Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 14 (2014) 465 – 472

Procedia Technology

2nd International Conference on Innovations in Automation and Mechatronics Engineering, ICIAME 2014

Text-Based Image Segmentation Methodology

Gupta Mehul[a*], Patel Ankita[b], Dave Namrata[c], Goradia Rahul[d], and Saurin Sheth[e]

[a,b]*U.G. Students, G. H. Patel College of Engineering and Technology, Vallabh Vidyanagar 388120, Gujarat, India*
[c]*Assistant Professor, Computer Engineering Department, GCET College, Vallabh Vidyanagar 388120, Gujarat, India*
[d]*Assistant Professor, Electronics & Communication Engineering Department, GCET College, Vallabh Vidyanagar 388120, Gujarat, India*
[e]*Associate Professor, Mechatronics Engineering Department, GCET College, Vallabh Vidyanagar 388120, Gujarat, India*

Abstract

In computer vision, segmentation is the process of partitioning a digital image into multiple segments (sets of pixels). Image segmentation is thus inevitable. Segmentation used for text-based images aim in retrieval of specific information from the entire image. This information can be a line or a word or even a character. This paper proposes various methodologies to segment a text based image at various levels of segmentation. This material serves as a guide and update for readers working on the text based segmentation area of Computer Vision. First, the need for segmentation is justified in the context of text based information retrieval. Then, the various factors affecting the segmentation process are discussed. Followed by the levels of text segmentation are explored. Finally, the available techniques with their superiorities and weaknesses are reviewed, along with directions for quick referral are suggested. Special attention is given to the handwriting recognition since this area requires more advanced techniques for efficient information extraction and to reach the ultimate goal of machine simulation of human reading.

Peer-review under responsibility of the Organizing Committee of ICIAME 2014.

Keywords: Handwritten text and printed text comparison, levels of segmentation, segmentation methodologies, text document image analysis.

1. Introduction

Text extraction is an important phase in document image analysis and it does not have a universal accepted solution [1] [2]. In order to segment text from a page document it is necessary to detect all the possible manuscript text regions. Text Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. It implies a labelling process which assigns the same label to spatially align units i.e. pixel, connected components or characteristic points such that a group of pixels with the similar label share specific visual features. The result of image segmentation is a set of segments that collectively cover the entire image, or a set of contours

*Corresponding author. Tel:+91 9726075579; E-mail:mehulgupta29@gmail.com

Peer-review under responsibility of the Organizing Committee of ICIAME 2014.
doi:10.1016/j.protcy.2014.08.059

extracted from the image (edge detection). Each of the pixels in a region is similar with respect to some characteristic properties, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristic(s). [3]

Prior performing segmentation, we need to bring the image in a specific format to simplify the subsequent processing. The preprocessing [4] [5] [6] includes digitization, noise removal [7], binarization, normalization. The preprocessing stage yields a "clean" document in the sense that a sufficient amount of shape information, high compression, and low noise in a normalized image is obtained. The next stage in the process of document analysis is segmentation. Segmentation can then be implemented into its subcomponents [8] [9]. Segmentation is an important stage because the extent one can reach in separation of words, lines, or characters directly affects the recognition rate of the script [10].

Further, based on the obtained labels/ regions, text is divided into different logical areas, each one representing a predefined set of semantics [8]. An ideal situation would be to segment the image into regions which depicts a text line. After text line segmentation is finished, it provides the essential information for the consecutive documents image steps such as skew detection and correction, text feature extraction and character recognition. Hence, it is prerequisite for the further process of document image analysis. Although some text line detection techniques are successful in printed documents, processing of handwritten documents has remained a key problem in optical character recognition.

Also, the need of segmentation triumphs the possibility of reduction in complexity to implement an efficient system. Segmentation has applications in various domains, like machine vision, object detection, medical imaging [21], recognition tasks, et al. Content-based image retrieval (CBIR), is one of the application of computer vision techniques to the image retrieval problem, that is, the problem of searching for digital images in large databases on the basis of syntactical image features (like color, texture, shape) [22].

There are various factors that hinder the process of text based image segmentation [1] [20].

A few are as follows:

- *Image Quality*: The quality of the image is a significant factor for text segmentation. Presence of noise in the image results in degradation of accuracy and efficiency [24].
- *Handwritten or Printed Document*: Most text line segmentation methods are based on the assumptions that distance between neighboring text lines is precise as well as that text lines are equitably straight. However, these assumptions are not characterized for handwritten documents. In case of handwritten document, text image segmentation is a leading challenge. The prior, is the case of the printed text document. For such a document segmentation is an easy task, due to the symmetric nature of the document. The line, word and even character spacing is defined, which abolish the challenges as faced with handwritten documents.
- *Orientation of text content*: For handwritten document if the individual lines are not straight or if there is a presence of skew then the overall complexity for text extraction increases [6] [23].
- *Textured document*: Presence of texture, like images, patterns, et al. in the text document makes the task of segmentation multifaceted.
- *Type of Text*: Cursive text provides additional difficulty during character segmentation, due to the presence of ligatures.

2. Levels of text segmentation

Text image segmentation can be achieved at three levels [1] [6] [8] [11]. As we move at different levels of text segmentation hierarchy, we obtain specifically finer details. Using all the three levels is not compulsory. Segmentation at any of these levels directly depends on the nature of the application. More the details required for the image, the more is the level of segmentation. The various levels in the hierarchy are as shown in figure 1a.

2.1. Line segmentation

Line segmentation is the first and a primilary step for text based image segmentation. It includes horizontal scanning of the image, pixel-row by pixel-row from left to right and top to bottom [8] [10] [12] [13]. At each pixel the intensity is tested. Depending on the values of the pixels we group pixels into multiple regions from the entire

image. The different region indicates different content in the image file. Subsequently the desired content can be extracted. Due to inaccuracies in the scanning process and writing style, the writing may be slightly tilted or within

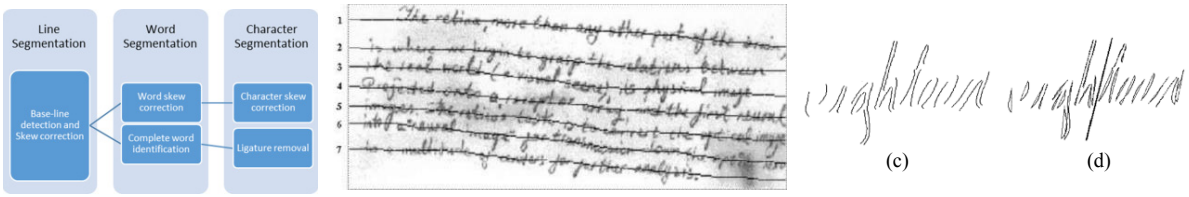


Fig. 1. (a) Levels of segmentation; (b) Baseline extraction using attractive and repulsive network; [10] (c) Slant angle estimation: Near vertical elements; [10] (d) Slant angle estimation: Average slant angle. [10]

the image. This can hurt the effectiveness of later algorithms and, therefore, should be detected and corrected. Additionally, some characters are distinguished according to the relative position with respect to the baseline (e.g., "9" and "g") [9]. Methods for baseline extraction include using the projected profile of the image [25], a form of nearest neighbours clustering [26], cross correlation method between the lines [27], and using the Hough transform [28]. In [29], an attractive repulsive NN is used for extracting the baseline of complicated handwriting in heavy noise (as shown in figure 1b). After skew detection, the character or word is translated to the origin, rotated, or stretched until the baseline is horizontal and retranslated back into the display screen space [23].

2.2. Word segmentation

Word segmentation is the next level of segmentation. It includes vertical scanning of the image, pixel-row by pixel-row from left to right and top to bottom [10] [16]. At each pixel the intensity is tested. Depending on the values of the pixels we group pixels into multiple regions from the entire image. The different region indicates different content in the image file. Subsequently the desired content can be extracted.

Figure 1c shows the slant angle estimation to perform skew correction for the extracted word in heavy noise. The skew correction can be performed by determining the angle and rotating the image in the opposite direction.

2.3. Character segmentation

Character segmentation is the final level for text based image segmentation. It is similar to in operations as word segmentation [10] [14] [15]. A few precautions should be followed while preforming character segmentation. Figure 2 shows one such problem. The segments as shown in figure 2c is not accurate, as "h" is extracted as "l" and "i". Such errors are undesirable. Another precaution is of ligatures. If the text image contains a cursive type font then while segmenting the ligature should be separated for better efficiency.

3. Segmentation Methodologies

In this section we discuss the various methodologies to segment a text document image. To achieve segmentation of a text based image depends greatly on the presence of guidelines in the document. Appearance of guidelines eliminates the possibility of skew. More over guides restricts the character size as a result of which the overall process of segmentation becomes plain sailing.

The methodologies can be thus evaluated on the basis of the following key factors:

Fig. 2. Segmentation using shortest path of a graph of gray level image (a) Segmentation intervals; (b) Segmentation paths; (c) Segments. [10]

- Appearance of the page: Appearance of the page indicates to the presence of guideline in the page. The presence of such guidelines eases the entire process.
- Level of Segmentation: Performing segmentation at higher levels requires additional advance methods for correct extraction.

The following are the techniques to perform segmentation of a text document image. Various segmentation algorithms have been proposed in [15]

3.1. Pixel counting approach

Reference [30] states this approach, the line separation procedure consists of scanning the image row by row. The row in the preceding line represents the pixel row and not the lines of the address, i.e. The entire image is scanned from left to right and top to bottom. Then the intensity of the pixel is tested for 0 or 1 (Here we consider a binarized image). In a binarized image, 0 represents black and 1 represents white. The algorithm would vary according to the image under consideration. Pixel counting approach is a simple technique to implement, but it cannot be used in situations when the text line in the document has a higher degree of skew, when the characters overlap, or when there is irregular spacing between the text lines. There are two ways to achieve line segmentation, first way can be used for a document without the guidelines, and second way can be used in the document with guidelines.

In the first way, the line separation is obtained by setting a threshold value for the number of white pixel rows between two address lines. This number of white pixel rows determine the space between two text lines. Two lines are separated if the number of white pixel rows between them is greater than the threshold value (If the image is binarized and complemented, then we consider a number of black pixels as threshold. Hence black pixels represents blank space between the text lines, whereas the white pixels would represent the actual text). Such a logic would be futile when letters such as ‘y’,’g’ etc. occur in the first line and letters like ‘f’,’d’, etc. occur in the second line without having white pixel rows in between. Due to such overlapping of the characters the pixel approach fails to provide accurate results. Such a bottleneck can be averted by designing the algorithm in such a way that it is tolerant to a certain minimum number of black pixels in a white row.

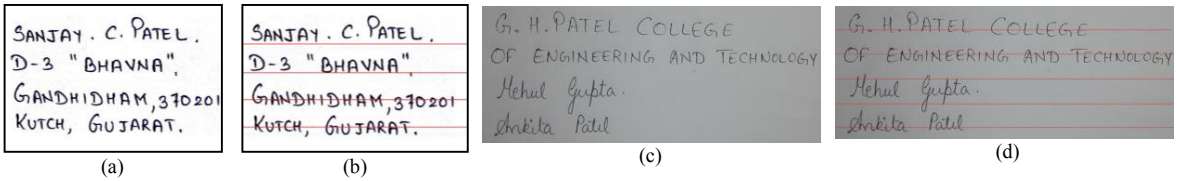
The second way is simple to implement. Due to the presence of guidelines the space between two lines is constant. We can use this information to perform line segmentation. The space between the two consecutive lines can be treated as a constant, using which the text image can be segmented at regular intervals. This method successfully address the problem of overlapping characters, as there is a visible demarcation between the two text lines. The problem arises when any the character extends the guideline boundary. In such case instead of getting an entire alphanumeric character, only a portion of it would be segmented.

Figure 3a and 3c are the original images. They are provided as input to the algorithm and fig 3b and 3d are obtained as outputs respectively. The region between the red lines represent the individual segments. The result of segmentation is unacceptable as the text in the segments contains only a portion of the original text line.

Higher level of segmentation can be achieved by minimizing changes in the algorithm logic. For Line segmentation, we perform horizontal cuts along the image length, for word and character segmentation, we have to perform vertical cuts along the width of the image.

3.2. Histogram Approach

Histogram approach is a method to automatically identify and segment the text line regions of a handwritten document [1] [8]. In the work of Marinai and Nesi [11], the projection curves are used to segment music sheets in order to extract the basic symbols and their positions. Manmatha and Rothfeder [31] used projection profiles in the



(a) (b) (c) (d)

Fig. 3. (a) (c) Original Text Based Image; (b) (d) Line Segmentation using Pixel Counting Approach.

horizontal direction to segment words of historical handwritten documents during the line segmentation stage. The feature extraction or binarization step is applied to the input image (figure 4a). Then, a Y histogram projection is obtained to detect the possible lines. Due to some noises, a text line separation is necessary. Once the false lines are found, they must be excluded. After that, the line region recovery step (figure 4j) is performed in order to recover some losses introduced by the preceding step. At this point (figure 4c) we have the coordinates of individual text lines, which can be extracted by cropping at the endpoints of the original digitized image [8].

Histogram method can very easily be extended to higher levels of segmentation. A Y histogram (figure 4b) is used to segment the text lines [1] [8] [13] [17], and an X histogram is used to segment words and characters. An X histogram projection that is applied to each line detected takes out possible words [8]. The points obtained are similar to those obtained from line segmentation. Each cut point reflects a rectangular region where the possibility of a text word/character is maximized. Using this rectangular coordinates, we can extract the words/characters from the digitized image.

3.2.1. Y Histogram Projection

Reference [1] [8] states that, once the preprocessing (binarization, noise removal, normalization) of the images is performed, the Y histogram projection of the whole image is obtained. The idea is to use a simple and fast method to correctly distinguish possible line segments in the handwritten text. In figure 4c it is clear that each text line corresponds to a peak in the histogram. The histogram represents the added pixels for each y value. So the empty spaces between the peaks represent possible regions between different text lines.

3.2.2. Text Line Separation

Reference [1] [19] states that, once all the potential lines are detected, a procedure to apply a threshold is performed to obtain a possible line separation in the text. This threshold is dynamically calculated and it is proportional to the average length of the lines in the text (Y histogram values). This procedure aims to remove the regions in the histogram that do not refer to the lines in the text, or the elimination of noises that confuses with the text lines. The choice of the parameter to be used as a threshold is intrinsic and is related to the information like the text. Such an approach restricts the algorithm, thereby utilizing minimum possible of heuristic techniques to determine the line separation points. Actually, this stage tries to identify the location of each text line. The separation of the possible text line regions using the histogram shows a deficit due to the upper and lower regions of some letters as shown in figure 4j.

3.2.3. False Line Exclusion

Reference [1] states this procedure as it tries to exclude the possible noises close to the text lines regions. Once the possible text line regions are separated by removing an offset from the histogram, we determine the average height of these regions to exclude false lines that might be detected. If the presence of noise is more than this region poses enough height it can be confused with a text line segment by the algorithm. The height of a line is obtained by taking the limit values of the corresponding region in the Y histogram and calculating the difference obtained by taking the limit values of the corresponding region in the Y histogram and calculating the difference between them.

The equation bellow provides the average height of the lines found in a page [1]:

$$\sum |Yinitial - Yfinal|/Np \quad (1)$$

Equation 1. Calculate average height of the line

Where, Yinitial is the y position where the text region begins, Yfinal is the y position where the text region ends and Np is the number of regions found in the page. The lines with height below a pre-determined threshold are removed. The value of this threshold is proportional to the average height of the text lines in the whole image.

Figure 4a, d are author's implementation of Histogram method and the result (from [1]) shown in fig 4g, i prove that the comparison is alike.

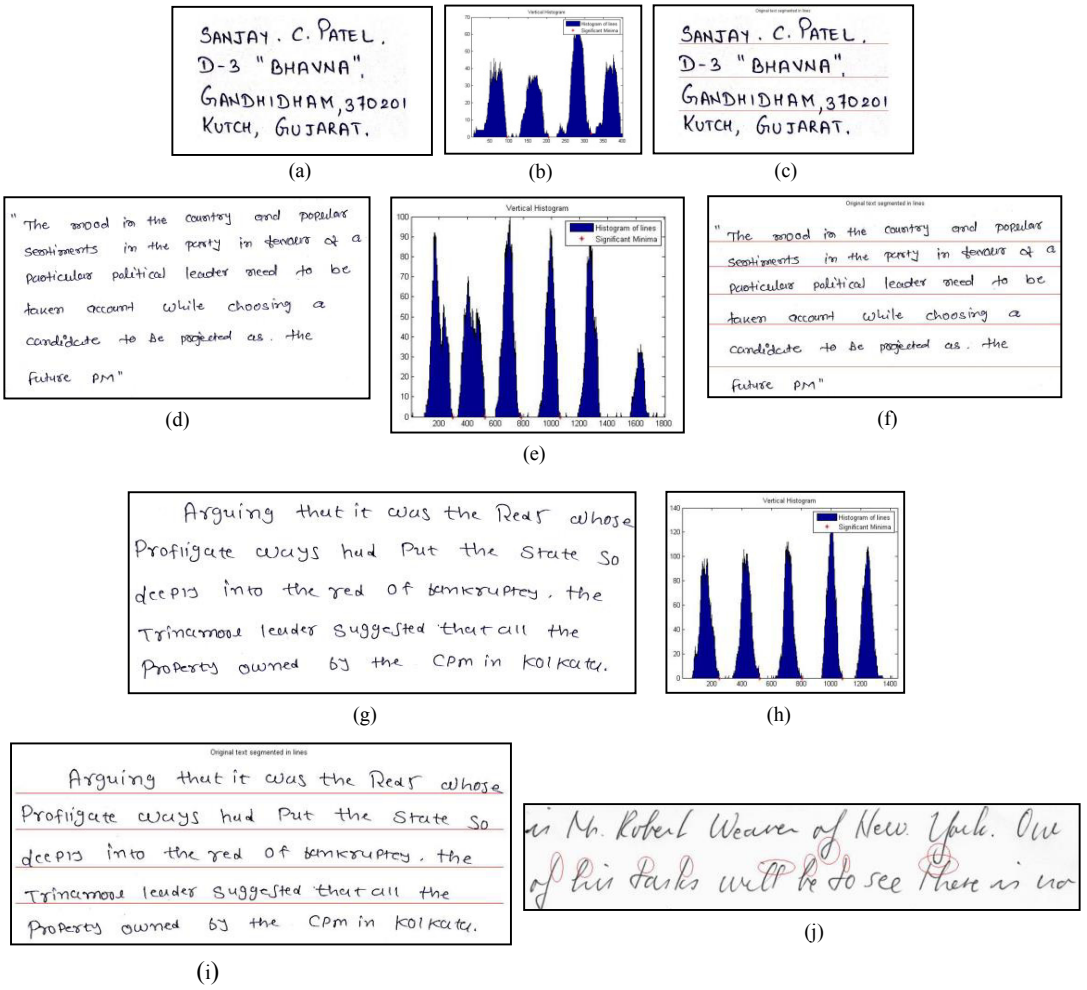


Fig. 4. (a) (d) (g) Original input Image; (b) (e) (h) Histogram; (c) (f) (i) Line Segmentation using Histogram Approach, (j) Region that provokes false lines [1]

3.2.4. Line Region Recovery

This procedure determines the average point between the regions found. The idea is to find the maximum area that each line might be inscribed, by determining the superior and inferior coordinates in the y axis. Figure 4c shows the limits of these regions after the exclusion threshold is applied. The red lines are the limits between two adjacent text line regions. In this way, the excluded regions are recovered [1].

3.3. Smearing Approach

Reference [12] describes smearing method. In this method the consecutive black pixels along the horizontal direction are smeared consequently; the white space between the black pixels is filled with black pixels. It is valid only if their distance is within a predefined threshold. This way, enlarged areas of black pixels around text are formed. It is so-called boundary growing areas. These areas of the smeared image enclose separated text lines. Thus, obtained areas are mandatory for text line segmentation.

3.4. Stochastic Approach

Reference [12] [13] [18] describes the stochastic approach for text based image segmentation. Stochastic method is based on probabilistic algorithm, which accomplished nonlinear paths between overlapping text lines. These lines are extracted through hidden Markov modelling (HMM). This way, the image is divided into little cells. Each one them correspond to the state of the HMM. The best segmentation paths are searched from left to right. In the case of touching components, the path of highest probability will cross the touching component at points with as less black pixels as possible. However, the method may fail in the case that contact point contains a lot of black pixels.

3.5. Water Flow Approach

The water flow algorithm assumes hypothetical water flows under a few angles of the document image from left to right and top to bottom [12]. In this hypothetically assumed situation, water is flowing across the image. For the water flows from left to right, (i) the situation is shown in figure 5a. Areas that are not wetted form unwetted ones. The stripes of unwetted areas are labelled for the extraction of text lines. Further, this hypothetical water flow is expected to fill up the gaps between consecutive text lines. Hence, unwetted areas left on the image indicates the text lines. Once the labelling is completed, the image is divided into two different types of stripes. First one contains text lines. The other one contains line spacing. The angle of the flow of the hypothetical water can be obtained using a mathematical function depending on the application. The united unwetted can be seen in figure 5b. The unwetted region describes the presence of text in the image.

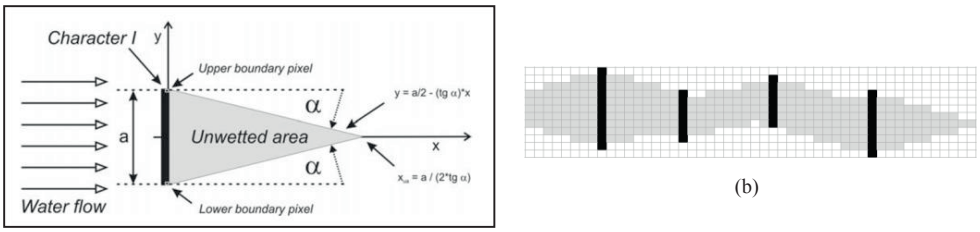


Fig. 5. (a) Unwetted area definition [12]; (b) United unwetted area [12]

4. Conclusion and future work

The work performed as discussed in the paper brings a conclusion that the algorithms that should be used for printed or handwritten text document image differs greatly. The pixel counting algorithm is simple to implement and we can conclude that it excels only for the printed text document. This algorithm can be used for a handwritten document if it has some kind of guidelines provided or when the document has even text size and uniform interline spacing, but it fails to provide satisfactory results while working with handwritten text images. Also, additional overhead like skew correction module is required.

A histogram approach being flexible outrivaled the previews for both printed and handwritten text documents. For printed document due the increase in the computation, it is slower compared to the pixel counting approach. The algorithm triumphs for handwritten text document and provides results with high level of accuracy. Skew correction can be coup easily using this technique. The only disadvantage of the proposed histogram algorithm as compared to the pixel counting approach is the increased computation and the resulting space complexity, thereby experiencing diminution in computational speed.

The future prospects is to implement the remaining algorithm and conduct a comprehensive comparison of the various techniques as discussed in the paper. This work will provide ease to researchers, scientists and engineers working with text, image and provides them with application specific algorithm selection knowledge to comprehend the need of achieving faster and efficient methodologies for segmenting the text image.

References

[1] Rodolfo P. dos Santos, Gabriela S. Clemente, Tsang Ing Ren and George D.C. Calvalcanti,"Text Line Segmentation Based on Morphology and Histogram Projection", in 10th International Conference on Document Analysis and Recognition, 2009.

[2] L. Likforman-Sulem, A. Zahour, B. Taconet, “Text line segmentation of historical documents: a survey”, International Journal on Document Analysis and Recognition, 2007, pp. 123-138.

[3] Linda G. Shapiro and George C. Stockman (2001): “Computer Vision”, pp 279-325, New Jersey, Prentice-Hall, ISBN 0-13-030796-3.

[4] B. M. Sagar, G. Shobha and P. Ramakanth Kumar,Converting printed Kannada text image file to machine editable format using Database, International Journal of Computers, 2, 2008, 173–175.

[5] S. N. Srihari, V. Govindaraju and A. Shekhawat, "Interpretation of Handwritten Addresses in US Mailstream" in proceedings second International Conference on Document Analysis and Recognition, Tsukuba, Japan, 1993, IEEE Computer Society Press, pp. 291-294.

[6] M. Maloo and K. V. Kale, "Gujarati Script Recognition: A Review" in International Journal of Computer Science Issues, Vol 8, July 2011.

[7] S. B. Patil, Neural Network based bilingual OCR system: experiment with English and Kannada bilingual document, International Journal of Computer Applications, 13, 2011, 6–14.

[8] M. Thungamani and P. Ramakhanth Kumar, "A Survey of Methods and Strategies in Handwritten Kannada Character Segmentation" in International Journal of Science Research, Vol 01, issue 01, June 2012, pp. 18-23.

[9] M. S. Das, C. R. K. Reddy, A. Govardhan and G. Saikrishna, Segmentation of Overlapping Text lines, Characters in printed Telugu text document images, International Journal of Engineering Science and Technology, 2, 2010, 6606–6610.

[10] Nafiz Arica and Fatos T. Yarman-Vural,"An Overview of Character Recognition Focused on Off-Line Handwriting" in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, May 2001.

[11] S. Marinai, P. Nesi, “Projection Based Segmentation of Musical Sheets”, Document Analysis and Recognition, ICDAR 1999, pp. 515-518.

[12] D. Brodić and Z. Milivojević,"A New Approach to Water Flow Algorithm for Text Line Segmentation" in Journal of Universal Computer Science, vol. 17, no. 1,2011.

[13] Z. Razak, K. Zulkiflee, R. Salleh, M. Yaacob and E. Mohd, Tamil: A real-time line segmentation algorithm for an offline overlapped handwritten jawi character recognition chip, Malaysian Journal of Computer Science, 20, 2007, 171–182.

[14] K. A. Kluever, Study report character segmentation and classification, http://www.tipstricks.org/example.asp, 2008, 1–21.

[15] T. V. Ashwin and P. S. Sastry, A font and size-independent OCR system for printed Kannada documents using support vector machines: Sadhana, 27, 2002, 35–58.

[16] R. S. Kunte and R. D. Sudhaker Samuel, A simple and efficient optical character recognition system for basic symbols in printed Kannada text: Sadhana, 32, 2007, 521–533.

[17] M. Thungamani and P. Ramakhant Kumar, Keshava Prasanna and S. K. Rao, Off-line handwritten kannada text recognition using support vector machine using zernike moments, International Journal of Computer Science and Network Security, 11, 2011, 128–135.

[18] I. Rios, A. de Souza Britto Jr, A. L. Koerich L. E. S. Oliveira, An OCR free method for word spotting in printed documents, Journal of Universal Computer Science, 17, 2011, 48–63.

[19] Pulagam Soujanya, Vijaya Kumar Koppula, Kishore Gaddam & P. Sruthi,"Comparative Study of Text Line Segmentation Algorithms on Low Quality Documents", in CMR College of Engineering and Technology Cognizant Technologies, Hyderabad, India.

[20] K. Junga, K.I. Kimb, A.K. Jain, “Text information extraction in images and video: a survey”, Pattern Recognition, 2004, pp. 977-997.

[21] Pham, Dzung L.; Xu, Chenyang; Prince, Jerry L. (2000). "Current Methods in Medical Image Segmentation". Annual Review of Biomedical Engineering 2: 315–337.

[22] Content-based Multimedia Information Retrieval: State of the Art and Challenges, Michael Lew, et al.,ACM Transactions on Multimedia Computing, Communications, and Applications, pp. 1–19, 2006.

[23] N. Venkateswara Rao, A. Srikrishna, B. Raveendra Babu and G. R. M. Babu, An efficient feature extraction and classification of handwritten digits using neural networks, International Journal of Computer Science, Engineering and Applications, 1, 2011, 47–56.

[24] Zhu Xiaoyan, Shi Yifan,"New Algorithm for Handwritten Character Recognition", Beijing, China.

[25] J. Kanai and A. D. Bagdanov, “Projection profile based skew estimation algorithm for JPIG compressed images,” Int. J. Document Anal. Recognit., Vol. 1, no. 1, pp. 43–51, 1998.

[26] A. Hashizume, P. S. Yeh, and A. Rosenfeld, “A method of detecting the orientation of aligned components,” Pattern Recognit. Lett., Vol. 4, pp.125–132, 1986.

[27] M. Chen and X. Ding, “A robust skew detection algorithm for gray-scale document image,” in Proc. 5th Int. Conf. Document Anal. Recognit.,Bangalore, India, 1999, pp. 617–620.

[28] G. Louloudis1, B. Gatos2, I. Pratikakis2, C. Halatsis1,"Line And Word Segmentation of Handwritten Documents".

[29] E. Oztop et al., “Repulsive attractive network for baseline extraction on document images,” Signal Process, Vol. 74, no. 1, 1999.

[30] C. I. Patel, R. Patel, P. Patel, "Handwritten Character Recognition using Neural Network" in International Journal of Scientific & Engineering Research, vol2, Issue 5, May-2011, ISSN 2229-5518.

[31] R. Manmatha, J.L., Rothfeder, “A scale space approach for automatically segmenting words from historical handwritten documents”, IEEE Trans. Pattern Anal. Mach. Intell., 2005, pp. 1212-1225.

[32] G. Louloudis1, B. Gatos, I. Pratikakis, C. Halatsis1,"Line And Word Segmentation of Handwritten Documents".

[33] B. M. Sagar, G. Shobha and P. Ramakanth Kumar, Character segmentation algorithm for Kannada optical character recognition, Proceedings of the International conference on Wavelet Analysis and Pattern Recognition, Hong Kong, 30–31, 2008, 339-342.