slope of the pixel projection represent the top and bottom lines of white line space. The maximum peak value in the profile is found. A line at a fraction of the maximum peak value (threshold value) is passed through the profile. The intersecting points on the stem of the profile projected back on to the image represent bounding lines of the line space. Three lines of different threshold values passing through the profile are shown in Fig 3.16. A large threshold increases the line space and smaller value results in bounding lines beyond the conjuncts and characters. A value 15% of the peak value of the profile is found to give approximate bounding lines.

The threshold value is subtracted from the 1-D HPP array leaving only absolute values. The consecutive values in the 1-D HPP array represent the smeared lines. The consecutive adjacent pixels are defined as Run-lengths. The Run-lengths are extracted and smaller Run-lengths indicate the conjuncts (smaller peaks near the three horizontal lines in Fig 3.16) in document image. They are filtered out after clustering the run-lengths into two groups and discarding the smaller value run-lengths in 1-D HPP array. The location of the
segmentation path computed from the algorithm is shown in Fig 3.18.

The algorithm is discussed in section 4.11.

3.4.4 AMBIGUOUS CASES IN MANUAL TYPESET DOCUMENTS

Conjuncts in the segmentation path: In this case the lowering of segmentation line to 0.7 times the line space instead of mid way between the lines traces the path correctly and moves around the conjunct (Fig 3.17).

Sub-image profile of touching conjuncts and vowel modifiers: The conjuncts of top line and vowel modifiers of bottom lines may touch in very few instances. The segmentation line reaches a black pixel and cannot move further (Fig 3.19). A sub image with height equal to the bounding line space and width equal to the half the height is analyzed for possible paths. Horizontal profile of the sub-image indicates a clear path. Clear opening in the sub-image profile points to the correct traversal path for the segmentation (Fig 3.20).
The algorithm proposed is suitable for scripts which are zonal based and the components extending into the top and bottom zones. The algorithm applied for Kannada and Devanagari scripts are shown in Fig 3.21 and Fig 3.22.
Fig. 3.21 Segmentation of Kannada script with proposed algorithm.
Fig. 3.22 Segmentation of Devanagari script with proposed algorithm.
3.5 SEGMENTATION OF TOUCHING CHARACTERS

3.5.1 CLASSIFICATION

The Study of touching characters points out that a single strategy is not adequate to segment all of them. Based on the location of touching, we propose to classify the touching characters into five groups: touching conjuncts Type-I, touching conjuncts Type-II, top zone touching characters, middle zone touching characters and multiple touching characters. The middle zone touching has three sub classes and top zone touching have two sub classes depending on the variation of structural properties (Fig 3.23).

Fig. 3.23 (a) Classification of touching characters.
Conjuncts written in bottom and middle zones touch with the base characters, in Type-I. In Type-II, conjuncts in bottom zone touch with base characters. In middle zone touching the base characters of the syllable touch with the next base character or conjunct written on left. The strokes or circular shaped connected components of two vowel modified consonants touch in top zone. In multiple touching all the above touching may occur but mostly conjuncts are observed to be touching with each other.

Touching characters in randomly selected documents revealed that about 60% of the touching characters are consonant conjuncts, touching characters in top zone contribute 10%, touching characters in middle zone contribute 15% and multiple touching contribute to the remaining 15%. Each specific font style is found to have a single type of touching characters. This is due to the specific enhanced visual style in that font variation.

Script orthography and structural features differentiate each touching class from other classes. Segmentation of the touching characters proposed in the following sections is dependent on the characteristics of side profiles and pixel projections.
3.5.2 Consonant Conjuncts: Type-I

The segmentation of line into words by spaces in vertical pixel profile of the line image is a straightforward exercise. The character segmentation with vertical pixel profile is not always successful owing to touching characters and shadowing of white space in the syllable by the overlapping of base character with the conjuncts in the vertical direction. The connected component analysis also fails in the above situation. Segmenting the overlapping bounding box of the touching characters with algorithm identifying the location of touching is presented below.

The characters separated with clear vertical white space are shown with bounded boxes and overlapping bounding boxes in Fig 3.24. The glyphs in the overlapping bounding boxes can be segmented with the connected component analysis. Four touching characters are present in the two line image. The Type-I touching characters can be
identified in three boxes with the touching between the base character and the conjunct in middle zone. Few examples of Type-I touching is shown in Figure 3.25.

3.5.2.1 OVERLAPPING BOUNDING BOX APPROACH

The motivation for this approach is the observation that the base character and the conjunct formative can be bounded with the overlapping boxes.

![Fig. 3.26 Overlapping bounding boxes.](image)

Pixel profiles taken for the entire width and height cannot recognize the individual box boundaries, but projections taken from horizontal and vertical sections covering only one of the images can identify the separate box (Fig 3.26).

3.5.2.2 LOCATING BOUNDING BOXES

A partial pixel profile is count of pixels taken at a section within the image. A horizontal pixel projection of the left part of the consonant conjunct (Samyuktashara) taken at 'α' times can demarcate the base of the first character.
A statistical analysis showed that a value of 0.3 times the width of the image from the left edge is appropriate for the purpose at section 1-1 in Fig.3.27.

\[
\text{php}_1 = \sum_{j=1}^{\alpha w} I_b(i, j) \quad 3.5.1
\]

Where \( w \) = width of the combined image

\( I_b(i, j) \) = binary image, \( \alpha = 0.30 \)

The first character is isolated by its bottom and right side. The partial pixel projection taken at section 1-1 has no pixel count beyond...
the first character. The projection is searched from the bottom of the image till the zero height of pixels ceases. This location is identified as the bottom of the first bounding box.

The touching location is at the right side (Fig. 3.27) of the first box. This location needs to be identified for the isolation of the first box. A partial vertical pixel count of touching character is taken at section 2-2.

\[
pvp_t = \sum_{i=1}^{bb1.h} I_b(i,j) \forall j
\]

3.5.2

Where \( bb1.h \) = height of first bounding box

\( I_b(i,j) \) = binary image

Two approaches are proposed to locate this section. After identifying the segmentation location, the first character is removed from the image. The location is padded with white pixels. The partial text pixel profile is taken at the line 2-2 downward to obtain the second character.

3.5.2.3 Forward search: (Method-I)

The partial vertical pixel projection taken at the base of the first character towards top (2-2) shows a valley at the point of touching due to drop in number of pixels. In many touching characters, it is observed that the width of stroke at the touching point is found to be about two times the stroke width. The valley locations are identified in the forward search. The width criterion is verified at all the valley locations.
The vowel modified consonants of vowel ‘aa’ have long horizontal strokes to the right (Fig 3.28). The binarization inaccuracies cause valleys in these strokes. The strokes are not uniform and have ‘neck regions’. These points cause false valley locations in pixel projections. An algorithm to eliminate these false locations with the width of touching location criterion is presented below.

Consider the image of consonant conjunct in Fig.3.28. The neck region or a false location is identified at point A and the correct location at point B. The forward search algorithm looks for all the valleys points in partial vertical pixel profile starting the search from the left. At each valley point the stroke width is assessed. The search for the edge of the black pixels on either side is started from mid-point of the stroke. The stroke width at ‘A’ is \( p \) on the left side and \( q \) on the right side. These two widths are large enough to be discarded. The next valley in the search is at point B. The widths of the stroke in horizontal direction is ‘\( r \)’ on the left and ‘\( s \)’ on the right side of the touching point. The criterion that is adopted is that one of the widths at the touching
point is about twice of the average stroke width. The point B is identified as correct location.

3.5.2.4 AVERAGE STROKE WIDTH

A simple but effective method that is based on morphological operations is proposed to estimate the average stroke width (Fig 3.29 a,b). The image is thinned to one pixel width with morphological thinning algorithm. The total pixel count gives the length of the character. The sum of pixels in the original image divided by the length of character assessed as above after rounding off gives the average stroke width.

\[ sw = \frac{\sum I_{org}}{\sum I_{thin}} \] 3.5.3

Where \( sw \) = stroke width

\( I_{org} \) = Original image of character

\( I_{thin} \) = Thinned image of character

3.5.2.5. BACKWARD SEARCH: CONCAVITY AT TOUCHING POINT (METHOD-II)

In this method the search is proceeded backwards from the right edge of the image along the line 2-2 (base of first character) in the image Fig 3.30.
Side profile is the summation of white pixels from the specified edge of an image or at a section, to the edge of character contour. In the bottom side profile taken at 2-2, concavity in the profile is observed at the touching location along 3-3. The method is described below.

The left edge of conjunct stroke is identified by moving from the right edge of the image towards left. In backward traversal from the right edge along 2-2, the transition of white pixels to black pixels identified the right edge of the conjunct stroke; and the transition of black pixels to white pixels identified the right edge. A side profile towards the top at 2-2 is computed. From the right edge of conjunct stroke, the side profile has a rise. After the touching point, the profile takes a concave shape. This location is identified by the concavity at the profile falling point. The touching conjuncts of Type-I are identified by the classifier in the recognition stage with the distance of the image pattern to the template pattern. Distances are found to be more than the pre-defined threshold value.

Fig. 3.30 Segmentation based on concavity at touching location.
3.5.3 Touching Consonants Conjuncts: Type-II

This type of touching characters constitute about 15% of the touching characters. Touching glyphs in three text lines are shown in Fig 3.32. The base consonant in a syllable touches with the conjunct formatives written in bottom zone. Script level characteristics and structural features of profile are used to locate segmentation point.

3.5.3.1 Minimum area of the bounding boxes

These conjuncts are written in a smaller size below the base component. This script level characteristic is used to locate the
segmentation line. Segmentation line is a horizontal line passing through the correct touching point. Initially a segmentation line is proposed dividing the touching character into top and bottom parts. Bounding boxes for both the parts are calculated as shown in Fig 3.33. As the segmentation line is moved down one pixel at a time, the bounding boxes keep varying in their sizes.

![Fig. 3.33 Bounding boxes for the top and bottom parts.](image)

Three properties of the image bounding boxes are considered for studying the variation of these properties on changing the location of the segmentation line. Total area $A$ of both the bounding boxes, total perimeter $P$ of both the bounding boxes and pixel density $D$ of the bounded image are the parameters considered. Pixel density is the number of pixels divided by the total bounded area.

$$A = A_1 + A_2$$  \hspace{1cm} 3.5.4

where $A_1$ is area of the top bounding box

$A_2$ is area of the bottom bounding box

$$P = P_1 + P_2$$  \hspace{1cm} 3.5.5

where $P_1$ is perimeter of the top bounding box

$P_2$ is perimeter of the bottom bounding box
\[ \text{total\_pixels} = \sum_{1}^{N \times M} I_{\text{inv}} \]  \hspace{2cm} 3.5.6

\[ D = \frac{\text{total\_pixels}}{(A_1 + A_2)} \]  \hspace{2cm} 3.5.7

where \( I_{\text{inv}} \) is the inverted image

Telugu characters have circular orthography and are wider at middle of the character. Initially the segmentation line is started at middle of the character height. As the segmentation line is moved down, the bounding box of the lower part gets diminished (Fig.3.33(b)). The total area and the total perimeter keep on taking the lower values. On reaching the correct segmentation location, the bounding boxes both fit snugly around both the characters. At this location both the parameters reach the lowest value.

Further moving down the segmentation line, increases the area parameter. The area of top box increases across the entire width, but the bottom box decreases across the width of the smaller conjunct. So the total area starts increase with the lowering of line.

On moving down the segmentation line, the increase in total perimeter is not substantial because lowering the segmentation line by one pixel increases the top box perimeter by two pixels and decreases the bottom box perimeter by two pixels.

Pixel density is the total number of character pixels multiplied by the inverse of total bounded area, the pixel density parameter increases to a maximum value on reaching the correct location and then decreases afterwards.
Of the three parameters, area and perimeter decrease rapidly and increase after attaining lowest value. The increase is rapid in case of the area and slow in case of the perimeter. We can use area or pixel density parameters for locating the segmentation line. The area and perimeter will start decreasing after reaching the wider part of the conjunct due to the shape of the conjunct. It is observed in several touching characters that the location is within the 0.8*H where H is the height of the touching image. So we initialize the segmentation location with 0.5*H and increment up to the 0.8*H. Figs 3.34, 3.35 and 3.36 show the variation of the area A, perimeter P and pixel density D parameters against the height of segmentation line from the initial value.

Fig. 3.34 Variation of the total area of the bounding boxes.
Fig. 3.35 Variation of total perimeter.

Fig. 3.36 Variation of density of pixels.
Table 3.1 Parameters of bounding box – highest and lowest values

<table>
<thead>
<tr>
<th>Type-II</th>
<th>Max. Area - pixels</th>
<th>Min. Area - pixels</th>
<th>Max. Perimeter - pixels</th>
<th>Min. Perimeter - pixels</th>
<th>Max. density - ratio</th>
<th>Min. density - ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>ཁི</td>
<td>11269</td>
<td>10430</td>
<td>618</td>
<td>574</td>
<td>0.4001</td>
<td>0.3703</td>
</tr>
<tr>
<td>ཆི</td>
<td>4689</td>
<td>4524</td>
<td>404</td>
<td>388</td>
<td>0.3393</td>
<td>0.3274</td>
</tr>
<tr>
<td>ཌྷི</td>
<td>4728</td>
<td>4032</td>
<td>416</td>
<td>356</td>
<td>0.3442</td>
<td>0.2936</td>
</tr>
<tr>
<td>ཤི</td>
<td>7260</td>
<td>6406</td>
<td>506</td>
<td>444</td>
<td>0.3893</td>
<td>0.3435</td>
</tr>
<tr>
<td>ཉི</td>
<td>11092</td>
<td>10678</td>
<td>592</td>
<td>584</td>
<td>0.4074</td>
<td>0.3922</td>
</tr>
<tr>
<td>ཥི</td>
<td>7308</td>
<td>6519</td>
<td>516</td>
<td>472</td>
<td>0.4212</td>
<td>0.3758</td>
</tr>
<tr>
<td>ཟི</td>
<td>5640</td>
<td>5216</td>
<td>442</td>
<td>414</td>
<td>0.4511</td>
<td>0.4172</td>
</tr>
<tr>
<td>སྨ</td>
<td>4292</td>
<td>3918</td>
<td>412</td>
<td>368</td>
<td>0.3606</td>
<td>0.3292</td>
</tr>
<tr>
<td>ཟྲ</td>
<td>6309</td>
<td>5488</td>
<td>446</td>
<td>424</td>
<td>0.5308</td>
<td>0.4617</td>
</tr>
<tr>
<td>དྲ</td>
<td>7728</td>
<td>6652</td>
<td>536</td>
<td>468</td>
<td>0.4039</td>
<td>0.3477</td>
</tr>
</tbody>
</table>

The touching characters, the maximum and minimum values of the parameters are shown in Table 3.1. We used area parameter in our algorithms. The algorithm fails if the conjunct consonants are of same width as that of the base character. This is a very rare case. In cases where the conjuncts are larger than the base character the algorithm also detects the correct location.

3.5.3.2 Peak values in side profiles

The binarization inaccuracies and fusing with blobs at touching location may prevent correct identification. The side profiles of the character image are used to locate the correct location. It is observed
that the meeting point of the base character and the conjunct has
sharp peak values due the circular shapes of the both characters (Fig
3.37, 3.38). The side profile is the summation of the white pixels from
the edge of the image to the edge of the character image.

\[ SP_L = \sum_{j=1}^{k-1} l_{i,j} \forall i \] 3.5.7

Where \( SP_L \) is side profile on left of image

\[ k = \text{the column index of edge of image} \]

The left and right side profiles are examined for a sharp peak value.
The search starts at few pixels above the identified location and ends
after a few pixels. The segmentation line passes through the peak
value positions of the left and right side profiles (Fig 3.37, Fig 3.38). If
they differ, the segmentation line is passed through the peak value
locations into the touching zone up to middle of the touching width,
and segmented as shown in Fig 3.39 and Fig 3.40.

---

Fig. 3.37 Bounding boxes.  Fig. 3.38 Peaks in side profiles.
3.5.4 **Touching Characters in Top Zone**

Among the vowel modifiers 6 extend from left to right including halant, 2 extend from right to left (Fig. 3.41(a)). These modifiers may touch at their end locations in the top.

![Fig. 3.39 Before segmentation.](image1) ![Fig. 3.40 After segmentation.](image2)

![Fig. 3.41(a) Vowel modifiers. (b) Vowel modifiers attached to consonant 'ka'. First and second line – attached at top zone, third line – attached in middle zone.](image3)

![Fig. 3.42 Two components in a single character.](image4)
In addition few characters have two components in a single syllable (Fig. 3.42) \( \text{gha, pa, pha, sha, sa, ha} \). The vowel modifiers when attached to these consonants form two separate components which are prone to touching with the preceding modifiers that extend from left to right. This results in two forms of touching characters with the following characteristics. Some of the touching characters are shown in Fig. 3.43.

In the first form of top zone touching, the middle zone part has clear separation with white spaces. This gives more pixel count in the bottom side profile at the central portion. In the second form the separate components of the single character (Fig. 3.44) may touch with the preceding modifier in top zone (Fig. 3.45 (c&d)). As the top component of the character is touching with the first character, we can observe a white space below the touching top component (Fig. 3.45). Due to this the white pixel count on right side profile is more from the base line up to the top of middle zone (Fig. 4.45(c),(d)). This is used as the distinguishing feature for the two forms of touching characters in the top zone.
Fig. 3.44 Separated top components of a single character.

Fig. 3.45 Pixel counts in bottom and side profiles for Form-I and Form-II top touching characters.

In the first form horizontal stroke of the second character is touching first character. Hence in the bottom side profile the high pixel count location is identified from the right side. After continuing
column wise search towards the left side, the end of constant height (flat portion of the stroke) or end of high pixel count is located. This is the correct segmentation location (Fig. 3.45(a),(b)). The traversal from the right side avoids the wrong segmentation of small holes, if present at the end of the stroke in first character. In the second form it is observed that the stroke component of the second character has concave shape or a stepped rise. Hence the bottom side profile is scanned from right side for the concavity in the white pixel count. The concavity shows a gradual lowering of the pixel count, then a possible constant count followed by a gradual higher pixel count (Fig. 3.45(c)). The stepped rise is a rise in height of profile more than a threshold (Fig. 3.45(d)). While scanning from right side, the first character ends at the end of the concavity. This is identified as correct location for segmentation.

3.5.5 Touching Characters in Middle Zone

All the characters in Telugu have circular shapes. The larger holes or circular shapes of a character may be full circle or part of a circle and occupy middle zone (‘sunna’, ‘ya’, ‘ra’, etc.). The characters may have small holes or part of a circle at the top or bottom portions of middle zone as vowel modifiers or as part of the character itself (‘ma’, ‘ba’ etc.).

Fig. 3.46 possible touching locations.
The circular segment of the next character then may touch at the middle, top or bottom areas of the middle zone. The possibilities of touching due to the structural shapes are shown in Fig. 3.46. Fig. 3.47 shows some of the middle zone touching characters.

![Fig. 3.47 Touching in middle zone.](image)

In the middle zone, the characters touch tangentially with the next character due to the circular nature of characters. The top and bottom profiles extend up the point of contact of the two characters (Fig. 3.48). The difference between the two profiles has low value at point of contact.

![Fig. 3.48 (a) Bottom and top side profiles, plot of difference of the profiles.](image)
Fig. 3.48(b) Bottom, top side profiles, plot of difference of the profiles.

\[ hce_i = \left( h - \sum_{r=1}^{k_1-1} i_{r,c} - \sum_{r=k_2+1}^{M} i_{r,c} \right) \forall c \]

3.5.8

Where \( hce_i \) = height of character edge

\( k_1, k_2 = \) pixel locations of upper and lower edge respectively

Some other locations also show low difference value. The distinguishing feature of the point of contact is that both bottom and top side profiles have peaks where as others do not have peaks. In some cases the peaks are not vertically above each other. They are separated by a small value ‘\( \Delta \)’. Therefore at the point of low difference value, we need to search for peaks in a small neighborhood ‘\( \Delta \)’ on left and right sides of the point. The segmentation line then, is the line
joining the peaks of top and bottom profiles which is slightly inclined to the vertical.