International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 8, No. 4 (2015), pp. 155-164
http://dx.doi.org/10.14257/ijsip.2015.8.4.14

Segmentation Methods for Hand Written Character Recognition

Namrata Dave

G H Patel College of Engg. & Tech
Gujarat Technological University Gujarat, INDIA
namrata.dave@gmail.com

Abstract

Hand written Character Recognition is area of research since many years. Automation of existing manual system is need of most industries as well as government areas. Recognition of hand written characters is a demand for many fields. In this paper we have discussed our approach for hand written character segmentation. This paper discusses various methodologies to segment a text based image at various levels of segmentation. This paper serves as a guide for people working on the text based image 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. Also, the available techniques with their advantages and weaknesses are reviewed, along with directions for quick referral are suggested. At last, we have given our approach to text segmentation in brief.

Keywords: *Hand Written Character Segmentation, Binarization, Segmentation*

1. Introduction

Segmentation of hand written text document into individual character or digit is an important phase in document analysis, character recognition and many other areas. Character segmentation has become a crucial step for mail address recognition in the automatic post mail sorting system. Also out of available text segmentation methods, we do not have a universal accepted solution [1]. The reason for not achieving satisfactory recognition rates is the difficult nature of cursive handwriting and difficulties in the accurate segmentation and recognition of cursive and touching characters [2].

In order to segment text from a given input document image, it is necessary to detect all the possible text regions. In the case of printed scripts, segmentation is a relatively simple task. In the case of overlapped scripts, broken characters, connected characters, loosely configured characters, and mixed scripts, segmentation is difficult. Overlapped, broken, connected and loosely configured characters are major causes of segmentation errors [3]. Segmentation of Text Image is used to locate each individual character and its boundaries. It involves process of labeling, which assigns the some 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.

There are five major steps for character recognition. They are pre-processing, segmentation, representation, training and recognition, and last is post processing. In this paper we have focused on first two stages, pre-processing of document image followed by segmentation phase.

To simplify process of segmentation, we need to convert our image in a specific format to process it further. The preprocessing [5, 6] includes several steps such as digitization of input image, removing noise from source image, converting image to

ISSN: 2005-4254 IJSIP
Copyright ⓒ 2015 SERSC

binary image and last step is normalization. The preprocessing stage gives normalized image with reduced or no noise. Segmentation of image is next step after preprocessing input image. Segmentation can then be implemented into its subcomponents. Character recognition accuracy totally relies on Segmentation procedure. Segmentation of words, lines, or characters plays major role in the recognition rate of the script [7-10].

Further, based on the obtained labels/regions, text is divided into different logical areas, each one representing a predefined set of semantics [9]. Ideally segmentation is carried out by segmenting the image into regions which depicts a text line. After completion of the line segmentation of given image, it provides the necessary information for further segmentation steps such as skew detection and correction, text feature extraction and character recognition. Unlike printed documents, Processing of handwritten documents has remained a key problem in character recognition. Also, the need of segmentation triumph 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]. 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 [21, 24]. 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 character spacing is defined, which abolish the challenges as faced with handwritten documents. 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, 22]. Presence of texture, like images, patterns, et al. in the text document makes the task of segmentation multifaceted. Cursive text provides additional difficulty during character segmentation, due to the presence of ligatures [35].

2. Levels of Text Segmentation

All 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.

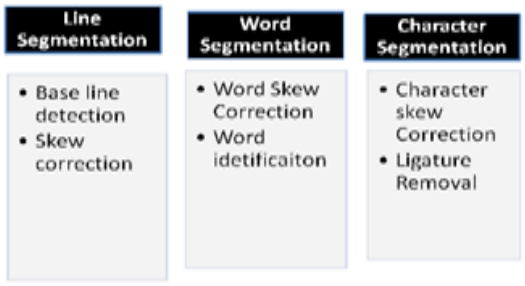
Segmentation at any of these levels directly depends on the nature of the application. More the details required for the image, the more is the level of segmentation. The various levels in segmentation are as shown in Figure 1(a).

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, 35]. 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

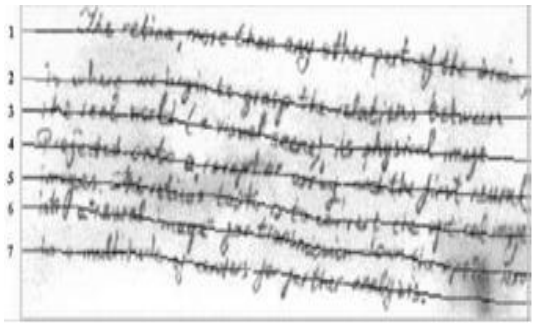
 Copyright ⓒ 2015 SERSC

scanning process and writing style, the writing may be slightly tilted or within the image. This can hurt the effectiveness of later algorithms and, therefore, should be detected and corrected.



(a)

Figure 1. Levels of Segmentation



(b)

Figure 2. Baseline Extraction [11]

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 1(b)). 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. Slant angle estimation is used 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 [33, 34].

Copyright ⓒ 2015 SERSC

Figure 2. Segmentation using Shortest Path of a Graph of Gray Level Image (a) Segmentation Intervals (b) Segmentation Paths (c) Segments [11]

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 3c 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. First, Appearance of the page indicates to the presence of guideline in the page. The presence of such guidelines eases the entire process. Another is 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 line 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.

 Copyright ⓒ 2015 SERSC

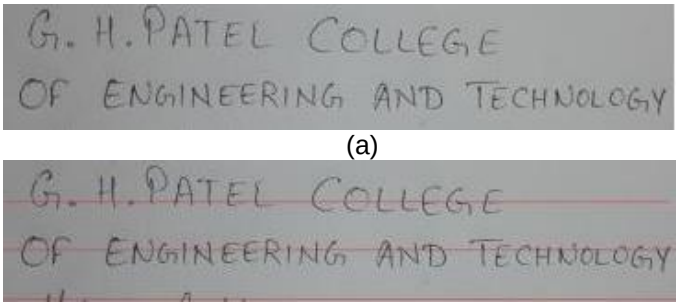


Figure 3. (a) Original Text Based Image; (b) Line Segmentation using Pixel Counting Approach

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 determines 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 represent 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 addresses 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 is the original images. They are provided as input to the algorithm and Figure 3b is obtained as output. The region between the red lines represents 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 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. Figure 4b shows histogram of input image. 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 is performed in order to

Copyright ⓒ 2015 SERSC

recover some losses introduced by the preceding step. At this point 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, 20, 21], 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 these rectangular coordinates, we can extract the words/characters from the digitized image.

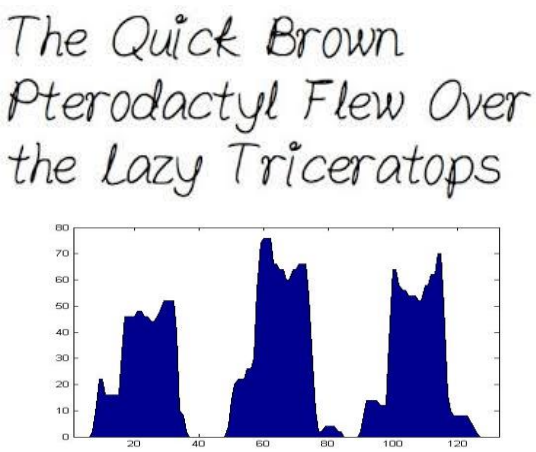


Figure 4. (a) Original Text Based Image; (b) Histogram of Given Input Image

3.2.1. Y Histogram Projection

Reference [1, 8] states that, once the pre-processing (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. 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.

 Copyright ⓒ 2015 SERSC

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.

$$\Sigma \mid Y_b - Y_f \mid / N_n$$

Where, Y_b is the y position where the text region begins, Y_f is the y position where the text region ends and N_n 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 and Figure 4b are author's implementation of Histogram method.

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.

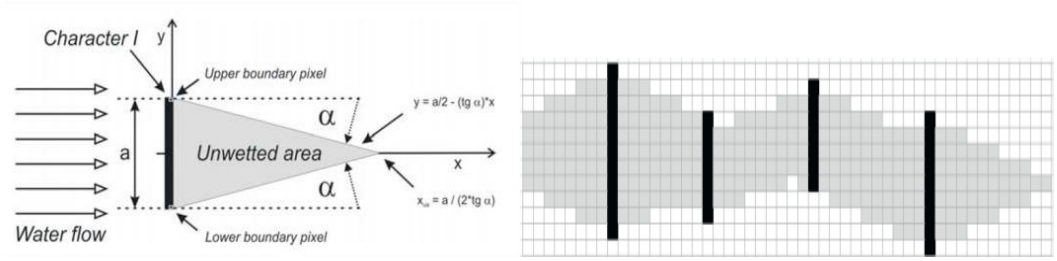
3.5. Water Flow Approach

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, the situation is shown in Figure 5a. Areas that are not wetted form unwetted

Copyright ⓒ 2015 SERSC

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 indicate 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.



(a) (b)

Figure 5. (a) Unwetted Area Definition [13]; (b) United Unwetted Area [13]

3.6. Mixed Approach

We have tries to employ a very basic approach for character segmentation by combining various image processing techniques. The image is first being converted to gray-scale image if it is colour image.

After converting it to gray-scale we have further converted our gray-scale image to black and white image followed by the threshing technique, which make the image become binary image. The binary image is then sent through connectivity test in order to check for the maximum connected component, which is, the box of the form. After locating the box, the individual characters are then cropped into different sub images that are the raw data for the feature extraction routine. Result of our approach is given in Figure 6a and 6b.

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.

 Copyright ⓒ 2015 SERSC

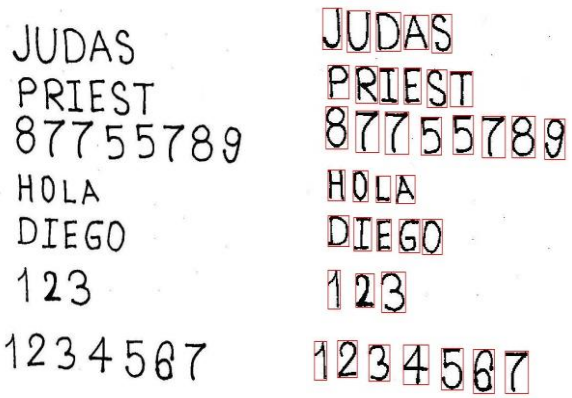


Figure 6 (a) Original Image (b) Segmentation Result

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 reduction in computational speed.

References

[1] R. R. P. dos Santos, G. S. Clemente, T. I. Ren and G. D. C. Calvalcanti, "Text Line Segmentation Based on Morphology and Histogram Projection", in 10th International Conference on Document Analysis and Recognition, **(2009)**.
[2] B. Verma and M. Blumenstein, "Pattern Recognition Technologies and Applications: Recent Advances", Information Science Reference, Hershey, New York, **(2008)**, pp. 1-16.
[3] H. L. Guo-Hong and S. Peng-Fei, "An approach to offline handwritten Chinese character recognition based on segment evaluation of adaptive duration", Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai, China, **(2003)**.
[4] L. G. Shapiro and G. C. Stockman, "Computer Vision", New Jersey, Prentice-Hall, ISBN 0-13-030796-3, **(2001)** pp. 279-325.
[5] B. M. Sagar, G. Shobha and P. R. Kumar, "Converting printed Kannada text image file to machine editable format using Database", International Journal of Computers, vol. 2, **(2008)**, pp. 173–175.
[6] 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, IEEE Computer Society Press, **(1993)**, pp. 291-294.
[7] M. Maloo and K. V. Kale, "Gujarati Script Recognition: A Review", in International Journal of Computer Science Issues, vol. 8, **(2011)** July.
[8] S. B. Patil, "Neural Network based bilingual OCR system: experiment with English and Kannada bilingual document", International Journal of Computer Applications, vol. 13, **(2011)**, pp. 6–14.
[9] M. Thungamani and P. R. Kumar, "A Survey of Methods and Strategies in Handwritten Kannada Character Segmentation", in International Journal of Science Research, vol. 01, issue 01, **(2012)** June, pp. 18-23.
[10] 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, vol. 2, **(2010)**, pp. 6606–6610.
[11] N. Arica and F. T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting", in IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, **(2001)** May.
[12] S. Marinai and P. Nesi, "Projection Based Segmentation of Musical Sheets", Document Analysis and Recognition, ICDAR, **(1999)**, pp. 515-518.
[13] 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)**.
[14] 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, vol. 20, **(2007)**, pp. 171–182.

Copyright ⓒ 2015 SERSC

[15] K. A. Kluever, "Study report character segmentation and classification", **(2008)**, pp. 1-21, http://www.tipstricks.org/example.asp.
[16] T. V. Ashwin and P. S. Sastry, "A font and size-independent OCR system for printed Kannada documents using support vector machines", Sadhana, vol. 27, **(2002)**, pp. 35–58.
[17] R. S. Kunte and R. D. S. Samuel, "A simple and efficient optical character recognition system for basic symbols in printed Kannada text", Sadhana, vol. 32, **(2007)**, pp. 521–533.
[18] M. Thungamani, P. R. Kumar, K. 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, vol. 11, **(2011)**, pp. 128–135.
[19] R. A. de Souza Britto Jr, A. L. Koerich and L. E. S. Oliveira, "An OCR free method for word spotting in printed documents", Journal of Universal Computer Science, vol. 17, **(2011)**, pp. 48–63.
[20] P. Soujanya, V. K. Koppula, K. Gaddam and P. Sruthi, "Comparative Study of Text Line Segmentation Algorithms on Low Quality Documents", in CMR College of Engineering and Technology Cognizant Technologies, Hyderabad, India.
[21] K. Junga, K. I. Kimb and A. K. Jain, "Text information extraction in images and video: a survey", Pattern Recognition, **(2004)**, pp. 977-997.
[22] D. L. Pham, C. Xu and J. L. Prince, "Current Methods in Medical Image Segmentation", Annual Review of Biomedical Engineering, vol. 2, **(2000)**, pp. 315-337.
[23] M. Lew, *et al.*, "Content-based Multimedia Information Retrieval: State of the Art and Challenges", ACM Transactions on Multimedia Computing, Communications, and Applications, **(2006)**, pp. 1–19.
[24] N. V. Rao, A. Szrikrishna, B. R. 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, vol. 1, **(2011)**, pp. 47–56.
[25] Z. Xiaoyan and S. Yifan, "New Algorithm for Handwritten Character Recognition", Beijing, China.
[26] J. Kanai and A. D. Bagdanov, "Projection profile based skew estimation algorithm for JPIG compressed images", Int. J. Document Anal. Recognition, vol. 1, no. 1, **(1998)**, pp. 43–51.
[27] Hashizume, P. S. Yeh and A. Rosenfeld, "A method of detecting the orientation of aligned components," Pattern Recognit. Lett., vol. 4, **(1986)**, pp. 125–132.
[28] M. Chen and X. Ding, "A robust skew detection algorithm for grayscale document image", In Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on IEEE, **(1999)**, pp. 617-620.
[29] G. Louloudis, B. Gatos, I. Pratikakis and C. Halatsis, "Line and Word Segmentation of Handwritten Documents", 11th Int. Conf. on Frontiers in Handwriting Recognition, **(2008)**.
[30] E. Oztop, *et al.*, "Repulsive attractive network for baseline extraction on document images," Signal Process, vol. 74, no. 1, **(1999)**.
[31] C. I. Patel, R. Patel and P. Patel, "Handwritten Character Recognition using Neural Network", in International Journal of Scientific & Engineering Research, vol. 2, Issue 5, **(2011)**, ISSN 2229-5518.
[32] R. Manmatha and 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.
[33] G. G. Louloudis, B. Gatos, I. Pratikakis and C. Halatsis, "Text line and word segmentation of handwritten documents, Pattern Recognition", vol. 42, Issue 12, **(2009)** December, pp. 3169-3183, ISSN 0031-3203, http://dx.doi.org/10.1016/j.patcog.2008.12.016.
[34] B. M. Sagar, G. Shobha and P. R. Kumar, "Character segmentation algorithm for Kannada optical character recognition", Proceedings of the International conference on Wavelet Analysis and Pattern Recognition, Hong Kong, vol. 30–31, **(2008)**, pp. 339-342.
[35] Z. Han, C.-P. Liu and X.-C. Yin, "A two-stage handwritten character segmentation approach in mail address recognition", Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on, vol. 1, **(2005)** August-September, pp. 111, 115.
[36] G. Mehul, P. Ankita, D. Namrata, G. Rahul and S. Sheth, "Text-based Image Segmentation Methodology", Procedia Technology, vol. 14, **(2014)**, pp. 465-472, ISSN 2212-0173.

Author

Namrata Dave

Assistant Professor,
Computer Engineering Department,
G H Patel College of Engg and Tech,
Gujarat, India.

 Copyright ⓒ 2015 SERSC