

Handwritten English Character Segmentation by Baseline Pixel Burst Method (BPBM)

* R. K. Mandal, **N R Manna

*Department of Computer Science & Application, University of North Bengal,
Siliguri, Distt : Darjeeling, West Bengal-734013, India (rakesh_it2002@yahoo.com)

** Department of Computer Science & Application, University of North Bengal,
Siliguri, Distt : Darjeeling, West Bengal-734013, India (nrmanna@sify.com)

Abstract

Segmentation of characters from English words is a very challenging task in the field of hand writing recognition. Difficulties arise because of complicated behavior of the handwriting patterns of different individuals. Different people write in different hand writing styles which is a mixture of joined and isolated characters. In this paper, a Base Line Pixel Burst Method (BPBM) has been tried out on a paragraph taken from an individual whose handwriting consists of both isolated and joined characters. For the isolated ones, the segmentation is done by identifying the gaps between characters but it becomes difficult to segment out joined characters as there is no standard method for this. A method has been developed here which actually draws baselines on various rows of the image word matrix and vertical lines tries to find out character boundaries by analyzing the densities of the pixels near the baselines.

Keywords: Segmentation, Handwritten Characters, Contour, Binary Image Matrix

1. Introduction

To recognize the characters of a handwritten text [28, 29, 30], it is important to segment out the characters accurately. The proper segmentation of the characters from the text directly puts an impact on the success of the handwriting recognition method forming a dilemma known as chicken-egg relationship [1]. The easiest way to segment the characters is to identify the gaps present between the characters, but this is possible only for those handwriting styles which

contains isolated characters [2, 3, 4]. The problem of joined characters arises in many non English scripts like Gurumukhi, Urdu, Bengali etc. [5, 6, 7]. Joined handwriting characters may be written using one or multiple strokes. This may create ambiguities while segmentation which is known as character-within-character problem, like ‘d’ can sometimes be recognized as ‘cl’, if written in separate strokes, [8]. Some sort of preprocessing may also be required before and after segmentation of the characters in order to remove or reduce the ambiguities.

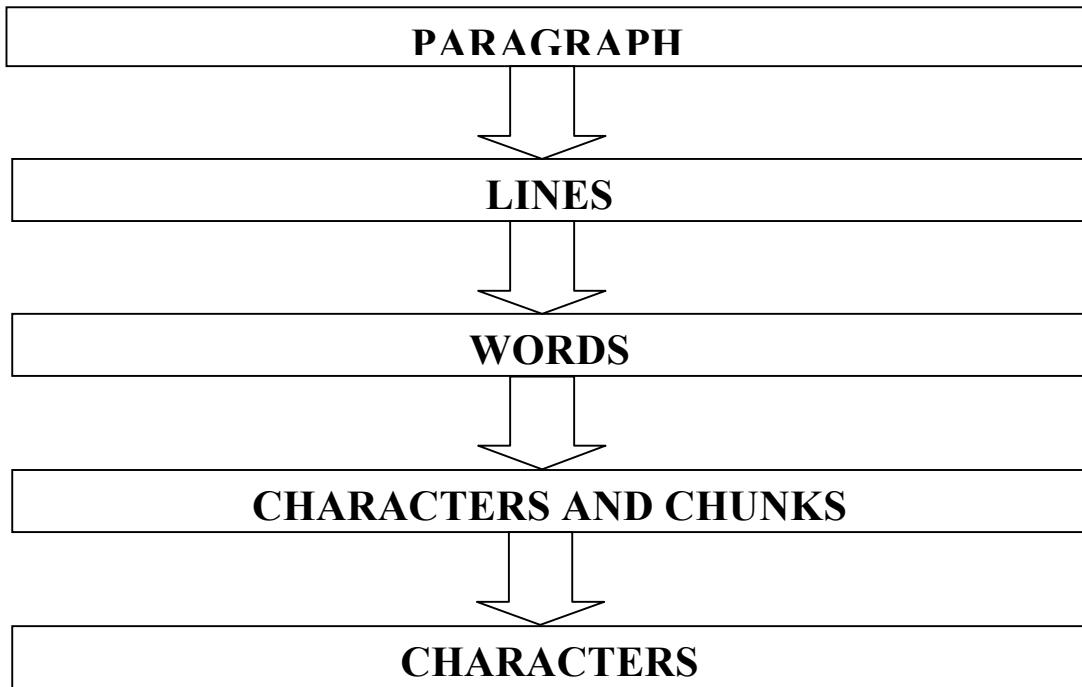


Figure 1. Conceptual diagram of BPBM

In this paper an approach, Base Line Pixel Burst Method (BPBM), simplifies the method by segmenting as many as characters by applying a suitable gap finding procedure. Joined characters can be segmented by drawing baselines at different rows of the word or part of the word image matrix containing joined characters and analyzing densities of the pixels near the baseline to find out the locations of the joints of two different characters.

Ram is a good boy
Cat likes to eat fish
Ram is a good boy
Cat likes to eat fish

Figure 2. Initial paragraph

The overall program is divided into five parts. Firstly, paragraph containing the text to be segmented is to be obtained and converted into a two dimensional matrix. Secondly, paragraph is segmented into separate lines of text. Then the lines are segmented into separate words. A suitable gap finding method is used to extract out the isolated characters as well as chunks of words containing joined characters, thereafter. Finally, the joined characters are extracted out. Figure 1 displays the conceptual diagram and working of the overall method.

2. Preparing the Input

A high definition scanner is used to scan a handwritten paragraph. The paragraph is converted into a two dimensional binary image matrix. Figure 2 shows a paragraph containing four sentences taken from two individuals. The paragraph is enclosed within a rectangular boundary by using contour tracing method, [9, 10, 11, 12].

3. Segmentation of lines from a paragraph

The binary paragraph image matrix just obtained is considered for the segmentation of lines and is presented to algorithm 1, for the extraction of the lines from the paragraph, (Figure 3).

Ram is a good boy

Figure 3. First line of the paragraph

The process is repeated till the end of the paragraph is reached extracting out all the lines of the paragraph.

Algorithm 1. Segmentation of lines from a paragraph

Step 1. Scan the paragraph and store it in bmp format.

Step 2. Convert the bmp format into a two dimensional binary matrix.

Step 3. Locate the topmost and bottommost rows of the binary matrix.

Step 4. Locate the extreme left and extreme right columns of the binary matrix.

Step 5. Form a contour touching the edges of the binary matrix by using a contour tracing algorithm.

[A contour is a rectangular or square boundary enclosing the text.]

[Here a contour tracing algorithm is used in which a contour is formed around the paragraph by drifting the topmost row downwards until the first black pixel, which is the part of the paragraph is found. Similarly, bottommost row is drifted upwards and extreme left and right columns are drifted towards right and left side of the paragraph to find the first black pixel in their respective sides.]

Step 6. Drift the topmost row of the contour downwards to find the first row containing no black pixel, locate the row and store the location using a variable.

[The row containing no black pixel is the end of the first line of the text.]

Step 7. Drift the row downwards from the current position to find the first row containing one or more black pixels, locate the row and store the location using a variable.

[The row containing one or more black pixels is the beginning of the next line of the paragraph.]

Step 8. Test the stopping condition.

[Here stopping condition is checked by testing the location of the current row having no black pixel. If the location exceeds the location of the bottommost row of the contour, which indicates the end of the paragraph, stopping condition is set to true.]

Step 9. If stopping condition is true GOTO Step 10 else GOTO Step 6.

Step 10. Using various locations of the lines of the paragraph stored in different variables extract out different lines of the paragraph using functions available in the MATLAB and store in different dimensional matrices having different names.

Step 11. Stop

4. Segmentation of the words from the lines of the paragraph

The lines obtained by applying Algorithm 1, is considered for the segmentation of words. These lines are presented to algorithm 2, for the extraction of the words from the lines of the paragraph, (Figure 4).

Algorithm 2. Segmentation of the words from the lines of the paragraph

Step 1. Read the lines of the paragraph stored in different matrices.

Step 2. For each line of the paragraph repeat Step 3 to Step 8.

Step 3. Locate the extreme left column of the line of the paragraph and drift the column towards the right to find the first blank column having no black pixel.

[The first blank column having no black pixel may be the end of the word or the character.]

Step 4. Test the successive columns for the presence of black pixels.

Ram	is	a	good	boy
Ram	is	a	good	boy
Cat	likes	to	eat	fish
Cat	likes	to	eat	fish

Figure 4. Segmented words of the first line of the paragraph

Step 5. Store the number of successive blank columns having no black pixel in a variable, say GAP_FOUND.

Step 6. Compare the value of the variable with a fixed value stored in a variable, which is chosen randomly, say GAP_MAX.

Step 7. If GAP_FOUND < GAP_MAX Then GOTO Step 4 Else GOTO Step 8

[The gap between two words is always greater than the gap between two characters of a word. GAP_MAX is the gap allowed between two words of a line.]

Step 8. Locate the column, store the location in a variable and extract out the word using MATLAB and store in a matrix.

Step 9. Stop.

5. Segmentation of the characters from the words of the line

The words obtained by applying Algorithm 2, is considered for the segmentation of isolated characters and group of characters joined together to appear as a single word. These words are presented to Algorithm 3, (Figure 5).

Algorithm 3. Segmentation of the characters from the words of the line

Step 1. Read the words of the lines of the paragraph stored in different matrices.

Step 2. For each word of the line of the paragraph repeat Step 3 to Step 4.

Step 3. Locate the extreme left column of the word of the line of the paragraph and drift the column towards the right to find the first blank column having no black pixel.

[The first blank column having no black pixel indicates the end of the character or gap between disjoint chunks of a word.]

Step 4. Locate the column, store the location in a variable and extract out the character or chunk of the word using MATLAB and store in a matrix.

Step 5. Stop.

6. Segmentation of the characters from the joint words appeared as characters

The words obtained after applying Algorithm 3 is considered for the segmentation of the individual characters. These words are presented to Algorithm 4, (Figure 6).

Ram	is	a	go	o	d	b	oy
Ram	is	a	go	d	boy	-	-
Cat	likes	to	eat	fish	-	-	-
Cat	likes	to	eat	fi	s	h	-

Figure 5. Segmented chunks of text from the different words of the paragraph

The idea in algorithm 4 is to locate densities of pixels at different locations of the column which partitions the character. Actually, it is found by experiments that only a single burst of pixels is found on the column near the baseline.

Figure 7, represents the extraction of single characters from a chunk of three joined characters using the following Algorithm.

Algorithm 4. Segmentation of the characters from the joint words appeared as characters

Step 1. Read the portion of the word, having joined characters, as a two dimensional matrix.

Step 2. Repeat Step 3 to 14, for all the characters or chunks, present in the matrix.

Step 3. Drift upwards, starting from the bottom row of the matrix, taking one row at a time, stop at the location, where, the row finds its first black pixel and draw a horizontal baseline at that location and store the location using a variable.

R	a	m	i	s	i	s	a
C	a	t	c	a	t		a
L	i	k	e	s		t	D
l	i	k	i	h		t	o
e	a	t	e	a	t		
f	i	s	h	t	i	b	h
g	o	o	d	j	o	d	d
b	o	r	b	o	r		

Figure 6. Segmented characters from the different joined words appeared as characters

Step 4. Drift towards the rightmost position, starting from the leftmost column taking one column at a time, stop at the location, where the column finds its first black pixel and draw a vertical solid line on that location and store the location using a variable.

[Vertical solid line shows the beginning of the character]

Step 5. Drift towards the rightmost position, starting from the current location of the solid line, taking one column at a time, stop at the location, where the column finds a single burst of black pixels near the baseline and draw a vertical line at that location and store the location using a variable.

[Vertical line may be the end of the character. While drifting towards rightmost position, the succeeding columns, show bursts of black pixels, showing the presence of the character image, the end of the character image shows the pixel burst only at the joint which is mostly found near the baseline.]

Step 6. Draw a dotted line with end points at the middle of the starting and ending lines of the suspected character.

Step 7. If numbers of bursts of black pixels on the dotted line is greater than 1 then convert the vertical line into a solid line and go to step 12 else go to step 8.

[The character may be ‘i’ or a portion of ‘u’, ‘w’, ‘y’ or ‘v’]

Step 8. Drift downwards from the current location of the upper horizontal line, stop at the first black pixel and draw another horizontal line at that location, with end points at the starting and ending position of the character under observation.

Step 9. Drift downwards from the current location of the horizontal line, stop at the location having no black pixel on the horizontal line and draw another horizontal line at that location.

Step 10. Drift downwards from the current location of the horizontal line, stop at the first black pixel and draw another horizontal line at that location. [Steps 8, 9 and 10 shows the presence of character ‘i’ or ‘j’]

Step 11. If Steps 8, 9 and 10 are met then convert the vertical line ending at the character having single burst of pixel into a solid line else go to step 4.

Step 12. Start from the current location and repeat Step 3 to find the starting location of the next character. Draw a solid line on that location.

Step 13. Start from the current location and repeat step 4 to find the end of the next character. Draw a solid line on that location.

Step 14. Extract out all the suspected characters using MATLAB.

7. Result Analysis

The experiment produces satisfactory results for the test paragraph. Following are the details of the results produced by the method after presenting a paragraph:

Number of paragraphs tested = 1

Number of lines present in the paragraph = 4

Number of lines identified by applying algorithm 1 = 4

Accuracy of identifying the lines = 100 %

Number of words present in the paragraph = 20

Number of words obtained after applying algorithm 2 = 20

Accuracy of identifying the words = 100 %

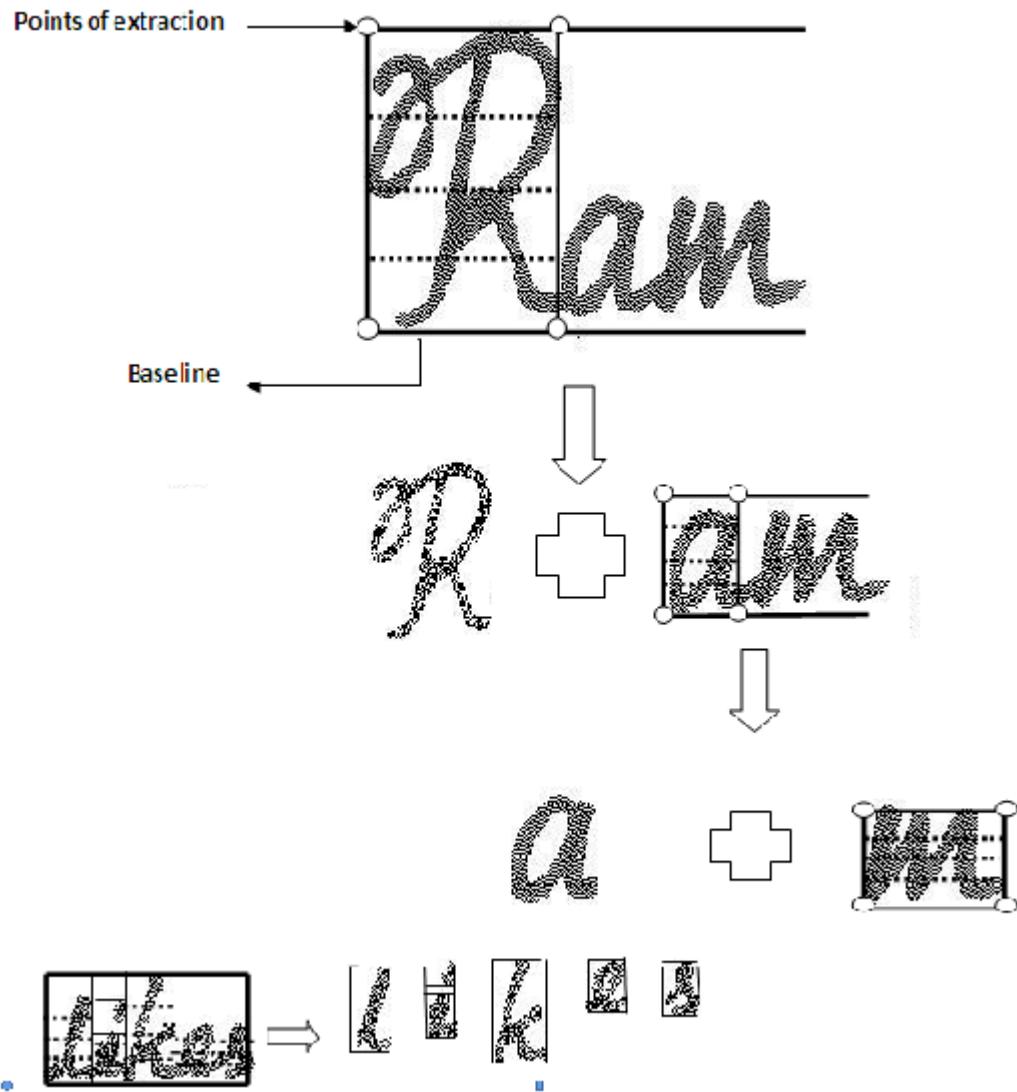


Figure 7. Extraction of single characters from joined chunks

Number of chunks present in the paragraph = 25

Number of chunks obtained after applying algorithm 3 = 25

Accuracy of identifying the chunks containing joined characters from words = 100 %

Number of characters present in the paragraph = 60

Number of characters identified with by algorithm 4 = 57

Accuracy of identifying the words with noise = 95%

8. Discussion

BPBM is applied for the paragraph containing characters which may or may not have spaces in between. The method shows very good results for the isolated characters and shows satisfactory results for the joined or cursive type of handwriting. But BPBM can recognize cursive type of handwriting with noise in those characters which are having ambiguous features like u and w where it is difficult to find the existence of single ‘w’ or double ‘u’ in case of the presence of two consecutive ‘u’. It is also difficult to extract out characters having overlapping boundaries if a vertical line segment is used to separate two characters. To recognize cursive type of handwriting, better methods can be developed which will even be helpful to recognize complicated handwriting types and reduce the noise in the characters. The idea behind BPBM is movement of a vertical column to find out the pixel densities near the baselines, so sometimes it has become very difficult to identify the boundaries in case of garbled characters. Non linear methods can be developed to find out the sharp boundaries.

BPBM is compared with other character segmentation methods in terms of their success rate and it is found that the performance is far better than other character segmentation methods. Table 1 displays the performance of other character segmentation method and BPBM.

Table 1. Comparison of BPBM with already developed segmentation method

S.No.	Method used	Success Rate (SR)	Script	Reference
1.	Neuro-Heuristic Approach for Segmentation	69.72%	Arabic	[13]
2.	Segmentation of Cursive English Handwriting	85.74%	English	[14]
3.	Component Based Segmentation	85.00%	Arabic	[15]

S.No.	Method used	Success Rate (SR)	Script	Reference
4.	Heuristic Technique	82.98%	Arabic	[16]
7.	Feature and Rule based	81.08%	Latin	[17]
8.	ANN	83%	English	[18]
9.	Heuristic Algorithm	85.7%	Latin	[19]
10.	ANN	90%	Latin	[20]
11.	ANN	75.9%	Cursive Latin	[21]
12.	ANN and Conventional Method	81.21%	Latin	[22]
13.	Rule based	86.9%	Latin	[23]
14.	Feature based ANN	91%	Latin	[24]
15.	Feature based ANN	78.85%	Latin	[25]
16.	Feature based ANN	84.87%	Latin	[26]
17.	Enhanced Feature based ANN	84.19%	Latin	[27]
18.	Baseline Pixel Burst Method (BPBM)	95.%	English	-

9. Conclusion

This work has been carried out to actually to find out those characters present in the paragraph which are joined and basically of cursive style. The method works very well in extracting out lines from the paragraph. The words can easily be extracted out from the lines. The isolated chunks of words and the isolated characters can also be easily detected. Most of the joined characters are identified. The difficulty arises while extracting out joined characters having

overlapping boundaries and ambiguous in nature in shape. Some noise detection methods are tried out here and results obtained was satisfactory for the sample paragraph. Work is going on to develop some better noise removal methods.

References

1. H Lee, B Verma, “Binary Segmentation algorithm for English cursive hand writing recognition”, Central Queensland University, Australia, ‘<http://dx.doi.org/10.1016/j.patcog.2011.09.015>, How to Cite or Link Using DOI’.
2. R K Sharma, A Singh, “Segmentation of Handwritten Text in Gurmukhi Script”, International Journal of Image Processing, Volume 2, Issue 3, pp 12-17.
3. Y Lu, “Machine Printed Character Segmentation – an Overview”, Pattern Recognition, Volume 29(1), pp 67-80, 1995.
4. V Bansal, R M K Sinha, “Segmentation of touching and Fused Devnagri characters”, Pattern recognition, Volume 35, pp 875-893, 2002.
5. M K Jindal, G S Lehal, R K Sharma, “Segmentation Problems and Solutions in Printed Degraded Gurmukhi Script, IJSP, Volume 2(4), 2005.
6. G S Lehal, C Singh, “Text Segmentation of machine pintaed Gurmukhi script”, Document Recognition and Retrieval VIII, Proceedings SPIE, USA, Volume 4307,pp 223-231, 2001.
7. U Pal, S Dutta, “Segmentation of Bangla Unconstrained Handwritten Text”, Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.
8. C C Tappert, S H Cha, “English Language Handwriting Recognition Interfaces”, Chapter 6 in Text Entry Systems, ed. Mackenzie and Tanaka-Ishii, Morgan Kaufman, pp 1-19, 2007.
9. Y E Saady, A Rachidi, M E Yassa, D Mammass, “Amazigh Handwritten Character Recognition based on Horizontal and Vertical Centerline of Character”, International Journal of Advanced Science and Technology, Volume 33, August, 2011, Available : <http://www.sersc.org/journals/IJAST/vol33/4.pdf>, October, 2012

10. Y Amrouch, Y E Saady, A Rachidi, M E Yassa, D Mammass, “Handwritten Amazigh Character Recognition System Based on Continuous HMMs and Directional Features”, International Journal of Modern Engineering Research (IJMER), Volume 2, Issue 2, pp 436-441, 2012, Available: http://www.ijmer.com/papers/vol2_issue2/BZ22436441.pdf, October, 2012.
11. S Nobesawa, J Tsutsumi, T Nitta, K Ono, S D Jiang, M K Nakanishi, “Segmenting a Sentence into Morphemes using Statistic Information between Words”, Available : <http://acl.ldc.upenn.edu/C/C94/C94-1036.pdf>, October, 2012.
12. D D. Palmer, “Tokenisation and Sentence Segmentation”, The MITRE Corporation, Available: <http://comp.mq.edu.au/units/comp348/ch2.pdf>, October, 2012.
13. A Hamid, R Haraty, “A Neuro-Heuristic Approach for Segmenting Handwritten Arabic Text”, Proceedings of ACS/IEEE International Conference on Computer Systems and Applications, pp 0110, 2001.
14. R Casey, E Lecolinet, “A survey of methods and strategies in character segmentation”, IEEE Trans. Pattern Analysis and Mach, Volume 18, pp 690-706, 1996.
15. H A Jawad, J Jianmin, R Jinchang, S I Stan, “Component-based Segmentation of Words from Handwritten Arabic Text”, Proceedings of World Academy of Science, Engineering and Technology, Volume 31, pp 1307–6884, 2008.
16. Hamad, A A Husam, “Over-segmentation of handwriting Arabic scripts using an efficient heuristic technique”, Proceedings of Wavelet Analysis and Pattern Recognition (ICWAPR), IEEE, pp 180-185, 2012.
17. C C Tappert, C Y Suen, T Wakahara, “The state of the art in on-line Handwriting Recognition”, IEEE Trans. Pattern Analysis. Machine. Intelligence. Volume 12: pp 787-808, 1990.
18. S N Srihari, “Recognition of handwritten and machine-printed text for postal address Interpretation”, Pattern Recognition Letters, pp 291-302, 1993.
19. K Han, I K Sethi, “Off-line cursive handwriting segmentation”, ICDAR 95, Montreal, Canada, pp 894-897, 1995.

20. S W Lee, D J Lee, H S Park, “A new methodology for gray-scale character segmentation and recognition”. IEEE Transaction on Pattern Analysis and Machine Intelligence, pp 1045-1051, 1996.
21. B Eastwood, A Jennings, A Harvey, “A feature based neural network segmenter for Handwritten words”, ICCIMA’97, Australia, pp 286-290. 1997.
22. M Blumenstein, B Verma, “A segmentation algorithm used in conjunction with artificial neural networks for the recognition of real-word postal addresses”, In Proceeding of International Conference on Computational Intelligence and Multimedia Applications (ICCIMA’97), Gold Coast, Australia, pp 155-160, 1997.
23. G Nicchiotti, C Scagliola, “Generalized projections: a tool for cursive handwriting Normalization”, In Proceedings of 5th International Conference on Document Analysis and Recognition, Bangalore, pp 729-733, 1999.
24. B Verma, P Gader, “Fusion of multiple handwritten word recognition techniques, Neural Networks for Signal Processing X”, In Proceedings of the IEEE Signal Processing Society Workshop, Volume 2, pp 926-934, 2000.
25. M Blumenstein, B Verma, “Analysis of segmentation performance on the CEDAR benchmark database”, In Proceedings of Sixth International Conference on Document Analysis and Recognition (ICDAR'01), pp 1142, 2001.
26. B Verma, “A contour character extraction approach in conjunction with a neural confidence fusion technique for the segmentation of handwriting recognition”, Proceedings of the 9th International Conference on Neural Information Processing, Volume 5, pp 18-22, 2002.
27. C K Cheng, M Blumenstein. “Improving the segmentation of cursive handwritten words using ligature detection and neural validation”. In Proceedings of the 4th Asia Pacific International Symposium on Information Technology (APIS 2005), Gold Coast, Australia, pp 56-59, 2005.
28. Al-Shridah, Nuhammad, Sharieh, Ahmad, “Recognition process of handwritten and typed Arabic letters”, AMSE Journal, Advances in Modelling, Signal Processing and Pattern Recognition, Volume 45, Issue 1, pp 1, 2002.

29. N Liolios, K Anastasiou, B Kostos, “A new shape transformation approach to handwritten character recognition”, AMSE Journal, Advances in Modelling, Signal Processing and Pattern Recognition, Volume 46, Issue 6, pp 55, 2003.
30. K G Srinivasagan, L Ganesan, S Arumugam, AMSE Journal, Advances in Modelling, Signal Processing and Pattern Recognition, Volume 50, Issue 2, pp 64, 2007.