Volume 3, No. 1, January 2012

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

LINE SEGMENTATION USING CONTOUR TRACING

Ashu Kumar*[1], Simpel Rani Jindal[2], Galaxy Singla[3]
[1]Department of CSE, Yadwindra College of Engineering, Talwandi Sabo, Punjab, India
ashu.software.engineer@gmail.com[1]
[2]Department of CSE, Yadwindra College of Engineering, Talwandi Sabo, Punjab, India
simpel_jindal@rediffmail.com[2]
[3]Department of CSE, Bhai Maha Singh College of Engineering, Sri Muktsar Sahib, Punjab, India
galaxy_bansal@yahoo.co.in[3]

Abstract: Text line segmentation is an important step because inaccurately segmented text lines will cause errors in the recognition stage. Text line segmentation of the handwritten documents is still one of the most complicated problems in developing a reliable OCR. The nature of handwriting makes the process of text line segmentation very challenging. Text characteristics can vary in font, size, orientation, alignment, color, contrast, and background information. These variations turn the process of word detection complex and difficult. Since handwritten text can vary greatly depending on the user skills, disposition and cultural background. The technique of Piece-wise projection alongwith contour tracing to segment a handwritten document into distinct lines of text is presented. The proposed method is robust to handle line fluctuation.

Keywords: OCR, Line Segmentation, Histograms, chunks, Piece-wise separating lines, Potential PSLs.

INTRODUCTION

A lot of research work has been investigated for character recognition of Gurmukhi script. For an optical character recognition (OCR) system, segmentation phase is an important phase and accuracy of any OCR heavily depends upon segmentation phase. Incorrect segmentation leads to incorrect recognition. Segmentation phase include line, word and character segmentation. Before word and character segmentation, line segmentation is performed to find the number of lines and boundaries of each line in any input document image. Incorrect line segmentation may result in decrease in recognition accuracy.

For segmentation of lines from handwritten text, survey papers are available [1,2]. Considerable amount of work has been carried out to segment lines of handwritten Roman script and there are varied and some well developed techniques [3-7]. But very little work has been carried out for Indic scripts like Devnagri, Bengali, Gurmukhi etc. Only a few papers are available for segmentation of handwritten Indic scripts [8-11].

The simplest and most widely used method to segment the lines is to use the inter-line gap in horizontal projection as line boundaries. This method does not work well on skewed, fluctuating or proximate images. Here, we are modifying the method to segment text lines based on histogram projection. Figures 1,2, and 3 shows three kinds of sample documents on which the line segmentation is performed. The rest of the paper is organized as follows. Section 2 describes problems associated with line segmentation. Section 3 describes the method to be proposed. Experiments and results are discussed in section 4 which is followed by conclusion in section 5.

SEGMENTATION CHALLENGES

When dealing with handwritten text, line segmentation has to solve some obstacles that are uncommon in modern printed text. Among the most predominant are:

Skewed lines: lines of text in general are not straight.

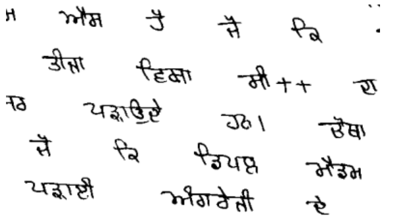


Figure 1: Skewed lines

Fluctuating lines:

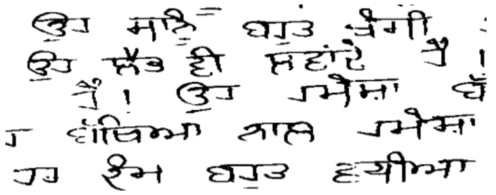


Figure 2: Fluctuating lines

Line proximity: Small gaps between neighbouring text lines will cause touching and overlapping of components, usually words or letters, between lines and irregularity in geometrical properties of the line, such as line width, height, distance in between words and lines, leftmost position etc.

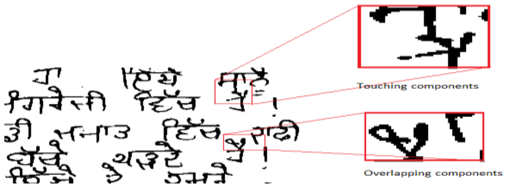


Figure 3: Line proximity

PROPOSED METHOD

There exist several methods for text line segmentation. which are roughly categorized as follows. Smearing methods, Horizontal projections, Hough transform, repulsive attractive networks, stochastic methods and text line structure enhancing [1,2]. Due to many challenges in handwritten text line segmentation, although many methods have been proposed, the problem still remains open.

Horizontal projections can't deal well with skewed, curved and fluctuating lines. The method of horizontal projection of the whole text is suitable for segmentation of the text with straight lines and with large gap in lines. For example, see the Fig 1,2 & 3.

So to segment this type of text, Here we are modifying the method to segment text lines based on histogram projection and this technique is called piece-wise projection [12]. For connecting intersecting components, we are using contour tracing.

At first, we divide the text into vertical chunks of width W. Width of the last chunk may differ from W. Computation of W is discussed later. Next, we compute piece-wise separating lines (PSL) from each of these chunks [12]. We compute the horizontal projection of each chunk. The projection profiles of the chunks of the image are shown in Fig 4.

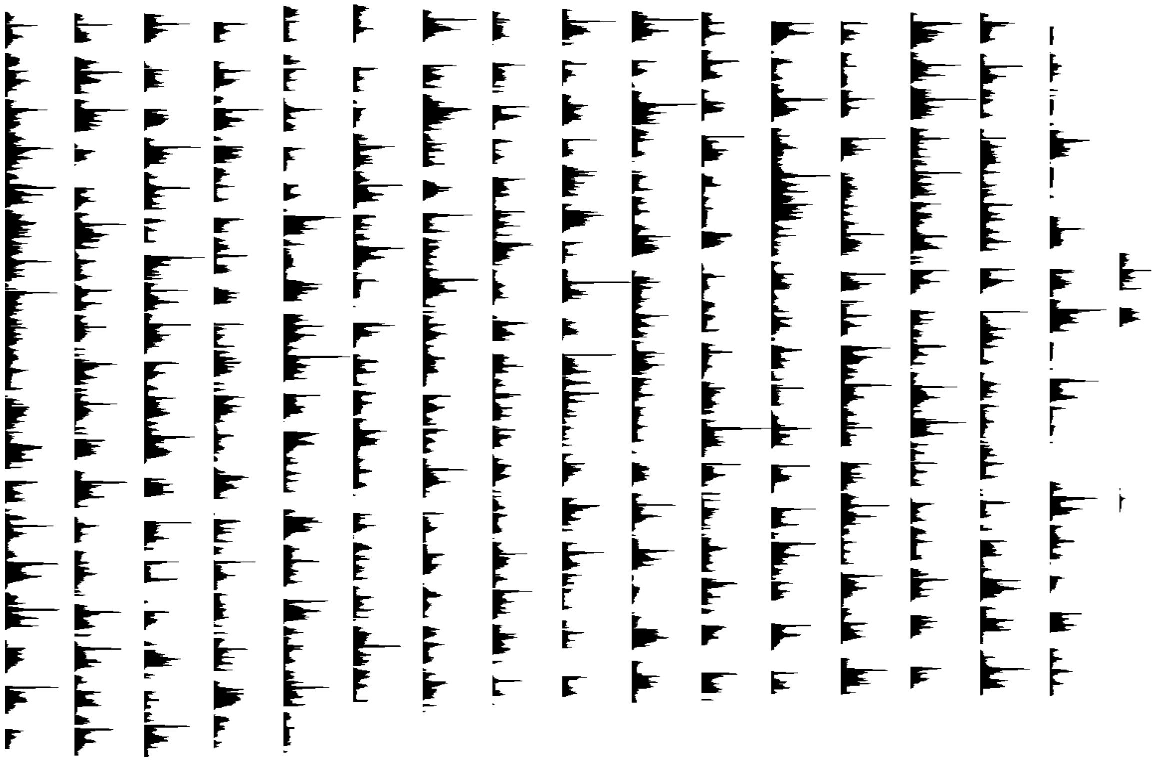


Figure 4: Projection profile of each chunk

The row where this HP is zero is a PSL. We may get a few consecutive rows whose HP is zero. Then the first row of such consecutive rows is the PSL. The PSLs of different chunks of a text are shown in Fig. 5 by horizontal lines.

All these PSLs may not be useful for line segmentation. We choose some potential PSLs as follows. For this, we compute the estimate height of word. If the distance between any two consecutive PSLs of a stripe is less than word height, we remove the upper PSL of these two PSLs. PSLs obtained after this removal are the *potential PSLs*. The potential PSLs obtained from the PSLs of Fig 5 are shown in Fig 6.

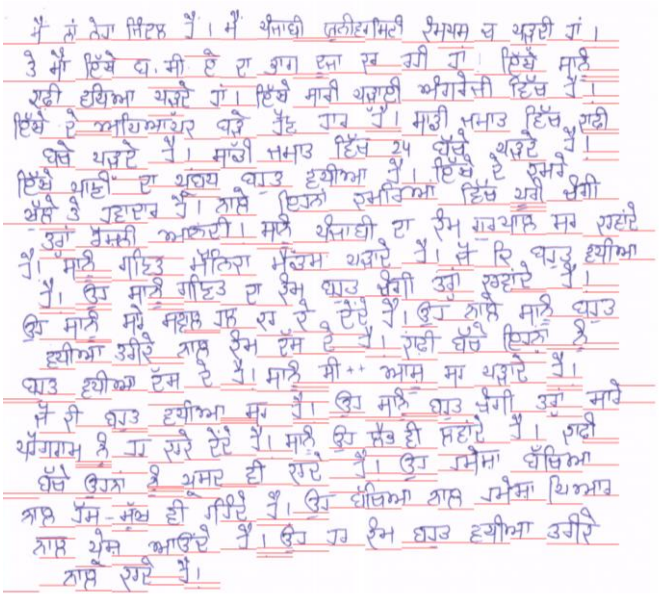


Figure 5: PSLs of each chunk

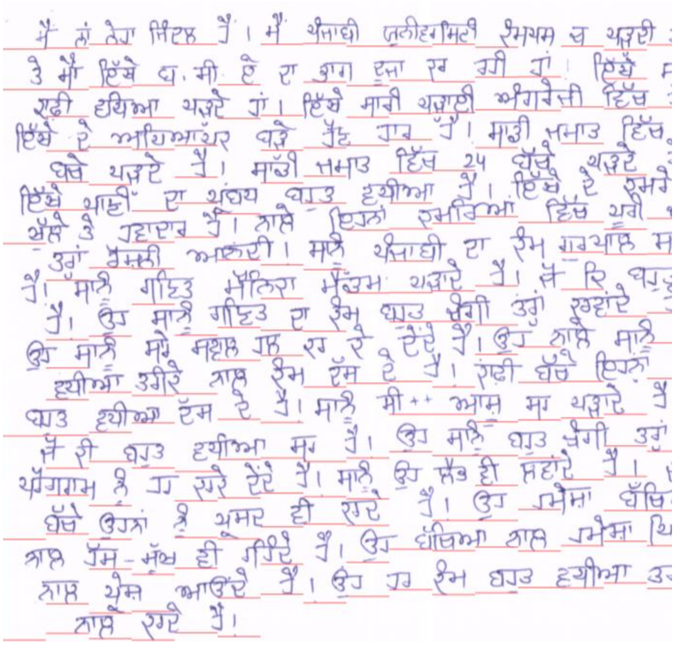


Figure 6: Potential PSLs of each chunk

We stored the y-coordinates of each potential PSL in an array for future use. By proper joining of these potential PSLs, we get individual text lines. It may be noted that sometimes because of overlapping or touching of one component of the upper line with a component of the lower line, we may not get PSLs in some regions. Also, because of some modified characters of Gurmukhi (e.g. *adhak, chandrabindu*) we find some extra PSLs in a chunk. We take care of them during PSL joining, as explained next.

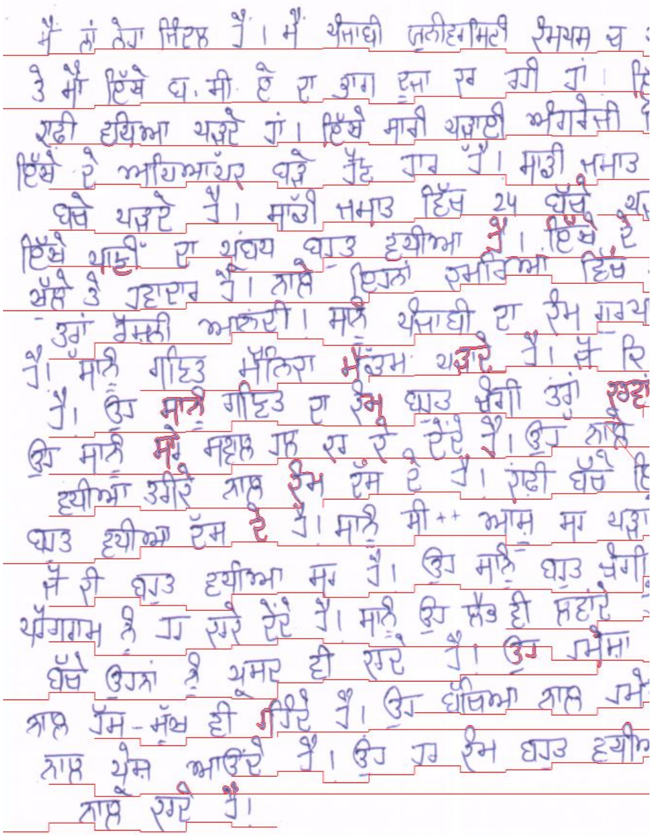


Figure 7: Segmented Text

First, we stored the y-coordinates of each potential PSL in an array for future use. By proper joining of these potential PSLs, we get individual text lines. It may be noted that sometimes because of overlapping or touching of one component of the upper line with a component of the lower line, we may not get PSLs in some regions. Also, because of some modified characters of Gurmukhi (e.g. *adhak, chandrabindu*) we find some extra PSLs in a chunk. We take care of them during PSL joining, as explained next.

To join a PSL of the *i*th chunk to a PSL of *(i + 1)*th chunk, starting with the first PSL of both ith and (i+1)th chunk, we check the distance between them that should be less than 72% of word_height. The word_height is probably 40-50. So the distance can be upto 3/4th of height of word. By experimenting, we have reached the conclusion of 72%.

If it exists, we check weather the PSL in (i+1)th chunk is on upper or lower side. Then we join the right co-ordinate of K_i with the left co-ordinate of the PSL in the *(i +1)*th chunk. Pointer to both PSLs is incremented by 1. If it does not exist and if PSL is in lower side, we extend the PSL_i horizontally in the right direction until it reaches the right boundary of the *(i + 1)*th chunk or intersects a black pixel of any component in the *(i + 1)*th chunk. The value of PSL in (i+1)th chunk array is also inserted. Otherwise we just increment the pointer to (i+1)th chunk by 1.

If the extended part intersects the black pixel of a component of the *(i + 1)*th chunk, we decide to trace the component whether in upper line or lower line. Based on the belongingness of this component, we extend this line in such a way that the component falls in its actual line[12]. Belongingness of a component is checked as follows.

We compute the distances from the intersecting point to the topmost and bottommost point of the component. Let d_1 be the top distance and d_2 the bottom distance and word_height is estimated to be 40 for A4 size paper having 18-20 text lines written. If $d_1<d_2$ and $d_1<$(word_height/2) then the component belongs to the lower line. If $d_2\leq d_1$ and $d_2<$(word_height/2) then the component belongs to the upper line. If $d_1>$(word_height/2) and d2>(word_heigh/2) then we assume the component touches another component of the lower line [12]. If the component belongs to the upper-line (lower-line) then the line is extended following the contour of the lower part (upper part) of the component so that the component can be included in the upper line (lower line) as shown in Fig 8.

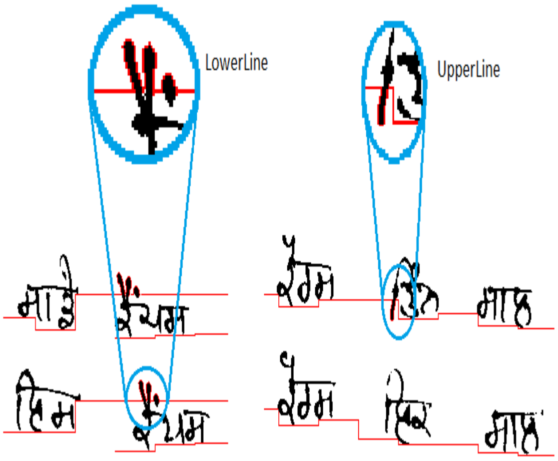
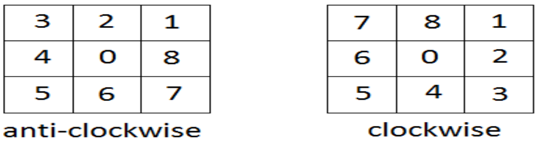


Figure 8: Upperline and Lowerline contours

To follow the contour, we are testing the 8-connectivity (8 neighbouring points). To test the connectivity, we have numbered the pixel to 0 and its neighbouring pixels from 0 to 8 depending upon the type of contour, whether it is upperline or lowerline contour. For upperline (lowerline) contour, we numbered the pixels in clockwise (anti-clockwise) direction as shown in Fig 9:



3	2	1
4	0	8
5	6	7

anti-clockwise

7	8	1
6	0	2
5	4	3

clockwise

Figure 9: Anti-clockwise contour for lowerline and clockwise contour for upperline

The method for both upper and lower contours is same. The method is that if the pixel at position 1 is black and at position 2, the pixel is white, then pixel at position 2 will become the current pixel. Similarly, if 2 is black and 3 is white then current pixel will be 3 and so on. Sometimes, the values are repeated and program stucks. At that point, increment current pixel's x coordinate by 2. The line segmented text is shown in Fig 7.

We have estimated the width size of chunk say W as 70. It is because if we take W = 50, it will make very small chunks and the chances of intersecting with the component are more and if we take W=100, number of chunks will be less and it can protect various lines to segment in case of the document in which text lines are very close to each other.

EXPERIMENTS AND RESULTS

The experiments are performed on various handwritten text images in Gurmukhi Script. The images with high skewness, less line gap, more gap in words etc. are considered. For experiments, we considered only single column document pages. By viewing the results on the computer's display, we calculate line segmentation accuracy manually by checking correctly segmented components. We have shown a segmented image with lines very close to each other. The words of lines are overlapping and touching highly in this example. We tried to use contour tracing to accurately segment the intersecting components as shown in Fig 8.

Table 1: Line Segmentation Accuracy

Documents	Components		Interline Gap	No. of lines: correctly segmented/Total	Line Fluctuation	Accuracy
	Intersecting	Touching				
Doc 1	9	0	low	16/16	Very low	100%
Doc 2	16	0	low	15/15	Very low	100%
Doc 3	21	0	Very low	13/14	low	92.8%
Doc 4	39	10	Very low	15/19	high	78.9%
Doc 5	12	0	high	19/20	high	95%
Doc 6	17	2	medium	12/14	high	85.7%
Doc 7	7	0	medium	15/15	medium	100%
Doc 8	14	0	medium	15/15	medium	100%
Doc 9	11	0	low	14/15	Very high	93%
Doc 10	13	0	Very low	8/10	high	80%
Doc 11	10	1	low	15/17	Very high	88%

The results of Line segmentation can be seen in Table 1. This method can be applied to other Indian scripts too. But its limitation is that it is size dependant. In future we plan to use different sized text. Figures of some documents are shown below:

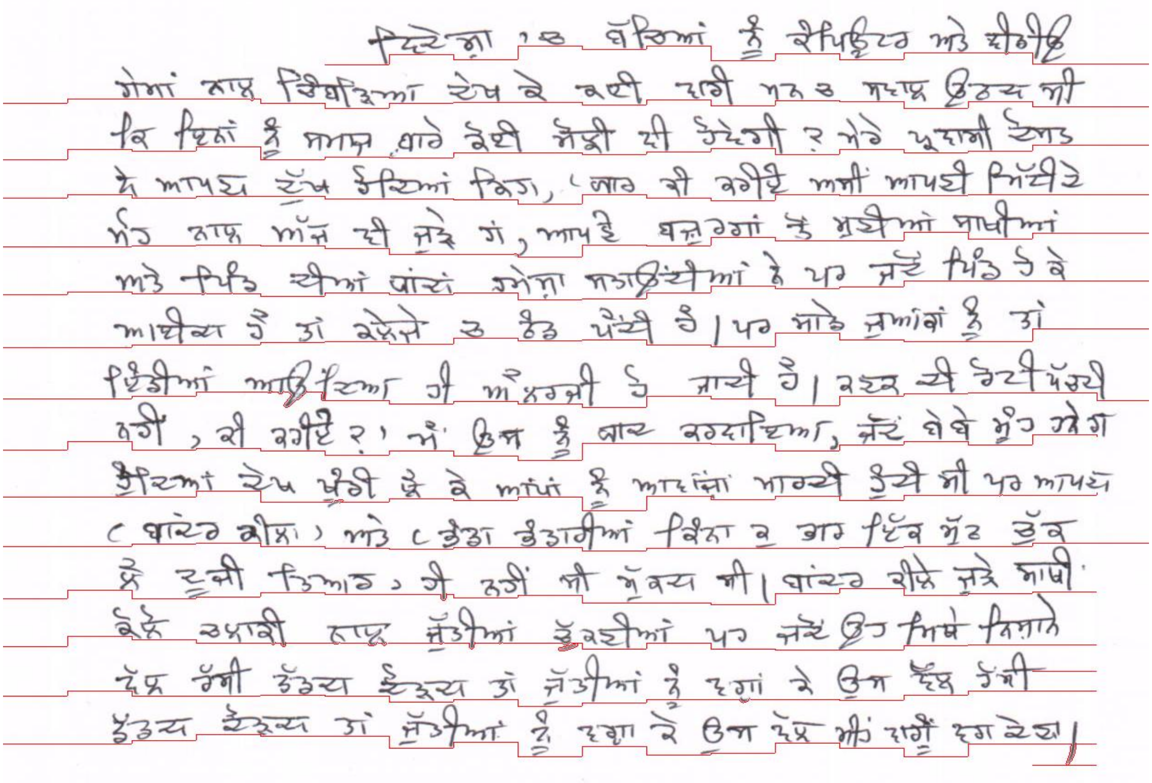


Figure 10: Output of Doc 7

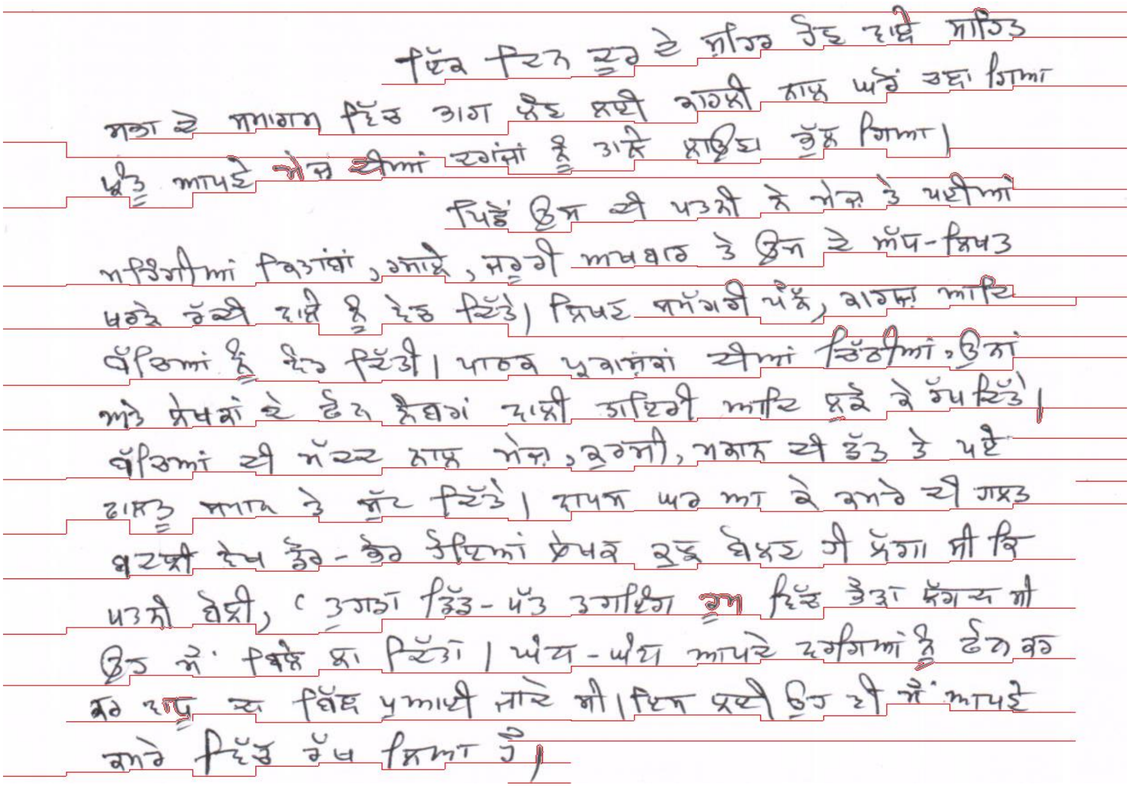


Figure 11: Outputs of Doc 9

REFERENCES

[1] Sulem, Zahour, Bruno Taconet, “Text line segmentation of historical documents: a survey”, pp. 123–138, IJDAR-9, 2007.

[2] Razak, Mohd, “Off-line Handwriting Text Line Segmentation : A Review”, pp. 12-20, IJCSNS, 2008.

[3] Manivannan Arivazhagan, Harish Srinivasan and Sargur Srihari, “A Statistical approach to line segmentation in handwritten Documents”, pp. 1-11, SPIE, 2007.

[4] Xiaojun Du, Wumo Pan, Tien. D. Bui, “Text Line Segmentation in Handwritten Documents Using Mumford-Shah Model”, pp. 3136-3145, Pattern Recognition, 2009.

[5] Yosef, Nate Hagbi, Dinstein, “Line segmentation for degraded handwritten historical documents”,pp. 1161-1165, ICDAR, 2009.

[6] A. Nicolaou and B. Gatos, “Handwritten Text Line Segmentation by Shredding Text into its Lines”, pp. 626-630, 10th ICDAR, 2009.

[7] Stéphane Nicolas, Thierry Paquet, Laurent Heutte, “Text Line Segmentation in Handwritten Document Using a Production System”, pp. 245-250, IWFHR-9, 2004.

[8] Rajiv K. Sharma & Dr. Amardeep Singh, “Segmentation of Handwritten Text in Gurmukhi Script”, pp. 12-17, IJIP-11, 2008.

[9] M. Hanmandlu and Pooja Agrawal, “A Structural Approach for Segmentation of Handwritten Hindi Text”, pp. 589-597, ICCR, 2005.

[10] Rajiv K. Sharma & Dr. Amardeep Singh, “Segmentation of Handwritten Text in Gurmukhi Script”, pp. 12-17, IJIP-11, 2008.

[11] Naresh Garg, M.K. Jindal, “Segmentation of Handwritten Hindi Text”, pp. 19-23, IJCA, 2010.

[12] N Tripathy and U. Pal, “Handwriting segmentation of unconstrained Oriya text”, pp. 306-311,IWFHR, 2004.

[13] Munish, R.K.Sharma & M.K.Jindal, “Segmentation of Lines and Words in Handwritten Gurmukhi Script Documents”, pp. 25-28, IITM-1, 2010.

[14] U. Pal and S. Datta, “Segmentation of Bangla unconstrained handwritten text”, pp. 1128-1132, ICDAR-7, 2003.