CHAPTER-4

4 Character Segmentation Techniques for Off-Line Cursive Handwritten Words

4.1 Introduction

A character is the smallest unit of any language script and the segmentation of characters is the most crucial step for any OCR (Optical Character Recognition) System. This system translates scanned or printed image of the document into a text document that can be edited. The selection of segmentation algorithm being used is the key factor in deciding the accuracy of the OCR system. If there is a good segmentation of characters, the recognition accuracy will also be high. Segmentation of words into characters becomes very difficult due to the cursive and unconstrained nature of the handwritten script.

Character segmentation and recognition has been an active field of research for many years. It still remains an open problem in the field of pattern recognition and image processing. There are mainly three phases of a character recognition system namely preprocessing, segmentation and recognition. Preprocessing aims at eliminating the variability that is inherent in printed words. The preprocessing techniques such as background noise removal, scaling, thinning skew removal etc. have been employed by various researchers in an attempt to increase the performance of the segmentation and recognition process. The role of segmentation is to find correct letter boundaries. Segmentation precedes character recognition, which means that the output of segmentation becomes the input to the character recognition module. Segmentation of off-line cursive words into characters is one of the most difficult and important process in handwriting recognition as it directly affects the result of recognition process (Broumandnia et al., 2008; Camstra, 2007; Fujisawa, 2010; Kherallah et al., 2008).

48
4.2 Challenges in the Proposed Segmentation Techniques

Off-line handwritten word segmentation is a subject of much attention due to the presence of many difficulties such as:

- Presence of skew and slant in the handwriting.
- Cursive nature of handwriting i.e. two or more characters in a word can be written connected to each other.
- Characters can have more than one shape according to their position inside the word image.
- Some characters (e.g. ‘u’ and ‘v’) in a handwritten word image can have similar contours.
- Some characters can give the illusion of presence of two similar characters e.g. ‘w’ can be segmented into two ‘v’ and ‘v’ characters.
- Two consecutive handwritten characters ‘i’ and ‘i’ may not get segmented and give the illusion of presence of character ‘u’.

This chapter proposes two techniques for segmentation and extraction of individual characters from the machine printed or handwritten word images.

4.3 Untouched Character Detection and Extraction from Off-Line Printed Text and Images

A robust segmentation technique is proposed for segmentation of untouched characters from the machine printed or handwritten words of varying length written on a noisy background having some images etc. This segmentation approach is based on connected component analysis of the foreground components (objects) in the binary image of the machine printed or handwritten words. Some images may also be there along with the words. These words are of different lengths (number of characters) and are printed by various cursive fonts of different sizes. Excellent results are achieved which reveals the robustness of the proposed untouched character detection and extraction technique.
4.3.1 Proposed Character Extraction Technique

Segmentation module in handwriting recognition plays a crucial role for successful performance of the overall recognition system. However, it is very difficult to find precise character boundaries without knowledge about characters. So to avoid the error prone process, this segmentation approach that is based on selecting the foreground objects or regions in the word image or text that meet the criterion of having area greater than some threshold value has been proposed as one of the segmentation strategies. The objects whose area is greater than 25 and less than 3000 are assumed to be the valid character objects. The objects of area less than 25 are assumed to be some noise and the objects having area greater than 3000 are assumed to be some picture or logo etc. that may be present in the scanned image of the document. Various steps followed in the proposed character detection and extraction process can be described as follows:

Step 1. The input scanned document image (having logo/ images and handwritten/ typed words) to be segmented into characters is obtained.

Step 2. After preprocessing i.e. removing the background noise etc., word image is converted into binary form.

Step 3. The image is inverted and all the connected components in binary image are computed using MATLAB's "bwlabel" function. The output of "bwlabel" is a label matrix.

Step 4. All foreground connected components (objects) are extracted by MATLAB's "regionprops" method.

Step 5. For each object in the scanned word image, repeat steps 5 to 7.

Step 6. If area of an object is greater than 25 and less than 3000, it is a valid connected component (object) and a smallest rectangular region with minimum area containing the complete connected component (object) is drawn on each object.

Step 7. Each sub-image containing a single object within a smallest rectangular region is inverted and displayed as segmented character image.
4.3.2 Implementation and Methodology

The word images of computer printed or handwritten words have been acquired through a scanner or a digital camera. These word images are written with different fonts of different size having different colors on the noisy background of different colors. The input word images are saved in JPEG or BMP formats for further processing. Some word image samples from our collected dataset are shown in Fig.4.1.

The word images to be segmented into character are on a noisy background. So, it is necessary to remove the background noise to improve the quality of the word image before further processing. The contrast adjustment is also done to overcome the problem raised due to the use of pens of different colors in case of handwritten words and fonts of different colors in case of the machine printed words. Fig.4.2 shows the original text image containing a logo and printed words and Fig.4.3 shows the resulting image after background noise removal and binarization using the grayscale intensity threshold.
The segmentation technique has been applied on the image as obtained in Fig.4.3. The input image is traced vertically from the upper left corner and all the connected components are identified based on their foreground area. All the valid connected components has been extracted and enclosed in a rectangular region with smallest possible area. All the rectangular regions enclosing segmented character images are obtained as shown in Fig.4.4.
It can be observed from these segmented character images that they have been cropped very well. After various preprocessing techniques, these character images can be used for training a neural network classifier if the recognition also has to be performed after the segmentation process, as it is normally done in an optical character recognition system.

### 4.3.3 Experimental Results and Analysis

The proposed character extraction technique performed exceptionally well as shown in Fig.4.4. This technique is based on the extraction of the smallest bounding box that contains the whole connected component. A connected component (object) is valid if the area of its rectangular region (bounding box) lies in the minimum and maximum range as specified in the algorithm of this technique. Due to this constraint, the dots (.) in characters “i” and “j” could not be extracted because the area of the bounding boxes containing the dot (.) is less than 25 as shown in Fig.4.6.
If the minimum area requirement for a bounding box containing a valid connected component is further decreased to become less than 25, the dots (.) in characters like “i” and “j” can be extracted but the foreground noise may also be extracted from the scanned word image. The criterion of minimum and maximum area requirement for deciding valid bounding box can be changed as per the requirement. If the word image is noise free, the minimum area criterion can be set to a value much lower than 25.

Further, it has been observed that this technique under-performed when the machine printed or handwritten characters / digits in the scanned word image are not in the form of a single connected component as shown in Fig.4.7. Here the last character’s image (image of ‘5’) is divided into two connected components as shown in Fig.4.8 where each component is a valid connected component (object) as both the components qualified the criterion of having the bounding box with a minimum specified area.
4.3.4 Conclusion

Very promising results are achieved by using this technique to extract character images. Although this technique is having a few shortcomings in cases where a character image is not completely connected, yet it can be extensively used in a variety of applications, like, a car number plate recognition system etc. where the extracted character images are used to train and test a character recognition system.

4.4 Touched Character Segmentation from Off-Line Cursive Handwritten Words of Varying Length

Off-Line handwriting segmentation and recognition has been a challenging and exciting area of research for many years. The popularity of this field of research is mainly due to the unconstrained and cursive nature of human handwriting. The segmentation and recognition of such type of handwritten script is still an open problem and is an active area of research these days. The character recognition accuracy of an OCR system can be improved remarkably if the characters within a word are correctly isolated. Hence, segmentation is the most crucial step in the off-line cursive handwritten script recognition process. Good segmentation results are always welcome.
4.4.1 Proposed Technique and Methodologies

The selection of a segmentation technique depends on the nature of the script to be segmented. The proposed segmentation technique is proposed for segmentation of touched characters from the handwritten words of varying length written on a noisy background. This new segmentation approach is basically a vertical dissection based segmentation technique in which the segmentation points are located after thinning the word image to get the stroke width of a single pixel. The knowledge of shape and geometry of English characters is used in the segmentation algorithm to detect ligatures. The proposed segmentation approach is tested on a local database and high segmentation accuracy is found to be achieved.

4.4.2 Preparation of Handwritten Words Local Database

For conducting the segmentation experiment by the proposed segmentation technique, handwriting samples from 10 different people (age 15-50 years) has been gathered. Some of these samples are written on white paper and others on a colored or a noisy background. Exactly 200 words have been selected randomly from these handwriting samples containing all shapes of English characters written by those persons. Some of the word image samples from the collected database are shown in Fig.4.9.

![Fig. 4.9 Handwritten Word Image Samples having Touched Characters](image-url)
4.4.3 Handwritten Word Image Acquisition

In image acquisition, the word images are acquired through a scanner or a digital camera. The input word images are saved in JPEG or BMP formats for further processing. Three such handwritten word images from the local database are shown in Fig. 4.10.

![Input Scanned Handwritten Word Image](image)

Fig. 4.10 Input Scanned Handwritten Word Image

4.4.4 Word Image Preprocessing

The aim of preprocessing is to eliminate the inconsistency that is inherent in cursive handwritten words. The handwriting samples may be written on a noisy or colored background and also the quality of the word images may be degraded due to the noise that is introduced in the process of scanning or capturing the word images. It is necessary to remove the background noise to improve the quality of the word images for the segmentation experiment. Preprocessing of the word images is of main concern so that the segmentation of the characters from the word images may be carried out correctly. After the segmentation process, it is also very important to deal with the preprocessing of each segmented character to improve accuracy of the recognition process.

The various preprocessing techniques that have been employed in an attempt to increase the performance of the segmentation process are as follows:

4.4.4.1 Thresholding

In this phase of preprocessing, the RGB images in BMP format shown in Fig. 4.10 are converted to grayscale format by retaining the illumination while eliminating the hue and saturation as shown in Fig. 4.11. This preprocessing step is necessary so as to overcome the problems that may arise due to the use of pens of different colors and different intensities on various noisy and colored backgrounds.
These grayscale images are then converted in a binary matrix format. The resultant binary images have values of 0 each for all the foreground black pixels and 1 each for all the background white pixels. The threshold parameter is so chosen that some negligible information of the characters is lost. The resulting word images after background noise removal and binarization using gray scale intensity threshold are shown in Fig. 4.12. It is advantageous to store these handwritten word images in this bi-level image format because it is easy to handle only two levels of colors. Also, these images take less storage space and are computationally less expensive. Hence, further processing becomes fast.

4.4.4.2 Thinning and Skeletonization

Skeletonization is a process in which the foreground region in a binary image is reduced to a skeletal remnant. During this process, the connectivity of the original region is preserved while removing a maximum number of original foreground pixels. Thinning is an image morphological operation in which selected foreground pixels are removed by eroding an image until it becomes one-pixel wide. It produces a skeleton of the object present in the image and makes it easier to recognize the object such as character. Thinning process is usually applied to a binary image and the output is also another binary image. The process of thinning erodes an object over and again, without breaking it, until the width remains to one-pixel wide.

A large amount of variability may be present among the handwritten words because writers can use different type of pens of unequal stroke width while giving their handwriting samples. The thickness that varied from one word to another must be
uniform. The thinning process delivers all the words used in the proposed experiment, a uniform stroke width of one-pixel.

Although thinning the word image is disadvantageous rather than beneficial because sometimes a huge amount of important information is lost e.g. some filled holes in the word images may get disappeared after thinning the word images. However, it is necessary to make the stroke width uniform for every handwritten word images so that the proposed segmentation technique can be employed properly. A simple thinning algorithm is applied to the handwritten word images so as to make the thickness of each word to one-pixel wide. Three such handwritten word images after thinning are shown in Fig.4.13.

![Called buffalo image](image)

**Fig. 4.13 Word Image after Thinning**

### 4.4.4.3 Noise Removal

Noise (small dots or foreground components) may be introduced easily into an image while scanning the handwritten word image during image acquisition. It is very necessary to eliminate the noise from the word images so as to make the word images fit for further processing. MATLAB's `bwareaopen` method is used to morphologically open the binary image by removing small objects that have less than a particular number (user specified) of pixels and producing another binary image. The small noise dots are removed by using `bwareaopen` but some small portions of the characters e.g. dot '.' of character 'i, j' etc. are also lost. The methods `bwlabel` and `regionprops` of MATLAB are used to highlight the pixels that are removed as shown in Fig.4.14.

![Called buffalo image](image)

**Fig. 4.14 Noise Detection in Word Images**
A logical AND operation of the dilated characters with the pixels removed by 'bwareaopen' is performed. The portions of the characters pixels which are very near to the character image are put back. The resultant image without noise dots while retaining the portions of the characters is shown in Fig.4.15.

Fig. 4.15 Word Image after Noise Removal

4.4.4.4 Cropping of Handwritten Word Images

Image cropping is a process in which the extra space around the handwritten word image is removed. The outcome of ‘imcrop’ is a rectangular region with minimum area but containing the complete word image. The final word images after cropping operation are shown in Fig.4.16.

Fig. 4.16 Cropped Word Images

4.4.5 Segmentation Technique

Many segmentation techniques have been developed by the researchers in the recent years. These techniques are basically script dependent and may not work well if applied for segmentation of words written in any other script. For example, the technique developed for segmenting touched characters in Roman script may not work well to segment touched characters of a word written in Arabic or Chinese script.

4.4.5.1 Overview

There are two types of characters in English language. First type of characters are called "Closed Characters" and contain a loop or a semi-loop such as ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘g’, ‘o’, ‘p’, ‘s’ etc. Second type of characters are termed as "Open Characters" and are without a loop or a semi-loop e.g. ‘u’, ‘v’, ‘w’, ‘m’, ‘n’, ‘i’ etc. In case of open
Chapter-4  

Character Segmentation Techniques...

characters, it is very difficult to differentiate between ligatures and characters because of the cursive nature of handwriting. In case of cursive handwritten words, a ligature is a link (small foreground component) which is present between two successive characters to join them. Two consecutive ‘i’ characters may give an illusion of the presence of a character ‘u’ and vice versa. Two consecutive characters ‘n’ and ‘i’ may look like ‘m’. Also, handwritten character ‘w’ may look like the presence of two consecutive characters ‘i’ and ‘v’. To overcome such type of challenges in the domain of cursive handwriting segmentation and recognition, a new segmentation approach is developed which is based on the analysis of the character’s geometric features, such as, the shape of the character to identify the characters and the ligatures.

4.4.5.2 Methodology

After the preprocessing of the input handwritten word image, the height and width of the word image is calculated for the analysis of the ligatures in an accurate manner (Rehman et al., 2009). The word image is scanned vertically, from top to bottom, column wise and the number of foreground pixels in the inverted word image are counted in each column. The positions of all these columns are saved for which the sum of foreground black pixels is either 0 or 1. All these identified columns are termed as PSC (Potential Segmentation Columns).

4.4.5.3 The Problem of Over-segmentation

Many consecutive PSC are present in various groups in the whole word image where sum of foreground pixels are 0 or 1. This situation can be termed as over-segmentation. The over-segmentation problem is occurred in three cases. First, when the two consecutive characters in the word image are not touching each other and the sum of foreground pixels of the columns in this area are 0. Second, when the two consecutive characters in the word image are connected by a ligature and the sum of foreground pixels in these columns crossing this ligature are 1. Third, when the characters are Open Characters like ‘u’, ‘w’, ‘m’, ‘n’ etc. without having any loop or semi-loop. A ligature is present within all of these Open characters. Due to the presence of this ligature-within-character, the characters of such type in the word image are still over-segmented and the sum of foreground pixels in these columns is also 1.
4.4.5.4 Solution of Over-Segmentation Problem

When there is a clear vertical space between two consecutive characters in a word image, the problem of over-segmentation is completely eliminated by taking average of all the PSC present in that area because the sum of foreground pixels for all these PSC is 0.

When there is a ligature between two consecutive characters or there is a ligature-within-character (open characters such as ‘w’, ‘m’ etc.), the over-segmentation is eliminated to a great extent by taking average of those PSC which are at a distance less than a particular value (threshold) and by merging them into a single SC (Segmentation Column). The threshold value is the minimum distance (along the width of the word image) between consecutive PSCs and is so chosen that its value must be less than the width of the thinnest possible character (e.g. ‘i’, ‘l’) in a word image. By experimenting several times, the value of threshold is set to a value 7. This means that all those PSCs which are separated by a distance of 7 pixels or less by another PSC will be merged to a single SC (Segmentation Column).

4.4.5.5 Implementation

The steps followed during the implementation of the proposed segmentation technique are mentioned below:

Step 1: In the first step, the input word image is preprocessed by using various preprocessing techniques such as thresholding, binarization, thinning, noise removal and cropping. The collective outcome of all these preprocessing technique are summarized and reproduced in Fig.4.17. This preprocessed word image is taken as input image to be segmented into characters as shown in Fig.4.18 (a).

Step 2: To minimize the computation complexity, the input word image is inverted for further processing. By complementing the input binary image, white pixels become the foreground pixels and the black pixels become the background pixels. Hence, it becomes easier to count the foreground white pixels represented by 1, in each column of the word image as shown in Fig.4.18 (b).
Chapter-4  Character Segmentation Techniques...

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(b)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(c)</td>
<td>Called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(d)</td>
<td>Called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(e)</td>
<td>Called</td>
<td>buffalo</td>
</tr>
</tbody>
</table>

Fig. 4.17 Word Image Preprocessing (a) Input Scanned Word Images; (b) Word Images after Gray Scale Intensity Threshold; (c) Word Images in Binary Format; (d) Word Images after Thinning; (e) Cropped Word Images after Noise Removal

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(b)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(c)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(d)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(e)</td>
<td>called</td>
<td>buffalo</td>
</tr>
<tr>
<td>(f)</td>
<td>called</td>
<td>buffalo</td>
</tr>
</tbody>
</table>

Fig. 4.18 Word Image Segmentation (a) Pre-processed Word Images; (b) Inverted Binary Images; (c) RGB Images; (d) Over-segmentation in Images; (e) Image after removing Over-segmentations; (f) Final Segmented Output Word Images
Step 3: This image is now converted from binary format to a RGB format as shown in Fig.4.18 (c). Now, it becomes computationally easier to display the PSC (Potential Segmentation Columns) in different color other than black and white.

Step 4: All PSCs over-segmenting the word image are printed in red color as shown in Fig.4.18 (d). It is clear from Fig.4.18 (d) that each column in the word image, for which the sum of foreground white pixels is 0 or 1, is a PSC and vertically cuts the word image.

Step 5: All PSCs, which are at a distance less than a threshold value (7 pixels) from each other, are merged into a single column termed as Segmentation Column (SC) as shown in Fig.4.18 (e).

Step 6: The final segmented word image is obtained by changing the black background of the image with a white background as shown in Fig.4.18 (f).

4.4.5.6 Result Analysis

For evaluation of the proposed segmentation approach, 200 handwritten word samples have been selected from the handwriting samples of 10 different writers. The performance of the proposed segmentation approach is judged on the basis of segmentation errors of the three types, namely, Number of Over-Segmentations, Number of Miss-Segmentations and Number of Bad-Segmentations and is shown in Table 4.1.

<table>
<thead>
<tr>
<th>Total of Handwritten Words</th>
<th>Number of Correctly Segmented Words (%)</th>
<th>Total Incorrectly Segmented Words (%)</th>
<th>Number of Words with various Segmentation Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>167 (83.5%)</td>
<td>33 (16.5%)</td>
<td>Over-Segmented: 14</td>
</tr>
</tbody>
</table>

Table 4.1 Segmentation Result of the Proposed Vertical Segmentation Approach
Out of 33 incorrectly segmented words, some words are over-segmented in one place as well as bad-segmented in some other place. While putting the results in Table 4.1, such types of words are counted in both of the error categories i.e. counted in over-segmented as well as bad-segmented. Similarly, in some words, the correct segmentation point is missed and shifted to some other place resulting in bad-segmentation. Some word images which are over-segmented or miss-segmented or bad-segmented are shown in Fig.4.19.

![Word Images Showing All Type of Segmentation Errors](image)

Fig. 4.19 Word Images Showing All Type of Segmentation Errors

It is very difficult to compare the segmentation results achieved by the proposed approach with the segmentation results of some other researchers because different researchers used different databases of handwritten words and reported the segmentation results under various constraints, such as, some researchers assumed the absence of noise, some researchers collected the handwriting samples from different number of writers and so on. Although some researchers (Marti and Bunke, 2002; Hull, 1994) used various benchmark databases e.g. CEDAR or IAM for their experiment but they used different number of words from the benchmark database. As the character segmentation in word images is done before the character recognition phase, most of the researchers mentioned only the recognition results and not the segmentation results.

4.4.6 Conclusion

A new vertical segmentation technique is developed to enhance the over-segmentation of the handwritten word image by thinning the word image to a single pixel width. The objective of the proposed approach is to over-segment the handwritten word image sufficient number of times to ensure that all possible character boundaries have
been dissected. Another technique is also developed to merge more than one successive segmentation points present between any two characters into a single segmentation point to enhance the segmentation performance.

The proposed segmentation approach guaranteed correct segmentation when characters in a word image are not touching each other. This approach also delivered excellent results in case of segmentation of ligatures present between consecutive closed characters. It also minimized the problem of over-segmentation that appeared during segmentation of open characters. The presence of over-segmentation in case of open characters is due to the presence of ligatures-within-characters. In case of open characters, ligature-within-characters sometimes appeared as they are ligature between two consecutive characters and are over-segmented by the proposed technique. Although the obtained segmentation accuracy of 83.5% is very favorably, yet there is always a scope of improvement.