A Robust Technique for Handwritten Words Segmentation into Individual Characters

Amit Choudhary and Vinod Kumar

Abstract Segmentation of individual characters from a scanned word image is the most critical step of a typical optical character recognition (OCR) system. A robust segmentation algorithm is proposed in this paper. The word images are segmented into individual characters after skew angle correction and the thinning process, to get the single pixel stroke width. Ligatures of the touching characters are detected by keeping in view the geometrical shape of the English alphabets. The proposed vertical segmentation technique is used to cut individual characters from the handwritten cursive words. The proposed algorithm delivers excellent segmentation accuracy when tested on a local database.

Keywords Segmentation · OCR · Word recognition · Preprocessing

1 Introduction and Historical Background

Nowadays researchers are trying to introduce human brain's intelligence and capability into a computer system to recognize the information written on paper. In an OCR system, good character recognition accuracy can be achieved if the characters in the handwritten script are well segmented. Many researchers had already achieved very good segmentation results [1], but the scope of improvement is always there and superior segmentation results are always awaited. Technological advancements during the last 40 years in the area of document and character recognition are presented [2]. A new technique to recognize handwritten as well as typewritten English text is presented [3]. It does not require the thinning process,

A. Choudhary (✉)
Maharaja Surajmal Institute (GGSIP University), New Delhi, India
e-mail: amit.choudhary69@gmail.com

V. Kumar
Delhi Technological University, New Delhi, India
e-mail: vinodkumar@dce.edu

S. S. Agrawal et al. (eds.), *Speech and Language Processing for Human-Machine Communications*, Advances in Intelligent Systems and Computing 664,
https://doi.org/10.1007/978-981-10-6626-9_11

and it delivered 80% accuracy. The slant and skew correction was not performed during preprocessing the word images by the authors [4].

2 Preprocessing Techniques and Database Preparation

To demonstrate the proposed segmentation algorithm, handwritten word samples written on colored or noisy background have been collected from 10 different persons aged between 15 and 40 years. From the collection of handwriting samples, we have selected 400 handwritten words randomly to perform the proposed experiment. Figure 1 displays few samples from the local handwritten word images database.

2.1 Image Acquisition

In image acquisition, a digital photocamera or a scanner is generally used to capture the handwritten word images, and these images are saved in .bmp or .jpg file format for preprocessing. Figure 2 shows two such image samples from the database.

2.2 Preprocessing

The main objective behind preprocessing is to remove the invariabilities existing in word images. While scanning the handwritten word images, the quality may be ruined as the noise is introduced due to dust or due to colored background.

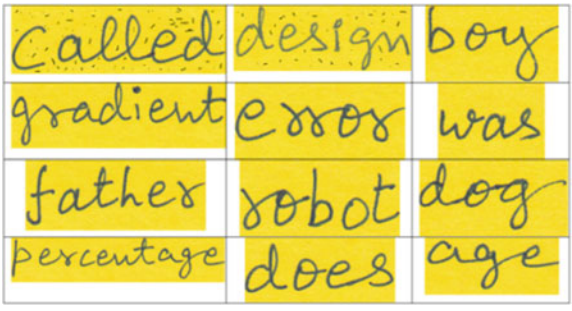

Fig. 1 Handwritten word image samples

Fig. 2 Input scanned handwritten word images

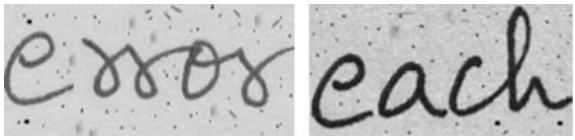

Fig. 3 Handwritten word images in grayscale format

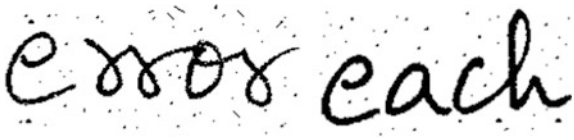

Fig. 4 Images in binary form

Fig. 5 Noise detection and word images after noise removal

Thresholding and Binarization.
Thresholding is necessary so that the problems can be avoided due to usage of pen of different colored ink on colored and noisy surfaces. Figure 3 shows two such grayscale images obtained after thresholding.

The grayscale images are then transformed to the binary matrix form in which a 0 represents a black pixel in the foreground and a 1 represents a white pixel in the background. Figure 4 displays such binary images.

Image De-noising and Skeletonization.
In this preprocessing stage, the noise (small foreground components and dots) induced in the image scanning process is optimally eliminated. Only the noise dots and other foreground components have been removed in this step while retaining the character components. Noise-free images are shown in Fig. 5.

As the pens of different stroke width can be used by the different writers, a lot of unevenness may exist. After the thinning process, all the handwritten word images

Fig. 6 Word image after thinning

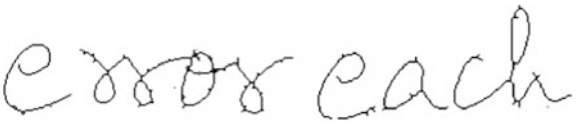

Fig. 7 Cropped image

were made to have stroke width of 1 pixel each. Such image samples following the skeletonization process are displayed in Fig. 6.

Cropping and De-skewing.
The images after de-noising are cropped to remove the extra space available around the rectangular region enclosing the handwritten noise-free word image. The skew correction is also performed on the handwritten word images. Figure 7 shows such cropped sample images after skew correction which was not performed earlier [5].

3 Proposed Segmentation Technique

The projected segmentation algorithm is designed for segmenting touching character present in the handwritten words of English language and may not work well if applied to some other languages such as Arabic or Chinese.

3.1 Overview

English language has closed as well as open characters. Closed characters have a semi-loop or a loop such as ‘g’, ‘o’, ‘p’, ‘s’, ‘a’, ‘b’, ‘c’, ‘d’, ‘e’. Open characters do not have any semi-loop or loop, e.g., ‘u’, ‘v’, ‘w’, ‘m’, ‘n’. Discriminating between ligatures and character segment is very hard in open characters. Ligature may be defined as a link between two or more consecutive characters used to join them. In written English language words, two ‘i’ characters side by side may look like ‘u’ and vice versa. Successive ‘n’ and ‘i’ may appear as ‘m’. Character ‘w’ may give the illusion of presence of two characters ‘i’ and ‘u’.

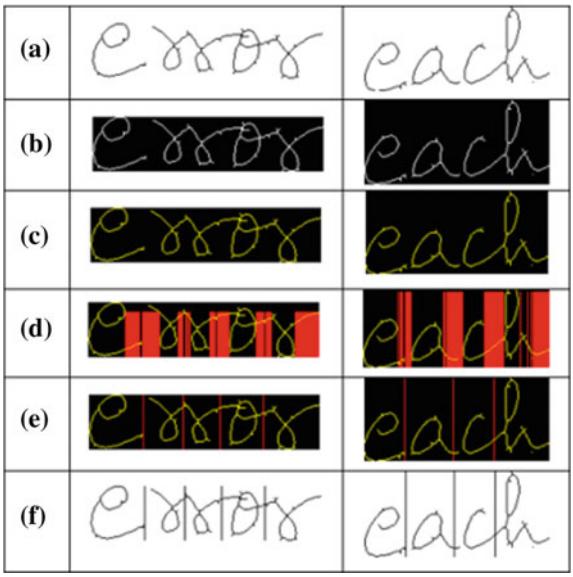

Fig. 8 **a** Images after preprocessing, **b** binary inverted images, **c** images in RGB, **d** over-segmentation, **e** solving over-segmentation problem, **f** output handwritten word images after segmentation

3.2 Methodology

After inverting the handwritten word image, the number of white pixels is counted in each column scanning the image from top to bottom. The columns having 1 or 0 as the count of white foreground pixels are termed as candidate segmentation columns (CSCs), and their positions are noted. Figure 8d shows all such acknowledged columns.

3.3 Problem of Over-Segmentation

Several successive CSCs have been grouped together at various places in the handwritten word image resulting in a situation called 'over-segmentation' and is displayed in Fig. 8d. There are three situations in which this problem of over-segmentation occurs. First, when there is a gap between two successive characters and for each column that lay in this gap, the count of the number of white pixels is 0. Second, when there is a ligature between two characters and the sum of white pixels is 1 for all columns through such ligatures in the whole word image. Finally, when there exist characters such as 'u', 'm', 'n', 'w', which contains loop or semi-loop and the count of all the white foreground pixels for each column which crosses the ligatures-within-characters is also 1. Hence, such types of characters are over-segmented.

3.4 Solving the Over-Segmentation Problem

In the situations, when there is a gap between successive characters, each and every CSC in this gap will have 0 white pixels. By taking mean of all CSCs lying in that gap and merging all the CSCs to a sole column, over-segmentation problem has been solved. In other situations, when ligature-within-character is present (e.g., characters 'u', 'v', 'm', 'w') or a ligature connecting two successive characters, a mean of all those CSCs in a group is calculated which are within a distance below threshold range, and these CSCs are merged to a sole segmentation column.

In horizontal direction, the least gap between successive CSCs is called threshold range, and its value is selected in such a manner that it should be less than the thinnest available character's width such as 'l', 'i'. By repetitive experiments performed many times, threshold's value is selected as '8'. Hence, all the CSCs that are within the 8 pixels range distance from another CSC would be merged into a single segmentation column.

4 Implementing the Proposed Technique

The handwritten word images obtained after various preprocessing steps as shown in Fig. 8a are complemented and taken as input to the segmentation algorithm. By inverting the input black and white images, black pixels form the background and white pixels form the foreground as displayed in Fig. 8b. White pixels have been represented by 1, and it is now easy to count the number of white pixels in each and every vertical column of the binary handwritten word images. Now, this binary image is converted to the RGB color arrangement and is displayed in Fig. 8c. It is convenient if we show CSCs in any color (say red) other than black and white as shown in Fig. 8d. It can be clearly seen that every column, whose total count of white pixels is zero or one, vertically dissects the word image and is termed as a CSC. All CSCs lying within the threshold range of 8 pixels from one another, are fused together to draw a single column representing that particular group of CSCs and is called Segmentation Column and is indicated by the Fig. 8e. Now, the image is then inverted again to get the white background and black foreground for the final segmented handwritten word image as displayed in Fig. 8f.

5 Discussion of Results

A random selection of 400 word images contributed by 10 different writers was used in this experiment. To evaluate the proposed segmentation technique, three types of errors were considered, i.e., number of bad-segmented, over-segmented, and miss-segmented words out of a total of 400 words used in the experiment.

Table 1 Segmentation results

Count of word images used in the experiment	Correct segmented word images (percentage)	Incorrect segmented word images (percentage)	Count of word images and type of error		
			Over-segmented images	Miss-segmented images	Bad-segmented images
400	346 (86.5%)	54 (13.5%)	19	11	40

Table 1 shows that 346 words were segmented correctly, and 54 words were segmented incorrectly. Some incorrectly segmented words were bad-segmented as well as over-segmented and are counted in each type of error category while displaying the results in Table 1. This is why $19 + 11 + 40 \neq 54$.

Comparing the results attained by the proposed segmentation technique with the results of other segmentation techniques developed by other researchers in the literature is not so easy because different researchers presented their segmentation results under different constraints and also they used different types of databases. Some researchers made the assumption that the word images are noise free while some researchers gathered the word image samples from different number of contributors. Although, some authors [5, 6] used popular benchmark databases such as IAM and CEDAR, but they selected different number of handwritten word images from these databases and they even rejected some particular complicated word images from the database as per their personal choices.

6 Conclusion and Future Directions

The proposed technique ensures to dissect each and every possible character boundary by over-segmenting the sample word image enough number of times. Another strategy is also adopted that detects groups of many candidate segmentation points that are lying between any two successive characters and then clubs them into a single segmentation point. Whenever a word image contains untouched characters, accurate segmentation is guaranteed by the proposed technique. It performs very well to dissect ligatures connecting two successive closed characters. This technique sometimes over-segments the open characters because the ligature-within-characters look like ligature connecting two characters. The segmentation accuracy of 86.5% delivered by the proposed segmentation technique is quiet excellent, but the scope of improvement is always there. In future work, there is a need to improve some of the preprocessing techniques, e.g., thinning.

References

1. Tan, J., et al.: A new handwritten character segmentation method based on nonlinear clustering. Neurocomputing **89**, 213–219 (2012)
2. Fujisawa, H.: Forty years of research in character and document recognition-an industrial perspective. Pattern Recogn. **41**, 2435–2446 (2008)
3. Saeed, K., Albakoor, M.: Region growing based segmentation algorithm for typewritten and handwritten text recognition. Appl. Soft Comput. **9**, 608–617 (2009)
4. Choudhary, A., Rishi, R., Ahlawat, A.: A New character segmentation approach for off-line cursive handwritten words. Proc. Comput. Sci. **17**, 88–95 (2013)
5. Marti, U., Bunke, H.: The IAM database: an english sentence database for off-line handwriting recognition. Int. J. Doc. Anal. Recogn. **15**, 65–90 (2002)
6. Hull, J.J.: A database for handwritten text recognition. IEEE Trans. Pattern Anal. Mach. Intell. **16**, 550–554 (1994)