



# **Elimination of splitting errors in printed Bangla scripts**

**Md. Abul Hasnat and Mumit Khan**

**Md. Abul Hasnat**

**Center for Research on Bangla Language Processing**

**BRAC University**



# Outline

- Overview of Bangla OCR and Segmentation
  - Zone based approach.
  - Segmentation errors.
- Analysis on splitting error
- Methodology of the proposed approach
- Result
- Conclusion



# Overview

## Bangla OCR and Segmentation

- Research started since 1994.
  - B. B. Chaudhuri, U. Pal and U. Garain
  - Indian Statistical Institute, Kolkata.
- Usage of Headline/Matraa/Sirorekha for character segmentation.
- Zone based approach for character segmentation.
- Similar technique also used for Devanagari script.
- Segmentation errors (over and under segmentation).
  - Problem addressed in few literature.
  - Variety of different solution.



# Zone based approach

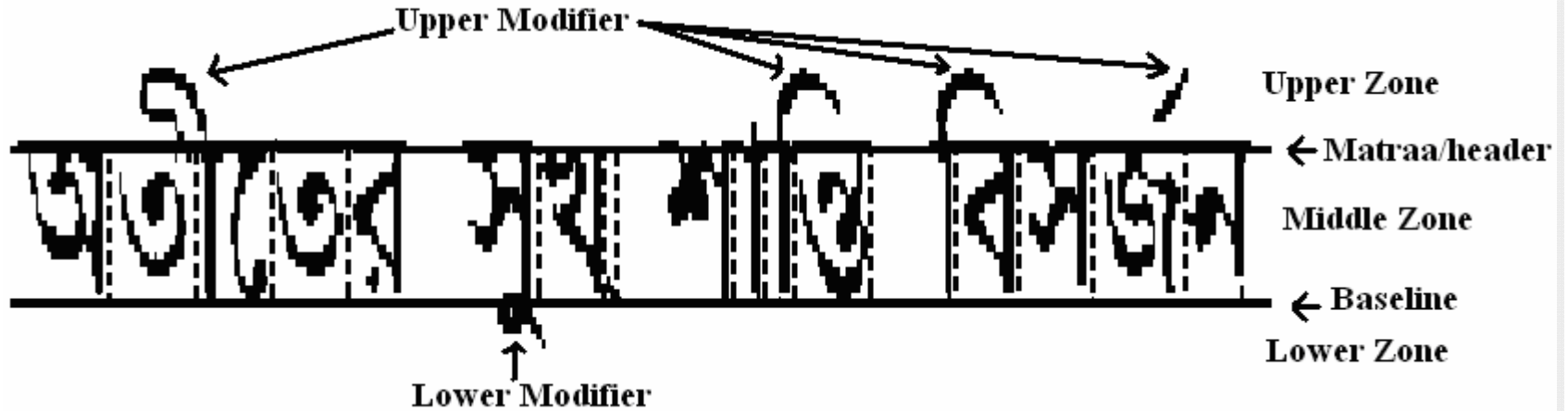
## Idea

- Divide the word into three zones – upper, middle and lower zones.
- Then using the vertical projection profile of the middle zone.
- The gaps between the characters are identified and used to segment the characters.



# Zone based approach

## Example



Zone based approach of Bangla character segmentation.  
Dotted lines show the character boundary



# Zone based approach

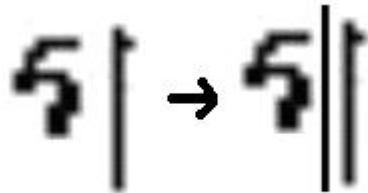
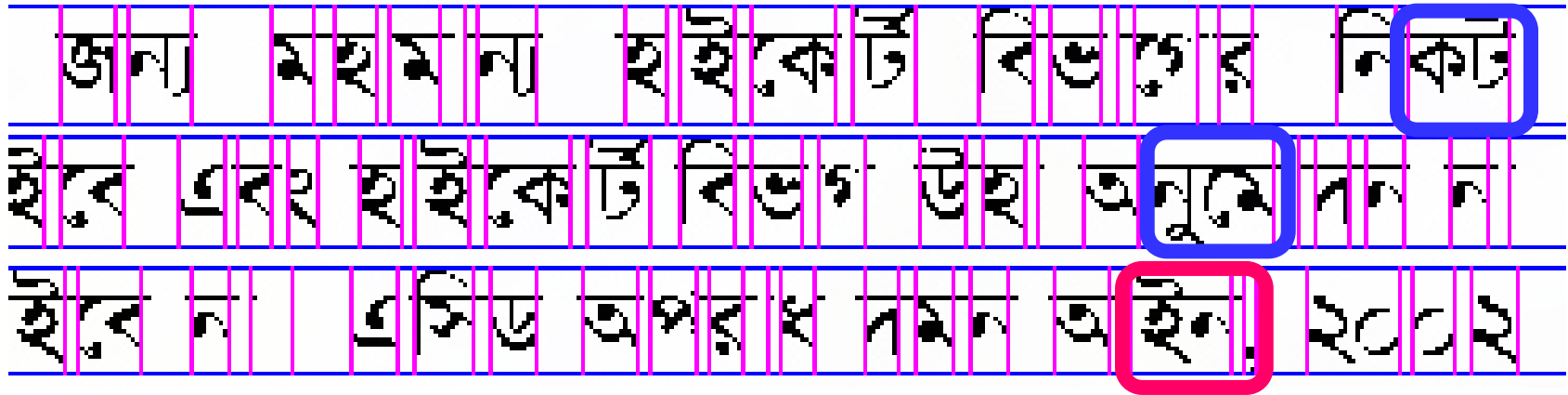
## Steps

- Find out the matraa (headline) of the line/word.
- Separate the upper zone from the middle and lower zone using the matraa.
- Find out the baseline (separator line which divides the middle and lower zone).
- Separate the middle and lower zone using the baseline.
- Take vertical histogram in the middle zone for segmentation of characters.
- Separate characters using the gaps identified in the histogram information.



# Zone based approach

## Output & Observation



Splitting error

— Joining error

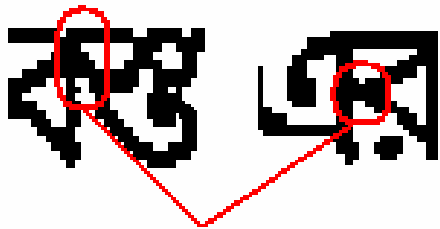
— Joining & splitting error

# Segmentation Errors

- Divided into two types
  - Joining error (Under Segmentation)
    - 77.634% of the total error (from my experiment)
  - Splitting error (Over Segmentation)
    - 22.365% of the total error (from my experiment)
- Joining error is further divided into two types
  - Shadow/overlapping error
  - Touching error

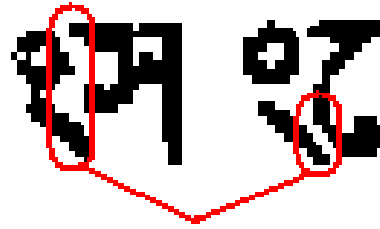


# Segmentation Errors



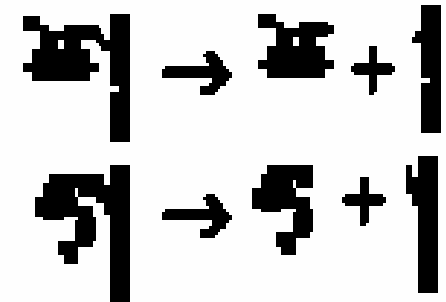
Two characters touching each other

(a)



Shadow of the second character cause segmentation fault

(b)



One character split into two parts

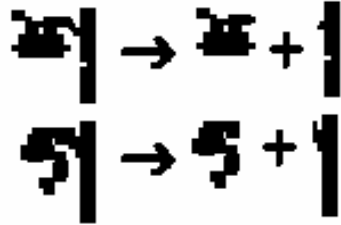
(c)

Example of segmentation errors after applying zone based approach of segmentation.

- (a) shows touching errors
- (b) shows shadow character problem and
- (c) shows splitting errors



# Splitting Error



One character split  
into two parts

Example of of splitting errors

- Characters those suffers from splitting error are:
  - গ গ প শ
- Reasons for splitting:
  - Over threshold.
  - Removal of noise.
  - Matraa/headline deletion.
- Break down into two parts.
  - Breaking occurs at weak joining portion - just before the aa-kar (া) like portion.

# Splitting Error

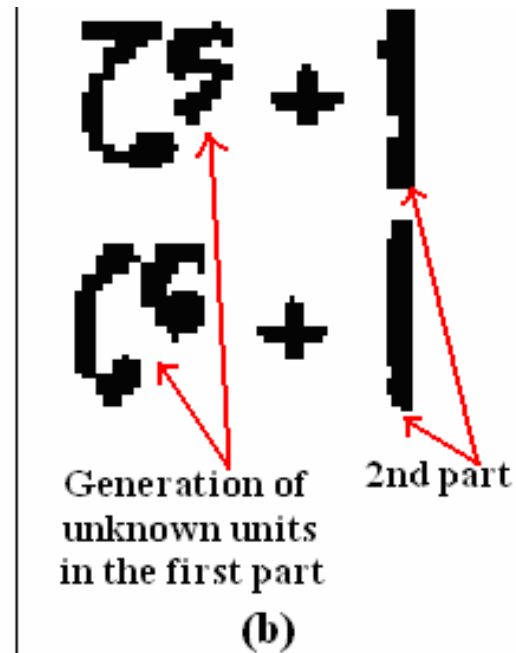
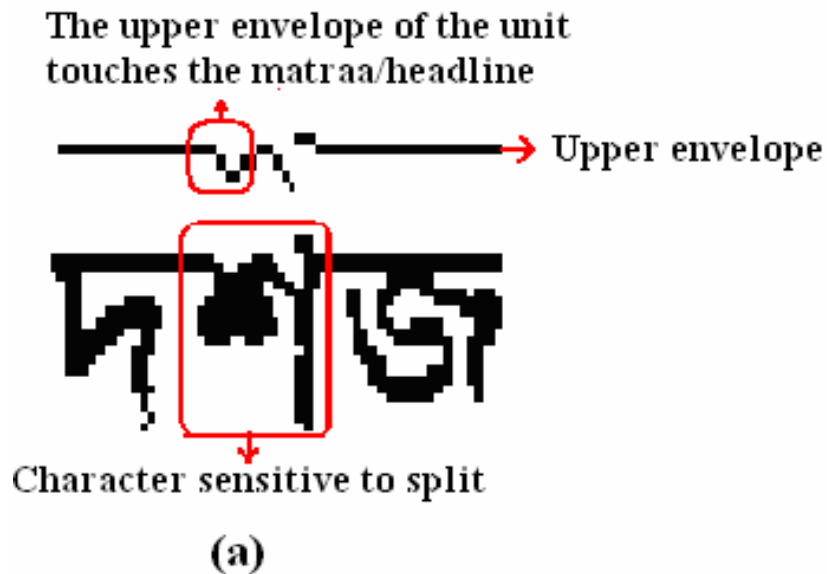
## Existing solutions

- Careful matraa deletion approach using the connected component analysis : *Matraa/headline deletion*.
- Training based techniques.
- Usage of the information: “Bangla characters do extend up to the lower zone”.



# Splitting Error

## Failure of proposed solutions

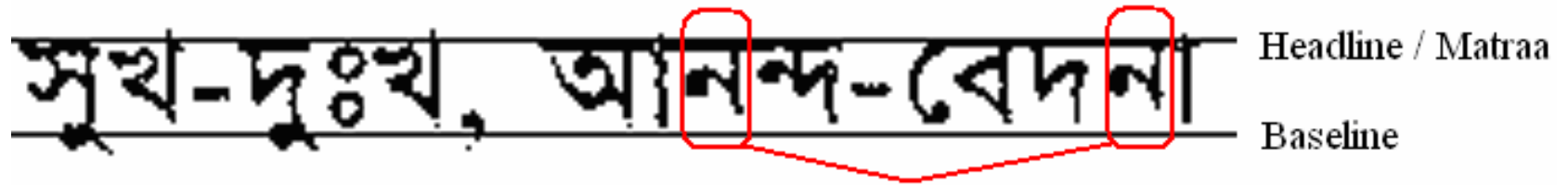


- (a) Unit where matraa clipping approach fails
- (b) Units where training based approach fail



# Splitting Error

## Failure of proposed solutions



Bottom of the units does  
not extend up to baseline

Units which fails to reach the baseline.

Should we combine these units with the next units?



# Proposed Approach

- We solved the problem of splitting error in two steps.
- **First step:** Resolved the problem of matraa clipping using connected component analysis on the envelope of the matraa.
  - Applied at the beginning of the segmentation process
- **Second step:** Performed rule based feature analysis on the extracted characters/units of each word.
  - Applied in between the shadow character elimination and touching character segmentation process.



# Methodology

**Clip the matraa/headline using histogram and connected component analysis.**

- Apply this task at the beginning of the character segmentation task. Steps are:
  - Locate and separate the matraa region.
  - Apply careful matraa deletion.



## Locate and separate the matraa region

- Take horizontal run length histogram (RLH) of the word image.
- Identify the location of the matraa - maximum valued location (maxValLoc) from RLH.
- Identify the probable candidate rows as a part of matraa. Search upward and downward location of the maxValLoc based on unit height.
- Store their run length value in a 1D matrix (MArr).
- Take the standard deviation (stdVal) of those run length values.
- Any row (i) in the matrix (MArr) will be considered as a part of matraa if it satisfies the following equation:
  - $\text{MArr}(\text{maxValLoc}) - \text{MArr}(i) \leq \text{stdVal} * 2$
- Locate the start and end position of the matraa.
- Separate the region of the matraa from the word image.

# Methodology

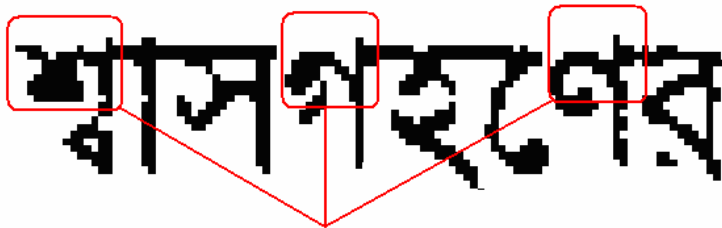
## Apply careful matraa deletion

- Take the upper envelope of the identified matraa region.
- Take the connected components from the upper envelope.
  - Store the ratio of the component width vs. word height after the matraa location ( $\text{ratCcwdWht}$ ) which is actually the aspect ratio of the units.
  - Check this ratio against a threshold value ( $\text{thVal} = 0.5$ ).
  - Take the bottom location ( $\text{bottomLoc}$ ) of the connected component and check it against the height of matraa.
- Keep the pixels below those connected components that satisfy the rules given below.
  - $\text{ratCcwdWht} < \text{thVal}$
  - $\text{bottomLoc} > \text{Height of Matraa} / 2$



# Methodology

Clip the matraa/headline using histogram and connected component analysis.



Units which are sensitive to matraa (headline) deletion

(a)



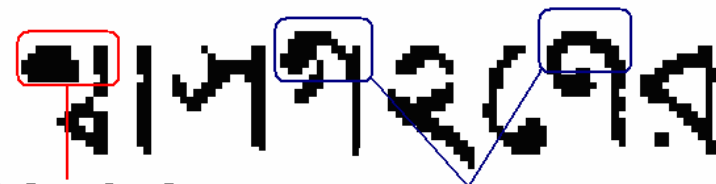
Units break down in more than one part

(b)



Upper Envelope of the matraa region

(c)



Unit breaks down

Units do not break down

(d)

Example of the matraa/headline deletion process

- character ঝ remains broken into two parts as the upper region of this character has the same alignments with the matraa.

# Methodology

## **Rule based feature analysis to identify the splitted units.**

- Necessary to detect and merge the characters that break down into more than one part.
- Applied after performing the shadow error elimination and before touching error elimination.
- Take the necessary information related to the character
  - Height
  - Width
  - Aspect ratio (aspRatio) and
  - Bottom location (bottomLocUnit)



# Methodology

## Rule based feature analysis to identify the splitted units.

Calculate the following feature measurement:

- Average bottom location of the units (avgUnitsBottom)
- Median of the bottom locations of the units (medianUnitsBottom)
- Standard deviation of the bottom location of the units (stdUnitsBottom)
- Maximum bottom location of the units (maxUnitsBottom)
- Z Value of the bottom location of each unit (zValueUnit)
- Ratio of the present character height vs. next character height (ratPhtNht)



# Methodology

## Rule based feature analysis to identify the splitted units.

Using these feature measurements we set four rules based on four threshold values:

- 1. Threshold for aspect ratio of the next unit (thUnitAspectRatio).
  - Value is set to 0.4
  - Ensure that the unit is likely to an aa-kar (া).
- 2. Threshold for ratio of the height of present and next unit (thUnitsHtHtRatio).
  - Value is set to 0.75
  - Ensure the eligibility of merging between two units.



# Methodology

## Rule based feature analysis to identify the splitted units.

Using these feature measurements we set four rules based on four threshold values:

- 3. Threshold for the z value of the unit (thUnitZValues).
  - Value is set as 0.
  - Required to select the preliminary candidate units to be merged with the next.
- 4. Threshold value of bottom of the units (thUnitBtLoc).
  - Value is calculated from the feature measurements of the units.
  - Necessary for the ultimate selection of the units which can be merged with the next.

# Methodology

## Rule based feature analysis to identify the splitted units.

A unit is marked as a valid unit to be merged with the next if it main the **following rules**:

- **Rule 1:  $zValueUnit(i) < thUnitZValues$** 
  - Performs the preliminary selection of the units.
  - Sometime selects non-splitted units.
- **Rule 2:  $bottomLocUnit(i) \leq thUnitBtLoc$** 
  - Applied on the selected units which passed rule 1.
  - Determines whether the candidate unit is eligible to merge with the next.



# Methodology

## Rule based feature analysis to identify the splitted units.

A unit is marked as a valid unit to be merged with the next if it main the following rules:

- **Rule 3:  $\text{aspRatio}(i+1)$  of next unit  $<$   $\text{thUnitAspectRatio}$** 
  - Ensure that the the next unit is quite similar with the broken part (০ত).
- **Rule 4:  $\text{ratPhtNht}(i) <$   $\text{thUnitsHtHtRatio}$** 
  - Ensures the elimination of few units which has height likely to the height of the first part of the broken units.



# Result

<b>Image Name</b>	<b>Total Units</b>	<b>Error rate (Generic Approach)</b>	<b>Error rate (After Matraa deletion)</b>	<b>Error rate (Feature based)</b>	<b>Error rate (Combined)</b>
BB1	2026	1.97	1.09	1.09	0.10
BB3	2095	0.97	0.58	0.10	0.00
BPD1	917	2.4	1.53	0.44	0.22
BT1	357	2.24	2.24	0.00	0.00
Average		1.68	1.04	0.52	0.07

- Reason for the rest of the error is the presence of noise at the bottom of the first part of the splitted units.

# Conclusion

- Technique complete its task in two stages.
  - In the first stage we put our effort in preventing the breaking of the sensitive units.
  - In the second stage we use a selection and merge approach to rebuild the splitted units into one character.
- Approach shows significant improvement compared to the other proposed approaches.
- Methodology outlined here is applicable to Devanagari as well.



**Thank You**



# Z-value

- A z-value (also known as z-score, standard score, or normal score) is a measure of the divergence of an individual experimental result from the most probable result, the mean. Z is expressed in terms of the number of standard deviations from the mean value.

$$z = \frac{X - \mu}{\sigma}$$

- $X = \textit{Experimental Value}$
- $\mu = \textit{Mean}$
- $\sigma = \textit{Standard Deviation}$
- Z-scores allow us to transform any normal distribution into a standard normal distribution. The standardized sets of data can then be compared with one another.

