Character Analysis Using Space in Handwriting Image

Subham Nagar, Sudiksha Chakraborty, Arka Sengupta, Joymallya Maji and Rajib Saha

Department of Computer Science and Engineering, RCC Institute of Information Technology, Kolkata – 700015, West Bengal, India; subhamnagar@gmail.com, brinda@gmail.com, asengupta526@gmail.com, joymallya1903@gmail.com, rajibsaha_4u@yahoo.co.in

Abstract

Handwriting analysis is an effective and reliable indicator of personality and behaviour. It occurs in several stages. Initially, the handwriting samples are collected on plain paper. For better appearance, pre-processing steps such as binarization and noise removal etc are performed. First step begins with considering colour image or gray scale image as an input, then thresholding converts the image into binary image and noise removal technique is also applied. Then line segmentation, word segmentation and character segmentation are performed. After each segmentation process, normalization techniques are applied for normalization purpose to find out space between lines, words and letters in handwriting images. Finally, the mean of the space between all the closed loops formed by the characters has been found out and compared with the word spaces to determine the character. This paper focuses on efficient method of space analysis in handwritten document. The proposed method based on determination of behaviour based on space analysis. The proposed method was tested on images of IAM database which detect the exact space in between lines, words and characters before and after skew normalization of a document. The experimental result shows that proposed algorithm achieves more than 63% accuracy.

Keywords: Document Image, Feature Extraction, Line Segmentation, Space Analysis, Word Segmentation

1. Introduction

Handwriting analysis has been a topic of intensive research interest. Handwriting analysis has four parts- pre-processing, segmentation, feature extraction and classification. Amongst them, image pre-processing[1,16] is used to improve the quality of the image for easy and efficient processing in future procedure. The steps of pre-processing are namely thresholding[6,7] and noise removal[8,30]. Thresholding process converts gray scale image into binary scale and noise removal technique improves the quality of the image. Noise can appear in the foreground or background of an image and can be generated before or after scanning.

**Author for correspondence*

Several types of the noises can appear in the scanned handwriting document. Feature extraction, the most crucial task which extracts the feature from the segmented image using various techniques. The last step is classification method which is the main attraction of the handwriting recognition technique.

This paper concentrates on space analysis i.e. determination of behaviour on the basis of comparison of space between words and the space between the closed loops formed by the characters. How the space analysis of a handwritten document before skew normalization differs from that after the skew normalization is also an important perspective of this paper. Whereas skew[2,10–13] in handwriting document can occur due to the humans' behaviour as well as by the scanner during the scanning process. After applying skew normalization, it was found that the space calculation before and after skew normalization was varying in case of same document. It may increase as well as decrease after the skew. In the case of correct orientation of the pages, handwriting document still consists of positive and negative skews[2,10–13] due to variation in writing. Thus on its application, the changed results on the now rotated lines are found to be more accurate. Accordingly, the space calculations state the behaviour characteristics of a person predefined in previous papers.

Technically, skew is defined as a deviation in the alignment of the straight line i.e. a slant. Skew can be either clockwise i.e. positive skew or anti-clockwise i.e. negative skew. But at the time of recognition of the procedure, handwriting document should be free from the skew for better appearance. Skew normalization is a compulsory requirement for character segmentation. There are four types of skew available - Negative, Positive skew, Normal and Wavy skew, which are shown in Figure 1.

Examples of positive skewed and negative skewed words are shown in Figure 2(a) and Figure 2(b). The movement of the positive skew is towards the clockwise direction and movement of the negative skew is towards the anti-clockwise direction.

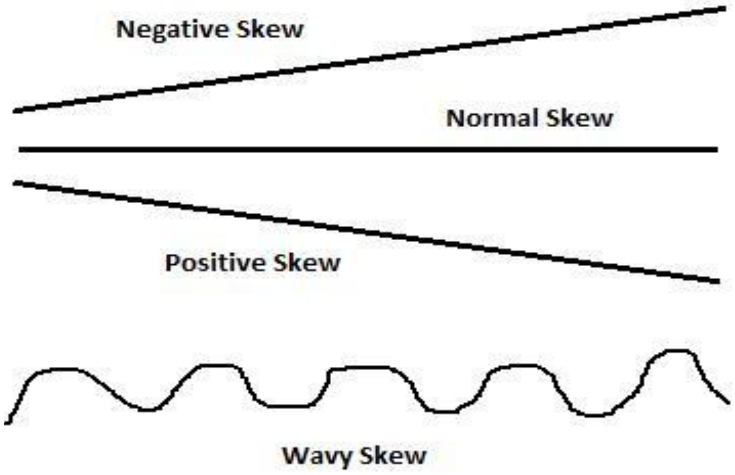


Figure 1. Orientation of skew.

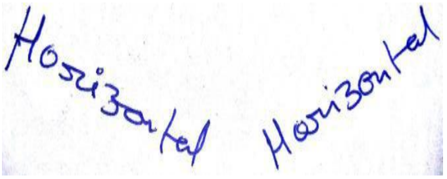


Figure 2. **(a)** Example of positive skewed and **(b)** negative skewed words.

Let us consider a document having positive or negative skew. When the segmentation of lines, word and characters is performed, it is often found that due to too much slanting of lines, the top or bottom or both sides of the lines are getting cut-off. As a result, the space calculations in between lines, words and characters are getting deviated. In case of overlapping lines in the original document, this segmentation was found to have errors. There are previous methods have been proposed in past research[2,10–13] for skew detection and normalization of document image. On applying any of these methods, skew normalization is possible. However, he have used a different method here[31,32].

This paper proposed an efficient method for behaviour or character classification based on space analysis on handwriting documents. The proposed method is based on space calculations before and after skew normalization[13] and comparison of the space between words and that between the closed loops of the letters in a document. This method efficiently can deal with higher as well as smaller skew of handwriting document. Proposed approach is much efficient to deal with the words like “pot”, “gauge”, “below” etc where more number of letters in the words consists of closed loops.

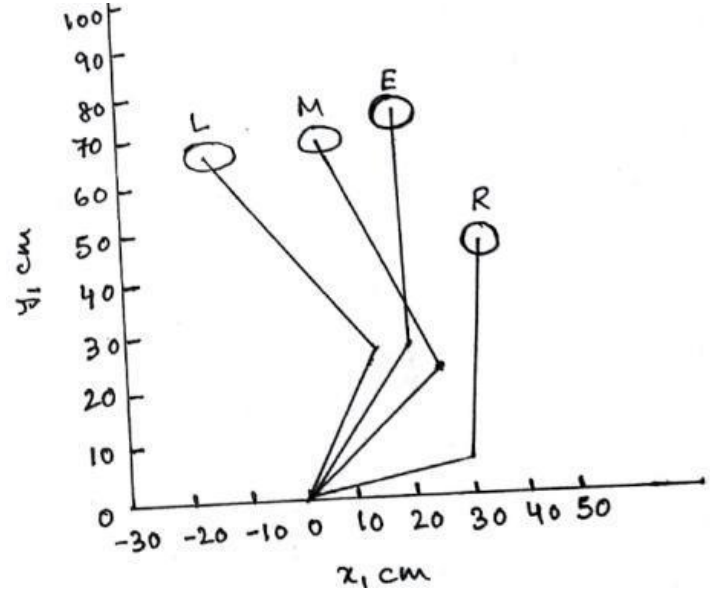


Figure 3. Closed loop of letters in words.

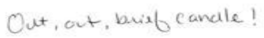
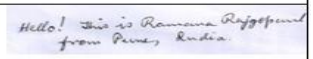
Figure 3 shows the closed loop of letters in words of the handwriting image. The handwriting is considered to be written on a blank plain A4 sheet of paper. Wherever words like “below”, “gauge”, “pot”, etc are found, the proposed method detects the closed loop of “b”, “o”, “g”, etc and calculates the radii of all such characters and stores in an array. Then the sum of the diameters of all such characters is calculated and their mean is found out[34]. Now, this mean space between the loops of these characters is compared with the space between the words of that document. If double the diameter of the loop space is found to be greater than the word space, then the person is narrow-minded and vice-versa[34].

2. Literature Review

U,-V. Marti and H. Bunke[9] introduced a system for reorganization English handwriting text based on large vocabulary. The proposed method segments complete lines of text lines into single words. The algorithm tested on 541 text lines containing 3899 word and performed correct segmentation rate of 95.56%.

In 2015, Seema Kedar, Ms. Vaishnavi Nair, Ms. Shweta Kulkarni[2] proposed a Personality Identification through Handwriting Analysis.

Table 1. Spacing and related personality traits

Sr. no	Type	Personality Traits	Writing Sample
1	Wide Spacing	discrimination, good taste, independence, exclusiveness, isolation, loneliness, snobbery, and pride.	 Out, out, brief candle!
2	Narrow Spacing	warmth, sympathy, gregariousness, obtrusiveness, poor taste, and inability to be alone	 Hello! this is Ramana Rajgopal from Pune, India

In graphology spacing implies the distance maintained between the lines, words and letters by the writer. Spacing can reveal various personality constructs like the writer's closeness with people, and also his intelligence. Spacing can be identified into two main groups as wide or narrow as shown in table below:

In 2013, A. Roy, T.K. Bhowmik, S.K. Parui and U. Roy[10] proposed novel approach to word spacing. If the spacing between words in your handwriting is very wide, it clearly shows that the writer has developed trust issues and therefore he is unable to establish a steady relationship with fellow humans around him. Such writers mostly keep quiet and avoid getting into discussions and arguments.

3. Proposed Work

Skew is defined as a deviation in the alignment of the straight line i.e. a slant. In general, skew is measured depending on the baseline. Space calculation between lines, words and characters are done on original handwritten document and that after skew normalization on same sample. According to the review[2,10–12] in the previous section, there are several pre-defined behavioural characteristics analyzed on the basis of spacing. Previously proposed assumptions[2] only stated the characteristics but have not showed it by any specific calculative method. The most important thing is that review methods only stated the traits based on assumptions and thus it is tested whether previous assumptions[11] are efficient to deal with actual documents.

The proposed work deals with identification of behaviour traits with the help of space analysis[11], thus comparing the space between the closed loops of letters with the space between the words and this proposed method is not concerned about the other characteristics of handwriting analysis.

At first, in our proposed work, the colour handwriting documents from surroundings or gray scale handwriting documents are collected from the IAM database. For better appearance, the gray scaled handwriting is converted to binary image. Handwriting text lines are then segmented from the handwriting images and the proposed method is applied on the segmented lines to find the space calculations between lines, words and characters on the original document. Skew normalization is applied on segmented lines to normalize them. Again it is applied to find out the space calculations on the modified skewed document similarly to get accurate results. The skew normalization technique[9] is applied on the segmented lines. On completion of these, the mean of the diameter of the loops between the letters of words is compared with the space between the words to determine the personality trait.

In the proposed method for calculating the space between lines, words and characters, firstly, it is required to find out the space between the words by segmentation[2,4] and projection profile[2]. The space values before and after skew is calculated by the previous proposed methods. Now, the mean of the diameters of all closed looped characters is calculated[32]. This mean space between the loops of these characters is compared with the space between the words of that document. The details approach for space calculations according to original and skewed documents are explained in the experimental section[34].

4. Algorithm

4.1 Algorithm 1

Line_spacing calculation on original document:
Steps:

```
1.  hor<- horizontal_projection(img); // hor is an array
    which stores sum of pixels of a row
```

```
2.  for (each element in hor)
3.  if value>10000     // checking presence of
    written pixel, value is
    received after trying out        on several samples
4.      start of new line
5.          for(each element in hor)
6.              if(value>10000)
7.                  continue;
8.              Else
9.                  line ends;
10.                 store starting and ending points
    in "lines" matrix (one-dimensional)
11.             Endif
12.         exit inner loop;
13. continue outer;
14. lines(3)-lines(2) gives the space between Line no. 1 and Line no. 2,
    lines (5)-lines (4) gives the space between Line no. 2 and Line no. 3
15. using this we calculate total space and the "total space/total line-
    count" operation gives the line space mean result
```

In the above Algorithm (*Algorithm 1*) we have calculated the space between the lines on the original document. During line extraction, we are storing the starting row pixel and the ending row pixel in an array named "Lines". For example, Lines (1) and Lines (2) depict the starting and ending row pixel respectively of Line no. 1 and likewise Lines (3) and Lines (4) depict line no. 2. Now we can calculate the space between lines by calculating the difference between Lines (3) and Lines (1).

4.2 Algorithm 2

Word_spacing calculation on original document: Steps:

Steps:

```
1.  linestart=1; pos=size_of(lines) // no. of lines*2
2.  while(linestart<=pos) do
3.    new=img(lines(linestart):lines(linestart+1)) // assigning image
      of a line in new matrix
4.        ver<- vertical_projection(new) //finding vertical projection
          in order to separate words
5.        for(each element in ver)
6.          if(value>450)              //checking presence of written
      pixel, value is tried out on other samples
7.              hash(i)=hash(i-1)+1;    //hash matrix
    defines consecutive written pixels depicting a word
8.          else
9.              hash(i)=0;
10.         endif
11.       endfor
12.       word_extract(new); //extraction of words
13.       wordstart=1; posw=size_of(words) //words contains starting
          and ending position of all words in the line
14. while(wordstart<=posw) do
15. new_word=img(words(wordstart):words(wordstart+1)) // assign-
    ing image of a line in new matrix
16. totalwordspace=totalwordspace+words(wordstart)-
    words(start-1) //calculation of word space
17. char_extract(new_word);
```

In the above Algorithm (*Algorithm 2*) we have calculated the space between the words of a line in the original document. During word extraction, we are storing the starting column pixel and the ending column pixel n a array named "Words". For eg, Words (1) and Words (2) depict the starting and ending column pixel respectively of Word no. 1 and likewise Words (3) and Words (4) depict word no. 2. Now we can calculate the space between words by calculating the difference between Words (3) and Words (1).

4.3 Algorithm 3

Char spacing calculation on original document:

Steps:

```
1.  ver<- vertical_projection(new_word) //finding vertical projection in order to separate words
2.  for(each element in ver)
3.      if(value>450)   //checking presence of written pixel, value is similar
4.      hashw(i)=hashw(i-1)+1;   //hash matrix defines consecutive written pixels depicting a character
5.        else
6.            hashw(i)=0;
7.        endif
8.     endfor
9.  /*same steps as in word_extract and here we store characters in "chars" array.*/
10. /* calculation of space*/
11. start=3;
12. while(start<=posc) //posc signifies no. of chars in the word
13.     totalcharspace=totalcharspace+chars(start)-chars(start-1);
14. endwhile
```

In the above Algorithm *(Algorithm 3),* we have calculated the space between the characters of a word in the original document. During character extraction, we are storing the starting column pixel and the ending column pixel n an array named "Chars". For example, Chars (1) and Chars (2) depict the starting and ending column pixel respectively of Character no. 1 and likewise Chars (3) and Chars (4) depict word no. 2. Now we can calculate the space between characters by calculating the difference between Chars (3) and Chars (1).

4.4 Algorithm 4

Line spacing calculation on skewed document:

Steps:

```
1.  hor<- horizontal_projection(rotated); // hor is an array which stores sum of pixels of a row
2.  for (each element in hor)
3.  if value>10000 // checking presence of written pixel, value is received similarly
4.      start of new line
5.      for(each element in hor)
6.          if(value>10000)
7.              continue;
8.          else
9.              line ends;
10.             store starting and ending pointst in "lines" matrix (one-dimensional)
11.         endif
12.     exit inner loop;
13. continue outer;
14. lines(3)-lines(2) gives the space between Line no. 1 and Line no. 2, lines(5)-lines(4) gives the space between Line no. 2 and Line no. 3
15. using this we calculate total space and the "total space/total line-count" operation gives the line space mean result
```

In the above Algorithm (*Algorithm 4*) we have calculated the space between the lines on the skewed document. This is same as *Algorithm 1*, the only exception is that it is for original image thus (img) and this is for skewed image thus (rotated) in Line No.1.

Table 2. Function description

Function Name	Function Description
imfindcircles	Find out the closed loops of word image
hash	To implement hash matrix
horizontal_projection	Horizontal Projection
lines	Assigning image of a line in a new matrix
vertical_projection	Vertical Projection to separate words
word_extract	To extract words
char_extract	To extract characters
rotate	Rotate the image with given degree

After this, *Algorithm 2* and *Algorithm 3* are repeated again to find the space between words and characters on the skewed document.

4.5 Algorithm 5

Calculation of radii of the closed loops of the letters in a skewed document:

Steps:

1. rgb=imread(img);
2. gray_image=convertToGrayScale(img);
3. [centers,radii]=imfindcircles(gray_image,[1 30]," Object Polarity", "dark", "Sensitivity", 0.9);
4. [centersBright,radiiBright]= imfindcircles(rgb,[1 30]," Object Polarity", "bright", "Sensitivity", 0.9);

In the above Algorithm (*Algorithm 5*) we use the function imfindcircles() to detect circular objects within the radius range of 1 to 30 units. By the „bright„ keyword we mean all the bright circular objects which we eventually take into the calculations of mean.

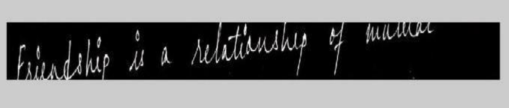


Figure 4. Sample original text line from IAM database.

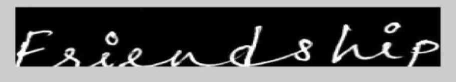


Figure 5. Sample original text word from IAM database.

Figure 6. Sample original text character from IAM database

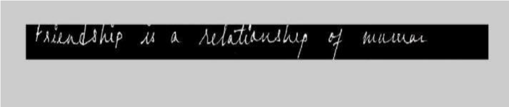
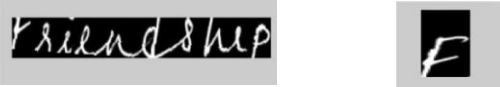


Figure 7. Sample normalized text line from IAM database.



(a) (b)

Figure 8. **(a)** Sample normalized word **(b)** Sample normalized character.

5. Experimental Result

The proposed work is implemented in MATLAB on IAM database. There are only previous proposed assumptions but no method to show their existence in actual and now proposed method can

Table 3. Space calculation result produced by proposed method on original document

Sl. no	Line Space	Word Space	Character Space
1	62.41	40.93	1.0996
2	14	11	3.42
3	28.25	19.68	0.8971
4	22	31.4	2.81
5	17.8	6.8	3.43
6	14.5	5.25	1.78
7	22.14	26.4182	1.04
8	32.37	11.59	1.53
9	7.2	23.59	1.63
10	14.21	11.24	1.72
11	13.18	13.97	1.14
12	16.37	17.94	1.53
13	9.171	10.9385	1.48
14	25.33	14.4615	0.71
15	8.07	3.27	1.21

analyze character based on calculation of the space between the lines, words and characters on the original document and on the skewed document from these segmented lines and those of closed loop of characters.

Example of original un-normalized text lines of IAM database are shown in Figure 4. First step involves segmentation of line from the original un-normalized document. Then words and characters are segmented from the text lines. Sample output for words and characters segmented from text line is shown in Figure 5 and Figure 6.

The text line, shown in Figure 4 contains the negative skew and that after applying skew normalization method where line is totally

Table 4. Space calculation result produced by proposed method on skew normalized document

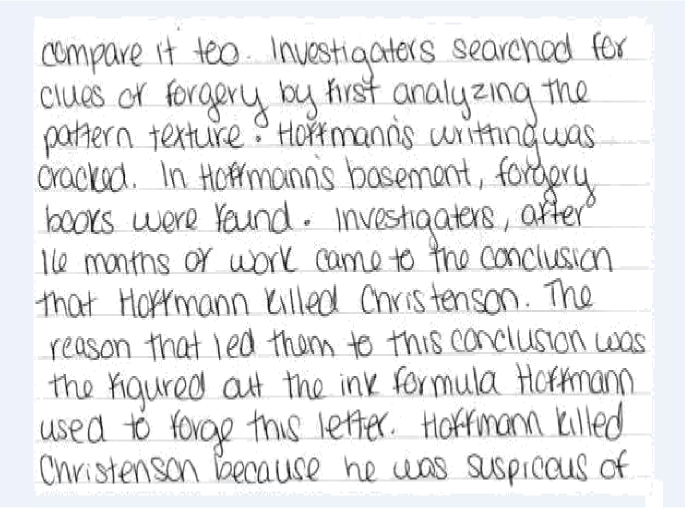
Sl. no	Line space = after Skew Normalization	Word Space after Skew Normalization	Character Space after Skew Normalization
1	33.25	13.95	1.0932
2	11	11.73	1.0932
3	38.40	22.86	1.21
4	18.52.	32.33	8204
5	9.4286	11.5625	3.7165
6	6.6429	6.3333	1.7964
7	14.1429	27.2581	1.1779
8	5.5	10.96	1.73
9	29.45	24.20	1.58
10	6.78	11.777	1.704
11	8.700	14.75	1.22
12	10.50	18.67	1.55
13	5.619	14.868	1.51
14	8.88	15.52	1.17
15	7	4.400	1.28

Table 5. Character analysis based on space calculation

Sl. No.	No. of Loops Detected	Radius (mean)	Diameter (mean)*2	Character Analysis
1	767	4.16	16.64	Narrow-minded
2	100	3.24	12.96	Narrow-minded
3	723	3.46	13.84	Broad-minded
4	872	5.10	20.40	Broad-minded
5	521	3.21	12.84	Narrow-minded

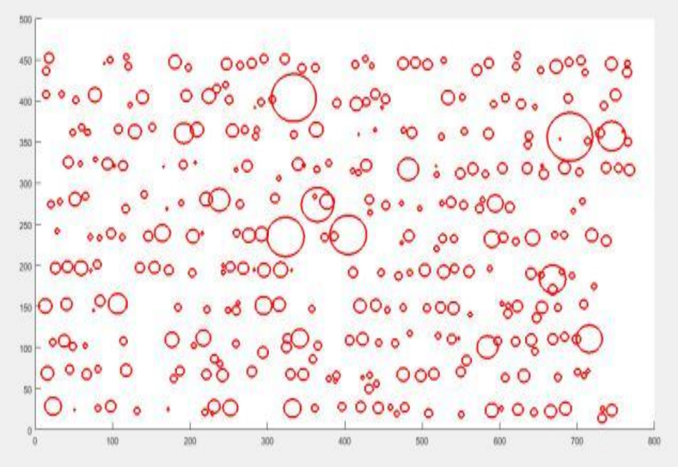
(*Contined*)

6	484	3.17	12.68	Narrow-minded
7	560	3.22	12.88	Broad-minded
8	540	3.05	12.20	Narrow-minded
9	691	4.31	17.24	Broad-minded
10	367	2.88	11.54	Narrow-minded
11	418	2.86	11.44	Broad-minded
12	449	2.50	10.00	Broad-minded
13	576	3.22	12.88	Broad-minded
14	454	3.84	7.68	Broad-minded
15	100	5.15	10.30	Narrow-minded



compare it teo. Investigators searched for
clues or forgery by first analyzing the
pattern texture. Hoffmann's writing was
cracked. In Hoffmann's basement, forgery
books were found. Investigaters, after
16 months or work came to the conclusion
that Hoffmann killed Christenson. The
reason that led them to this conclusion was
the figured out the ink formula Hoffmann
used to forge this letter. Hoffmann killed
Christenson because he was suspicous of

(a)



(b)

Figure 9 **(a)** Sample original text image from IAM database. **(b)** Sample recognized closed loops of the words of image from IAM database.

balanced and normalized with respect to baseline is shown in Figure 7. After normalization, the words and characters are segmented from the text lines and samples of segmented word and character are shown in Figure 8(a) and Figure 8(b). Training set of all the experiment has been carried out on total number of 500 text images. The experimental results before and after skew normalization process, are shown in Table 3 and Table 4.

In above Figure 8 (a) and Figure 8 (b) contains extracted word and character from skew normalized document respectively after applying the skew normalization method.

The space between the lines, words and characters are thus calculated from these images. The experiment results which were tested on the dataset are shown in Table 3. The significant results of the proposed method are shown in Tables 3 and 4.

From the above Table 5, it is cleared that proposed method is much efficient for behaviour analysis and can also deal with different skew.

If double the diameter mean of the closed loops of characters is less than word space after skew normalization, the person is broad-minded and vice-versa[35]. The success rate for this process of the proposed method is 63% better than the other methods. The failure cases of the proposed method are mainly caused by misclassified to lower case letters.

6. Conclusion and Future Work

The method has been proposed to detect the skew and normalize the skew for the writers' handwriting in current scenario. Using the proposed method 96% lines and words were normalized perfectly with very small error rate and the character analysis based on space calculation is found with 63% accuracy.

The future work can include more handwriting features with the proposed method removing the overlapping loops detected and some constant factor and obtain the more robust system.

7. Acknowledgment

The authors are very much grateful to the Department of Computer Science and engineering for giving the opportunity to work on An Efficient Methods for Character Analysis Using Space of Handwriting Image. All the authors sincerely express his gratitude to Dr. Arup Kumar Bhaumik, Principal of RCC Institute of Information Technology College for giving constant encouragement in doing research in the field of image processing.

8. References

1. Nicolas S, Paquet T, Heutte L. Text line segmentation in handwritten document using a production system. IEEE Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004); 2004.
2. Kedar S, Nair V, Kulkarni S. Personality identification through handwriting analysis. IJARCSSE. 2015.
3. Sarfraj M, Rasheed Z. Skew estimation and correction of text using bounding box. Fifth IEEE Conference on Computer Graphics, Imaging and Visualization; 2008. p. 259–64.
4. Srihari SN, Govindraju. Analysis of textual image using the Hough transform. Machine Vision Applications. 1989; 2:141–53.
5. Andria G, Savino M, Trotta A. Application of Wigner-Ville distribution to measurements on transient signals, instrumentation and measurement. IEEE Transactions. 1994 Apr; 43(2):187–93.
6. Otsu N. A threshold selection method from Gray level histogram. IEEE Transaction on System, Man, Cybernetics. 1979 Jan; SMC-9:62–6.
7. Niblack. An introduction to digital image processing. Englewood Cliffs, New Jersey: Prentice-Hall; 1986.
8. Plamondon R, Srihari SN. On-line and off-line handwriting recognition: A comprehensive survey. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2000 Jan; 22(1).
9. Marti U-V, Bunke H. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. IEEE. 2001.
10. Roy A, Bhowmik TK, Parui SK, Roy U. A novel approach to skew detection and character segmentation for handwritten Bangla words. IEEE Proceedings of the Digital Imaging Computing: Techniques and Applications; 2005.
11. Panwar S, Nain N. A novel approach of skew normalization for handwritten text lines and words. IEEE Eighth International Conference on Signal Image Technology and Internet Based Systems; 2012.

12. Gupta JD, Chanda B. Novel methods for slope and slant correction of off-line handwritten text word. 2012 IEEE Third International Conference on Emerging Applications of Information Technology (EAIT); 2012.
13. Panwar S, Nain N. Handwritten text documents binarization and skew normalization approaches. IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction; Kharagpur, India. 2012 Dec 27-29.
14. Pastor-Pellicer J, Espana-Boquera S, Zamora-Martınez F, Castro-Bleda MJ. Handwriting normalization by zone estimation using HMM/ANNs. 2014 IEEE 14th International Conference on Frontiers in Handwriting Recognition; 2014 .
15. Roy PP, Dey P, Roy S, Pal U, Kimura F. A novel approach of bangla handwritten text recognition using HMM. 2014 IEEE 14th International Conference on Frontiers in Handwriting Recognition; 2014.
16. Mohammed bin Abdl K, Mohd Hashim SZ. Handwriting identification: A direction review. IEEE International Conference on Signal and Image Processing Applications; 2009.
17. Abdul Rahiman M, Varghese D, Manoj Kumar G. HABIT: handwritten analysis based individualistic traits prediction. International Journal of Image Processing (IJIP). 2013.
18. Simistira F, Papavassiliou V, Stafylakis T. Enhancing handwritten word segmentation by employing local spatial features. IEEE International Conference on Document Analysis and Recognition. 2011.
19. Zimmermann M, Bunke H. Automatic segmentation of the IAM Off-line Database for Handwritten English Text. IEEE. 2002.
20. Liwicki M, Bunke H. IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard. IEEE Proceedings of the 2005 Eight International Conference on Document Analysis and Recognition (ICDAR'05); 2005.
21. Luthy F, Varga T, Bunke H. Using Hidden Markov Models as a tool for handwritten text line segmentation. Ninth International Conference on Document Analysis and Recognition (ICDAR'2007); 2007.
22. Stafylakis T, Papavassiliou V, Katsouros V, Carayannis G. Robust text-line and word segmentation for handwritten documents images. IEEE Greek Secretariat for Research and Technology under the Program; 2008.
23. Suarez SPD, Mello CAB, Oliveira ALI, Alves VMO. Text line segmentation in images of handwritten historical documents. IEEE Image Processing Theory, Tools and Applications; 2008.
24. Gupta JD, Chanda B. A model based text line segmentation method for off-line handwritten documents. 2010 IEEE 12th International Conference on Frontiers in Handwriting Recognition; 2010.
25. Javed M, Nagabhushan P, Chaudhuri BB. Extraction of line-word-character segments directly from run-length compressed printed text-documents. IEEE; Jodhpur. 2013 Dec 18-21. ISBN:978-1-4799-1586-6.
26. Zhang X, Tan CL. Text line segmentation for handwritten documents using constrained seam carving. 2014 IEEE 14th International Conference on Frontiers in Handwriting Recognition; 2014 .
27. Zimmermann M, Bunke H. Hidden Markov Model length optimization for handwriting recognition systems. IEEE Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR'02); 2002.
28. Champa HN, Ananda Kumar KR. Automated human behavior prediction through handwriting analysis. 2010 IEEE First International Conference on Integrated Intelligent Computing; 2010.
29. Papavassiliou V, Stafylakis T, Katsouros V, Carayannis G. Handwritten document image segmentation into text lines and words. Pattern Recognition. 2010 Jan; 43:369–377. DOI: 10.1016/j.patcog.2009.
30. Solihin Y, Leedham CG. Noise and background removal from handwriting images. IEEE; 1997.
31. Bal A, Saha R. An effective method for skew normalization in handwriting analysis. Proceedings of the IEEE International Conference CSNT-2016; Mar 2016.
32. Bal A, Saha R. An improved method for text segmentation and skew normalization of handwriting image. Proceedings of the Springer International Conference (ICACNI-2016); 2016 Sep.
33. Story G, O'Gorman L, Fox D, Schaper L, Jagadish H. The right pages image-based electronic library for alerting and browsing. Computer. 1992 Sep; 25(9):17–26.

34. Dounskaia N, Van Gemmert A, Stelmach GL, Leis BC. Biased wrist and finger coordination in handwriting analysis. Pattern Recognition. 2009 Oct; 9:351–60.

35. Rice L. Practical graphology or the science of reading character through handwriting. Kessinger Publishing; Sep 2010. ISBN: 10:1163173592, ISBN: 13:978-11631735.