International Journal of Engineering Research and Development
e-ISSN: 2278-067X, **p**-ISSN: 2278-800X, www.ijerd.com
Volume 6, Issue 11 (April 2013), PP. 92-110

Font and Size Identification in Telugu Printed Document

K.Ram Mohan Rao[1], B.Ramesh[2], G.Indrasena Reddy[3]
Assistant Professor, Dept of ECE, Sri Indu College of Engineering and Technology, Hyderabad, AP, India

Abstract: Telugu is the official language derived from ancient Brahmi script and also one of the oldest and popular languages of India, spoken by more than 66 million people, especially in South India. While a large amount of work has been developed for font and size identification of English and other languages, relatively not much work has been reported on the development of OCR system for Telugu text. Font and size identification of Indian scripts is much more complicated than other scripts because of the use of huge number of combination of characters and modifiers. Font and size identification is the pre-processing step in OCR systems. Hence Telugu font and size identification is an important area if interest in the development of Optical Character Recognition (OCR) system for Telugu script. Pre processing tasks considered here are conversion of gray scale image to binary image, image clearing, and segmentation of the text into line, separation of connected components. Zonal analysis is applied and then top zone components are identified and checked for the presence of tick mark. Aspect ratio and pixel ratio for component are calculated. Comparing these two parameters with the database we identify the font and size. Simulation studies were carried out using MATLAB with GUI for all the segmentation methods for given Telugu text and the results it observed were good enough for identification of different fonts and sizes in the given Telugu text.

Index Terms: Optical Character Recognition (OCR), Telugu script, Segmentation, Zonal Analysis

I INTRODUCTION

1.1 Optical Character Recognition (OCR)

Character Recognition or Optical Character Recognition (OCR) is the process of converting scanned images of machine printed or handwritten text, numerals, letters and symbols into a computer processible format (such as ASCII). OCR is a technology that allows us to scan and convert printed, typewritten or even hand-written text on sheet of paper to a readable format, either in the form of a plain text file, a Word document or even an Excel spreadsheet. All this and more is done, in a fraction of the time that would normally take to manually type the document. OCR technology has today evolved to become both accurate and easy to use. The accuracy rate can be as high as 99.99% for a good quality printed text and even advanced artificial intelligence techniques help in recognizing scanned characters.[1]

1.2 Historical Perspective

Modern OCR technology is said to be born in 1951 with M. Sheppard's invention, GISMO-A Robot Reader Writer. In 1954, J. Rainbow developed a prototype machine that was able to read uppercase typewritten output at the fantastic speed of one character per minute. Several companies, including IBM, Recognition Equipment Inc., Farrington, Control Data and Optical Scanning Corporation, marketed OCR systems by 1967. Today, OCR systems are less expensive, faster and more reliable. It is not uncommon to find PC-based OCR system capable of recognizing several hundred characters per minute if different languages. More fonts can be recognized than ever with today's OCR systems advertise themselves as omni font-able to read any machine printed font. Increased productivity by reducing human intervention and the ability to efficiently store text are the two major selling points. Current research areas in OCR include handwriting recognition and form reading in different languages. Sufficient amount of work has been done on some languages e.g., English (Roman), Japanese (kanji & Kana), Chinese (Kanji).

1.3 Font Type and Size Recognition

1.3.1 Need for High Recognition Rate

For any OCR software to be really useful it must have at least 99% accuracy. The running text printed on a A4 size paper can easily contain an average of 2000 characters per page. That means OCR software with 99% recognition rate will produce 20 errors per page. In manual typewriting, this is the worst case error rate. A good typist will

commit an average of 4 errors per page. If we really want to replace a typist with OCR, it must have at least 99.9% accuracy. One way we can achieve this recognition rate is by using a Font Type and Size Recognition system as a part of OCR [2].

1.3.2 Existing OCR Technologies

Current OCR technologies are largely based on one of the following approach:

i) **Template Matching:-** It is the most trivial method. Character templates of all the characters from most commonly used fonts are collected and stored in a database. The recognition consists of finding the closest matching template using one of the minimum distance matching algorithms. Template matching technique assumes the prior knowledge of the font used in the document and is highly sensitive to noise, skew etc. in the scanned image. This method is not suitable for omni font OCR system, because character templates of all the variants of the characters in all the fonts must be stored in the database.

ii) **Structural Approach:-** In this approach, characters are modeled using their topological feature. The main concentration will be on structural features and their relationship. The most common methods under this category are:

a. String matching methods where characters are represented by feature string
b. Syntactic methods where character features are determined by the vocabulary and grammar of the given language
c. Graph based methods consists of graph construction where nodes contain features

All of the above methods are superior to template matching but with respect to omni font OCR we cannot achieve desirable recognition rate using this approach.

iii) **Approach: -** This approach is based on the statistical decision theory where each pattern is considered as a single entity and is represented by a finite dimensional vector of pattern features. The most commonly used methods in this category are based on Bayesian classification, stochastic and nearest neighbor classification. In the recent past, classification based on Neural Networks is also used significantly to enhance the recognition rate.

1.3.3 Font Type and Size Recognition approach

Font Type and Size recognition approach can be used to overcome the limits of existing omni font-type OCR technologies. As stated previously mono font OCR will give high recognition rate. If we were able to discriminate the text in various fonts in a document, then they can be submitted to the corresponding mono font OCR engine. This approach is based on the features extracted from global properties of the text image such as the text density, letter size etc. [2]. Global features can also be tolerant of the image condition, i.e. they can be extracted from binary image scanned at low resolution.

In recognition of machine printed characters on printed documents, a high accuracy can be obtained due to regularity within a limited range of fonts known to an OCR system. The detection of font type and size can improve the character segmentation because the identification of type and size provides information on the structure and typographical design of character.

1.4 Problem of the Present Work

The research work in Font type and size identification has mainly concentrated for Roman, Chinese, Arabic, Japanese and other major languages of the world. Very little work has been done for Indian languages / scripts, especially for telugu language, and not much work is done for telugu script. Therefore, our proposed problem is an attempt to solve the problem of font type and size identification of telugu language.

1.5 Project Definition and Limitations

The objective of this project is to develop a Font Type and Size Identification system for machine printed Telugu characters. This system will be based on the feature extraction of the various fonts. We assume that the document contains a single character and is of good quality, noise free and less distortion is there.

Although this technique works for all the fonts we considered the fonts *Anupama, Brahma, Pallavi and Priyanka* of sizes 14, 16 and 19, because these are the fonts and size that are most commonly used in all printing purposes and daily use.

1.6 Applications

One uses different fonts in a document in order to emphasize some parts of the text in such a way that the reader notices them easily. In a document, font changes may occur at particular points (titles, Indexes, references, etc. They may be done by choosing another typeface, changing the style of the size of the same typeface. In reality, we do few and systematic font changes in a single structured document. In fact Telugu Font Type and Size Recognition is useful in different domains [17]:

i) Recognition of logical document structures, where knowledge of the font used in a word, line or text block may be useful for defining its logical label (chapter title, section title or paragraph).
ii) Document reproduction, where knowledge of the font is necessary in order to reproduce (reprint) the document.
iii) Document indexing and information retrieval, where word indexes are generally printed in fonts different from those of the running text.
iv) Text font knowledge may also improve recognition rate of OCR (Optical Character Recognition) systems.
v) Document Database System is important application for converting printed documents into electronic form on which search and retrieval can be carried out.
vi) Automatic archiving of Telugu document

1.7 Organization of Paper

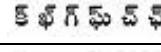
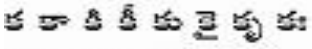
In this paper chapter 1 gives the introduction about OCR and the necessity of Font Type and Size identification. Chapter 2 gives a brief description and nature of the Telugu script. Chapter 3 explains about the Digital Image Processing, and describes the main section, segmentation. Chapter 4 Zonal Analysis and Connected component segmentation of the text image. Chapter 5 gives the entire algorithm and applications of the Telugu Font Type and Size Identification. Chapter 6 shows the results. Finally, Chapter 7 concludes and gives the future scope.

II TELUGU SCRIPT

2.1 Telugu Script

Telugu is one of the ancient languages of India that is more than 5000 years old. The basic alphabet of telugu consists of 16 vowels and 38 consonants totaling to 54 symbols. Each vowel in telugu is associated with a vowel mark. A consonant when combines with any of these 16 vowel marks results in a new combination called a compound character i.e, a total of 38*16 combinations. More complicated combinations are shown in Table2.1. The combinations are also typically referred to as compound characters and may be written using two or more disconnected symbols. A compound character in telugu follows the phonetic sequence. The phonetic sequence can be represented in grammatical form. When any character or compound character in telugu is represented in grammatical sequence, it belongs to one of the grammatical combinations shown in Table 2-1.

Figure 2-1: Vowels and Consonants
in Telugu Script

	Combination	**Examples**
1	Vowel	
2	Base Consonant	
3	Base Consonant + Vowel	
4	Base Consonant + Base Consonant + Vowel	
5	Base Consonant + Base Consonant+ Base Consonant + Vowel	

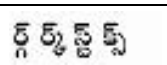
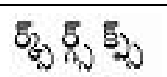
6	Base Consonant + Base Consonant	
7	Base Consonant + Base Consonant + Base Consonant	

Table 2-1: The various combinations forming compound characters

There are 18 vowels in telugu. Out of these some are out of usage in modern script. Therefore, in modern script only 15 vowels are employed. All of them are constructed as simple connected symbols except  and  . telugu language defines 38 consonants. Consonants are defined in telugu as combination of base consonant and the short vowel  /A/. Base consonants do not exist in the Telugu character set but are used when words like 'GOAL' are written i.e,  . Base consonant is a vowel suppressed consonant. Some of the consonants and their corresponding base consonants in are: 

The third combination i.e, of a base consonant and a vowel is an extremely important and often used combination in telugu script. Logically, 608 combinations of different characters can be generated with this combination. The combinations from the fourth to seventh are categorized under conjunct formation. Telugu has a special feature of providing a unique symbol of dependent form for each of the consonants. In all conjunct formations, the first consonant appears in its actual form. The dependent vowel sign and the second (third) consonant act as dependent consonants in the formation of the complete character.

The four combinations from the fourth to seventh generate a large number of conjunct in Telugu script. The fourth combination logically generates 23,104 different compound characters. This is an important combination of conjunct ligatures in the script. First consonant and the vowel combination is rendered first and the second consonant is rendered as the dependent consonant sign. The fifth combination is similar to the fourth combination. The second and the third consonants act as the dependent consonants. Logically 877,952 different compound characters are possible in this combination, but their frequency of appearance in the text is less when compared to the previous combinations. 1444 combinations and 54,872 combinations are logically possible in the sixth and seventh combinations respectively. The sixth and seventh combinations are used when words from other languages are written in telugu script. In these combinations, the vowel is omitted. The first consonant appears as a base consonant and the other consonants act as dependent consonants.

Sample	Vowels	Consonants	CV Core	Conjuncts	Others	Total
1	179	805	1723	738	344	3789
2	170	755	1615	607	376	3523
3	109	574	1312	471	326	2792
4	76	389	829	394	130	1818

Table 2-2: Analysis of different groups of characters in Telugu Language

Table 2-2 gives an analysis of the frequency of occurrence of different compound characters. The samples are from newspaper, a small part of the book on patent act in India, a children science book, and poems from a popular magazine. The occurrence of vowels, consonants, characters that represent CV core (consonant + dependent vowel sign), conjunct formations and other characters that include nasal sounds and base consonants are tabulated in Table 2-2.

The occurrence of these combinations in terms of percentages is as follows:

i) (4-5)% of the script contains vowels
ii) (21-22)% are consonants
iii) (45-46)% of the characters belong to the basic unit of the CV core
iv) Around (27-30)% of the characters belong to the other category of CCCV structure

2.2 Characteristics of Telugu Script

Telugu is a syllabic language. Similar to most languages of India, each symbol in telugu script represents a complete syllable. There is very little scope for confusion and spelling problems. In that sense, it is a WYSIWYC script. This form of script is considered to be the most scientific by the linguists [3].

The syllabic telugu script has been achieved by the use of a set of basic character symbols, a set of modifier symbols and rules for modification. Officially, there are eighteen vowels, thirty-six consonants, and three dual symbols. Of the vowels, sixteen are in common usage. These are combined in various fashions to generate numerous compound characters making the OCR problem a difficult one. Some of these compound characters are connected regions in the image whereas others are comprise of several smaller connected images.

2.3 Telugu Character set

Telugu script, as other Indian scripts, is generally written in non-cursive style, unlike English handwriting, which is normally written in cursive style rendering recognition difficult. However, Indian scripts pose a peculiar problem non-existent in European scripts – the problem of composite characters. Unlike in English alphabet where a single character represents a consonant or a vowel, in Indian scripts a composite character represents either a complete syllable, or the coda of one syllable and the onset of another. Therefore, although the basic units that form composite characters of a script are not that many (O(102)), these units by various combinations lead to a large number (O(104)) of composite characters. For example, the word*s tree*, a single syllable comprising three consonant sounds and a vowel, is represented by a single composite character in which graphical elements corresponding to the 3 consonants and the vowel are combined to form a complex structure. Moreover,, Telugu script is the most complex of all Indian scripts for two reasons: a) it has largest number of vowels and consonants and, b) it has complex composition rules. The telugu script consists of 14 vowels, 36 consonants, 3 special characters. As a general rule when the vowel appears at the beginning of the word, the full vowel character is displayed. When a vowel appears inside a word, a vowel modifier is added to the preceding consonant creating a composite character. The number of combinations possible here to form a composite character is very large. For example, all the above 36 consonants can combine with vowels (i.e. consonant-vowel type approximately 576 in number). Also the 36 consonant modifiers can combine to form composite characters of consonants-consonant-vowel (CCV) type (this number runs approximately to 20,736). Also a character can have more than one consonant modifier. Word processing with such a complex script using the QWERTY keyboard can be a harrowing experience, considering the cumbersome keyboard mappings the user has to learn. A pen-based interface in such a situation can prove to be a most convenient tool.

2.4 Telugu Writing System

The earliest evidence for Brahmi script in South India comes from Bhattiprolu in Guntur district of Andhra Pradesh [4]. A variant of Asokan Brahmi script, called Bhattiprolu Scrip, the progenitor of old telugu script, was found on the Buddha's relic casket. Telugu script is written from left to right and consists of sequences of simple and/or complex charaters. The script is syllabic in nature – the basic units of writing are syllables. Since the number of possible syllables is very large, syllables are composed of more basic units such as vowels and consonants. Consonants in consonant clusters take shapes which are very different from the shapes they take elsewhere. Consonants are presumed to be pure consonants, that is, without any vowel sound in them. However, it is traditional to write and read consonants with an implied 'a' vowel sound. When consonants combine with other vowel signs, the vowel part is indicated orthographically using signs known as vowel "maatras". The shapes of vowel "maatras" are also very different from the shapes of the corresponding vowels.

The overall pattern consists of sixty symbols, of which 16 are vowels, three vowel modifiers, and forty-one consonants. Spaces are used between words as word separators. The sentence ends with either a single bar | ("purna virama") or a double bar ‖ ("deergha virama"). Traditionally, in handwriting, telugu words were not separated by spaces. Modern punctuation (commas, semicolon etc.) were introduces with the advent of print.

2.6 Sample Page of Telugu Script Considered in This Project

The following script is one of the samples taken in this project.

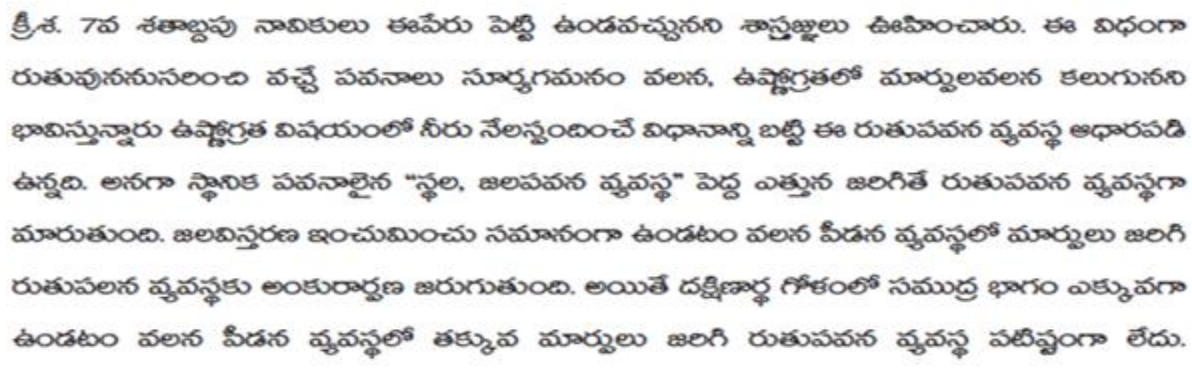


Figure 2-2: One of the sample texts used in this project

III IMAGE PROCESSING

3.1 Applications of image processing

Interest in digital image processing methods stems from 2 principal application areas:
(1) Improvement of pictorial information for human interpretation, and
(2) Processing of scene data for autonomous machine perception.
In the second application area, interest focuses on procedures for extracting from image information in a form suitable for computer processing. Examples include automatic character recognition, industrial machine vision for product assembly and inspection, military recognizance, automatic processing of fingerprints etc.

3.2 Introduction to Segmentation

Segmentation refers to the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain visual characteristics. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours extracted from the image. Each of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s) [7].

3.3 Various Segmentation Methods

Some of the segmentation methodologies are Clustering methods, Histogram-based methods, Edge detection methods, Graph partitioning methods, Neural networks segmentation etc. [3,8,9].

3.3.1 Clustering methods

The K-means algorithm is an iterative technique that is used to partition an image into *K* clusters. The basic algorithm is:

i) Pick *K* cluster centers, either randomly or based on some heuristic
ii) Assign each pixel in the image to the cluster that minimizes the variance between the pixel and the cluster center
iii) Re-compute the cluster centers by averaging all of the pixels in the cluster
iv) Repeat steps ii) and iii) until convergence is attained (e.g. no pixels change clusters)

This algorithm is guaranteed to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of *K*

3.3.2 Histogram-based methods

Histogram-based methods are very efficient when compared to other image segmentation methods because they typically require only one pass through the pixels. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are used to locate the clusters in the image. Color or intensity can be used as the measure. A refinement of this technique is to recursively apply the histogram-seeking method to clusters in the image in order to divide them into smaller clusters. This is repeated with smaller and smaller clusters until no more clusters are formed.

33.3 Edge detection methods

Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often discontinued. To segment an object from an image however, one needs closed region boundaries. Discontinuities are bridged if the distance between the two edges is within some predetermined threshold.

3.3.4 Graph partitioning methods

Graphs can effectively be used for image segmentation. Usually a pixel or a group of vertices and edges define the (dis)similarity among the neighborhood pixels. Some popular algorithms of this category are random walker, minimum mean cut, minimum spanning tee-based algorithm, normalized cut etc. the normalized cuts method was first proposed by Shi and Malik in 1997. In this method, the image being segmented is modeled as a weighted, undirected graph. Each pixel is a node in the graph, and an edge is formed between every pair of pixels. The weight of an edge is a measure of the similarity between the pixels. The image is partitioned into disjoint sets (segments) by removing the edges connecting the segments. The optimal partitioning of the graph is the one that minimizes the weights of the edges that were removed (the *cut*).

3.3.5 Neural networks segmentation

Neural networks segmentation relies on processing small areas of an image using an artificial neural network or a set of neural networks. After such processing the decision-making mechanism marks the areas of an image accordingly to the category recognized by the neural network. A type of network designed especially for this is the Kohen map.

In our project we used Histogram-based segmentation methodology. Histogram-based methods are very efficient when compared to other image segmentation methods because they typically require only one pass through the pixels. One disadvantage of the histogram-seeking method is that it may be difficult to identify significant peaks and valleys in the image. In this technique of image classification distance metric and integrated region matching are familiar.

3.4 Histrogram

In statistics, a histogram is a graphical display of tabulated frequencies, shown as bars. It shows what proportion of cases fall into each of several categories: it is a form of data binning. The categories are usually specified as non-overlapping intervals of some variable. The categories (bars) must be adjacent. The intervals (or bands, or bins) are generally of the same size, and are most easily interpreted if they are. Histograms are used to plot density of data, and often for density estimation: estimating the probability density function of the underlying variable. The total area of a histogram always equals 1. In a more general mathematical sense, a histogram is a mapping mi that counts the number of observations that fall into various disjoint categories (known as bins), whereas the graph of a histogram is merely one way to represent a histogram. Thus, if we let n be the total number of observations and k be the total number of bins, the histogram mi meets the following conditions:

$$n = \sum_{i=1}^{k} m_i .$$

3.5 Image Histrogram

An image histogram is type of histogram which acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance. If one views an image as depicting a scene composed of different objects, regions, and etc. then segmentation is the decomposition of an image into these objects and regions by associating or "labeling" each pixel with the object that it corresponds to. Most humans can easily segment an image. Computer automated segmentation is a difficult problem, requiring sophisticated algorithms that work in tandem. "High level" segmentation, such as segmenting humans, cars etc., an image is a very difficult problem. It is still considered unsolved and is actively researched.

Based on point processing, histogram based image segmentation is a very simple algorithm that is sometimes utilized as an initial guess at the "true" segmentation of an image. The basic approach of the segmentation method is dissecting the scanned image into individual blocks to be recognized as character. The Histogram-based

segmentation first segments the lines using Horizontal Projection Profile. The second step in this approach is, segmenting the segmented line into words and then into characters using Vertical Projection Profile (VPP) [10,11].

3.6 Calculating Image Histrogram

An image histogram is a chart that shows the distribution of intensities in an indexed or grayscale image. You can use the information in a histogram to choose an appropriate enhancement operation. For example, if an image histogram shows that the range of intensity values is small, you can use an intensity adjustment function to spread the values across a wider range.

To create an image histogram, use the ***imhist*** function in MATLAB. This function creates a histogram plot by making n equally spaced bins, each representing a range of data values. It then calculates the number of pixels within each range. IMHIST(I) displays a histogram for the intensity image I whose number of bins are specified by the image type. If I is a grayscale image, IMHIST uses 256 bins as a default value. If I is a binary image, IMHIST uses only 2 bins.

3.7 Projection Profile

Features are extracted from the projection profiles of text lines. Let S(N, M) be a binary image of *N* lines and *M* columns [12].

i) **Vertical projection profile:** Sum of black pixels perpendicular to the *x* axis; this is represented by the vector *Pv* of size *N* and defined by:

$$P_v[i] = \sum_{j=1}^{M} S[i,j]$$

ii) **Horizontal projection profile:** Sum of black pixels perpendicular to *y* axis; this is represented by the vector *Ph* of size *M* and defined by:

$$P_h[j] = \sum_{i=1}^{N} S[i,j]$$

3.8 Line Segmentation

The algorithm first generates the horizontal histogram of the entire image. The horizontal histogram is the number of black pixels on each row. When the horizontal histogram is plotted of the document image, the rows with the maximum number of black pixels will have high peak values. Thus in an horizontal histogram of a text document, each text line will produce a patter consisting of a peak and lower histogram values (number of black pixels) it's neighborhood, corresponding to characters on lower side and ascenders on upper side of this peak. Thus the patters (profiles) formed by the text blocks are characterized by ranges of thick black peaks, separated by white gaps. Any graphics or images by contrast have relatively uniform patter, with no prominent peaks. This method is largely independent of font size as no heuristic threshold value is used [13]. Another main feature of a text block is that the histogram corresponding to that block possesses certain frequency and orientation and shoes spatial cohesion i.e, adjacent lines are of similar height. Since text lines appear adjacent to each other, we look for adjacent patterns which have identical shape and size distribution. If this profile is found, then this portion of document must contain a text. Any variation from this characteristic must be an image with or without surrounding text [14]. The projection profiles have valleys of zeros between the text lines. Line segmentation is done at this point . The situation is shown in the Figure 3-5.

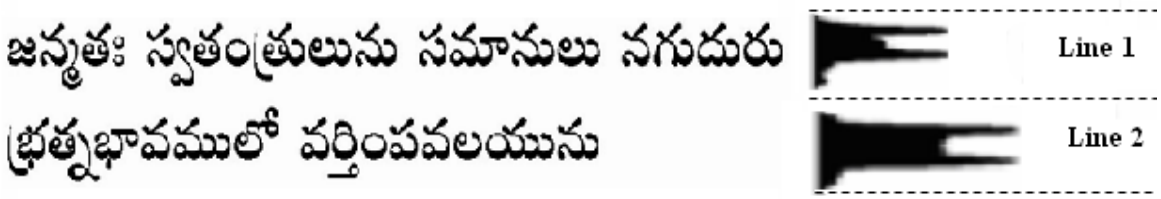


Figure 3-3: figure showing Horizontal Projection Profile (HPP)

3.9 Algorithm of Line Segmentation

Step 1 If the document is a grey scale image, and then binarize the document
Step 2 Find the horizontal histogram of the document
Step 3 Locate all the minima in the histogram
Step 4 Cluster the histogram in to blocks so that each block is classified as wither a regular block or an irregular block. Blocks of image whose histograms exhibit the following criteria are classified as regular blocks. All other blocks are classified as irregular blocks [15].

i) **Shape and Width** For every Telugu text line, the portion above the head line contains the ascenders. Usually few characters in a line have ascenders thus the number of black pixels in these rows is less. Then there is a large increase in histogram since head line is one of the most dominant horizontal line in text. In the portion below the head line, characters are present. Since characters contribute fewer pixels in each row, the number of black pixels in this portion decreases drastically from headline portion. Thus each text line produces a shape in histogram, in which the height of graph increases drastically after some rows and then decreases to a lower value. Printed texts consist of characters with approximately same size and thickness and are located at a regular distance from each other. In a single block, usually the text is of same font size and spacing. Thus the pattern produced by each text line in horizontal histogram has almost the same width (number of rows) and is equally spaced.

ii) **Intervals between peaks:** Since the headline is present at the top of each word, adjacent headlines are equally distanced in the document, due to spatial cohesion. Thus the peaks in horizontal histogram corresponding to adjacent headline repeat after regular intervals. This type of regularity is very rare in any image.

iii) **Width of the Peaks:** Since adjacent text is of same font type and size, the thickness of the peaks will be of the similar widths (number of rows) in the horizontal histogram.

Step 5 All the regular blocks contain only text. Regular blocks can be further sub divided into text blocks based on vertical projection profile

3.10 Some of the Sample Outputs of Line Segmentation

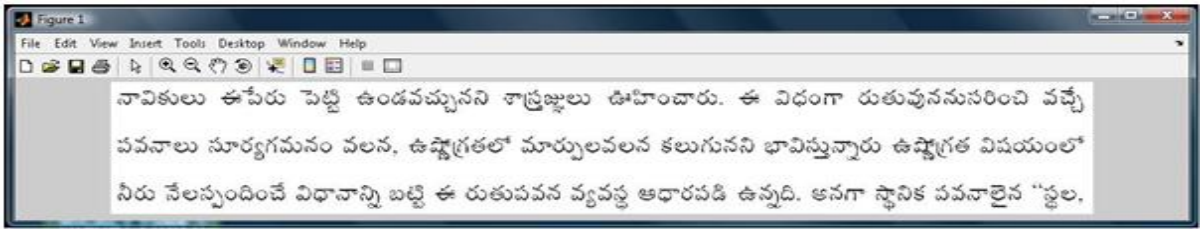


Figure 3-4 : Input text image for line segmentation

The figure 3-6 shows the input text image of line segmentation. It consists of three lines that have to be segmented using horizontal projection profile. It is expected to segment three individual lines. In the segmented figure of the total image is as separating first line image and first line without image. Again the image without first line image segmented as first line image and first line without image.

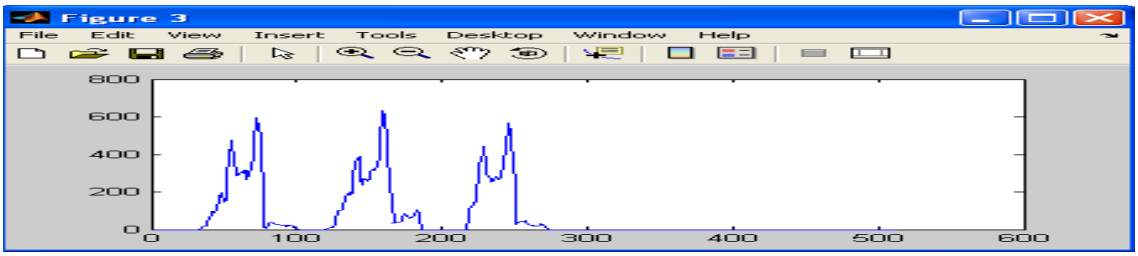


Figure 3-5: figure showing the histogram of the input text image

The figure 3-7 represents the histogram of the input text image shown in figure 3-6 based on line segmentation method. The first peak represents the first line, second peak represents the second line and the third peak represents the third line. The peaks shown in figure 3-7 indicate the core components representing to each line of the input text image. The minor peaks shown in the histogram are due to connected components in each of the lines.

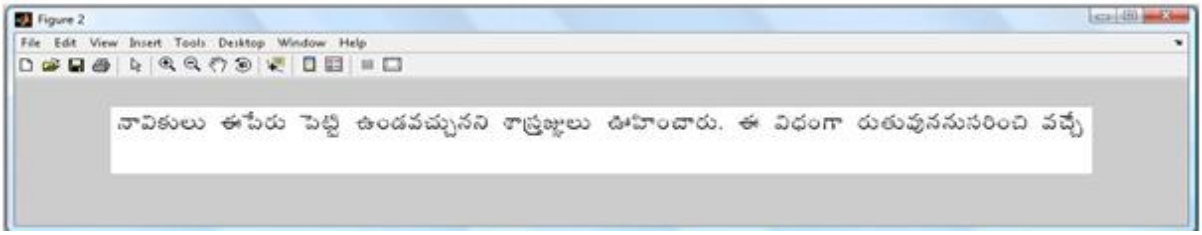


Figure 3-6: First line segmented image

The figure 3-8 shows the first line image segmented using the horizontal projection profile. Similarly figure 3-9 shows the second line image and third line image is shown in figure 3-10 respectively re obtained using horizontal projection profile method on input text mage.

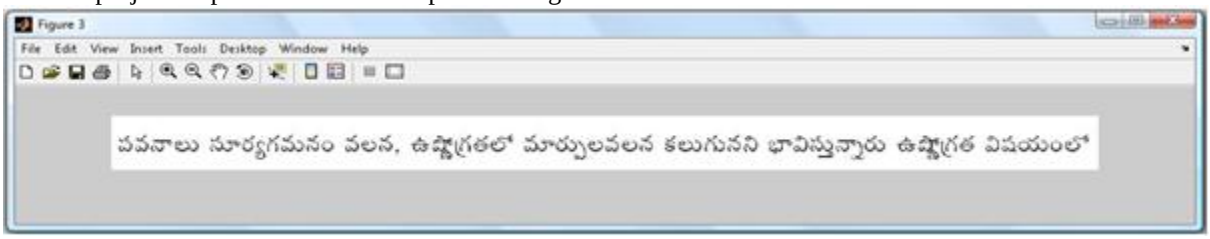


Figure 3-7: figure showing second line segmented

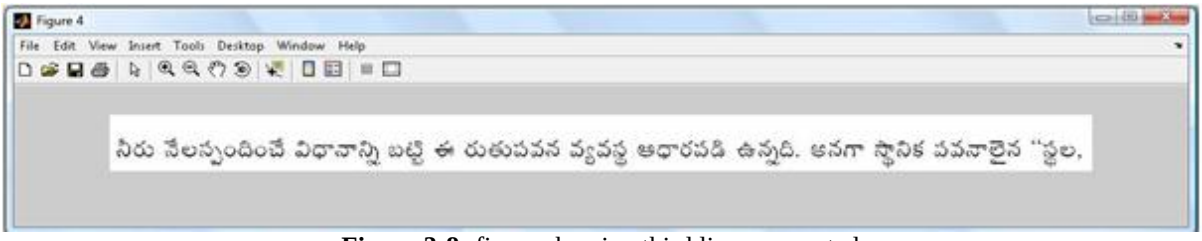


Figure 3-8: figure showing third line segmented

After the line segmentation of a input text the first line image is considered and the top-zone non-core components are separated. To extract top-zone non-core components we first identify different zones of the character by using zonal analysis algorithm.

IV ZONAL ANALYSIS AND CONNECTED COMPONENT

4.1 Zones in Telugu Script

The concept of zones is a useful knowledge that is inherent in all writing systems derived from the Brahmi script during the learning process of the script, which is a common phenomena observed in the formal education system. Extraction of zonal information from the text differs from script to script due to the inherent dissimilarities among Indic scripts. The Telugu word is characterized into three zones – upper zone, middle zone and lower zone. These zones are separated based on the top-line, head-line, base-line and bottom-line. In our approach, the head line is detected by counting the sum of gray values in a row wise scan. The row where the sum is highest is denoted by a head line. By using this head line we can easily separated the upper zone from middle zone of a text line. The process of base line detection is more complicated one than the head line detection. Here, we also perform row wise scanning. The base line is denoted when we find abrupt changes in the sum of gray values between two consecutive rows. The row containing the highest value between these two rows is denoted by the base line [16].

The text line may be partitioned into three zones. The upper-zone denotes the portion above the head-line, the middle-zone covers the portion of basic (and compound) characters below head-line and the lower-zone is the

portion below base-line. Those texts where script lines do not contain head-line, the mean-line separates upper-zone and middle-zone. The base-line separates middle-zone and lower-zone. An imaginary line, where most of the uppermost (lowermost) points of characters of a text line lie, is called as mean-line (base-line). We call the uppermost and lowermost boundary lines of a text line as upper-line and lower-line [17]. The following two cases are considered:

I. A word without subscripts
II. A word with subscripts

Case I: consider a word which does not have a subscript character. The imaginary horizontal line that passes through the top most pixel of the word is the top line. Similarly, the horizontal line that passes through the bottommost pixel of the main character is the baseline. The horizontal line passing through the first peak in the profile is the headline. The word can be divided into top and middle zones. Top zone is the portion between the top and headlines. Middle zone is the portion between the head line and the base line. This is shown in the figure 4-1

Figure 4-1: figure showing the zones in the word without subscripts

Case II: Consider a word which is having subscript characters. In this case the word is divided into three horizontal zones namely top, middle and bottom zones. The top and middle zones are chosen similar to that in the case word without subscripts. The bottom portion is chosen in between the base line and bottom line. The bottom line is the horizontal line that is passing through bottommost pixel of the word. The figure 4-2 shows the above situation.

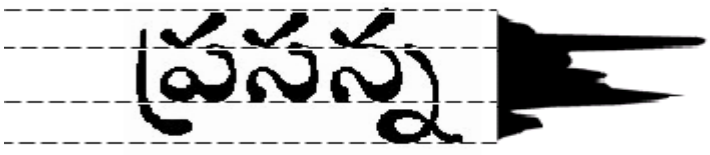


Figure 4-2: figure showing the zones in the word with subscripts

From the above two figures we can analyze the following things:

i) There are several peaks in the HPP's of both sample Telugu words
ii) However, there are only two peaks which are of equal size in both the HPP's
iii) In the HPP of word with subscripts, there are two peaks which are approximately equal in size in the top and middle zones of the word, also the occurrence third peak after the second peak in the bottom zone of the word which is due to the subscripts in the word
iv) Whereas, in the HPP of the word without subscripts, there are two peaks of approximately equal in size in the top middle zones of the word, here the absence of third peak indicates that the word has no subscript characters
v) Thus by checking for the presence or absence of the third peak present in the bottom zone of the HPP of the segmented Telugu word it is possible to know whether the word has the subscript or no

4.2 Text line structure

A text line can be considered as being composed of three zones: the upper zone, the middle zone and the lower zone. These zones are delimited by four virtual lines: the top-line, the head-line, the base-line and the bottom-

line. Each text line has at least a middle zone; the upper zone depends on capital letters and letters with ascenders; the lower zone depends on letters with descenders [18]. This structure allows the definition of four kinds of text line:

1. **full line**, with character parts present in all three zones
2. **ascender line**, with character parts present in the upper and middle zones
3. **descender line**, with character parts present in the *lower* and middle zones
4. **short line**, with character parts present in the middle zone

The middle zone is the most important part of a text line, where we find the main information. Its height is commonly called the *x-height* because it corresponds to the height of a lowercase x. The proportions of the different zones in the font size differ from one type face to another.

4.3 Profile Heights

The profile heights are calculated as follows

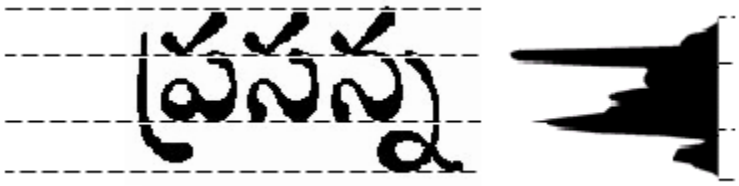


Figure 4-3: features extracted from horizontal profile

(i) Htop = max{i such that $Ph[i] > 0$}
(ii) Hbottom = min{i such that $Ph[i]>0$}
(iii) Hbase = *i* such that $Ph[i] - Ph[i-1]$ is maximal
(iv) Hhead = *i* such that $Ph[i] - Ph[i-1]$ is minimal

The following three features are extracted from *Ph*

1. The height of the whole profile, h1 = | Htop – Hbottom|
2. The height of the upper part of the profile, h2 = | Htop – Hbase|
3. The height of the middle part of the profile, h3 = | Hhead – Hbase|

4.4 Density of black pixels

The weight of a font is reflected by the density of black surfaces on the white background. This density (*dn*) is extracted from the horizontal profile Ph. It is computed pn the central part of the line located between Hhead and Hbase, in order to be independent of the text line structure. This feature is defined by:

$$dn = \frac{1}{n} \sum_{x=1}^{n} Ph[x]$$

4.5 Zonal Analysis Algorithm

Step 1: Enhance the image quality by making all the pixels which are not 255 to 0
Step 2: Perform Line segmentation based on HPP
Step 3: Calculate the top line and bottom line of the complete line. Call it as topline and bottomline
Step 4: Calculation of *headline* – First peak between topline and middle row which is greater than 1.5 * average row intensity
Step 5: Calculation of *baseline* – First peak between middle row and bottomline which greater than 1.5 * average row intensity
Step 6: Do character segmentation based on VPP
Step 7: Calculate character top line and bottom line – chartopline and charbottom line
Step 8: Depending on the chartopline, charbottom line, topline, and bottomline the zone is determined
Step 9: Repeat the steps 7 and 8 for each character in the line

4.6 Connected component (CC)

Detection of Connected Components (CCs), as shown in figure 4-4, between pixels in binary image is a frequently used technique for segmentation of objects and regions within the image. A CC is denoted as a set of black pixels where each pixel has at least one black 8-neighbor. A unique value is assigned to each CC that separates it from other blobs. The original algorithm for CCs extracting and labeling was developed by Rosenfeld and Pfaltz in 1996. It performs two passes through the binary image. In the first pass the image is processed from left to right and top to bottom to generate the labels for each pixel and all the equivalent labels assigned to its equivalence class.

Figure 4-4: Figure showing two different connected components

Some of the recent algorithms generate Bounding Boxes (BBs) of the CCs using the connectivity between the segments of black pixels. Here the binary image is scanned line bye line and the arrays of black segments are composed. Then the connectivity between these black segments for each pair of adjacent scanlines is examined. BBs are extended to include all black pixels connected to the previous scanline.

In our examinations a simple procedure for Connected Components labeling is used. The binary image is scanned from left to right by rows. The CCs of each row are detected and labeled consequently using 8-connectivity as shown in figure 4-5. The background is assigned 0 and never changes since it is not associated with any other region. Then the equivalent components of adjacent rows are merged and the image is relabeled to obtain a consecutive set of labels for the CCs. The procedure is fast and does not require much space in memory.

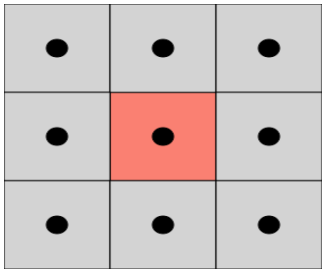


Figure 4-5: figure showing 8-connectivity

A data structure is appointed to each CC. it holds the number of black pixels, the coordinates of the corresponding Bounding Box, the width and height of the CC, the coordinates of its mass center etc. this structure facilitates the manipulation of the components as objects and their further analysis.

4.7 Algorithm for Connected Components

The algorithm makes two passes over the image: one pass to record equivalences and assign temporary labels and the second to replace each temporary label by the label of its equivalence class.

The input data can be modified or labeling information can be maintained in an additional data structure.

On the first pass:

1. Iterate through each element of the data by column, then by row
2. If the element is not the background

i) Get the neighboring elements of the current element
ii) If there are no neighbors, uniquely label the current element and continue
iii) Otherwise, fine the neighbor with the smallest label and assign it to the current element
iv) Store the equivalence between neighboring labels

On the second pass:

1. Iterate through each element of the data by column, then by row
2. If the element is not background

i) Relabel the element with the lowest equivalent label

Here, the background is a classification, specific to the data, used to distinguish salient elements from the foreground. If the background variable is omitted, then the two-pass algorithm will treat the background as another region.

V ALGORITHM

5.1 Font Type and Size Recognition

Font type and size recognition is a fundamental issue in document analysis and recognition, and is often a difficult and time-consuming task. Numerous optical character recognition (OCR) techniques have been proposed and some have been commercialized, but few of them take font type and size recognition into account. Font type recognition has great influence over automatic document processing (ADP) in at least two aspects. Font is an important factor both to character recognition and to script identification. Font classification can reduce the number of alternative shapes for each class, leading to essentially single-font character recognition. Secondly, the ideal output of an ADP system includes not only the content of the document but also the font used to print this document in order to achieve automatic typesetting. The research work in Font Type Recognition has mainly concentrated for Roman, Chinese Arabic, Japanese and other major languages of the world. Very little work has been done on Indian languages and no work is done for Telugu script. Therefore, our proposed problem is an attempt to solve the problem of Font Type and Size Recognition of printed Telugu script.

5.2 Approach and Objectives

The objective of this project is to develop a font and size identification for machine printed Telugu characters. This system will be based on the feature extraction of the various fonts. We assume that the document is of good quality, noise free and no distortion. In our approach first we perform the line segmentation process and extract a single line from the document. Next, connected components are extracted using 8x8 connectivity approach and their zones are identified. From these top zone characters are taken and checked for the tick mark character. Aspect ratio and Pixel ratio for this tick mark is calculated and the type and size of the font are identified. The fonts and Aspect Ratio & Pixel ratio considered in this project are shown in Table 5.1

Font	Size	Aspect Ratio	Pixel Ratio
1. Brahma	14	2.1429	0.5909
	16	0.5238	2.0000
	19	2.5000	0.4815
2. Pallavi	14	2.2857	0.4359
	16	2.5000	0.4286
	19	2.4444	0.4667
3. Priyanka	14	1.7500	0.3333
	16	1.7143	0.3333
	19	1.5556	0.3125
4. Anupama	14	1.5000	0.3714
	16	1.5556	0.3548
	19	1.7273	0.2822

Table 5.1: Table showing different fonts used in this project

5.3 Algorithm of the project

The whole algorithm is divided into the following section:

A. Pre-processing of input Telugu document image
B. Line Segmentation
C. Calculation of top, head, base and bottom lines
D. Connected component segmentation
E. Classification of each component into core and non-core components based on zonal information
F. Identification of tick character among the top zone non-core components
G. Calculation of pixel ratio and aspect ratio of identified tick character
H. Identification of font and size based on the pixel ratio and aspect ratio

(A) Pre-processing of input Telugu document

1. Conversion of the RGB (color) OCR Telugu document into Gray document.
2. Conversion of Gray document into black and white document based on a threshold value.
3. The image consists of pixels whose value is either 0 (black pixel) or 255 (white pixel).
4. Addition of twenty rows in top and bottom of image. These forty rows will have only white pixels.
5. Addition of twenty columns in left and right side of image. The forty columns will have only white pixels.

(B) Line Segmentation

1. Calculate the Horizontal profile of each row. It is the plot of intensity versus rows. For each row the number of black pixels are counter and cumulated. The plot of number of black pixels in each row w.r.t the row gives the horizontal profile.
2. Based on the horizontal profile each line in the Telugu document can be segmented.
3. For each line follow the below procedure.

(C) Calculation of Top, Head, Base and Bottom Lines

1. Calculate the Horizontal profile of the line.
2. The first row which contains the minimum or more than one black pixel is the TOP line
3. The last row which contains the minimum or more than one black pixel is the BOTTOM line
4. The MIDDLE line is calculated between TOP and BOTTOM lines
5. The row which is has maximum intensity between TOP and MIDDLE line HEAD line
6. The row which is has maximum intensity between TOP and BOTTOM line is BASE line.

(D) Connected Component Segmentation

1. Using 8x8 Connectivity approach group the connected components
2. Each connected component will represent a alphabet of the telugu language. Along with the modifier components (Votthulu, Deergalu) will also be extracted

(E) Classification of each component into core and non-core component based on zonal information

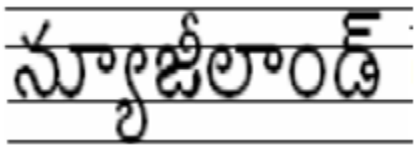


Figure 5-1: Figure showing different zones and lines of Telugu script

1. The space between Top and Head lines is TOP zone
2. The space between Head and Base lines is MIDDLE zone
3. The space between Base and Bottom line is BOTTOM zone
4. Each component will occupy the majority of a zone. The core components of Telugu language (i.e. alphabets) occupy the Middle zone. Hence connected components occupying the middle zone are classified as core components.
5. The other components which occupy majority in Top or Bottom zone are classified as Non-core components.
6. The Non-core components are further classified into top zone non-core components and bottom zone non-core components
7. Each Top zone non-core component is considered and following procedure id followed.

(F) Identification of tick character among the top zone non-core components

1. Each Top zone character is checked whether it is TICK character or not.
2. The TICK character has a unique shape in Telugu language and does not vary a lot with fonts and sizes
3. Additionally the TICK character is easier to be identified.

4. The right profile of the component is calculated. Right profile is calculated by the following procedure:
i. For each row traverse from extreme right side calculate the pixel which the first black pixel is encountered.
ii. The plot of the pixel volume value versus the row gives the right profile.
5. The top profile of the component is calculated. Top profile is calculated by following procedure:
i. For each column traverse from extreme top side calculate the pixel at which the first black pixel is encountered.
ii. The plot of the pixel row value versus the column gives the top profile.
6. The right profile of TICK character will have a descending plot
7. The top profile of TICK character will have a descending plot followed by ascending plot.
8. If the right and top profile of the component satisfies the above two conditions it is TICK character.
9. If the component is not TICK character the next component is checked and the procedure is followed till all the components are checked.
10. If there is no TICK character in the selected line the above procedure is repeated for the components in the next line and stopped once a TICK character is identified.

(G) Calculation of pixel ratio and aspect ratio of the identified tick character

1. Pixel ratio is the ratio of the black pixels to white pixels in the TICK character.
2. Aspect ratio is the ratio of the height of the component to the width of the component.

(H) Identification of font and size based on the pixel ratio and aspect ratio

1. Each Telugu font and size will have unique pixel ratio and aspect ratio of the TICK character.
2. Based on the calculated pixel and aspect ratio in previous steps the Telugu Font and Size can be identified.

VI SIMULATION AND STUDIES

6.1 Introduction

The proposed algorithm has been tested on different fonts: Anupama, Brahma, Pallavi and Priyanka of sizes 14, 16 and 19. The program has be developed in Matlab language and simulated. The following are the test results of the font Anupama of size 16.

6.2 Simulation

The input text image is shown in the following figure.

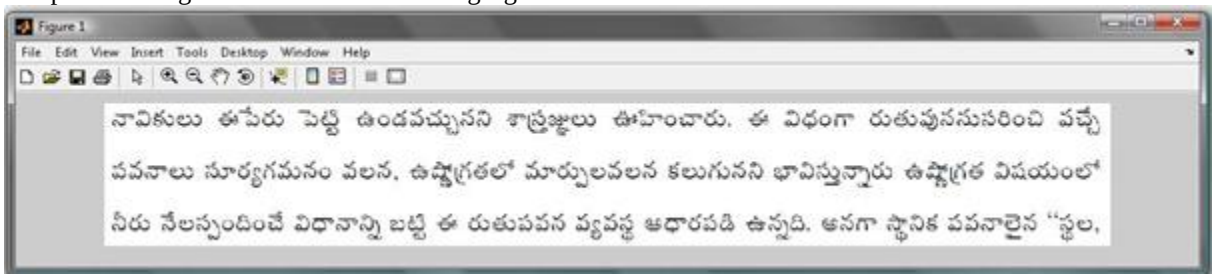


Figure 6-1: Input text Image

The input image consists of three lines those have to be segmented using horizontal projection profile. This input image is first converted from RGB to gray image and then to black and white image (black `pixels and white pixels). The following figure shows the converted black and white image of the input RGB image.

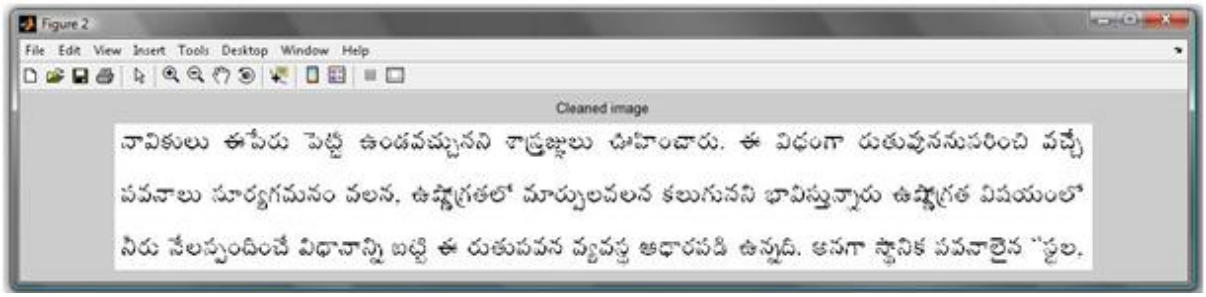


Figure 6-2: Black and White image of the input text

Using Horizontal Projection Profile on this black white text image the first line is extracted. The following figure shows the horizontal projection of the above image.

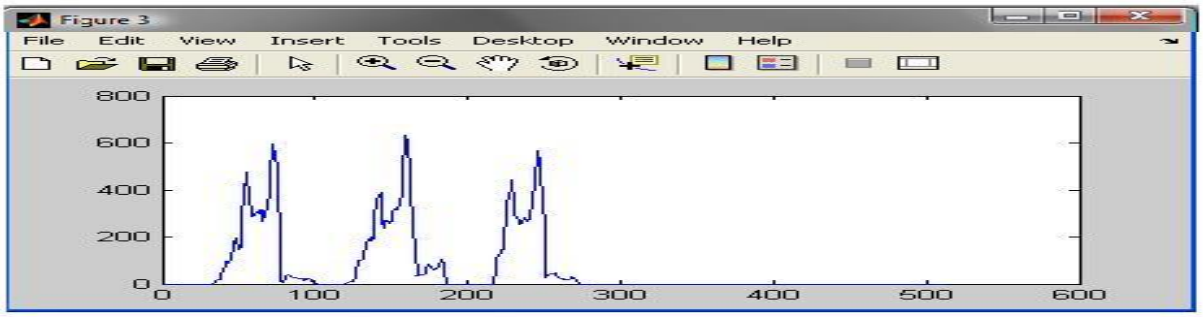


Figure 6-3: Horizontal Projection Profile of Input Image

Figure 6-4: First line image of the input text

This first line image is then used to find the connected components using 8x8 connectivity approach. The following figure shows the connected components labeling in first line image.

Figure 6-5: Connected components labeling image of first line

Now the zonal information id identified and core and non-core components are separated. Again from the non-core components the top-zone non-core components are separated. The following figure shows the top-zone non-core components extracted from the first line of the text.

Figure 6-6: Top-zone non-core components of first line in the text

From this each character is identified for the TICK-mark character and is extracted. The following figure shows the extracted TICK-mark from the top-zone non-core components.

Figure 6-7: Extracted TICK-mark image

Now the Pixel Ratio and Aspect Ratio are identified based on the pixel density of black pixels and white pixels and the width and height of the TICK-mark.

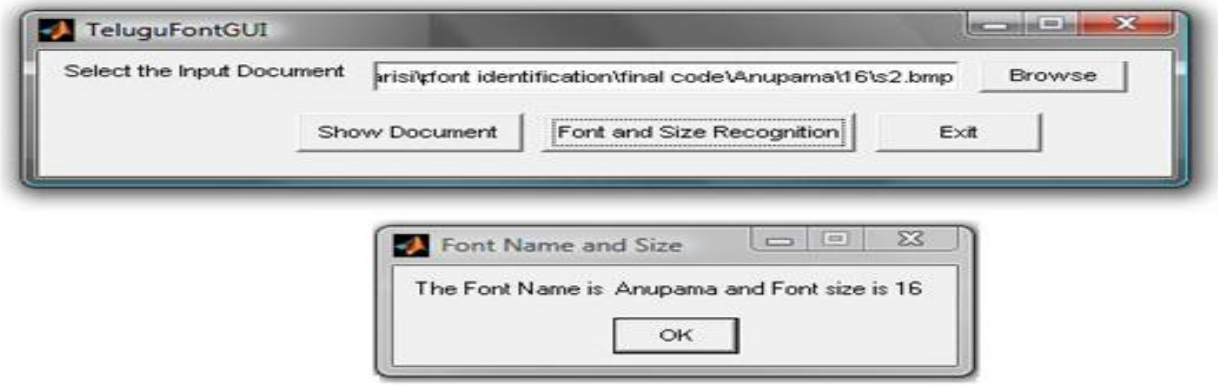


Figure 8. Final result of Font Type and Size

The above figure shows the final result of the algorithm. It shows the name and size of the font. This algorithm has been implemented as Graphical User Interface.

VII CONCLUSION

7.1 Conclusion

An algorithm for Font and Size Identification in Telugu Printed Document. Unlike existing methods, this new algorithm is based on aspect ratio and pixel ratio of a special character (TICK – mark) in the text. In this algorithm we have used projection profiles and 8x8 connectivity approach for extracting this special character. As we are using Histogram-projection profiles method for segmentation of the text, this approach is faster than other conventional methods. This algorithm is tested on different types of fonts (Anupama, Brahma, Pallavi and Priyanka) and different sizes of 14, 16 and 19.

7.2 Future Scope

The Font and Size Identification can be extended to other type of fonts and also with noise added documents. This algorithm can also be extended to identify the touching, overlapping and broken characters also.

REFERENCES

[1]. B. Anuradha Srinivas, Arun Agarwal and C. Raghavendra Rao – "An Overview of OCR Research in Indian Scripts", IJCSES International Journal of Computer Science and Engineering System, Vol.2, No.2, April 2008
[2]. Abdelwahab Zramdini and Rolf Ingold – "Optical Font Recognition from Projection Profiles", Computer Science Institute, university of Fribourg, Switzerland, Electronic Publishing, Vol.6(3), 249-260 (September 1993).
[3]. C. Vasantha Lakshmi and C. Patvardhan - "A Multi-font OCR System for Printed Telugu Text", Proceedings of the Language Engineering Conference (LEC'02) 2003 IEEE
[4]. http://www.buddhavihara.in/ancient.htm

[5]. R. Gonzalez and R. Woods - "Digital Image Processing", Addison-Wesley, Reading. MA, 91:1992
[6]. R.C. Gonzalez and R. E. Woods – "Digital Image Processing", Addison Wesley, 1993.
[7]. Zaidi Razak, Khansa Zulkiflee and Mohd Yamani Idna Idris – "Off-line Handwritting Text Line Segmentation: A Review", IJCSNS, Vol.8 No.7, July 2008
[8]. Laurence Likforman-Sulem, Abderrazak, Bruno Taconet – "Text Line Segmentation of Historical Documents: a Survey", Special Issue on Analysis of Historical Documents, International Journal on Document Analysis and Recognition, Springer, 2006.
[9]. J. Hochberg, L. Kerns, P. Kelly and T. Thomas - " Automatic Script Identification from Images using cluste-based templates", IEEE Trans. Pattern Anal. Mach. Intell. 19(2) (1997) 176-181
[10]. Atul Negi, K. Nikhil Shanker and Chandra Kanth Chereddi - "Localization, Extraction and Recognition of Text in Telugu Document Images", Proceedings of the 7th International conference on Document Analysis and Recognition (ICDAR 2003) IEEE
[11]. A. Spitz - "Determination of the Script and Language content of Document Images", IEEE Trans. Pattern Anal. Mach. Intell. 19(3) (1997) 235-245
[12]. Zaidi Razak, Khansa Zulkiflee and Mohd Yamani Idna Idris – "Off-line Handwritting Text Line Segmentation: A Review", IJCSNS, Vol.8 No.7, July 2008
[13]. C. Vasantha Lakshmi and C. Patvardhan – "Optical Character Recognition of Basic Symbols in Printed Telugu Text", IE (I) Journal-CP, Vol 84, November 2003 pp 66-71
[14]. Florian Kleber and Robert Sablatnig – " Ancient Document Analysis Based on Text Line Extraction", 2008 IEEE
[15]. Yong Zhu, Tieniu Tan and Yunhong Wang – "Font Recognition Based on Global Texture Analysis" NLPR, Institute of Automation, Chinese Academy of Sciences
[16]. U. Pal, S. Sinha and B.B. Chaudhuri – "Multi-Script Line Identification from Indian Documents", Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR'03) IEEE
[17]. M.C.Padma and Dr.P.A. Vijaya – "Language Identification of Kannada, Hindi and English Text words thorough Visual Discriminating Features", International Journal of Computational Intelligence Systems, Vol.1, No.2 (May, 2008), 116-126
[18]. Florian Kleber and Robert Sablatnig – "Ancient Document Analysis Based on Text Line Extraction", 2008 IEEEj

AUTHOR PROFILE

K.Ram Mohan Rao received B.Tech degree in Electronics and Communication Engineering from JNTU, Hyderabad and M.Tech degree in Electronics and Communication Engineering from JNTU, Hyderabad. He is working as Assistant Professor in the Department of Electronics and Communication Engineering, Sri Indu College of Engineering and Technology, Hyderabad, India. His research interests includes digital Communication, Image Processing and wireless communications

B.Ramesh received B.Tech degree in Electronics and Communication Engineering from Kakatiya University, Warangal and pursuing M.E in Electronics and Communication Engineering from Osmania University, Hyderabad. He is working as an Assistant Professor in the Department of Electronics and Communication Engineering, Sri Indu College of Engineering and Technology, Hyderabad, India. His research interests include digital Systems and VLSI Technology.

G.Indrasena Reddy received B.E degree in Electronics and Communication Engineering from Osmania University, Hyderabad and pursuing M.E in Electronics and Communication Engineering from Osmania University, Hyderabad. He is working as Assistant Professor in the Department of Electronics and Communication Engineering, Sri Indu College of Engineering and Technology, Hyderabad, India. His research interests includes Signal processing and Image Processing