BANGLA HANDWRITTEN TEXT SEGMENTATION FOR OPTICAL CHARACTER RECOGNITION

a dissertation submitted in partial fulfilment of the requirements for the M.Tech. (Computer Science) degree of the Indian Statistical Institute

By

ARIJIT BISHNU
(MTC9610)

Under the guidance of

Prof. B.B. CHAUDHURI
Head,
Computer Vision & Pattern Recognition
INDIAN STATISTICAL INSTITUTE
203, B. T. Road
Calcutta-700 035

1998

ACKNOWLEDGEMENT

First things first. In writing this, my mind goes back to the day I first met my guide, Prof. B.B.Chaudhuri. I was interested in Image Processing which led me to him. On the very first day I met him he had a very detailed talk with me in which he apprised me of what research is and how it is to be approached. To this day I can recall the very minutest details of that discussion. It had a profound impact on me. He only introduced me to the field of OCR. From that day onwards it has been a period of developing heuristics one after another. In this I would be ever grateful to my guide for showing me the correct path. At times he would chide me in his mildest manners which would goad me deeper into my work. I hope, in earnest, that I can be associated with him in further research interests. Sir, let me thank you for what you have been to me, for all your guidance and help.

To Umapada Pal and Utpal Garain, I would be highly thankful for the many discussions we had on OCR related topics. I would like to thank P. S.Umesh Adiga for helping me with the keys of the CVPRU lab at odd hours on holidays. To Arnab Nandi, my hostel mate, I owe a lot for it was his own PC on which the final part of the work was done.

Certificate of Approval

This is to certify that the dissertation work entitled **Bangla Handwritten Text Segmentation for Optical Character Recognition** submitted by Arijit Bishnu, in partial fulfilment of the requirements for M. Tech. in *Computer Science* degree of the *Indian Statistical Institute, Calcutta,* is an acceptable work for the award of the degree.

Date : July 29 , 1998 .

B. B. Chaudhuri
(Supervisor)

P. K Nandi 29/7/98
(External Examiner)

ABSTRACT

This dissertation work puts forward, to be specific, two methods for segmentation of Bangla handwritten text into characters for Optical Character Recognition, OCR in brief. Given a text, we propose a method to segment words from text. Now, with each word we proceed towards its segmentation into characters. We detect different zones across the height of the word based on certain characteristics of Bangla writing methods. These zones give certain structural information about the respective word and its constituent characters. Thereafter, we approach segmentation of words into characters - two methods are proposed. One based on vertical histogram and distance concepts, and the other one on recursive contour following and bounding box method. Limitations of these methods are also discussed with examples.

Key words: *OCR, character segmentation, histograms and distance, contour follow and bounding box.*

CONTENTS

TOPICS

1. INTRODUCTION

Digital Document Processing is gaining popularity for its immense potential in office and library automation, bank and postal services, publishing houses. Coupled with speech processors, where a text document after recognition can be converted into speech, has also got use as hearing aid for the blind.

Machine reading of optically scanned text is usually called OPTICAL CHARACTER RECOGNITION, OCR in brief .It is a process of automatic computer recognition of characters in optically scanned and digitised pages of text. OCR is one of the most fascinating and challenging areas of pattern recognition with various practical applications, a few of which has already been mentioned. The input of an OCR system consists of text on paper. The output is a coded file with some character code representation. Talking about some practical applications of OCR system, here are some:

- Reading aid for the blind, it is a joint application of OCR and speech synthesis.
- Automatic text entry into the computer, desktop publications, library cataloguing, ledgering.
- Automatic reading for sorting of postal mail, bank cheques, and other documents.
- Document data compression, in the sense that, say an ASCII formatted text requires less space than a document image.
- Language processing.
- Multimedia system design.

1.1 *About Bangla , the language*

Bangla is a traditionally rich language, being the national language of Bangladesh, and the secondmost popular language in India. It is spoken and written widely in West Bengal, a state of India. Over and above it is the fifth most popular language in the world. There are a number of popular publishing houses publishing in Bangla. Bangla script alphabet is used in texts of Bangla, Assamese and Manipuri languages. About 200 million people in the eastern part of Indian subcontinent speak this language.

1.2 *Properties of the Bangla script*

All major Indian scripts including Bangla are mixtures of syllabic and alphabetic scripts. They are varied in character and forms. The Bangla script is derived from the Brahmi script through various transformations. Writing style of Bangla is from left to right in a horizontal manner.

Many Bangla characters have a horizontal line at the upper part. This line is called *'matra'* or *headline. Henceforth*, the *headline* will be referred to as *matra* in our discussion. It plays an important role in our Bangla OCR system. It helps us to locate the script line, to segment the characters in a word and can also be used in further recognition schemes.

A vowel following a consonant sometimes takes a modified (allographic) shape, which depending on the vowel is placed at the left, right (or both) or bottom of the consonant. These are called *vowel modifier.*

Printed Bangla has got many significant characteristics, like presence of headline to name one, which are missed by people when they write themselves. The absence of these characteristics in handwriting makes OCR of Bangla handwriting a trifle difficult over OCR of
printed Bangla text documents.

1.3 *Approach*

In printed Bangla script the word can be partitioned generally into three zones[1]. The upper zone denotes the position above the headline, the middle zone covers the portion of basic (and compound) characters below matra and the lower zone is the portion where some modifiers, as has been previously mentioned, can reside. But hand-written documents have not got the uniformity that is a characteristic of printed documents. So in our work we have proposed the word to be partitioned into four zones[4]. The upper zone remains the same as printed characters. After that we define a zone below the matra which is supposed to be the zone of joining one character to the next character. Thereafter comes the middle zone which covers the basic character portion, and the lower zone is the part where some modifiers can reside.

Having found the zones which are structurally very informative about the connection and expanse of characters, our next step is to proceed for character segmentation based on the detection of zones in the word. The first method is basically is a vertical segmentation method, where the projection of the object pixels of the word are summed columnwise and the relative deep minima points give us an idea as to where the segmentation point or zone lies. The distance of an object pixel from a certain zone, considered to be its weight, prevents a valid character to be further segmented. The second method recursively follows a character contour within certain bounds and finds out the extent within which the character lies. This extent is called the bounding box of the character.

To the best of our knowledge, this is the first work of its kind in Bangla handwritten text segmentation.

2. IMAGE DATA COLLECTION AND DISPLAY

2.1 *Image data collection*

We collected several handwritten documents. We asked people to write on blank papers with a shadow of lines underneath the paper so that they could write on a guided straight line. This we did to exploit a particular feature of Bengali writing.

Thereafter, we scanned the documents on a flat-bed scanner and stored the document as an image in tiff file format. The documents were scanned at 200 or 300 dpi with maximum contrast and medium brightness.

2.2 *Displaying and thresholding*

The image file is displayed after proper thresholding on the screen so that binary images are seen. Here the black pixels belong to characters and the white pixels form the background.

3. WORD SEGMENTATION

3.1 *Detection of text lines and zones*

First of all we discriminate the text lines from the image displayed. We sum up the black pixels on each row. Now, there must be a gap between two lines of a document. So if we
see the values of the number of black pixels on a row we will find minimum values almost nearing zero in rows where a gap between two lines exists. We thus pick up the zones of written lines from the displayed document.

3.2 *Estimation of the matra in the selected zone*

Here we make a tacit approximation that the probability of a person giving strokes of matra in a detected zone of line is much higher than giving strokes of matra in a single word. So we determine the zone or row, to be particular where the concentration of black pixels is the greatest and call this the matra zone. Let us denote this value as 'R1'.

3.3 *Detection of words*

Now, we go for picking up the words from the zones of written lines. Here again we expect gaps between words. Now these gaps are classified as follows:

a) Inter word gaps: The gaps between valid words are known as inter word gaps.
b) Intra word gaps: The gaps between the characters of a word are known as intra word gaps.

Here again we cannot have an apriori knowledge about the inter and intra word gaps. So to circumvent the problem we take into account the white background gaps between written documents in the zone already detected as a line of writing. We count

1) the number of such white gaps in a zone

and 2)the range of the values of the width of the continuous white zones (By range we mean the difference between the maximum and minimum gaps of continuos bands of white zones),

and 3) we also keep into account the total sum of the said gaps.

Based on the above attributes we decide the optimum word gap and thus segment the words from a zone. Thus picking up of words from a document is done.

3.4 *How the words are stored*

The words are stored in separate files with the following information as header:

1. The width of the word, W1.
2. The height of the word, H1.
3. The value of the overall matra in the selected zone, R1.

The values of the rows and columns of the black pixels are only stored. Thus we can achieve some sort of compression on the size of the word files.

4. ZONE DETECTION IN A WORD

In order to proceed for the character segmentation, we take up the segmented word files. Taking the values of the width, height of the word from its header, the word is displayed.

4.1 *Smoothing and filtering of the displayed image*

The displayed word image might have spurious noise in forms of one or two isolated black pixels. The cleaning of the image is done in the following way:

Steps ::

1. *Scan the image for black pixels.*
2. *For each black pixel count the number of black pixels in its neighbourhood .*
3. *If the number < = 2, turn the pixel white i.e. make it a neighbourhood point.*

Thereafter, we proceed to determine the matra zone, which gives immense structural information about the connection of one character to another character. As hand-written documents have wide variations across the writers, it becomes increasingly difficult to locate exact positions of different zone. We first try to form an idea about certain
zones of a word.

4.2 *Determination of matra location*

It had been mentioned earlier we made people write with the guideline of a straight line underneath the paper. This was done purposefully to exploit the feature of matra in a word. While we had stored the file, we had kept the information as to where the possible matra zone lies in a selected zone which actually forms a line of the document. This value was referred to as 'R1'.

Now to determine where the possible matra lies in the word, we take into account three factors which are as follows:

4.2.1 *Formation of horizontal histogram*

We first form the horizontal histogram of the displayed word. The horizontal histogram basically is an array whose x th value is calculated as the sum of black pixels in
that particular x th row . Now we find the maximum value from this horizontal histogram.

We assume that this value (say, M1) will be a good indicator of the matra location.

Steps : :

for (j = 1 ; j < = rows ; j++)
for (k = 1; k<=columns;k++)
if the (j,k) th pixel is black
histogram[j] = histogram [j]+ 1
end if
end for
end for.

M1 = *maximum (histogram [ ]).*

4.2.2 *Formation of run length histogram*

Next we take into account the run length of the black pixels in a particular row. The run length is defined as the continuous run of black pixels. We count the length of each such continuous run of black pixels with weights assigned, say L1 and also count the number of runs, say N1. We now find the run of each row as,

Steps::

for (j = 1 ; j < = rows ; j++)
run_length=0;
run=0;
for (k = 1; k<=columns;k++)
if the (j,k) th pixel is black
run_length = run_length + extent of the black pixel run ;
if a white pixel is found
run = run + 1;
endif
endif
end for
if run ≠ 0
run_histogram = run_histogram [j] / run ;
endif.
else
run_histogram = 0 ;
endelse.
end for.

M2 = *maximum (run_histogram [ ]).*

4.2.3 *Matra of the text line*

We also refer to the value of 'R1' which was earlier defined. This we do as the overall matra of a line should have an influence over the word matra zone value.

4.2.4 *Final determination of matra*

Now the matra zone is treated as a function, f of the above three variables .

That is the matra zone, M = f (M1, M2, R1). We take f to be an average function of the variables.

Therefore, M = *(floor)* (M1 + M2 + R1) / 3.

The significance of determination of the matra zone lies in the fact that in Bangla words the connection between the consecutive characters lies very near to the matra zone. Next we proceed for determining the zone which is a band location in the word where the actual joining between characters take place.

4.3 *Determination of the cutting zone band*

This is the most important part of our segmentation approach. It actually tries to locate in a word the band where we can segment the words. To be very precise, in order to segment characters we search for the co-ordinate points where we can cut the characters. Now we consider all the discrete points in the word diagram as our universal space and thereafter we try to shrink that set of points to a set of points which are our candidate segmentation points. We consider the horizontal direction in the word as our x - direction (positive - left to right) and vertical direction as y - direction (positive - top to bottom). The cutting zone band gives a lower and an upper bound on the y- co-ordinates of the candidate segmentation points. The matra value, M gives the upper bound.

In order to find the lower bound we first delete whatever part of the word is present above the matra zone. Now we perform edge detection on the part of the word below the matra zone. Here we classify the points on the edges as points on the upper part of the contour and lower part of the contour. Now we try to find structures like ⌒ in the upper zone of the word . Obviously those points forming the said contour will be edge points on the upper part of the edge contour near and below the matra zone. To find the said contour we, as earlier has been said, delete whatever is present above the matra zone. Now we scan the image from top to bottom, left to right. As we scan in the said direction, we will be coming vertically down on to the upper contour edge points, we record the pixel which we have hit first in the said direction of scanning and remove all other pixels in the same column not having continuity with it and below the recorded edge point. After this scanning is over we do a contour following on the recorded edge pixels and thus form an almost continuous envelope describing the lower envelope of the cutting zone of the word. Then we go along the said contour formed and find out the lower tips of the said structures, the tips representing the local minima in the contour following and keep a record of them. After having found out those tips we make a weighted sum of the recorded tips. Thus we get an average value, L of the tips of the said structure. This average value finally gives us the lower bound of the cutting zone.

We also determine an estimate of the thickness, 'T' of the word.

So finally our lower bound of cutting zone, 'B' = 'L' + 'T'.

Thus we have got a zone which is our possible zone of character segmentation.

Steps : :

1. *Remove the word zone above matra i.e. turn the black pixels above the matra to white.*
2. *Do edge detection on the remaining part.*

```
for ( j = M ; j < row ; j ++)
    for ( k = 1 ; k < column ; k ++)
        if the ( j,k) th pixel is black
            check the pixels in its 8 neighbourhood ;
            if there are no black pixels
                ( j,k ) th pixel is an edge point
            endif
        endif
    endfor
endfor.
```

3. *From the edge detected image successive edge points in a column are calculated and the weighted average of such differences give an estimate of the thickness,* T.
4. *Start scanning the edge detected image from top to bottom, left to right.*

```
for ( k = 1 ; k < column ; k ++)
    for ( j = 1 ; j < row ; j++)
        Label the first black pixel hit.
        Check for any continuity of other pixels with the labelled pixel in a
vertical direction i.e. along j,
        If there are black  pixels give them the same label .
        Remove all other pixels beneath the last labelled pixel.
    endfor
endfor .
```

5. *follow the contour of the labelled pixels obtained from the previous step from left to right of the present image. Mark the pixels which are located below the nearmost pixels on left and right . This pixels gives the lowermost tips of the said structure.*
   Add the y - co-ordinates of such pixels and count the number of such pixels . Let the sum be 'Y' and the count be 'n'.
   So, L = *(floor) Y / n where* L *is the estimate of the lower boundary of the cutting zone band.*
6. *So,* B = L + T, *where* B *is the final estimate of the lower boundary of the cutting zone.*

Now after having found out the y co-ordinate bounds of the cutting zone we proceed for the possible x co-ordinate values or bounds where the actual joining of the characters has taken place. It suffices to say that if we can locate such x values we can uniquely determine the segmentation points or zones.

5. CHARACTER SEGMENTATION

5.1 *Segmentation by vertical histogram and distance based weights*

5.1.1 *Formation of vertical histogram* , (H) *and weighted vertical histogram* , (Mo)

We first delete whatever portion is there over the matra zone. Then we sum all black pixels in a column across the rows. The portion above matra is deleted as it was found that the vertical histogram thus formed was rich in information compared to the histogram formed with the portion above matra.

We also take the weighted vertical histogram of the displayed image. The weight of a black pixel is defined to be the distance of the black pixel from the matra zone, M.

Next we find the local minima in 'H'. An average value of the said minima is taken. Let that value be A.

Now the index of all the values in H less than A are taken to be the possible points of segmentation i.e. index is the set of all values such that H[index] < A. Let this set of values of index be S. Now we will try to prune the set 'S' so that we can arrive at the optimum set consisting of only the required segmentation points.

Consider weighted vertical histogram, Mo of the image for the above mentioned pruning.

Let d be the width of the cutting zone band i.e. d = B - M. Therefore, the weighted vertical histogram of the cutting zone band will be C = (d *(d - 1) / 2).

The concentration of black pixels in the region between characters (which is basically the zone of segmentation) should be minimum. Now there will be almost consecutive values in S, say ranging from 'x 1' to 'x n' such that the range, r (= 'x n' - 'x 1') is the cutting zone.

Now we define x-mid = ('x 1' + 'x n') / 2. The value, m = Mo [x-mid] is taken.
If m > C, we reject x-mid as a segmentation point. We calculate then the 'first point weights' at
'x-mid'. The 'first point weight' at any point is defined as the distance a black pixel from M. Clearly, had we segmented at x-mid there would be over segmentation. This 'first point weight' concept is used specifically to do away with the over segmentation of many Bengali characters supposed to be written with matra but actually not written with matra e.g. 'Aa', 'Aaa', 'Kha', 'Gha' 'Ta', 'Tha', 'Da', 'Na','Pha', 'Bha', 'Ma', 'dantya Sa'.
'Jya', 'La'. Also some extended parts of characters like 'short U', 'long U', 'ae', 'oi', 'da','murdhya na' may give rise to over segmentation.

Here a definition of over and under segmentation will be in place:

If a legitimate character is further segmented into smaller parts we call it over segmentation. In our case the distance based weight will help to overcome it. We will also deal with it by a post segmentation merging.

If a segmented character can be further segmented into legitimate characters, we call the initial segmentation to be under segmentation. We will circumvent it by a post segmentation scheme.

Steps : :

1. *for (j = 1 ; j < = rows ; j++)*
for (k = 1; k<=columns;k++)
if the (j,k) th pixel is black
H [k] =H [k]+ 1
end if
end for
end for.

2. *for (j = M ; j < = rows ; j++)*
for (k = 1; k<=columns;k++)
if the (j,k) th pixel is black
Mo[k] =Mo [k]+ j
end if
end for
end for.

3. *Let* U = *set of all points along x-axis* = $\{ x_1, x_2, \ldots\ldots, x_{column} \}$

4. *Find out all minima,* x_j *in* H .
Form S, *such that*
S = $\{ x_j \mid x_j$ *'s are minimas in* H *and* $x_j \in$ U $\}$
Let cardinality of S = n

5. *Average minima* , A = $(\Sigma\, x_j) / n$, *where* $x_j \in$ S.

6. *Form* X *such that* X = $\{ x_j \mid x_j \leq$ A *and* $x_j \in$ A$\}$.

7. *It is obvious that in* X *we will find consecutive* x_j *'s indicating zones between characters where we can have segmentation points or zones. Select such consecutive x's from* H .*Let the values be* x_1 *to* x_n. *Let* r = *range of the x's values i.e.* $r = x_n - x_1$.

8. *x-mid* = $(x_n + x_1) / 2$.

9. d = B – M .

10. C = $(d * (d - 1) / 2)$.

11. m =Mo [x-mid].

12. *If* m > C, we *reject* x-mid as a segmentation point.

13. *If* m ≤ C, *check the 'first point weight' at* x-mid, where *first point weight is a measure of the distance the first black pixel in a column is from* M. *If 'first point weight' at* x-mid >> B, *discard* x-mid *as a segmentation point, else accept it.*

14. *We thus form again a set of points of* x-mid *to be the probable cutting points. Let this set be* Y.

15. *From* Y *we take consecutive values,* y_j *and* y_{j+1}, *in between these two values a probable character lies. We determine the lowermost black pixel in the range of* y_j *and* y_{j+1}. *All such lowermost points in the ranges are calculated. Let the sum of all such values be* N *and the number of the ranges be* w.

E = *estimate of the end line = (floor)* N / w.

16. *Now we proceed for the final segmentation of characters. The final heuristic used for segmentation is that in a zone between* y_j *and* y_{j+1} *the lowermost black pixel should be near the end line ,* E . *If not, reject* y_{j+1} *and pick up* y_{j+2} *as the next segmentation point and proceed . This was the case where over segmentation was circumvented.*

17. *To tackle under segmentation , the space between the characters are calculated and if there is a space from the zone of* B *to* E , *we go for a further cut.*

18. *Now comes the segmentation part of the modified characters , known as allographs which might be present under the end line ,* E. *They can be characters like ' long and short U', ' Ri kar' etc. For these we check for the concentration of black pixels in the region of our segmentation and below* E .*If that is considerable we segment out whatever is present below* E *in the current zone of segmentation.*

19. *We had started at the outset by deleting the contents of the image above matra , M . Now those parts are to be taken into account. The lowermost pixels of each structure above matra,* M *are found out . Let the location of the said pixel is (j,k) . It is checked in which zone of* Y , *to be precise we find* y_j *and* y_{j+1} *such that* $y_j \leq j \leq y_{j+1}$, *j lies. Next the extents of the structure are found in horizontal directions and let the structure be bounded within x-left and x-right .Therefore, the component above the matra ,* M *bounded within x-left and x-right is joined with the structure between* y_j *and* y_{j+1}.

5.1.2 *Limitations*

The performance of the above algorithm is good on hand-written texts with proper spacing in between characters but its performance is not satisfactory on written texts where spacing is not adequate. It also fails for touching characters. The problem of a character getting into the convex hull of the next character also can't be taken into account. The success and failures of the proposed method is discussed in section 6 titled *Results and discussion.*

5.2 *Recursive contour following and bounded box method*

As already mentioned, the characters of words in Bangla are connected mostly in the upper part of the word zone. It is very rare that one comes across connection between characters in the lower part of the word. So the connections lie mostly in between the zone bounded up by M and below by B, where M and B have already been defined to be the matra estimate and the lower bound, respectively. So the connection free zone will lie below B and within the boundary zone of the word. In this method of segmentation we exploit the above properties inherent in a word structure.

5.2.1 *Contour following and bounded box formation*

We detect the edges of the word. The rest processing is done on the edge detected image . This is done to save time and space.

We scan the image from left to right, bottom to the zone bounded up by B. We record the first black pixel hit. Starting with this pixel a recursive contour following is done in the zone bounded above by B and below by the word boundary. From this contour following, a bounded box is developed for the contour followed. This bounded box is open ended at the top, bounded below by the lowermost pixel in the region which encompasses the contour followed, bounded left and right by the leftmost and rightmost black pixel in the contour followed region. This box is defined as the bounding box of the character.

Recursive contour following is basically tracing the contours of the edge detected image within a certain range in x and y directions. From recursive following, we determine the boundary of the bounded box which is bounded up by B.

5.2.2 *Merging of bounded boxes*

If the bounded box is contained in x direction or y direction in another bounding box, we merge the smaller bounded box into the larger bounded box.

If the lower bound of any box is much above E, as has already been defined, we merge it with the next bounding box to its immediate right. Pixels in separate bounded boxes are given separate labels.

Thus after finishing the findings of extent of the bounded boxes, extend the bounded boxes upwards from the zone of B to M.

The difference of this method with the earlier method lies in the fact that here vertical histograms and distance as a weight concept information is not at all used for segmentation .

Steps : :

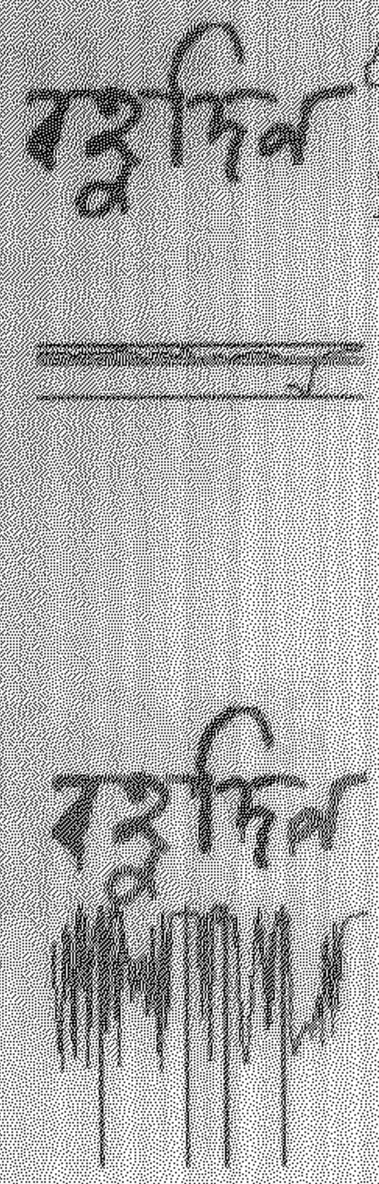
Steps 1 *to* 15 *are the same as the earlier method.*

16. *for (j =1; j≤column; j++)*
for(k = row ; k ≤ B *; k++)*
if (j,k) th pixel is black
recursive_follow (j , k , up_bound , low_bound , label);
if the lowest boundary of the bounded box almost touches E
if the lowest boundary of the previous bounded box is away from E *and near to* B
merge the previous bounded box to the current one;
endif
endif
else
if the lowest boundary of the previous bounded box is away from E *and near to* B
merge the previous bounded to the current one;
endif
endelse
if the current bounded box is bounded by the previous bounded box
merge the current bounded box to the previous one which engulfs the current one ;
endif
extend the bounded box upwards to M *with the current label. This extension takes care of the zone of the character in the cutting zone between* B *and* M *;*
check for structures above M *;*
if the start point (x,y) of any such structure is between the rightmost extent of the previous bounded box and the rightmost extent of the current bounded box
*recursive_follow (x , y, upper part of the word,*M,*label);*
extend the current bounded box from M *to the extent obtained from the just concluded recursive contour following with the same label ;*
endif
store the boundaries of the current bounded box (which may or mayn't be modified);
label ++;
endif
endfor
endfor

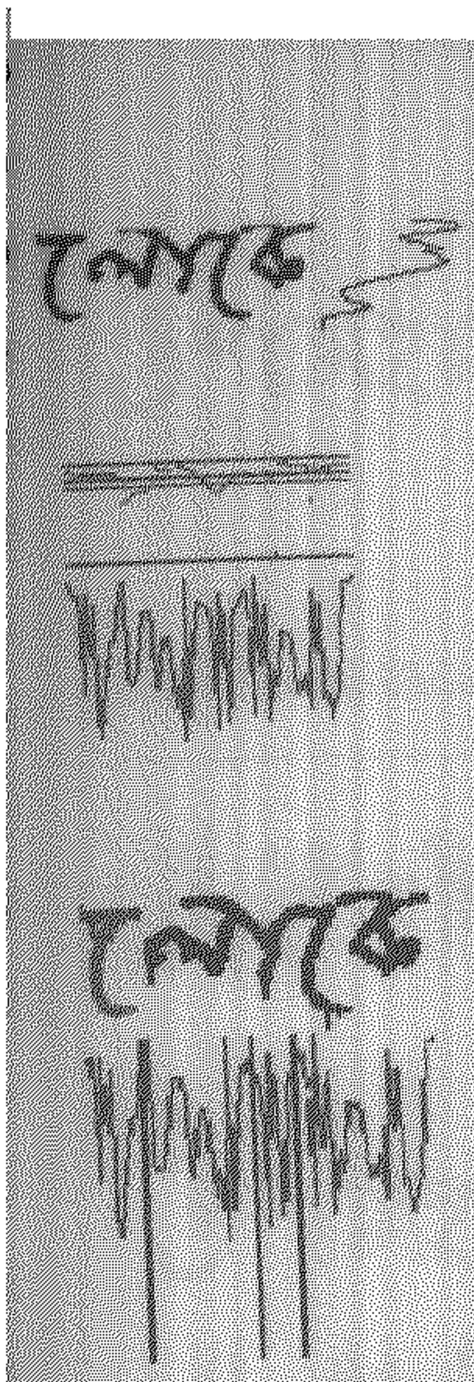
5.2.3 *Limitations*

This method works fine for the words not having any connection between its characters in the lower part of the word zone. If there are touching characters in the lower part of the zone this method will fail. But there are only a few cases where the connection between characters comes in the lower part of the word zone. A glaring example which comes to mind immediately is the signature of Rabindranath Tagore. However, Tagore's normal writing does not contain such connections. The success and failures of the proposed method is discussed in section 6 titled *Results and discussion*.

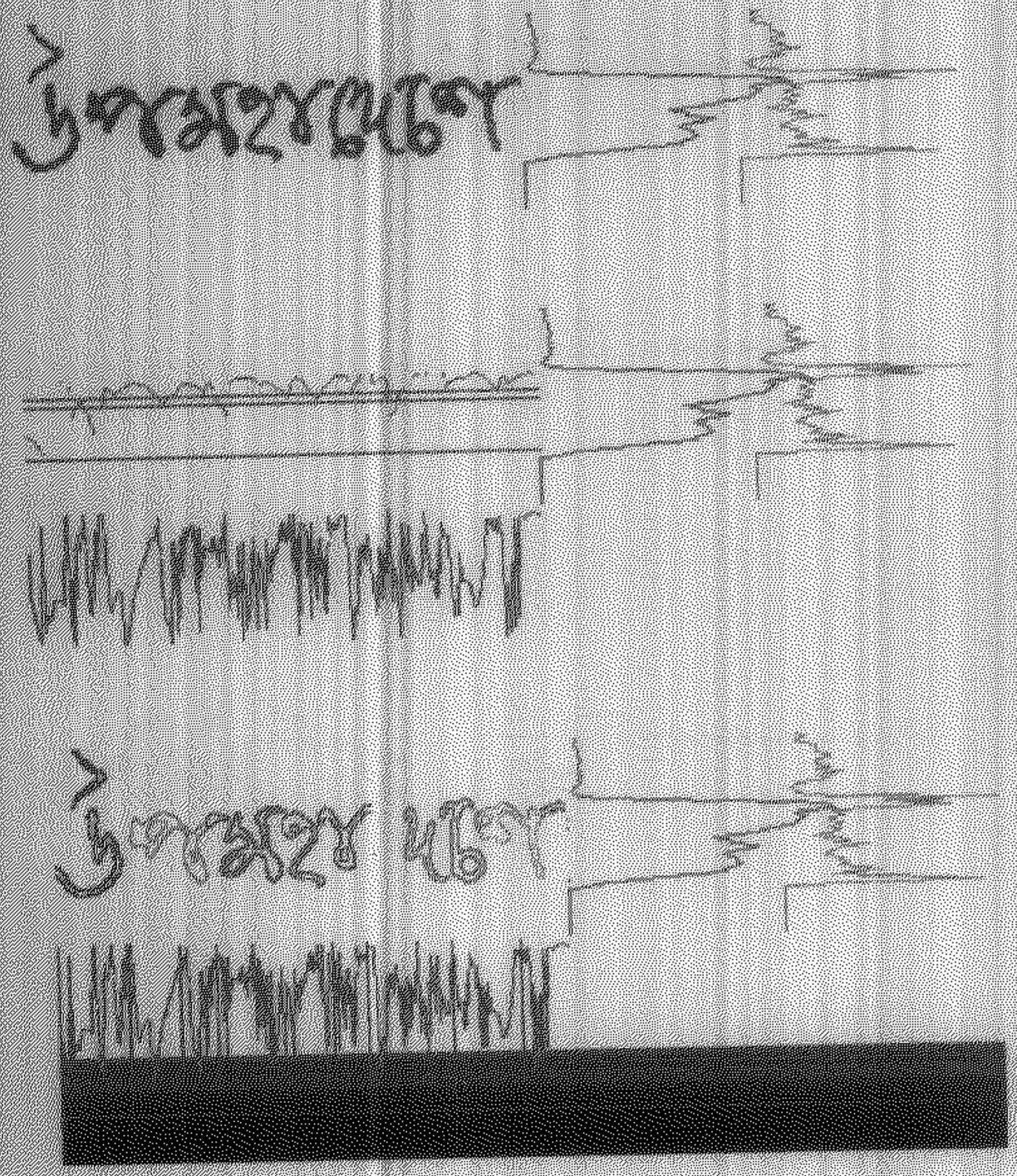
6. RESULTS AND DISCUSSIONS



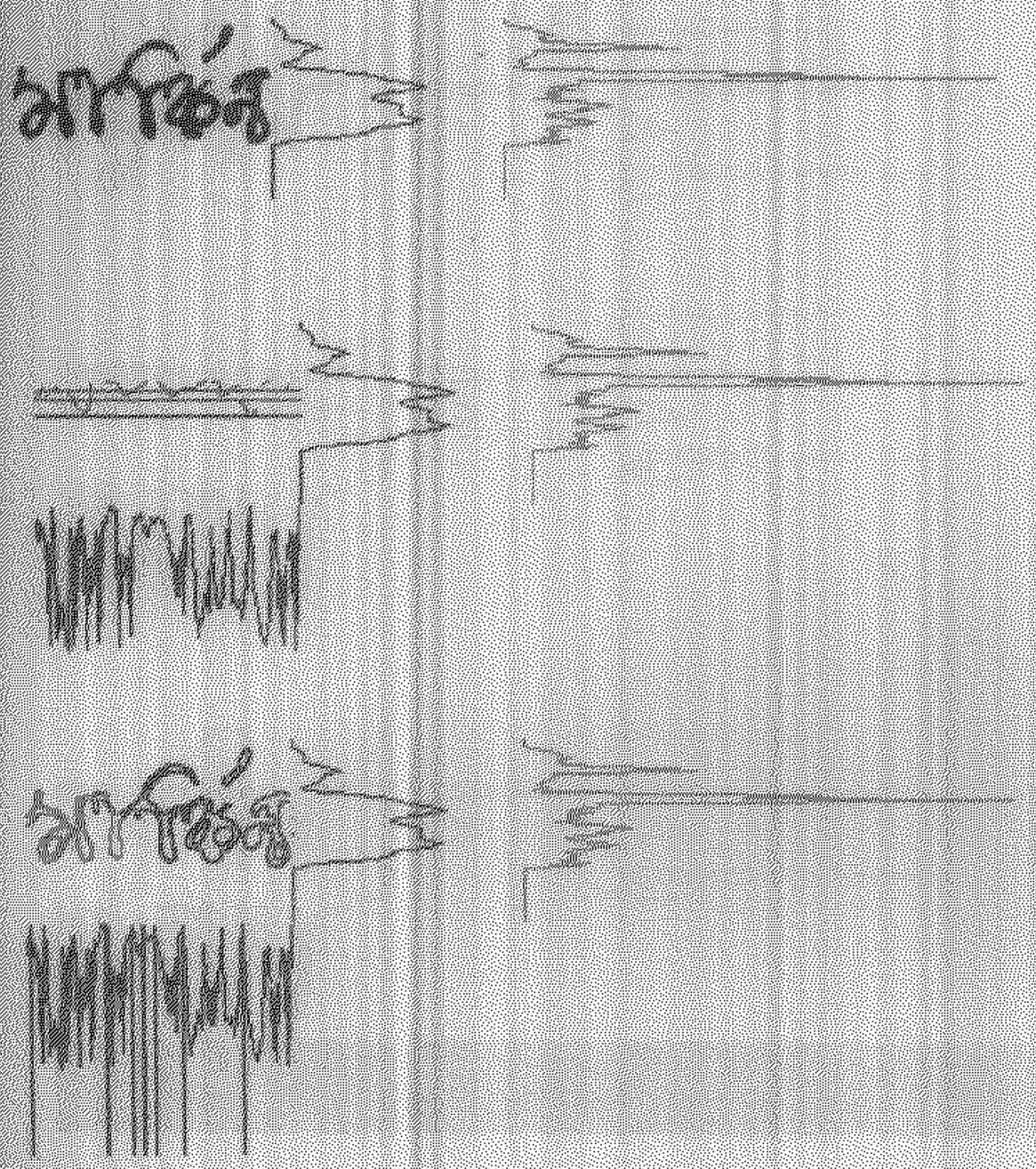
Successful in segmenting by the method of 5·1



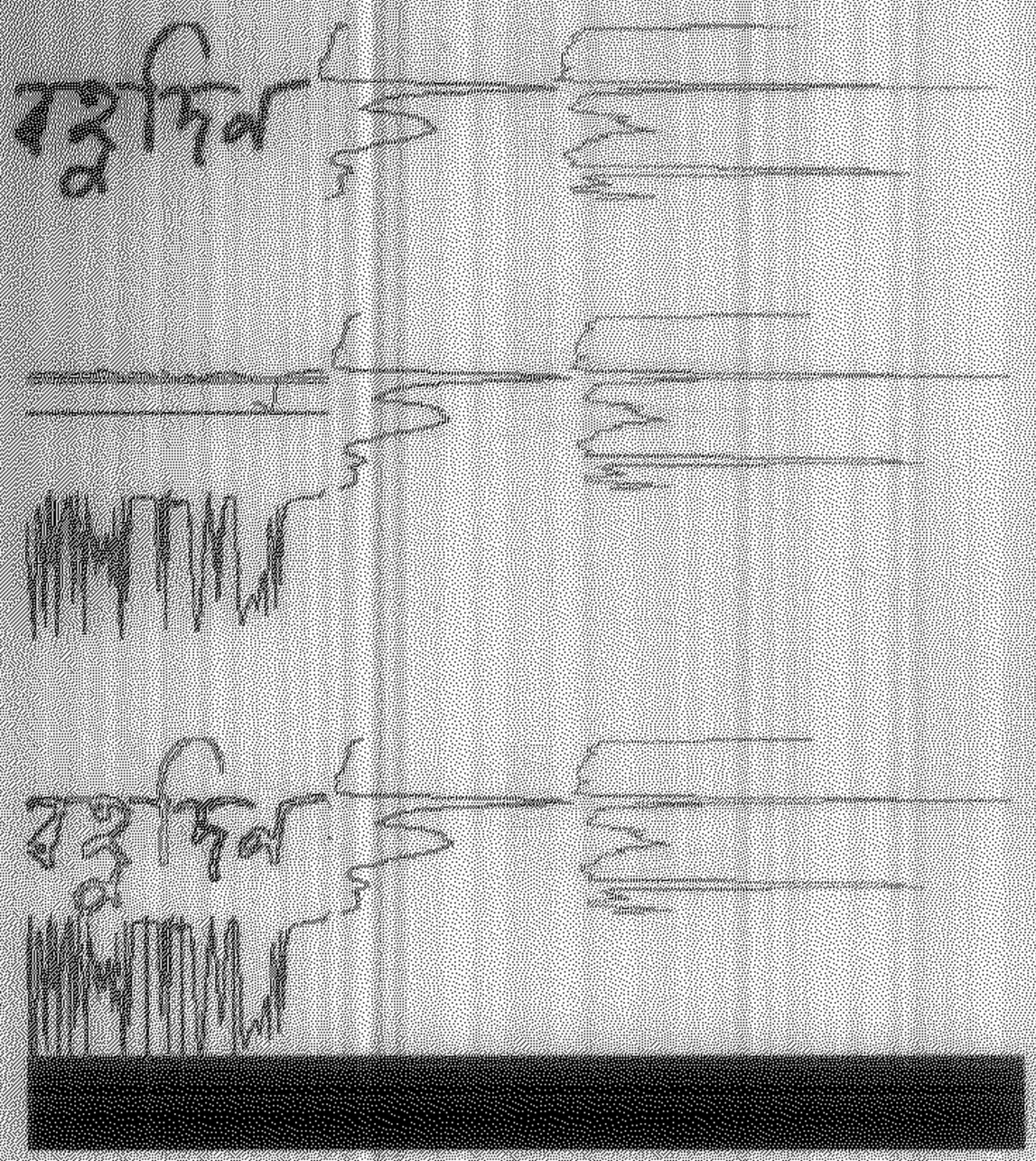
Failed in segmenting by the method of 5.1
Owing to the character 'ಪ' getting into the
convex hull of the character 'ಲ'.



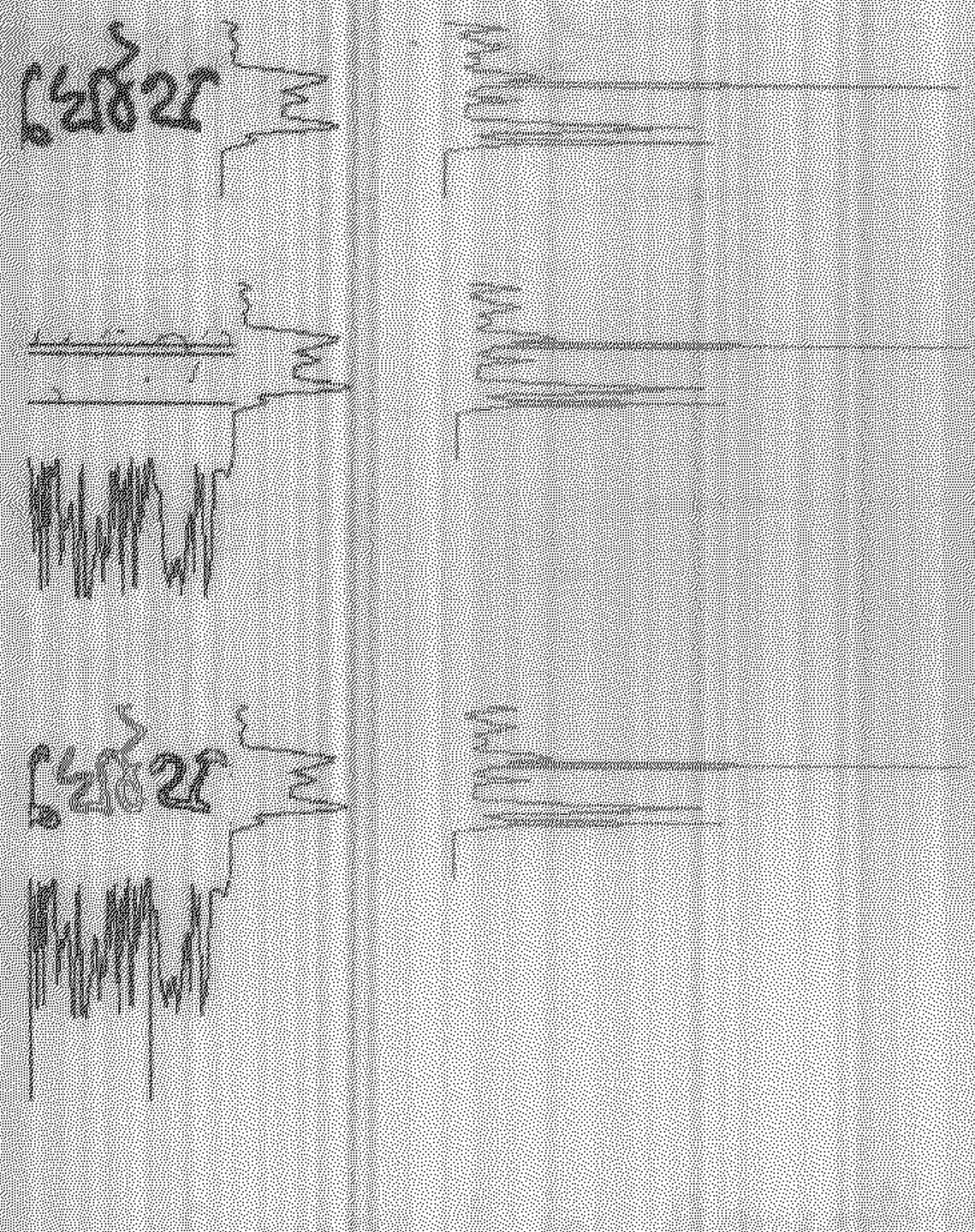
Successful in segmenting by the method of 5.2.



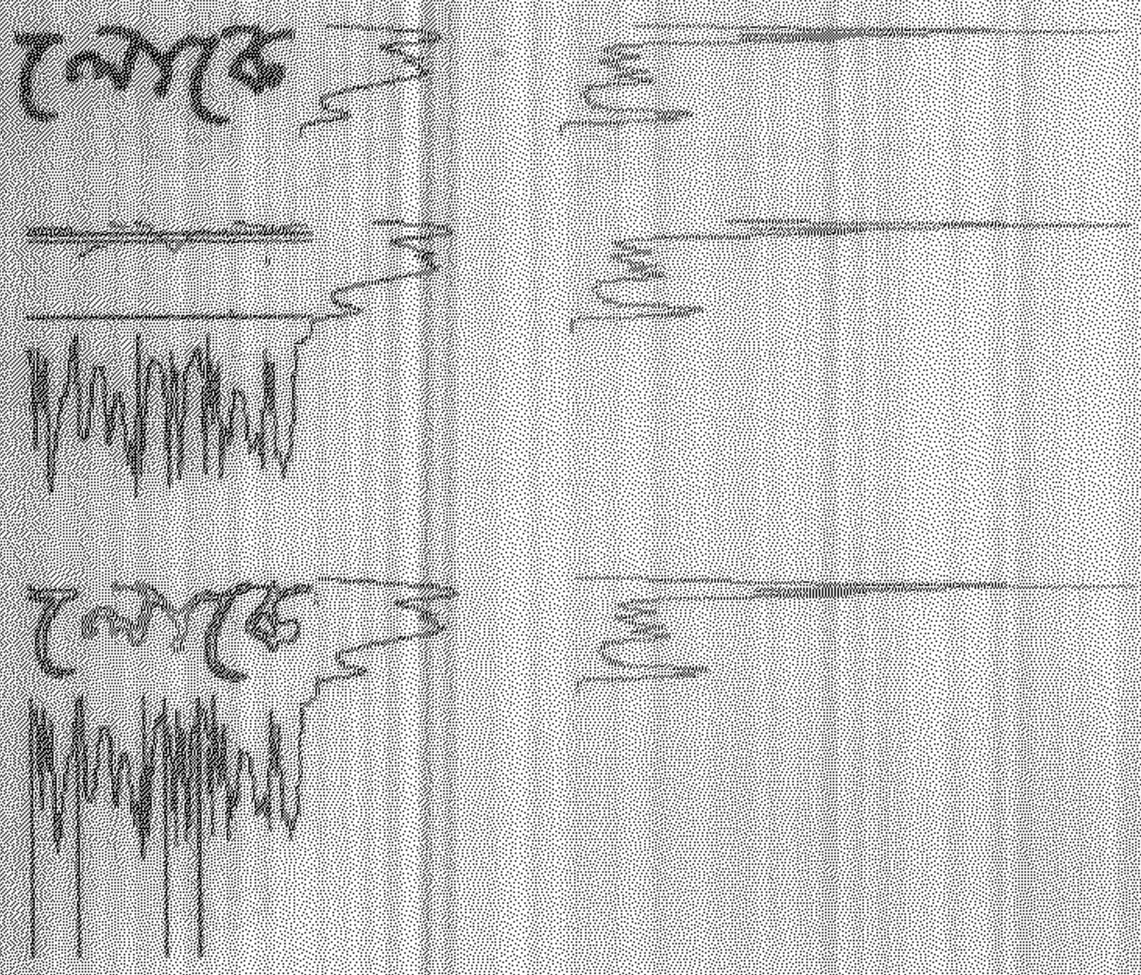
Successful in segmenting by the method of 5·2.



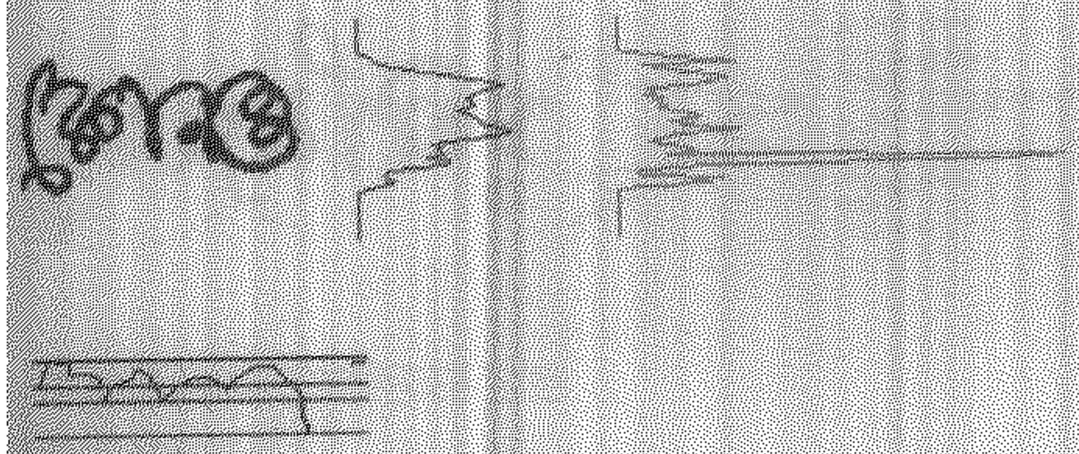
Partial success in segmenting by the method of 5.2 owing to the character '4' touching '2' in the lower zone.

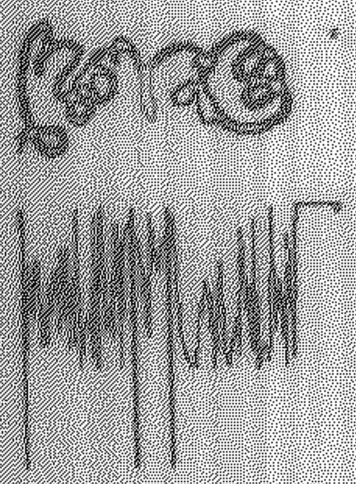


Succesful in segmenting by the method of 5·2

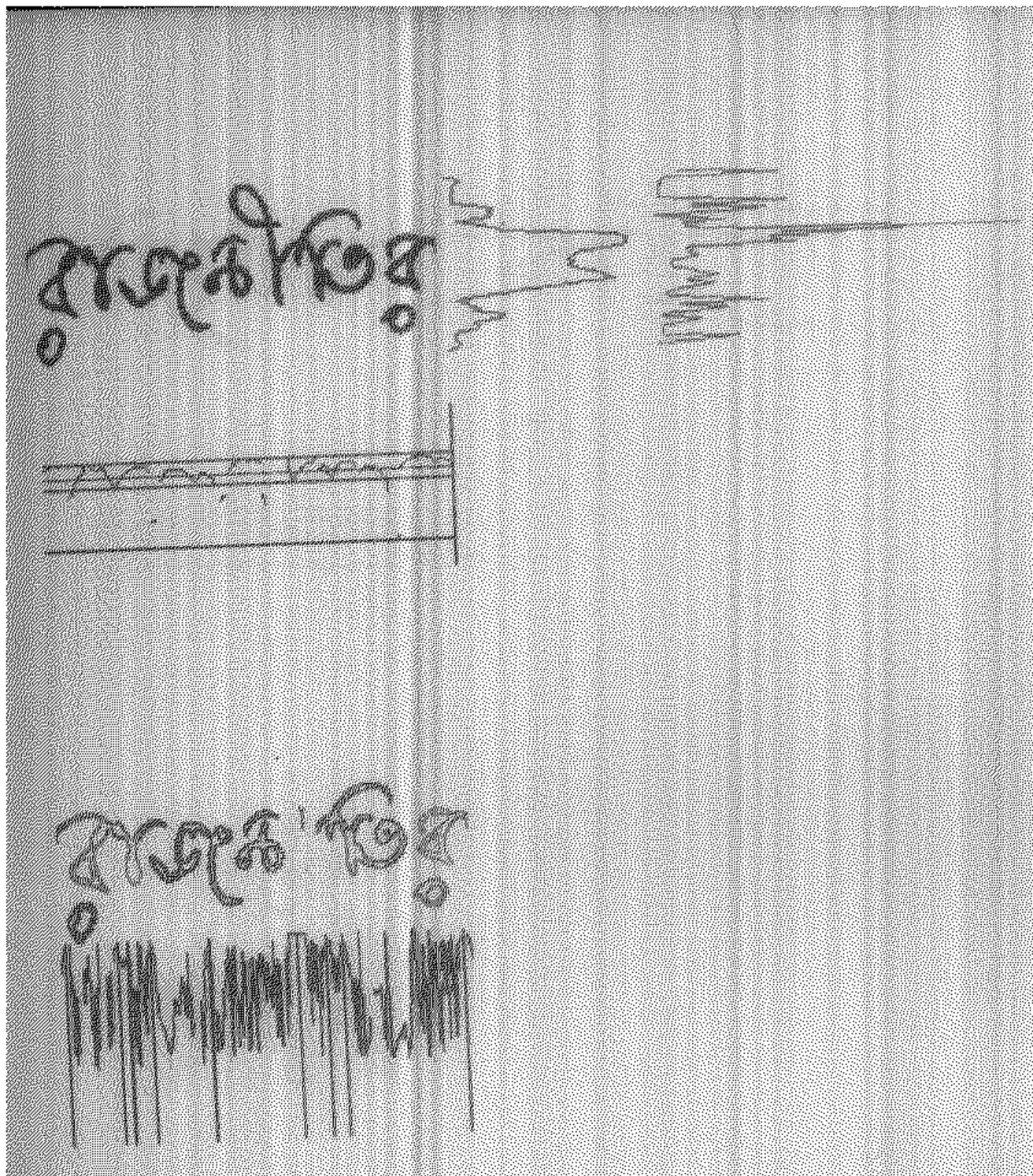


Successful in segmenting by the method of 5.2.

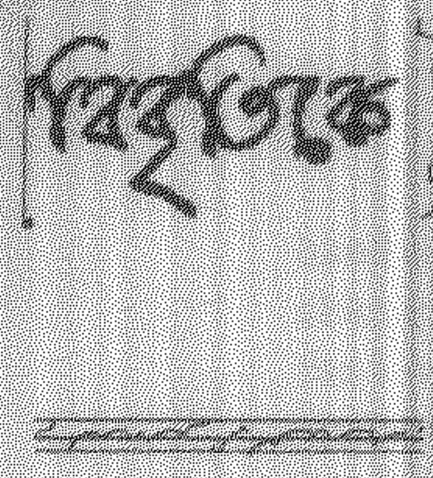


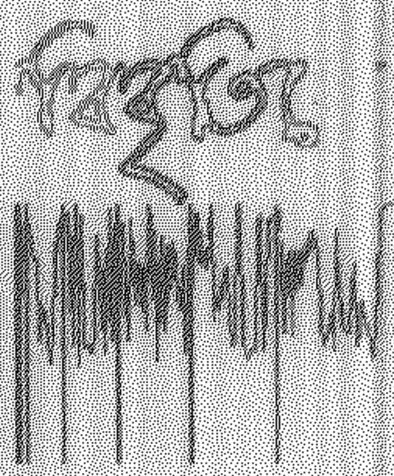


Failed in segmenting by the method of 5·2 owing to character '3' touching 'd' at the lower zone.



Failed in Segmenting by the method of 5.2 owing to character 'ರ' touching 'ಗ' at the lower zone; and as the character 'ಗ' doesn't have connection with its own part at the lower Zone.





Failed in segmenting by the method of 5·2
owing to 'ഗ' touching 'ി' at the lower zone.

CONCLUSIONS

We have approached character segmentation of Bangla handwritten words with two methods . The first method works fine with well spaced characters in a word but is not having satisfactory results with touching characters at the lower half of a word zone . Whereas, the second method works well with well spaced words it also takes care of one character protruding in the convex hull of another character , but the problem of touching character remains . The scope of further work in this field should encompass the problem of recognition and segmentation of touching characters .

REFERENCES

[1] Umapada Pal , 'On the Development of an Optical Character Recognition (OCR) System for Printed Bangla Script', Ph.D. thesis , Indian Statistical Institute , 1997.

[2] Hirobumi Yamada and Yasuaki Nakano , 'Cursive Handwritten Word Recognition Using Multiple Segmentation Determined by Contour Analysis', IEICE Transactions on Information and Systems , Vol. E79-D , No. 5 , May , 1996.

[3] B.B. Chaudhuri and U. Pal , 'Skew Angle Detection of Digitized Indian Script Documents', IEEE Transactions on Pattern Analysis and Machine Intelligence , Vol. 19 No. 2 , Feb. , 1997.

[4] A.F.R. Rahman , M. Kaykobad , M.A. Sattar , 'A Novel Hybrid Approach to Handwritten Bengali Character Recognition', Proceedings of the ICCLSDP, 1998 held at Indian Statistical Institute, Calcutta, India.

[5] R. Plamondon , C.G. Leedham edited 'Computer Processing of Handwriting', Singapore,WorldScientific,1990.