Pattern Anal Applic (2004) 7: 190–204
DOI 10.1007/s10044-004-0217-2

THEORETICAL ADVANCES

C. Vasantha Lakshmi · C. Patvardhan

An optical character recognition system for printed Telugu text

Received: 1 June 2003 / Accepted: 13 May 2004 / Published online: 1 July 2004

Abstract Telugu is one of the oldest and popular languages of India, spoken by more than 66 million people, especially in South India. Not much work has been reported on the development of optical character recognition (OCR) systems for Telugu text. Therefore, it is an area of current research. Some characters in Telugu are made up of more than one connected symbol. Compound characters are written by associating modifiers with consonants, resulting in a huge number of possible combinations, running into hundreds of thousands. A compound character may contain one or more connected symbols. Therefore, systems developed for documents of other scripts, like Roman, cannot be used directly for the Telugu language.

The individual connected portions of a character or a compound character are defined as basic symbols in this paper and treated as a unit of recognition. The algorithms designed exploit special characteristics of Telugu script for processing the document images efficiently. The algorithms have been implemented to create a Telugu OCR system for printed text (TOSP). The output of TOSP is in phonetic English that can be transliterated to generate editable Telugu text. A special feature of TOSP is that it is designed to handle a large variety of sizes and multiple fonts, and still provides raw OCR accuracy of nearly 98%. The phonetic English representation can be also used to develop a Telugu text-to-speech system; work is in progress in this regard.

C. V. Lakshmi (✉)
Department of Physics and Computer Science,
Dayalbagh Educational Institute,
282005 Agra, India
E-mail: cvasantha@rediffmail.com

C. Patvardhan (✉)
Department of Electrical Engineering,
Dayalbagh Educational Institute,
282005 Agra, India
E-mail: cpatvardhan@hotmail.com

Keywords Pattern recognition · OCR · Telugu · Basic symbols · Printed text

Introduction

During the past few decades, substantial research efforts have been devoted to optical character recognition (OCR) [1, 2]. The object of OCR is automatic reading of optically sensed document text materials to translate human-readable characters into machine-readable codes. Research in OCR is popular for its various potential applications in banks, post offices and defence organisations. Other applications involve reading aids for the blind, library automation, language processing and multi-media design [3].

Commercial OCR packages are already available for languages like English. Considerable work has also been done for languages like Japanese and Chinese [1]. Recently, work has been done in the development of OCR systems for Indian languages. This includes work on recognition of Devanagari characters [4], Bengali characters [5], Kannada characters [6] and Tamil characters [7]. Some more recent work on Indian languages is also reported [8, 9, 10, 11, 12].

Telugu is one of the oldest and most popular languages of India. It is spoken by more than 66 million people, especially in South India. Historically, Telugu has evolved from the ancient Brahmi script. It also used features of the Dravidian (Pali) language for script generation. In the process of evolution, this script was carved with needles on palm leaves, and so, it favoured rounded letter shapes. Work on Telugu character recognition is not substantial [13, 14]. Therefore, development of an OCR system for Telugu is an important area of current research.

Most document analysis systems can be visualised as consisting of two steps: the pre-processor and the recogniser. In pre-processing, the raw image obtained by scanning a page of text is converted to a form acceptable to the recogniser by extracting individually recognizable characters. This step is also called segmentation. The recogniser typically works as follows; the pre-processed image of the character is processed to obtain meaningful elements, called features; recognition is completed by searching for a feature vector in a database of stored feature vectors of all possible Telugu characters that matches with the feature vector of the character to be recognised. Thus, the system has three steps: segmentation of individual characters in the document image, features extraction and character recognition. This approach is followed in the current work.

Thus, segmentation of the images of individual characters from the image of the page is the first step. This step requires the answer to a simply posed question: "What constitutes a character?" Many researchers who try to provide an algorithmic answer to this question find themselves in a catch-22 situation. A character is a pattern that resembles one of the symbols that the system is designed to recognise. But, to determine such a resemblance, the pattern must be segmented from the document image. Each stage depends on the other, and, in complex cases, it is paradoxical to seek a pattern that will match a member of the system's recognition alphabet of symbols without incorporating detailed knowledge of the structure of those symbols into the process [15]. These observations are crucial in motivating the approach presented in this paper.

In Indian scripts, one or more vowel and consonant modifiers are attached to the consonant forms in a variety of combinations forming compound characters. The total number of possible compound characters is in of the order of hundreds of thousands. Therefore, the question, "What constitutes a character?", assumes many new dimensions for Indian languages. Is a modifier an independent character or not? Does being treated as an independent character depend on the way it is written, i.e. whether it is written touching the character it is to modify or separated from it? A more detailed discussion of these issues for Telugu script is provided in Sect. 2.

In this paper, an approach has been presented for Telugu. The unit of segmentation has been defined as a basic symbol. The separated basic symbols are then recognised and their association information determined to output the text in phonetic English, which can be transliterated into Telugu text and edited. The algorithms are implemented to create a Telugu OCR system for printed text (TOSP). A special feature of TOSP is that it is designed to handle a variety of sizes and multiple fonts, and still provides a recognition accuracy of more than 98% in most cases. In contrast, most of the work on OCR for Indian languages is restricted to a single font and a single size [4]. The phonetic English representation can be used to develop a text-to-speech system for Telugu; work is in progress in this regard.

The rest of the paper is organised as follows. Description of the Telugu script and motivation for the approach presented in this paper is given in more detail in Sect. 2. Sections 3 and 4 discuss the algorithms and implementation of TOSP. Results on various test data are also presented. Determination of the association information is described in Sect. 5. Some discussion on the development of post-processing module for improving the recognition accuracy is presented in Sect. 6. Conclusions and the scope for future work are given in Sect. 7.

Structure of Telugu characters and segmentation issues

Telugu is a syllabic language. There is very little scope for confusion and spelling problems. In that sense, it is a WYSIWYG (what you see is what you get) script. This form of script is considered to be most scientific by linguists.

Characteristics of Telugu script

The Telugu script consists of 18 vowels, 36 consonants and two dual symbols. Of the vowels, sixteen are in common usage. Table 1 lists some of the vowels in Harshapriya and Godavari fonts. All vowels and consonants, along with their modifiers and phonetic equivalent symbols, are listed in Tables 2 and 3, respectively.

Compound characters in Telugu follow some phonetic sequences that can be represented in grammatical form, as shown in Table 4.

Base consonants are vowel-suppressed consonants. These are typically used when words of other languages are written in Telugu. The third combination, i.e. of a base consonant and a vowel, is an extremely important and often used combination in Telugu script. As there are 38 (36 + 2 dual symbols) base consonants and 16 vowels, logically, 608 (38×16 = 608) combinations are possible.

The combinations from the fourth to the seventh combinations 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

Table 1 Vowels in Harshapriya and Godavari fonts

అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	అం	అః
అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	అం	అః

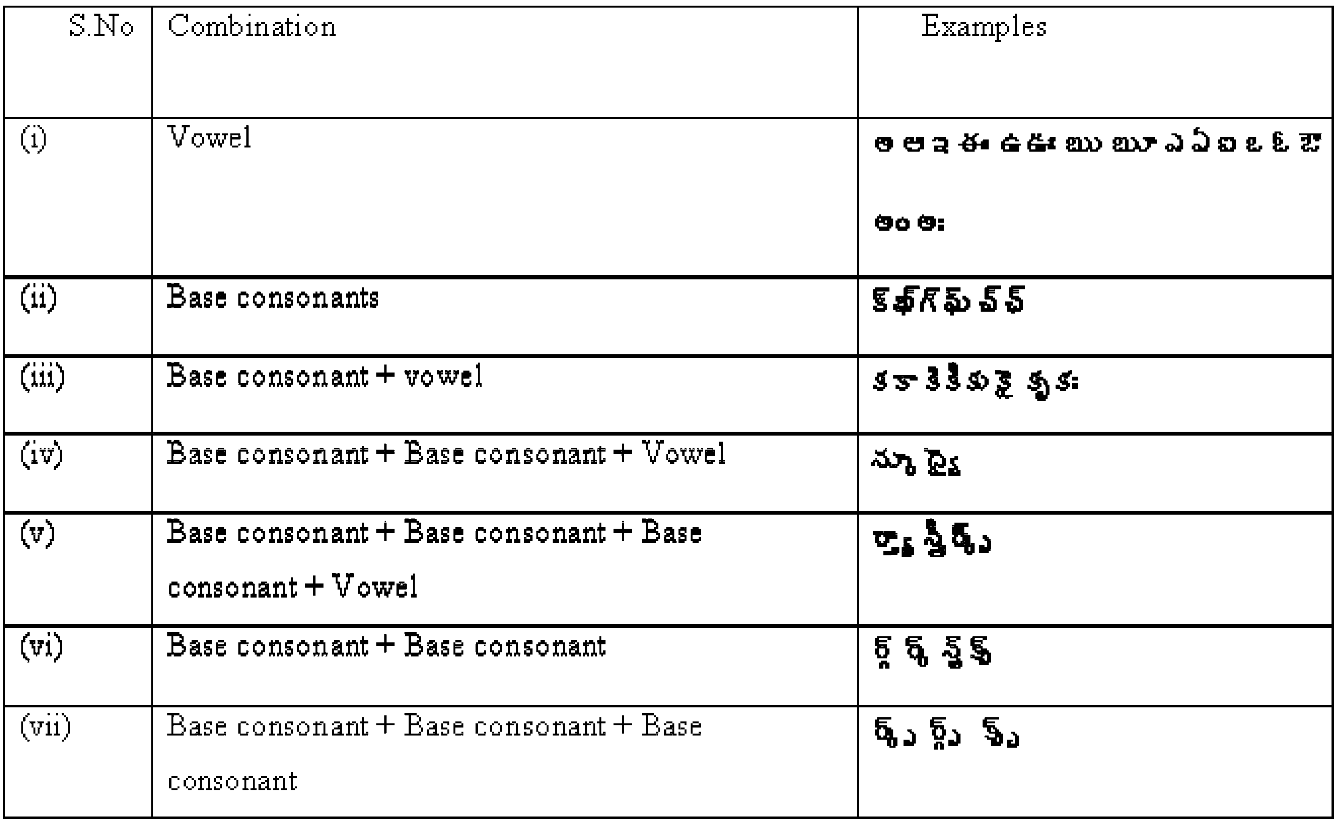
అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఋ	ౠ	ఎ	ఏ	ఐ	ఒ	ఓ	ఔ	అం	అః
✓	ా	ి	ీ	ు	ూ	ృ	ౄ	ె	ే	ై	ొ	ో	ౌ	ం	ః
a	A	i	I	u	U	HRI	NR	E	e	AY	O	o	AW	M	:

Table 2 Vowels, their associated modifiers (matras) and their phonetic English representation

combinations from the fourth to seventh combinations generate a large number of conjuncts in Telugu script. The fourth combination logically generates (38×38x16) 23,104 different compound characters. This is an important combination. The fifth combination is similar to the fourth combination. The second and the third consonants act as the dependent consonants. Logically 746,496 different compound characters are possible in this combination, but their frequency of appearance in the text is less when compared to the previous combination. In the sixth and seventh combinations, 1,296 combinations and

Table 3 Consonants, their associated modifiers (voththus) and their phonetic English representation

క	ఖ	గ	ఘ	ఙ
్క	్ఖ	్గ	్ఘ	్ఙ
ka	kha	ga	gha	Ng
చ	ఛ	జ	ఝ	ఞ
్చ	్ఛ	్జ	్ఝ	్ఞ
cha	chha	ja	jha	Nj
ట	ఠ	డ	ఢ	ణ
్ట	్ఠ	్డ	్ఢ	్ణ
Ta	Tha	Da	Dha	Na
త	థ	ద	ధ	న
్త	్థ	్ద	్ధ	్న
ta	tha	da	dha	na
ప	ఫ	బ	భ	మ
్ప	్ఫ	్బ	్భ	్మ
pa	pha	ba	bha	ma
య	ర	ల	వ	శ
్య	్ర	్ల	్వ	్శ
ya	ra	la	va	sha
ష	స	హ	క్ష	త్ర
్ష	్స	్హ	NR	NR
Sha	sa	ha	kSh	tr
జ్ఞ	ఱ	ళ		
NR	్ఱ	్ళ		
dny	NR	La		



S.No	Combination	Examples
(i)	Vowel	అ ఆ ఇ ఈ ఉ ఊ ఋ ౠ ఎ ఏ ఐ ఒ ఓ ఔ అం అః
(ii)	Base consonants	క ఖ గ ఘ చ ఛ
(iii)	Base consonant + vowel	క కా కి కీ కు కూ కృ కః
(iv)	Base consonant + Base consonant + Vowel	స్మా బ్లై
(v)	Base consonant + Base consonant + Base consonant + Vowel	ష్ట్రా స్త్ర్య
(vi)	Base consonant + Base consonant	క్గ క్థ క్త క్క
(vii)	Base consonant + Base consonant + Base consonant	క్థ్స క్గ్స క్క్స

Table 4 The various combinations forming compound characters

46,656 combinations, respectively, are logically possible. 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.

Presence of these combinations is less common in writing native Telugu words.

Examples are స్ట /ST/ in టెస్ట (TEST) and ర్క్స /RX/ in మార్క్స (MARX).

An analysis of the frequency of occurrence of different compound characters is given in Table 5. The samples are from a newspaper, a small part of the book on the Patent Act in India, a children's 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. The percentage occurrence of these combinations is as follows:

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

Table 5 Analysis of different groups of characters in the Telugu language

Sample	Vowels	Consonants	CV core	Conjuncts	Others	Total
1	76	389	829	394	130	1818
2	109	574	1312	471	326	2792
3	170	755	1615	607	376	3523
4	179	805	1723	738	344	3789
5	160	862	1854	600	647	4123
6	226	1191	2552	794	909	5672

The high percentage of incidence of compound characters implies that recognition of these must be given due importance. This is in contrast with another Indian language, Bangla, where compound characters are of secondary importance due to their low incidence percentage [5].

Segmentation issues in OCR of Telugu text

A connected region in an image of Telugu text may be:

(a) A part of a character or a compound character

(b) A character
(c) A compound character

This complicates the segmentation issues. The areas occupied by individual characters in a line of text are not in a horizontal line, unlike in English text, and in some cases, the area of a single complex character formation can be equal to the sum of the areas of two individual characters. The segmentation algorithm has to take these factors into consideration. The basic question to be answered in segmentation is: "What are the symbols that will be isolated during segmentation and provided to the recogniser for completing the OCR?"

The first approach is to treat all types of conjuncts, together with the base consonants, as units for the purpose of segmentation and further recognition. This is not preferable for a number of reasons. The first reason is that the sheer number of possibilities has been shown to be enormous. The second reason is that, in compound characters like క్రై "KRAI ", we have to identify all the three parts, i.e. కె Ke, ౖ Ai below and ్ర Ra on the left, as being together in the same compound character, although they are not connected in the image. This is, in general, difficult because the association information is difficult to generate until the recognition process is at least partially completed, and the reason we are segmenting is to perform this recognition process. This is the catch-22 situation referred earlier, and, therefore, treating all types of conjuncts together is not possible. The second alternative is to attempt to isolate the base consonants, vowel modifiers, etc. This is difficult and leads to unmanageable complications at the segmentation stage where the symbols are yet to be recognised. This is primarily because the symbols are full of curves and their separation is not clear. However, this is a popular approach for Indian scripts like Devanagari and Bangla [4, 5].

This motivates a more in-depth analysis of segmentation issues. In Telugu, the vowel modifier is written attached to the symbol it modifies, but the consonant modifier is written separately (without touching). Thus, a consonant modifier can be treated as an independent entity for the purpose of segmentation and subsequent recognition. This reduces the number of possible connected entities in the image of Telugu text drastically. These connected entities in Telugu script are defined as basic symbols and are the unit of segmentation and recognition. This is a logical approach, provided that the total number of such symbols is not excessive. If the number of basic symbols were large, this approach would simply transfer the complications from the preprocessing to the recognition stage. It turns out that there are less than 400 basic symbols in Harshapriya font of 30 size. The basic symbols are font- and size-specific to a very small extent because, in some fonts and sizes, some vowel modifiers are placed slightly separated from the character they modify, and in others, these are placed connected to it. This leads to a slight deviation in the number of basic symbols for each font. However, this deviation is very small and the number of basic symbols changes only slightly from font to font and from size to size. Creating a recognizer for 400 symbols is not difficult. Therefore, in this work, the objective of pre-processing is to extract these basic symbols from the image of a page of Telugu text. The basic symbols are now identified. This analysis is a pre-requisite for the development of a recognizer for these symbols. The basic symbols identified in Telugu script for Harshapriya 30 font are shown in Fig. 1.

The steps in the proposed approach are, therefore:

1. Segment the image of text into basic symbols as defined above
2. Compute features of the basic symbols
3. Recognise each basic symbol separately by matching with a features database of basic symbols and establish the association information between them

This relatively simple approach enables recognition of Telugu text with high levels of accuracy.

The pre-processing phase

The basic pre-processing steps are explained in this section. Figure 2 depicts a sample page of text which consists of a poem made up of five lines. This sample text is used as a running example in the rest of the section.

Thresholding and noise cleaning

Pages containing Telugu characters are scanned and digitised. A histogram-based thresholding approach is used to convert this image into a binary image. The output binary image has values of 0 (black, which implies the object) for all pixels in the input image with luminance less than 0.5, and 1 (white, which implies the background) for all other pixels. The digitised image shows protrusions and dents in the characters as well as isolated black pixels over the background. These are cleaned by a logical smoothing approach. Some discontinuities might be introduced in the images because of scanning defects. Figure 3 shows a Telugu character 'ka' in three different sizes. The character 'ka' actually is a single-connected entity, but the middle character image of 'ka' has a discontinuity because of a missing pixel at the point where the arrow is pointing. This, if not rectified, will be identified as two connected components. If two different connected components are close to each other by a pixel difference, the missing pixel is filled with a black pixel, resulting in one connected component.

Skew detection and removal

Casual use of the scanner may lead to a skew in the document image. Skew angle is the angle that the text

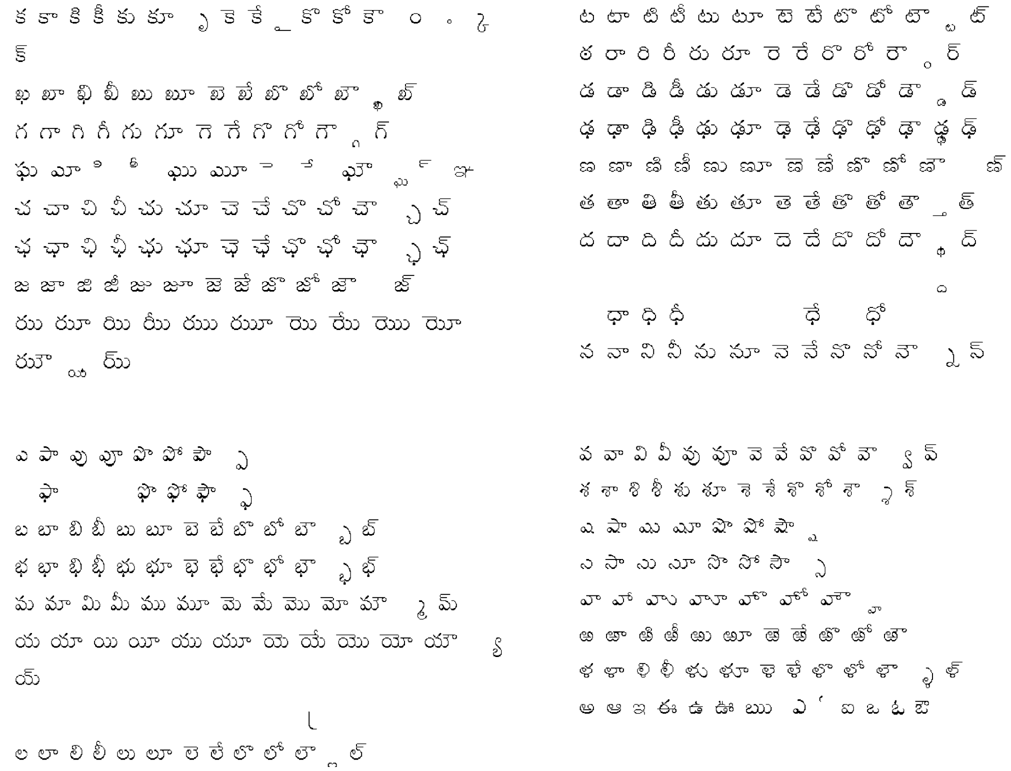


Fig. 1 Basic symbols in Harshapriya 30 font

చుకు చుకు బండి వస్తోంది	chuku chuku baMdi vastondi
దూరం దూరం జరగండి	dooram dooram jaragaMdi
ఆగినాక ఎక్కండి	Aaginaaka Ekkandi
జోజో పాప ఏడవకు	jojo papa eDavaku
లడ్డు మిఠాఇ తినిపిస్తా	laDDu miThai tinipistA

Fig. 2 A Telugu poem consisting of five lines of text. Transliterated version in English is given alongside the Telugu version

lines of a recorded digital document make with the horizontal direction. This may cause problems in segmenting the image to extract its layout structure [16, 17, 18, 19, 20]. Skew detection in Telugu text documents is made complicated by the presence of the vowel and consonant modifiers above and below the modified characters. Because of these, symbols belonging to a line of text do not fall on the same horizontal line. Further, there is no top bar as in other Indian scripts, like Devanagari, to aid the skew detection process.

There exist a wide variety of skew detection algorithms based on the projection profile [16, 17], Hough transform [18, 19], line correlation [20], etc. The algorithm devised in this paper is a combination of the profile method and the Hough transform method. The following algorithm is proposed for skew detection and removal in TOSP:

1. Find the bottom-most black pixel in each column of a text region of the image.
2. For each such black pixel, determine the connected component and select its bottom-most pixel if it is of adequate size. Otherwise, reject the bottom pixel as noise. Select the next bottom-most pixel in the column and repeat the processing.
3. In some columns, the pixel identified as the bottom-most pixel may actually belong to some line above the bottom-most line (because of gap in the bottom-most line at that column). Drop such pixels from further consideration.
4. Invoke the Hough transform on the bottom-most pixels of each connected component identified in step 2 and not dropped in step 3.

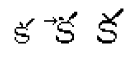
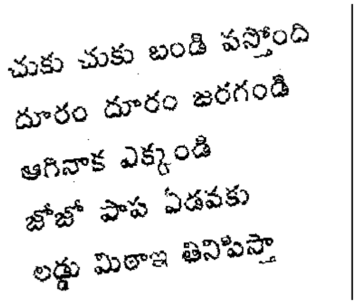


Fig. 3 Missing pixel of middle Telugu character shown by an *arrow*



చుకు చుకు బండి వస్తోంది
దూరం దూరం జరగండి
ఆగినాక ఎక్కండి
జోజో పాప ఏడవకు
లడ్డు మిఠాఇ తినిపిస్తా

Fig. 4 A picture with a skew

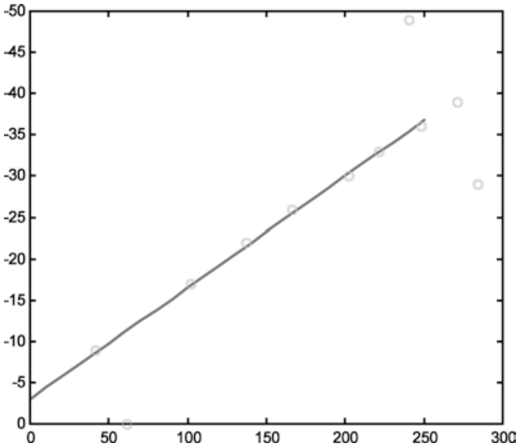


Fig. 5 Skew determined with 8 out of 11 pixels falling on the line $m=-0.135$ and $c=-3$

5. The angle and offset from the origin (i.e. the $\tan^{-1}m$ and c in $y=mx+c$) for which the maximum number of pixels in step 4 are in a straight line is the estimated skew angle.
6. Confirm this by determining the angle made by the left-most pixels of the first connected component of each row of the image.
7. In case of discrepancy, ignore the connected components of the bottommost line and repeat the same process for the last but one row.
8. Correct the skew by tilting the image back by the negative of the skew angle.

Several computational experiments are performed with a text image to test the robustness of the proposed algorithm across variations in skew angle. The skew angles vary between 8° in one direction and 7° in the opposite direction. These are taken to be worst case values for the skew that can be introduced due to scanning error. Figure 2 shows an image of a small and popular poem in Telugu. It is scanned at an angle of 7° to yield Fig. 4. The algorithm explained above is then used for the determination of the skew angle. The points that actually lie on the line determined for the sample are shown in Fig. 5.

The results presented in Table 6 show that the algorithm is quite robust and finds the skew angles with reasonable accuracy. The slight error is due to the discretisation. The accuracy of the algorithm can be improved further by attempting more combinations of m and c so that the discretisation error is minimised. The other steps of segmentation can now be performed on the document image that has been rendered free of skew.

Table 6 Actual and estimated skew angles for various input skew angles for another sample

Actual skew angle (°)	Estimated skew angle (°)
7	7.1250
5	5.14
3	3.4336
1	1.14
−2	−2.0045
−4	−4.0042
−6	−5.7106
−8	−7.6884

Line, word and character segmentation

For convenience of recognition, the OCR system should automatically detect individual text lines as well as segment the words and then the basic symbols accurately. The procedures employed for this purpose are described in this section.

Extraction of lines of text from the document image

Extraction of lines of text is comparatively easy in the case of English text once the skew has been detected and corrected. The different lines of text are identified by computing a histogram of the number of black pixels in each row and then finding the valleys in the projection profile. The spaces between the lines can be identified by the presence of rows with no black pixels identified in the histogram. The rows with the non-zero number of black pixels indicate the rows of text. The situation in Telugu text is different. The presence of modifier symbols above and below the main character implies that the continuous rows with non-zero values in the histogram are the result of three varieties of symbols:

(a) Vowel modifiers encountered above the main characters
(b) Main characters
(c) Vowel or consonant modifiers encountered below the main characters

This situation is illustrated in Fig. 6.

The histogram for the sample page consisting of five lines of text is plotted. The histogram indicates that there are seven sets of rows with non-zero values. Therefore, the above three cases have to be suitably distinguished. For this purpose, it is observed that the number of blank rows in the histogram is smaller between the main character and a modifier, and larger between the different lines of text. In the first case, the line gap is approximately a third of the height of the main

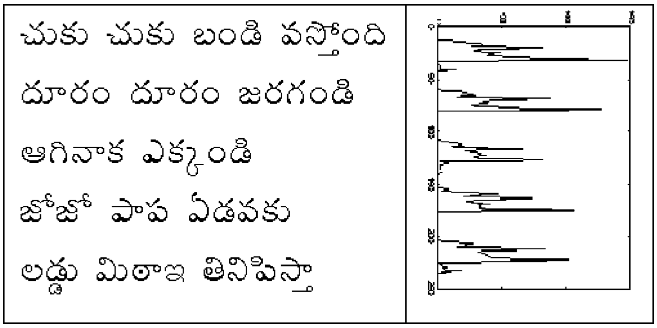

చుకు చుకు బండి వస్తోంది
దూరం దూరం జరగండి
ఆగినాక ఎక్కండి
జోజో పాప ఏడవకు
లడ్డు మిఠాఇ తినిపిస్తా

Fig. 6 Histogram of the number of black pixels in individual rows of the sample text

Table 7 Justification for proposed line segmentation approach

Sample	Font size	Height of main character	Actual gap between main character and modifier	2/7ths of the height of the main character	Word gap
1	12	29	4	8	17
2	15	36	4	10	22
3	18	44	4	12	26
4	20	49	6	14	29
5	25	50	9	14	36
6	28	55	9	15	41
7	30	60	11	17	45
8	32	67	11	19	48
9	35	70	13	20	54

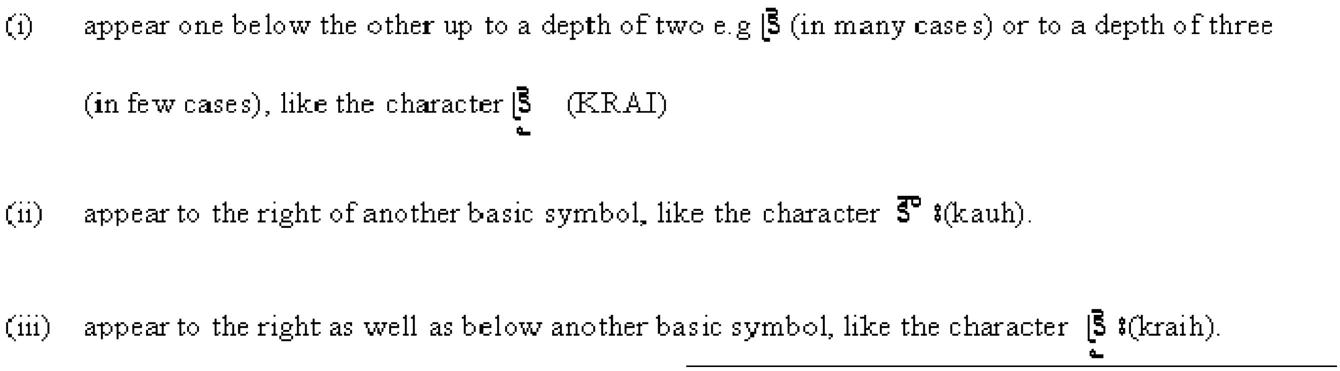
character. The number of continuous rows with histogram count >0 indicates the height of the characters in that set of rows. The maximum of this number for the first three sets of rows is taken as the height of the main character. So, if the line gap is less than or equal to a third of the size of the main character, it is not treated as a separation between the lines. Otherwise, it is treated as a separation between the lines.

Extraction of words of text from each line in the document

To break a line into its constituent words, advantage is taken of the gap between the words (word gap) and a smaller gap between the characters (character gap) constituting a word. The character gap for Telugu characters was found to be approximately 2/7ths of the height of the main character. To find the word gaps in a line, a histogram of the number of black pixels in individual columns is plotted. If the count of the continuous blank columns in the projection profile is less than 2/7ths of the height of a character, it is a character gap. If it is greater than or equal to that, it is a word gap. The justification for this logic is illustrated in Table 7.

Isolation of basic symbols from each word in the document

The next step is to identify the basic symbols of each character. The basic symbols of a Telugu character may:



(i) appear one below the other up to a depth of two e.g క్ర (in many case s) or to a depth of three

(in few cases), like the character క్రై (KRAI)

(ii) appear to the right of another basic symbol, like the character కౌ ః(kauh).

(iii) appear to the right as well as below another basic symbol, like the character క్రై ః(kraih).

Therefore, once a basic symbol is detected, the next symbol that is to be detected is the one that appears below it (if any) up to any depth in the same line. Only then are the ones to the right of this basic symbol detected. The method for isolating the basic symbols is as follows. A word is looked at column-wise from top to bottom in each column until a black pixel is encountered. This is called the seed pixel. Breadth-first search is initiated from this seed pixel to identify a connected component (i.e. basic symbol). The process is repeated till all the black pixels are assigned to some connected component and all the basic symbols are isolated.

Features computation and recognition

A database of the basic symbols and their features is created for all the basic symbols in three fonts, i.e. Godavari, Hemalatha and Harshapriya, and three sizes, i.e. 25, 30 and 35. The total number of basic symbols present in all the three different fonts and three different sizes mentioned above is approximately 3,000. A preliminary classification scheme is implemented. All the symbols are converted to a size corresponding to 36 columns whilst maintaining aspect ratios and the row sizes for this column size are used to classify the basic symbols into 15 different sets.

The image of the basic symbol is divided in to $N_1 \times N_2$ partitions. The features used in this paper are real-valued direction features (RDF) [21]. These are based on the percentage of pixels in each direction range within each partition. An adaptive gradient magnitude threshold, $\bar{r}>$, is computed as the average gradient magnitude over the whole character image gradient map. This threshold is used to filter out spurious responses to the Sobel operator used to find gradients. Threshold value, $\bar{r}$, is computed as follows:

$$\bar{r} = \sum \frac{r(i,j)}{D_1 D_2}$$

Thresholding is performed to nullify pixels whose gradient magnitude value lies below the computed threshold:

if $r(i,j) \geqslant \bar{r}$, then $r'(i,j) = r(i,j)$ and $\theta'(i,j) = \theta(i,j)$
else $r'(i,j) = \text{NULL}$ and $\theta'(i,j) = \text{NULL}$

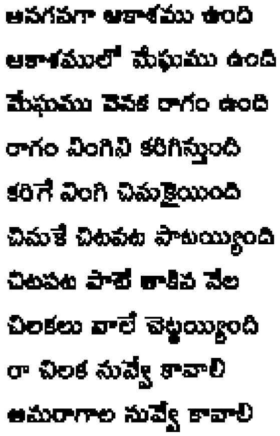
Pixel gradient directions are quantized into K ranges. The number of pixels in each partition that have pixel gradients in each of the K ranges is computed. Thus, exactly K features are computed in each partition of the image. The total number of features is $N_1 \times N_2 \times K$. In this work, N_1 and N_2 are both taken to be 4 and K is set at 18. These values are experimentally selected. Larger values increase computational expense and have been found to be unnecessary. Smaller values result in lower recognition accuracies. In the feature vector, F_i, element $F_i(k) = \alpha_k$, where α_k is the percentage of pixels in this partition with direction quantised to k. Feature vectors from all of the partitions are concatenated to obtain the complete feature vector. These feature vectors are real-valued.

The algorithm for feature extraction is represented succinctly in the following pseudo-code:

```
1. For each word in every line of a scanned printed page
   of Telugu text,
2. Isolate the next basic symbol from the word being
   recognised.
   Repeat steps 3 to 6 for each basic symbol:
3. Obtain the bounding box eliminating the blank sur-
   rounding space.
4. Calculate the gradient magnitude and direction at
   each pixel location.
5. Calculate the adaptive threshold of gradient magni-
   tude and perform thresholding to obtain the new
   threshold gradient direction at each pixel location.
6. Partition the adaptive gradient direction map and
   extract the complete feature vector.
```

All the feature vectors are stored in a database to aid the recogniser. As mentioned above, the database is created with three different fonts and three different sizes. The database stores the feature vectors divided into 15 sets, as described above.

The OCR of a text page begins with the scanned image. The pre-processing steps, described in Sect. 3, are performed on this image to isolate the image of each basic symbol. The feature vector of the symbol to be recognised is computed as given above. This is then provided to the recogniser. Advantage is taken of the preliminary classification scheme. The number of rows corresponding to the size of 36 columns (keeping aspect ratio unchanged) for the symbol to be recognised indicates the set to which it belongs. The recogniser uses a minimum distance classification scheme utilising both versions, i.e. the nearest neighbour (NN) classifier and the K-nearest neighbour (KNN) classifier [19] on the feature vectors stored in the set determined by preliminary classification to identify the basic symbol. The process is repeated until all the basic symbols are recognised.

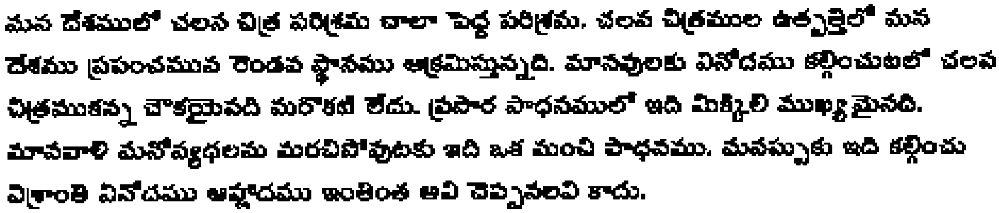


అనగనగా ఆకాశము ఉంది
ఆకాశములో మేఘము ఉంది
మేఘము వెనక రాగం ఉంది
రాగం నింగిని కరిగిస్తుంది
కరిగే నింగి చినుకైయింది
చినుకే చిటపట పాటయ్యింది
చిటపట పాటే తాకిన వేల
చిలకలు వాలే చెట్టయ్యింది
రా చిలక నువ్వే కావాలి
అనురాగాల నువ్వే కావాలి

Fig. 7 Sample image, DS6, in Hemalatha font, size 18

Artificial neural networks (ANN) based on the well-known back-propagation algorithm [22] are also trained for the recognition of the basic symbols. A different network is trained for each of the sets identified by the preliminary classification scheme as described above. The features used for training these networks are the RDF described above. The features of the basic symbol to be recognised are determined and the appropriate network is determined by the preliminary classification method. The neural network then recognises the basic symbol and provides the appropriate output.

Results of extensive testing of the NN, KNN and ANN recognisers are presented in this section. The input text is printed using HP Laserjet 6L Pro or HP Laserjet 5P printers. Images of this text are scanned at 300 dpi for sizes 15 and above, and at 600 dpi for sizes 9 to 12, and are provided to the OCR system. Although the smaller fonts are scanned at a higher resolution, the same recognisers created with images scanned at 300 dpi are used for recognition. The pre-processing module segments the image into its constituent basic symbols and passes the same to the recognisers. The recognisers complete the recognition task and provide the output. The test data sets consist of the text of a recent popular film song and a paragraph of text taken from the Corpora of text created by the Central Institute for Indian Languages, Mysore, printed in 3 different fonts and 13 different sizes. Therefore, each leads to 39 data sets. Sample images for each of these are shown in Figs. 7 and 8, respectively. Results of recognition on the page of text shown in Fig. 77 taken in different fonts and sizes



మన దేశములో చలన చిత్ర పరిశ్రమ చాలా పెద్ద పరిశ్రమ. చలన చిత్రముల ఉత్పత్తిలో మన
దేశము ప్రపంచమున రెండవ స్థానము ఆక్రమిస్తున్నది. మానవులకు వినోదము కల్గించుటలో చలన
చిత్రముకన్న చౌకైనది మరొకటి లేదు. ప్రచార సాధనములో ఇది మిక్కిలి ముఖ్యమైనది.
మానవాళి మనోవ్యథలను మరచిపోవుటకు ఇది ఒక మంచి సాధనము. మనుష్యులకు ఇది కల్గించు
విశ్రాంతి వినోదము ఆహ్లాదము ఇంతింత అని చెప్పనలవి కాదు.

Fig. 8 Another sample image, DS45, in Hemalatha font, size 18

Table 8 Results of RDF features on a sample page of Hemalatha text of different fonts and sizes

Data set	Size	Number of basic symbols	NN		KNN		ANN	
			Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)
DS1	9	127	122	96	123	97	122	96
DS2	10	127	122	96	123	97	122	96
DS3	11	127	122	96	124	98	123	97
DS4	12	127	123	97	124	98	123	97
DS5	15	127	124	98	125	98	122	96
DS6	18	127	126	99	126	99	123	97
DS7	20	127	122	96	124	98	124	98
DS8	23	127	124	98	126	99	127	100
DS9	25	127	125	98	127	100	127	100
DS10	28	127	126	99	124	98	121	95
DS11	30	127	125	98	125	98	126	99
DS12	32	127	126	99	124	98	124	98
DS13	35	127	124	98	127	100	125	98

are presented in Tables 8, 9, 10 and those on the sample shown in Fig. 8 are presented in Tables 11, 12, 13.

Cross validation of the test results is performed as follows. Not only are test images different from images used to create the database, but they also contain text of different sizes that is, in some cases, much smaller. The generalisation capability of the recognisers is clear from the fact that recognition accuracies are very high, even for the very small fonts.

The test results show that even raw OCR provides good results for multiple fonts and sizes that compare favourably with existing results in [14]. Recognition fails only in the case of symbols that are very similar to each other. A list of such symbols is given in the appendix, in a table referred to as the Confusion Table. These results can be further improved by post-processing. Some discussion regarding this is provided in Sect.6.

Determination of association information

An OCR system must provide output in a form that enables downstream operations like reading, editing, etc. to be performed with convenience. TOSP produces output in phonetic English that can be transliterated into Telugu text using standard transliteration software. This text can then be directly edited with any Indian language software. Construction of the phonetic English code from the list of basic symbols recognised is a non-trivial task as the association information is to be determined.

The need for the determination of association information and the complications in doing so can be understood as follows. The basic symbols may be vowels, consonants or modifiers. The modifiers are separated out and recognised independently of the symbol they are to modify in the order top to bottom and left to right and the context information is lost. In the absence of this information, it is impossible to reconstruct the original text by piecing together the various basic symbols and finding out the compound characters. Therefore, it is necessary to ensure that this information is available.

There can be at least two strategies for doing so. Some data structures may be created during the segmentation

Table 9 Results of RDF features on a sample page of Harshapriya text of different fonts and sizes

Data set	Size	Number of basic symbols	NN		KNN		ANN	
			Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)
DS14	9	127	121	95	122	96	121	95
DS15	10	127	121	95	122	96	122	96
DS16	11	127	122	96	123	97	123	97
DS17	12	127	122	96	123	97	123	97
DS18	15	127	115	90	118	94	114	90
DS19	18	127	127	100	122	96	125	98
DS20	20	127	124	98	125	98	120	95
DS21	23	127	124	98	124	98	125	98
DS22	25	127	123	97	123	97	123	97
DS23	28	127	123	97	126	99	123	97
DS24	30	127	123	97	126	99	127	100
DS25	32	127	126	99	126	99	124	98
DS26	35	129	120	95	125	98	125	96

Table 10 Results of RDF features on a sample page of Godavari text of different fonts and sizes

Data set	Size	Number of basic symbols	NN		KNN		ANN	
			Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)
DS27	9	127	122	96	122	96	122	96
DS28	10	127	122	96	123	97	122	96
DS29	11	127	123	97	124	98	122	96
DS30	12	127	123	97	124	98	123	97
DS31	15	126	124	98	123	97	120	95
DS32	18	127	125	98	124	98	124	98
DS33	20	128	124	98	122	96	127	99
DS34	23	129	127	100	127	100	124	97
DS35	25	127	124	98	124	98	126	99
DS36	28	127	124	98	124	98	124	98
DS37	30	127	124	98	125	98	124	98
DS38	32	127	125	98	125	98	123	97
DS39	35	127	124	98	126	100	118	93

Table 11 Results of OCR using RDF features on another sample of Hemalata text of different fonts and sizes

Data set	Size	Number of basic symbols	NN		KNN		ANN	
			Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)
DS40	9	215	208	97	210	98	208	97
DS41	10	215	211	98	211	98	211	98
DS42	11	215	213	99	213	99	213	99
DS43	12	215	213	99	213	99	213	99
DS44	15	215	212	98	211	98	211	98
DS45	18	215	213	99	212	99	210	98
DS46	20	215	206	96	213	99	211	98
DS47	23	215	209	97	209	97	210	98
DS48	25	215	204	95	209	97	208	96
DS49	28	216	212	98	211	98	213	99
DS50	30	219	209	96	209	96	216	99
DS51	32	215	210	98	212	99	208	96
DS52	35	215	211	98	209	97	211	98

stage and information regarding the association of the various basic symbols carried right through the recognition process. This is in effect a tedious task. The reason is as follows. When a modifier is segmented, the decision as to whether it is construed to be attached to the right of the previously segmented consonant or to the left of the consonant yet to be segmented depends on which modifier it is. Similarly, a modifier may be attached to a consonant above it or below it. Some of the decisions are aided by line and word segmentation, but not all. At the segmentation stage, the identity of the modifier is not yet known (recognition is yet to follow!). Further, there may be more than one modifier that is attached to the same consonant. If the results of the recognition of all the possibly relevant symbols are to be carried through for determination of the association information and analysed, the required data structures and logic become complicated.

An alternative strategy is to output the basic symbols into a file as and when they are recognised. End of word is also output accordingly. It turns out that the association information can be reconstructed for each word independently at the end of the recognition process by defining a set of simple rules. This splits up the problem into independent recognition and association information determination phases, and considerably simplifies the implementation. Therefore, this approach is followed in this work.

A line of text is scanned for basic symbols from top to bottom and left to right. The basic symbols, therefore, are processed in this order. The association relationship between basic symbols is handled for every word. If the word contains more than two basic symbols, the association could be one of the several possibilities listed below:

(a) The first basic symbol itself is a modifier that modifies the second basic symbol.

Example: characters ప ‘pa’, సి ‘si’ etc.

(b) The first as well as the second basic symbols are modifiers, which modify the third basic symbol.

Table 12 Results of OCR using RDF features on another sample of Harshapriya text of different fonts and sizes

Data set	Size	Number of basic symbols	NN		KNN		ANN	
			Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)
DS53	9	215	209	97	209	97	209	97
DS54	10	215	209	97	210	98	209	97
DS55	11	215	211	98	211	98	210	98
DS56	12	215	211	98	211	98	211	98
DS57	15	215	194	90	203	94	204	95
DS58	18	215	215	100	206	96	206	96
DS59	20	215	211	98	211	98	210	98
DS60	23	215	209	97	206	97	208	96
DS61	25	215	211	98	204	95	210	97
DS62	28	216	208	97	205	95	209	97
DS63	30	219	214	98	208	97	217	99
DS64	32	215	208	97	211	98	212	99
DS65	35	215	204	95	212	99	209	97

Table 13 Results of OCR using RDF features on another sample of Godavari text of different fonts and sizes

Data set	Size	Number of basic symbols	NN		KNN		ANN	
			Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)	Number of basic symbols recognised	Accuracy (%)
DS66	9	215	208	97	209	97	208	97
DS67	10	215	208	97	210	98	208	97
DS68	11	215	211	98	211	98	210	98
DS69	12	215	211	98	211	98	211	98
DS70	15	215	211	98	208	97	207	97
DS71	18	215	212	99	211	98	210	98
DS72	20	215	212	99	204	96	206	96
DS73	23	214	211	99	210	98	211	99
DS74	25	215	211	98	210	98	213	99
DS75	28	214	210	98	211	99	212	99
DS76	30	216	214	99	211	98	215	99
DS77	32	216	210	97	210	97	211	98
DS78	35	215	212	99	215	100	214	99

Example: characters ప్ర 'pra', స్ర 'sra' etc.

(c) The first as well as third are modifiers, which modify the second basic symbol.

Example: characters ప్త 'pta', స్త 'sta' etc.

(d) The first, third as well as fourth basic symbols are modifiers, which modify the second basic symbol.

Example: ప్తై 'ptai', స్లై 'slai' etc.

(e) The first, second as well as fourth basic symbols are modifiers, which modify the third basic symbol.

Example: ప్రై 'prai', హ్రై 'hrai' etc.

(f) The first, second, fourth and fifth basic symbols are modifiers, which modify the third basic symbol.

Example: ప్ప్రై 'pprai'

(g) The second basic symbol alone is a modifier, which modifies the first basic symbol.

Example: క్క 'kka', క్త 'kta' etc.

(h) The second as well as third basic symbols are modifiers, which modify the first basic symbol.

Example: త్త్రై 'ththrai' or గ్గ్రై 'ggrai' etc.

(i) The second, third as well as fourth basic symbols are modifiers, which modify the first basic symbol.

Example: త్తై 'ththai', గ్గై 'ggai' etc.

The association relationships are determined taking into consideration all these different cases and combinations thereof.

The text of a paragraph of a popular Telugu song is shown in Fig. 9. The various basic symbols that are recognised in the OCR of this paragraph are shown in Fig. 10. These basic symbols are combined with the help of rules to determine the compound characters and the phonetic English representation of the song is output, as shown in Fig. 11. This phonetic English representation is then transliterated into editable Telugu text shown in Fig. 12 using standard transliteration software.

Development of post-processing module—some discussion

A standard technique for improving recognition accuracies is the development of post-processing modules. One possible strategy for this purpose is to match the recognised words with a dictionary of "valid" words in the language. Such a dictionary match may suggest possible "close" alternative words for the words erroneously recognised by OCR system but not found in the dictionary. The size of the dictionary used has a large impact on the efficacy of this method. Further, the organisation of the dictionary, i.e. how root words and various extended forms are stored, impacts the speed. It has been observed that the number of distinct words in Telugu is relatively much larger than other languages

```
anaganagA  AkAshamu  uMdi
  AkAshamulo  meghamu  uMdi
  meghamu  vEnaka  rAgaM  uMdi
  rAgaM  niMgini  karigistuMdi
  karige  niMgi  chinukE\yiMdi
  chinuke  chiTapaTa  pATayyiMdi
  chiTapaTa  pATe  tAkina  nela
  chilakalu  vAle  chETTayyiMdi
  rA  chilaka  nuvve  kAvAli
  anurAgAla  nuvve  kAvAli
```

Fig. 11 Phonetic English code generated after performing association

అనగనగా ఆకాశము ఉంది
ఆకాశములో మేఘము ఉంది
మేఘము వెనక రాగం ఉంది
రాగం నింగిని కరిగిస్తుంది
కరిగే నింగి చినుకైయింది
చినుకే చిటపట పాటయ్యింది
చిటపట పాటే తాకిన నేల
చిలకలు వాలే చెట్టయ్యింది
రా చిలక నువ్వే కావాలి
అనురాగాల నువ్వే కావాలి

Fig. 9 A popular Telugu song

అనగనగా ఆకాశము ఉంది
ఆకాశములో మేఘము ఉంది
మేఘము వెనక రాగం ఉంది
రాగం నింగిని కరిగిస్తుంది
కరిగే నింగి చినుకైయింది
చినుకే చిటపట పాటయ్యింది
చిటపట పాటే తాకిన నేల
చిలకలు వాలే చెట్టయ్యింది
రా చిలక నువ్వే కావాలి
అనురాగాల నువ్వే కావాలి

Fig. 12 Transliterated output in Telugu

Fig. 10 Output of the recogniser

```
a  na ga na gA . A   kA sha mu   .  u  M  di    .
 A kA sha   mu lo .  me amatra  gha  geeta   mu  .   u    M    di     .
 me  amatra  ghgha   geeta   mu  . vE  na  ka  . rA  ga M . u M   di     .
 rA ga M . ni  M  gi ni . ka ri gi  amatra  su  tta   M  di   .
 ka   ri  ge . ni  M   gi .  chi nu kE \ yi   M     di    .
 chi  nu   ke . chi  Ta  amatra  pa  Ta .  pA  Ta  yi  yya  M di     .
 chi Ta amatra  pa  Ta  . pA  Te . tA  ki na  .   ne     la     .
 chi la ka lu . vA le .  chE  Ta TTa  yi  yya  M    di     .
 rA  . chi la  ka     .   nu      ve      vva    .   kA     vA li     .
 a  nu  rA  gA  la     .  nu   ve   vva    .   kA     vA     li     .
```

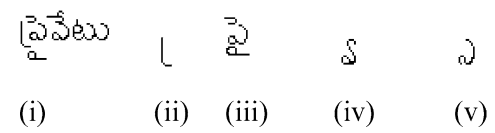


Fig. 13 Use of association information for post-processing

because of the way extended forms of words are coupled with the root word [23]. So, dictionary matching may be computationally cumbersome.

In Sect. 5, logic for association information has been described. Different cases classifying the various combinations of symbols have been identified and rules have been designed to decide the association relationship.

For example, consider the word shown in Fig. 13(i). Suppose that, due to defects in the image, the modifier in Fig. 13(ii), which modifies the compound character in Fig. 13(iii), is recognised as the one in Fig. 13(iv) or (v). These modifier symbols occur to the right of the character they modify. However, in this case, there is no character to the left of the symbol being recognised as it is the first symbol in the word when read from left to right. Thus, the first character cannot be Fig. 13(iv) or (v) and a case of mis-recognition is detected.

Rules can be constructed as given in example above for detecting cases of mis-recognition based on the association logic. Further, rules may also be constructed for suggesting possible current alternatives for the mis-recognised symbol based on the association logic. The symbol in doubt can be processed with more detailed logic to perform correct recognition. Work is in progress in this regard.

Conclusions

A Telugu OCR System for printed text (TOSP) that can perform the character recognition for multiple fonts in multiple sizes has been presented. It is based on a novel basic symbol approach that is in contrast to the other approaches for Indian languages reported in the literature. The idea of basic symbols is that it is neither convenient nor necessary to segment the compound characters into their constituent elements by separating out the various modifiers. It is the complications in these tasks that have traditionally been considered to be problematic in development of the OCR systems for Indian languages. It is suggested that the whole symbol that is a connected entity in the text can be processed as one. This simplifies the segmentation tasks considerably.

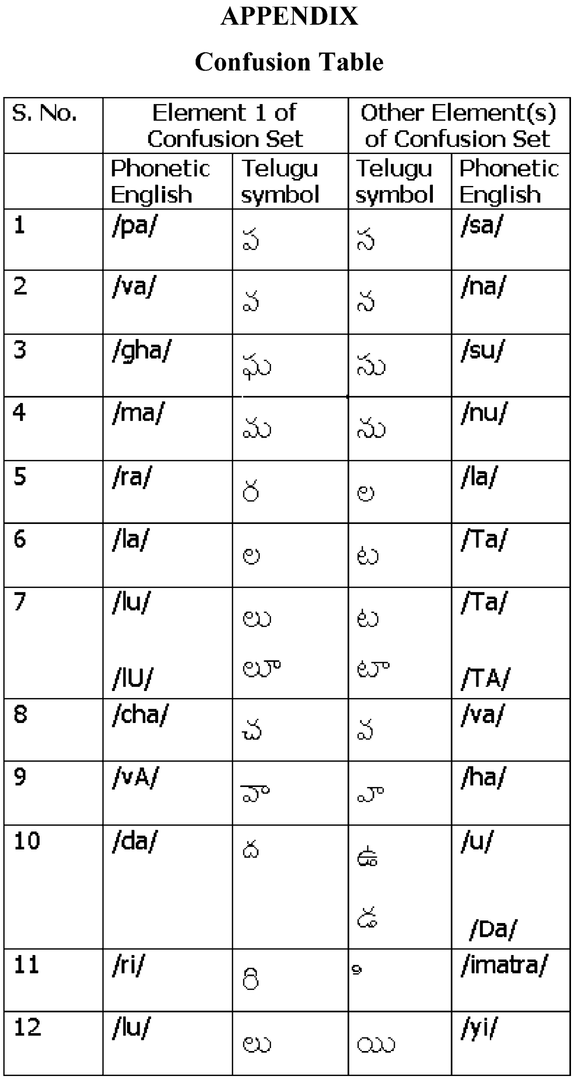
The proposed system takes scanned images of printed text as input and performs pre-processing on the image to separate out the basic symbols. A two-phase recogniser then recognises these. The preliminary recognition phase divides the symbols into 15 sets on the basis of the size of the symbols. The classifier operating on radial distance feature (RFD) vectors completes the recognition. Results are provided on various test images. The recognition accuracy is more than 98% for most of the images. This is very good considering that these are raw OCR results for a very complicated Telugu script and that the system handles multiple fonts and multiple sizes. Further, the computational effort is quite restricted. Improvement in recognition accuracies may be achieved by incorporating logic based on association information determination for which work is in progress. The output of the recogniser is in phonetic English. This output may be advantageously utilised for the development of a text-to-speech module. Work is in progress in this regard.

Originality and contributions

This work describes a Telugu optical character recognition system for printed text (TOSP). The main hindrance in the development of OCR systems for Telugu is the enormous number of symbols produced by the combinations of various consonants, vowels and modifiers thereof. An elegant system based on the identification and recognition of basic symbols is described in this work. Pre-processing and recognition tasks are considerably simplified in this approach. A modified Hough transform method is proposed for skew detection. Gradient-based features are used for recognising the basic symbols. These have been used for the first time for OCR of an Indian language. The segmentation scheme necessitates determination of association information between basic symbols. Ideas regarding the determination of association information are novel and have not been discussed elsewhere in the literature. Logic developed for determining association information can be extended for developing a post-processing module to improve recognition accuracies. Pointers are provided for this. A special feature of TOSP is that it is designed to handle multiple sizes and multiple fonts and still provides a high raw OCR accuracy of more than 98%, which is much better than the existing results. Further, the output produced by TOSP is in phonetic English that can directly be transliterated into Telugu text and edited. Software for such transliteration is available freely. The phonetic English representation can be also used to develop a Telugu text-to-speech system. Work is in progress in this regard. This aspect has also not been considered in existing works. The ideas presented in this paper may also be useful for the development of OCR systems for other Indian languages.

Appendix

Table A Confusion table.



APPENDIX

Confusion Table

S. No.	Element 1 of Confusion Set		Other Element(s) of Confusion Set	
	Phonetic English	Telugu symbol	Telugu symbol	Phonetic English
1	/pa/	ప	స	/sa/
2	/va/	వ	న	/na/
3	/gha/	ఘు	సు	/su/
4	/ma/	ము	ను	/nu/
5	/ra/	ర	ల	/la/
6	/la/	ల	టు	/Ta/
7	/lu/ /lU/	లు లూ	టు టూ	/Ta/ /TA/
8	/cha/	చ	వ	/va/
9	/vA/	వా	హా	/ha/
10	/da/	ద	ఉ డ	/u/ /Da/
11	/ri/	రి	ి	/imatra/
12	/lu/	లు	యు	/yi/

References

1. Nagy G (2000) Twenty years of document image analysis in PAMI. IEEE T Pattern Anal 22(1):38–63
2. Mori S, Suen CY, Yamamoto K (1992) Historical review of OCR research and development. P IEEE 80(7):1029–1058
3. Govindan VK, Shivaprasad AP (1990) Character recognition: a review. Pattern Recogn 23(7):671–683
4. Bansal V, Sinha RMK (2001) A survey of OCR in Indian languages and a Devanagari OCR scheme. In: Proceedings of the symposium on translation support systems (STRANS-2001), Kanpur, India, February 2001
5. Chaudhuri BB, Pal U (1998) A complete printed Bangla OCR system. Pattern Recogn, 31:531–549
6. Nagabhushan P, Radhika A (1997) Improved region decomposition method for the recognition of non-uniform sized characters. In: Proceedings of the 1st international conference on cognitive science , Seoul, Korea, August 1997 1:36–42
7. Anna Durai S et al (1995) Tamil character recognition using multilayer neural network. In: Proceedings of the Indian conference on pattern recognition, image processing and computer vision, Kharagpur, India, December 1995, pp 155–160
8. Bishnu A, Chaudhuri B (1999) Segmentation of Bangla handwritten text into characters by recursive contour following. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 402–405
9. Pal U, Chaudhuri B (1999) Script line separation from Indian multi-script documents. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 406–409
10. Bansal V, Sinha R (1999) On how to describe shapes of Devanagari characters and use them for recognition. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 410–413
11. Anatani S, Agnihotri L (1999) Gujarati character recognition. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 418–421
12. Sundaresan C, Keerthi S (1999) A study of representation for pen based handwriting recognition of Tamil characters. In: Proceedings of the 5th international conference on document analysis and recognition (ICDAR'99), Bangalore, India, September 1999, pp 422 – 425.
13. Sukhaswami MB, Seetharamulu P, Pujari AK (1995) Recognition of Telugu characters using neural networks. Int J Neural Syst, September, 1995, 6(3):317–357
14. Negi A, Bhagvati C, Krishna B (2001) An OCR system for Telugu. In: Proceedings of the international conference on document analysis and recognition (ICDAR 2001), Seattle, Washington, September 2001
15. Casey RG, Lecolinet E (1996) A survey of methods and strategies in character segmentation. IEEE T Pattern Anal 18:690 – 706
16. Pavilidis T, Zhou J (1992) Page segmentation and classification. Computer Vision Graph 54:484–496
17. Akiyama T, Hagita N (1990) Automatic entry system for printed documents. Pattern Recogn 23:1141–1154
18. Le DS, Thoma GR, Wechsler H (1994) Automatic page orientation and skew angle detection for binary document images. Pattern Recogn 27:1325–1344
19. Sonka M, Hlavac V, Boyle R (1998) Image processing, analysis, and machine vision, 2nd edn. PWS, New York
20. Yan H (1993) Skew detection of document images using interline cross-correlation. CVGIP–Graph Model Im 55:538–543
21. Srikanthan G, Lam SW, Srihari SN (1996) Gradient-based contour encoding for character recognition. Pattern Recogn 29(7):1147–1160
22. Fausett L (1994) Fundamentals of neural networks. Prentice Hall, Englewood Cliffs, New Jersey
23. Vasantha Lakshmi C (2003) PhD thesis (unpublished), Dayalbagh Educational Institute, Agra, India