



JnUJSci

Jagannath University Journal of Science

Volume 09, Number I, June 2022, pp. 27–42

<https://jnu.ac.bd/journal/portal/archives/science.jsp>

ISSN 2224-1698



A Segmentation and Labeling Tool for Constructing a Dataset for Bangla Optical Character Recognition

Research Article

Mahir Mahbub¹, Ahmedul Kabir², Selina Sharmin^{3*}, Sumon Ahmed⁴ and Md. Shariful Islam⁵

¹Department of IoT and Robotics Engineering, Bangabandhu Sheikh Mujibur Rahman Digital University, Gazipur, Bangladesh

^{2,4,5}Institute of Information Technology, University of Dhaka, Dhaka-1000, Bangladesh

³Dept. of Computer Science & Engineering, Jagannath University, Dhaka-1100, Bangladesh

Received: 3 August 2022, Accepted: 28 January 2023

ABSTRACT

The two main parts of Optical Character Recognition (OCR) systems are segmentation and recognition. In an OCR system, the input text is first segmented into distinctive character images before these segments are fed to the recognition system. The efficiency of a recognition system largely depends on the dataset. To build a good dataset, a highly interactive data collection tool proves to be extremely useful. This paper presents a tool that eases the segmentation of Bangla characters as well as the labeling of unlabeled character images. In this tool, a new segmentation strategy for characters and modifiers is introduced. A filtering-based decision-making strategy is also proposed, which gives better flexibility in character segmentation. Finally, the paper proposes an interactive tool using HCI principles that eases the labeling of the segmented data.

Keywords: Segmentation, Preprocessing, Matra/headline, Binarization, Character segmentation, Line segmentation, Word segmentation, OCR, Image processing

I. Introduction

Optical Character Recognition (OCR) is the process of converting text images into computer-editable text format. OCR can be constructed for printed images, typewritten images, hand-written images, signboards, or art-text images. The construction of an OCR system involves image processing, pattern recognition, and artificial intelligence. Due to the usefulness of OCR systems, extensive research on OCR has been conducted in various languages. Proto-Germanic languages, such as English, German, and Proto-

Italic languages, such as Spanish, Italic, and French, which have a common root of the Proto-Indo-European family (Gamkrelidze & Ivanov, 1990), have OCR systems (A. Chaudhuri et al., 2016) that are very well developed. The Persian language, which belongs to the Indo-Iranian languages family whose root also goes to the Proto-Indo-European language family, and Arabic, a Proto-Semitic (Huehnergard, 2017) language, also have their OCR systems (Alginahi, 2013; Khosravi & Kabir, 2009; Märgner & El Abed, 2012; Mehran et al., 2005). OCR systems have been developed for Indian sub-continental

*Corresponding author: Selina Sharmin
Email: selina@cse.jnu.ac.bd

languages. For example, Telegu (Jawahar et al., 2003; Kumar et al., 2011; Mathew et al., 2016; Negi et al., 2001) whose ancestry is traced back to the Dravidian languages' family and Hindi (Bairagi & Dulal, 2018; Bansal & Sinha, 2001; Govindaraju & Setlur, 2009; Mathew et al., 2016), Assamese (Borpuzari & Mahanta, 2021; Ghosh et al., 2012; Srinivas et al., 2008), Gujarati (A. Chaudhuri et al., 2017) whose ancestry is tracked to Indo-Iranian languages family.

Bangla originated from the Indo-European language family. It has lexical and constructional similarities with its sibling languages, such as Hindi, Gujarati, Assamese, etc. Bangla also inherits the structural and linguistic complexities of its ancestor languages. These complex patterns of Bangla languages pose a challenge for the construction of an OCR. The issues with the Bangla script add complexity while constructing the Bangla OCR (Pal & Chaudhuri, 1994). Firstly, Bangla has conjunct consonants where multiple characters combine to make one character. Secondly, there are multiple modifiers (য-ফলা, র-ফলা, ি, ী, রেফ, etc.) which vary in position and shape. Thirdly, there is a linear connector between characters (matra/headline) which makes it difficult to demarcate the start and end of characters. For these reasons, Bangla OCR is still a developing field of research. Traditionally, the Bangla OCR research paradigm can be divided into two parts. Firstly, the researchers prioritized resolving the complexities of the linguistic properties so that machines can process them efficiently. These include processing the structure of the characters and modifiers in an understandable form, simplifying the interaction among the characters and modifiers that form a new complex structure, etc. Secondly, the researchers focused on the recognition of those processable entities which translate human-readable characters to machine-readable codes. With the emergence of new computer vision technologies, the recognition process is far better developed. However, the first step is still a challenge, which can be tackled by a suitable segmentation method.

A. Segmentation

Initially, the focus is on the segmentation process of Bangla OCR. Improvement of the segmentation of printed/typewritten optical text images is the priority so that the linguistic

complexity can be reduced before they are fed into the machine for recognition. There are research works where the authors improved various sub-operations of the segmentation process. However, because of the variations in the input and output of those sub-operations, sometimes they face integration issues with other sub-operations. Those conflicting sub-operations are integrated into the segmentation pipeline with necessary modifications.

B. Data Labeling Tool

Another focus of our work is to ease the construction of a good training dataset, as it is a huge factor in getting satisfactory results in the future recognition process. So, in this literature, the design for the data labeling tool is described from the Human-Computer Interaction (HCI) perspective by accessing the standard design criterion (Zimmerman et al., 2007) and analyzing the users' experience feedback. The tool is designed with artificial intelligence so that the user experience can be better, and the data labeling can be more correct. The design details of the tools are described in the later sections.


II. Literature Review

Over the last decade, there has been interest in the construction of Bangla OCR, particularly in the segmentation and recognition of Bangla text. Though this process is challenging for multiple reasons which have been mentioned before. A. Chaudhuri et al. (2016) described mechanisms to build OCR including the segmentation process for multiple languages like English, Latin, French, German, Hindi, and Gujarati. Assamese language may be the closest compared to Bangla Language. There are works about the Assamese language (Borpuzari & Mahanta, 2021; Ghosh et al., 2012; Srinivas et al., 2008). For obvious reasons, they are dependent on the work of Bangla OCR system-related literature.

In the literature, multiple strategies are proposed for the segmentation process of Bangla printed text. As an early effort, B.B. Chowdhury and U. Pal (B. Chaudhuri & Pal, 1997a, 1998; Pal & Chaudhuri, 1994) proposed their method for a complete Bangla OCR. In the proposed approach, they coalesce into multiple sub-operations to construct the Bangla OCR. A piece-wise linear scan, eight stroke-based techniques, and a fill-

circle-based technique are presented to segment the Bangla printed text.

However, there is an issue with touching characters. As an early effort to address this problem, Garain & Chaudhuri (2002) presented a method based on statistical analysis for the segmentation of touching letters in printed Bangla script. They assumed that the touching characters were in the middle zone of the word. They used pattern analysis to separate the touching characters.

S. M. Mahmud et al. (2003) used a two-phase strategy to tackle the issues associated with the segmentation of printed Bangla characters. The proposed method detects merging characters ‘’ (here it is the upper part of “ি” and “ী”) using a recognition-based approach.

M.A. Sattar et al. (2007) proposed the identification-based separation of characters and modifiers like “ি, ী, টি, টী, ে, ো, ঠি, ঠী, ং” etc. However, their method did not segment “ঔ, ঐ”. Akter et al. (2008) also proposed the separation of characters and modifiers like M.A. Sattar et al. (2007) with a scanning-based approach. This approach seems to be pattern specific and constrained with lesser adaptability.

A.F. Rownak et al. (2016) proposed a curved scanning-based character separation approach to preserve the holistic structure of the characters. They used two types of matra/headline zones: the word matra/headline zone, and line matra/headline zone. Although this helps them to separate the numerical characters in the text, their approach sometimes did not segment “ি, ী”. Therefore, the segmentation approach also introduces noise.

A connected component analysis based on a *two-zone approach* was suggested by Zahan et al. (2018). They used a chain approximation algorithm with 8-connectivity for the separation of overlapping characters. Nevertheless, the approach has issues with the segmentation of characters that have a standing part over the matra/headline line inclusively. The suggested approach segments “গ, ঙ, ঞ, ঑” in two distinct parts. Also “ি, ী” get segmented into two distinct parts.

One of the most closely related languages to Bangla is Assamese. Regarding the Assamese

(Borpuzari & Mahanta, 2021; Ghosh et al., 2012; Srinivas et al., 2008), works are done, a large number of which are dependent on the literature connected to the Bangla OCR system.

The character image dataset is another essential part of our OCR construction pipeline. There are Bangla hand-written datasets (Das et al., 2014; P. K. Singh et al., 2018). People worked on various data collection tools from an HCI perspective. Though the purposes and motivations of those are different. In (Ozok, 2007), the authors described the survey design and implementation from the HCI perspective.

Kim et al. (2013) presented an authoring environment that enables users without programming experience to create mobile data management and collection applications for citizen science. To create Bangla printed textual isolated character image dataset, an interactive, semi-automatic data collection tool is built.

Khan et al (2022) designed a tool to simplify the process of collecting data, annotating it, and developing a useful computer agent that could overcome cognitive impairment and dementia of the users with generalized modeling.

The segmentation approach proposed by the above literature can be improved. The segmentation approach of “ি, ী, টি, টী, ে, ো, ঠি, ঠী, ং, ঔ, ঐ” sometimes create conflicting scenario among them so that all of them cannot be segmented. Finally, an intelligent and interactive data collection tool for collecting labeled character images can help in constructing the character image classification model. In our work presented in this paper, we improved the segmentation algorithm for Bangla text and proposed a data collection tool for labeling the segmented image. The data collection tool is built from the HCI perspective which has the desired features (Bass & John, 2003; Bass et al., 2001; Juristo, Moreno, & Sánchez, 2003; Juristo, Moreno, & Sanchez, 2003; Juristo et al., 2007) that meet the HCI criteria.

III. Methodology and Implementation

In this section, the proposed method for segmenting is described for digitally printed Bangla images along with the design of the data collection tool. The proposed method is developed in different

steps, which are described in detail in the following sub-sections.

A. Preprocessing

1) *Binarization, and Skew Correction*: The text document is scanned as an image and then supplied as the input to the segmentation process. Binarization means the conversion of a grayscale image to a binary image. There are various binarization methods like the Fixed Thresholding Method (Pavlidis, 1993; Sezan, 1990), Otsu Method (Otsu, 1979), Niblack Method (Khurshid et al., 2009), Sauvola Method (Sauvola et al., 1997), etc. Otsu Method (Otsu, 1979) is applied here. Otsu is a global binarization method (N. Garg & Garg, 2013) that proved to be a robust method for the OCR segmentation pipeline. The skew correction is achieved using the proposed method in (B. Chaudhuri & Pal, 1997b). This approach is relatively fast and accurate.

2) *Noise Elimination*: Scanned documents often have noise that arises due to printing quality, scanning quality, age of the document, etc. Therefore, it is necessary to filter out those noises. Several types of noise can occur in documents (Ali, 1996) and different methods have been proposed regarding them (Farahmand et al., 2013; Solihin & Leedham, 1997). Noise can be generated from decoding errors or even from noisy channels. Salt-and-pepper noise and background noise (B. Singh et al., 2012) are the most common that occur in scanned documents. In this work, an improved *median filtering approach* (Zhu & Huang, 2012) is used for noise-cleaning followed by applying Contrast Limited Adaptive Histogram Equalization (CLAHE) (Pizer et al., 1990). Using this process, local information details of an image can be enhanced even in regions that are darker or lighter than other parts of the image. To stretch or shrink the intensity levels of the image, *sigmoid correction* (Braun & Fairchild, 1999) is applied to the image. The image orientation is detected by scanning the image based on black pixel frequency and by detecting the interval of the frequency.

An added operation is also applied that clears noise or unnecessary structures in the image after the line segmentation process. This step is necessary because of the other noises that are generated from the thinning process which is described in Section III-B2. In this operation, round/semi-round shaped architectures with area <

x^2 where $x = 0.05 \times \text{Line Segmented Image Width}$, are discarded. It ensures that the differences between ব and র or ড and ড় are preserved while discarding other unnecessary area structures in the image.

B. Segmentation

Segmentation is the most challenging task after the preprocessing. There are steps for segmentation, each of which is described below.

1) *Line Separation*: The OCR system should first find and separate different lines of text from the image file. For this purpose, the digitized image file is scanned horizontally to detect black pixels to find out the starting (left top corner) point and the ending (bottom right corner) point of each line. As the document is now devoid of any kind of skew, lines will be separated with the vertical span of spaces (Alam & Kashem, 2010).

To detect the orientation of the image. The counting for horizontal and vertical scanning in rows and columns is conducted individually. Figure 1 shows an example result of the scanning.



Fig. 1: The black pixel counts of the corresponding orientations are represented by the vertical and horizontal histograms.

The rotation of the image is dependent on two conditions. Firstly, the count of zero black pixels can be found at regular intervals in the horizontal scan. Secondly, the analogous patterns of black pixel count are right-screwed. The image is rotated, as necessary.

After correcting the orientation of the image, the horizontal scanning of images is used for line detection as shown in Figure 1. Row histograms are constructed by counting the frequency of the black pixels for each row. The row count of zero is interpreted as the gap between two lines. This approach is well known among the OCR research

community and has been used in various literature (Ahmed & Kashem, 2013; Ahmed et al., 2012; B. Chaudhuri & Pal, 1997a, 1998; Omeo et al., 2012). The image size is readjusted to a fixed vertical pixel length of $vl(x)$, where x is the input line image, with the proportion of the earlier horizontal pixel length to the new horizontal pixel length $hl(x)$. This size readjustment of the line image is used so that the next steps in the segmentation pipeline can be easily applied. Here, the value of $vl(x)$ is used as 90 pixels.

2) *Thinning*: At this stage, the thinning process (Hakro et al., 2015) is applied to the image. An example line image after noise elimination and thinning is shown in Fig. 2. This thinning approach (N.K.Garg et al., 2010) is well-known about the segmentation of hand-written images and feature extraction (Bag & Harit, 2011; Yadav et al., 2013).

এখন তাঁরা প্রশস্ত কি না। উত্তরে এই দম্পতি বলেন,

Fig. 2: An example of a separated text line after thinning based on skeletonizing.

3) *Matra and Baseline Detection*: A Bangla word can be divided into 3 different portions: upper portion, mid-portion, and lower portion as shown in Fig. 3. The upper-portion and mid-portion are partitioned by the headline or *matra/headline*.



Fig. 3: Sections for a line-segmented image are shown. The “baseline” distinguishes between the characters and the modifiers below the characters; The “upper portion” is the upper slice that is divided by the headline or *matra*; The “mid portion” is the area between the *matra/headline* and baseline. The “lower portion” is the area that lies below the baseline.

In the traditional approaches (Ahmed et al., 2012; B. Chaudhuri & Pal, 1997a, 1998; Pal & Chaudhuri, 1994), the *matra/headline* is detected by maximum pixel count in the constructed horizontal histogram of the line-segmented image. It is not done over the word image but on the line image because it is hard to detect the *matra/headline* for some words (for example খগ) (Pal & Chaudhuri, 1994). the ratio of the candidate

matra/headline row count proportion to the other row’s count is too low for these types of words. To solve this issue, the *matra/headline* is detected over the line segmented thinned picture instead of word segmented thinned images. The detection of the *matra/headline* occurs based on horizontal scanning of the line-segmented image. The line-segmented image is further divided into two equal portions: the upper half portion and the lower half portion. The histogram bar with the highest row count in the upper portion is detected as the *matra/headline*. This entire process partially resolves the error in detecting *matra/headline* for words that have a low ratio of *matra/headline* row count that is already described. It is rare to find a line with a low ratio in a single line image, In the experiments, the method described here successfully detects the *matra/headline* 99.8% of the time. Then the scanning of the lower portion of the line-segmented image is taken into consideration. The detection of the baseline is described by Omeo et al. (2012) with details.

4) *Word Separation*: In multiple proposed methods for word segmentation, the area of a single connected component is considered as belonging to a word. In other approaches, the histogram-based vertical scanning approach is used to separate the words. But words can be disjointed within themselves in terms of the *matra/headline*, for example, খগ. To solve the issue, the horizontal length of the word x is extended by 10% in proportion to the vertical length. Then a filled rectangle is drawn over the word with the newly calculated vertical length and original horizontal lengths of the word. We find that the inter-character gap of a word is 6-10% of the vertical length of the word. Also, the inter-word gap is 14-25% of the vertical length of the word. Because of this approach, the disjointed parts are merged as one single word. Then we picked the area of filled rectangles to separate the word. The resulting image of this process is shown in Figure 4. For example, As can be seen in Figure 4, The two disjoint sub-segments (and) of the word “আগে”, are merged into a single word through the filled rectangle. Finally, the same area is projected from the thinned image to get the final word image.



(a) Connected component

(b) Extended and filled

Fig. 4: An illustration of word segmentation using the filled rectangle. (i) The two words are divided into three parts. The three parts are displayed inside the rectangles; (ii) the rectangle enclosing the word is filled and stretched along its horizontal axis. As a result, the terms' subdivisions are combined and made into one word.

5) *Character Separation*: For segmenting the characters and character modifiers, the *matra/headline* has been removed from the thinned image with some preconditions described below. the baseline detection approach described in section III-B3 is used to separate the modifiers like ষ্, ষ্, র-ফলা, য-ফলা, etc., by erasing the intersection point of the baseline and continuous characters. The intersection example is shown in Figure 5.



Fig. 5: Example of intersection for the baseline and the continuous characters

It is found that some specific patterns are present after the thinning operation is applied. The patterns

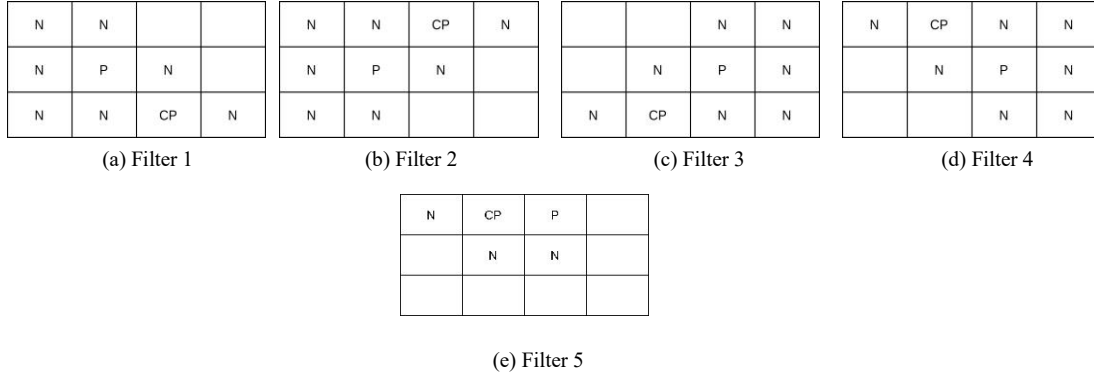


Fig. 6: Filter for *matra/headline* erasing. The matching of filters indicates that the *matra/headline* is found.

In Algorithm 1, the procedure for removing lines from the image is given. The inner procedure named *FindLineFilter* has a list of filter templates inside it to search. It is a simple filter-matching procedure.

Algorithm 1 Remove *matra/headline* from OCR image.

```

procedure ERASINGLINE(imageArray, lineRowIndex)
  lineArray  $\leftarrow$  &ImageArray[LineRowArray]
  lineLength  $\leftarrow$  length(lineArray)
  for i = 0 to lineLength - 1 do
    isLineExist, P Indexes = FindLineFilter(CP = &LineArray[i])
    if isLineExist then
      EraseLine(CPIndex = i, PIndexes = PIndexes)
    end if
  end for
end procedure

```

need to be recognized to remove the *matra/headline* and detect the presence of modifiers in general. So, the filters are introduced here to detect those generalized patterns. Five different filters are shown in Figure 6, which are used for removing the *matra/headline* by a greedy searching approach. In the filters, CP means the starting point of the search, P means the positive finding points and N indicates the negative finding points. When searching, The CP and P have the same pixel color of black in the image. Here, the pixel is the fundamental unit of programmable color on a computer display or in a computer image. The filters in Figure 6a to 6e indicate that there is only *matra/headline* exist. So, the CP and P values are erased with N.

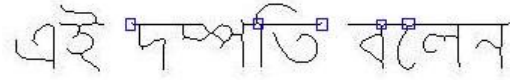


Fig. 7: The sections enclosed by blue-colored rectangles depict the volatility of the *matra/headline*. The filters in Figure 6 are proposed to detect and manage the variations in the *matra/headline*.

The address of the *CP* is passed to the procedure *FindLineFilter*. *FindLineFilter* returns the flag for the existing line declared as *isLineExist* and Positive pixels described as *P* above, their index list. When the value of *isLineExist* is true, the *CP* and *P* pixels are erased.

These multiple filters are used to detect the *matra/headline* because the *matra/headline* line after thinning is not always a straight line. In Figure 7, multiple fluctuations can be seen in the *matra/headline* that is marked by the blue rectangle. Those patterns are greedily searched to obtain the filters in Figures 6a to 6e for removing the *matra/headline*.

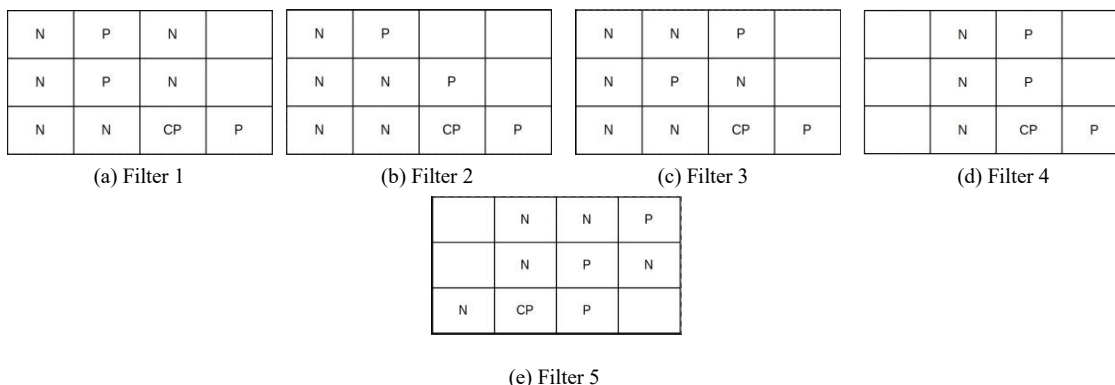


Fig. 9: Upper modifier detection filters.

To separate the characters and modifiers like *রেফ*, *ি*, *ী*, *ঠ* and upper portion of *ট*, *ই*, *ঈ*, *উ*, *ঊ*, etc. with *matra/headline*, an added set of filters are used. Figures 9a, 9b, and 9c indicate the probable presence of *ি*. When these patterns are found, an “up”, “up-right”, and “right” pattern-based custom Depth First Search (DFS) is conducted. It means the first search is conducted for “up” from the starting point. If it does not find the pixel with the same color as the starting point, it searches for up-right and thereon. If the match is found, the starting point is changed to the current positive finding point. If an intersecting point or a horizontal pattern with six pixels count is found, a filled rectangle of the white color is drawn. It separates the merging example in Figure 8a and *টি*, *ঠি*.



(a) Merging example 1 (b) Merging example 2

Fig.8: Example of the merging of characters. (i) The top of *ট* and *রেফ* are connected. (ii) *ি* and *ট* are overlapped.

Moreover, obvious merging issues like those shown in Fig.8 have been resolved and separated in this proposed method. Previously, a fuzzy multifactorial analysis-based approach (Garain & Chaudhuri, 2002) was proposed on the issue. That method only works for touching characters. This problem also can be overcome by applying a contour tracing mechanism (Bishnu & Chaudhuri, 1999) or a greedy search algorithm (J.U. Mahmud et al., 2003).

When the filter in Figure 9e is found, there probably exists a *রেফ* merged with *matra/headline*. CP and P are erased with N. A custom DFS with “up”, and “up-left” patterns is also applied. When an intersection point is found, the second last current positive search pixel is erased, thereby separating *রেফ*.

After searching for filters in Figures 9a, 9b, and 9c, If the filter in Figure 9d is found, it is probable that *ী*, *ঠ* and upper portion of *ট*, *ই*, *ঈ*, *উ*, *ঊ* may exist here. So, a white eraser line is drawn one pixel above the CP pixel in the backward horizontal direction where the eraser line’s length is 20% of the line segmented image’s width. It separates the *ী*. An up, up-left, left pattern-based DFS is conducted here. Upon finding the

intersection point, the second last current positive search pixel is erased. By using these filters, all possible mergers are resolved, and the characters and modifiers are separated. The proposed approach does not separate baseline modifiers that

reside inside the baseline. Also, the process separates ট, ই, ঙ, উ, ঊ in two different portions which can be seen in Figure 10.



Fig. 10: Example of a fully segmented text line.

Finally, the segmented characters are collected as connected components based on their centroidal position. In Algorithm 2, the procedure for separating modifiers and characters is given.

Algorithm 2 Separate characters and modifiers from the OCR image

```

procedure SEPERATINGCHARACTERSANDMODIFIERS (imageArray,lineRowIndex)
  lineArray  $\leftarrow$  &ImageArray[LineRowArray]
  lineLength  $\leftarrow$  length(lineArray)
  for i = 0 to lineLength - 1 do
    filterType, hCount, erasingIndex  $\leftarrow$  FindSeperatorFilter (CP = &LineArray[i])
    if filterType in [a, b, c] then
      if hCount  $\geq$  6 then
        EraseByRectangle (rectangleStartIndex = rectangleIndex)
      end if
      innerFilterType, _, erasingIndex  $\leftarrow$  FindSeperatorFilter (CP = &LineArray[i])
      if innerFilterType is d then
        EraseByBackwardLine (index = ErasingIndex)
        isMergeFound, mergerPixel  $\leftarrow$  SearchForMerge (CP &LineArray[i])
        if isMergeFound then
          ErasingMergerPixels (mergerPixel = mergerPixel)
        end if
      end if
    else if filterType is e then
      PIndexes  $\leftarrow$  GetPIndexes (CP = &LineArray[i])
      isMergeFound, mergerPixel  $\leftarrow$  SearchForMerge (CP = &LineArray[i])
      if isMergeFound then
        ErasingMergerPixels (mergerPixel = mergerPixel)
      end if
      ErasingPositivePixels (CP = i, PIndexes = PIndexes)
    end if
  end for
end procedure

```

B. Image Data Collection and Image Labeling Tool

In addition to the segmentation process, recognizing segmented images is an important task for OCR. The labeled dataset is essential to train a recognition model. In this work, a labeling tool named “OCR Bangla Data Collection Tool (OBDCCT)” is developed to help the creation of the dataset. From the perspective of HCI, usability is one of the main concerns. Over time, the standards related to usability have been developed (Bevan, 2001). These standards can be categorized as follows:

- The use of the product (effectiveness, efficiency, and satisfaction).
- The user interface and interaction.
- The process used to develop the product.

- The capability of an organization to apply user-centered design.

Functional usability features (Juristo et al., 2007) can help us to build usable software. In (Juristo et al., 2007), the functional usability features with design implications are described that include Feedback, Undo/Cancel, User input error correction and prevention, Wizard, User profile, Help, Command aggregation, Shortcuts, and Reuse information. In this sub-section, it is discussed how these features are applied to our tool. The results of a survey are discussed here to understand the usefulness and ease of use of the tools.

1) *Tool Design:* The goal of the proposed labeling tool is to annotate the segmented image to its correct class. The user interface (UI) is designed simply as the user experience is valued the most.

The UI of the proposed system is shown in Figure 11.



Fig. 11: User Interface of OBDCCT

On the UI, the tool supplies c example images related to a class. Top u unlabeled images are shown on UI for labeling. The user tries to match the unlabeled image with the c example images. The tool design, as well as the value of c and u , is based on the requirements and the continuous feedback of the users. The example class images are selected from all the labeled images of the corresponding class. The kmean++ (Vassilvitskii & Arthur, 2006) clustering, preceded by principal component analysis (PCA), is applied to those images where the number of clusters is equal to c . The images on the centroid of clusters are selected as example images. This clustering activity is scheduled for every 6 hours which recalculates the clusters after the addition of newly labeled images. To select the unlabeled images for UI, the

structural similarity of the unlabeled images is analyzed with the example class images using the structural similarity index measure (SSIM) (Hore & Ziou, 2010; Sara et al., 2019; Wang et al., 2004). SSIM is calculated on u unlabeled images and the images are marked with an incremental calculation marker (ICM). The selection of top m unlabeled images for calculation is based on their ICM value. The m unlabeled images with minimum ICM value are selected. ICM ensures an equal probability of all the unlabeled images being shown on UI before the similarity calculation. It also increases the probability of the unlabeled images being labeled beside the first m images.

When an image is labeled, two extra markers are added to the image, l , and w . l denotes the number of times an image is labeled. w denotes the number

of times an image is labeled to the current temporary label. these two markers are used for two different purposes:

- to confirm the labeling
- to discover new classes

For the validation of labeling of an image, a value of l is incremented on the event of labeling, and the current label of the image is checked with the given label on an image labeling event. Initially, for an image that was never labeled before, the value of w is set to 0. When the two labels match or the image is initially not labeled before, the value of w is incremented by 1 and a temporary label is added. Otherwise, the value of w is decreased by 1 if $w > 0$. When the value of w becomes 0, the temporary label gets reset as unlabeled. The temporary label gets passed as a permanent label when $w \geq W$, where W is the voting number threshold. \bar{l} is the average value of l .

$$\bar{l} = \frac{\sum l'}{\|l'\|} \quad (1)$$

To discover new classes, the admin can manually inspect the non-permanent labeled images with a higher value of l . This gives the admin the privilege to look at images to discover a new class or to discard a problematic image.

2) *Functional Usability Features for OBDCT*: According to “Guidelines for Eliciting Usability Functionalities” by N. Juristo et al. (2007), ensuring functional usability features can help to develop usable software from an HCI perspective. In this section, those features (Bass & John, 2003; Bass et al., 2001; Juristo, Moreno, & Sánchez, 2003; Juristo, Moreno, & Sanchez, 2003) are discussed from the perspective of the proposed tool. This tool was developed by continuous feedback from users during requirement analysis, system design, and UI design. The usability elicitation and specification guides described by Juristo et al. (2007) are discussed here.

- **Feedback**: It ensures to keep the users informed about what is going on in the internal system (Constantine & Lockwood, 1999; Nielsen, 1994). There are mechanisms to deal with the implementation of the features on the proposed system, such as showing system status, interaction status,

warning, etc. After the submission, the user is given a status to display (Tidwell, 1999).

- **Undo/Cancel**: It allows the user to cancel completed or ongoing operations (Constantine & Lockwood, 1999; Nielsen, 1994). These features can be implemented by mechanisms like global undo, object-specific undo, abort the operation, go back, etc. Users can Undo (Welie, 2007) the label button to change the image status in the proposed system.
- **User Input Error Prevention/Correction**: It helps prevent the users from making data input errors (Constantine & Lockwood, 1999). Input is taken from a toggle button which is one kind of structured format (Tidwell, 2005).
- **Wizard**: This helps the users to do tasks that require diverse levels with input and correct them (Constantine & Lockwood, 1999). The presented UI is a single-page application and does not have scope for multiple levels of inputs.
- **Help**: This function provides useful help options for users on how to do tasks (Constantine & Lockwood, 1999; Nielsen, 1994). A user manual and command-driven UI instruction are provided with an added similarity-based label image prediction system that helps users label the data.
- **Command Aggregation**: This feature allows the user to create a command to execute more than one task at a time (Constantine & Lockwood, 1999; Nielsen, 1994). This functionality is of no interest to the proposed tool.

IV. Result

In this section, first, the result of the data labeling tool is discussed. Then the result of the segmentation approach is described.

A. Result of OBDCT

The experiment is conducted on 250 API responses for each value of m in Table I. The API responses with images like class labels based on optimistic matching among m images are shown in Figure 11.

TABLE I: Average response time(\bar{t}) and average correct class instance (\overline{cc}) for different values of m . \bar{t} is computed based on the API response time. A rigorous human examination is used to manually

m	\bar{t}	\overline{cc}
50	1.52 sec	2.56
100	2.8 sec	4.89
200	5.48 sec	5.28

\bar{t} is the average response time of the API that calculates the similarity of c class images with u unlabeled images. Based on API response time, \bar{t} is computed. On the other hand, \overline{cc} is the correct class instance which is calculated using manual human assessment. Though the calculation time is the least when m is equal to 50, the number of average correct images shown on UI using artificial intelligence for manual labeling is lower than the other two values of m . Similarly, the number of the average correct images is the highest when the value of m is 200. So, taking the concern of minimal response time and maximal correct class image, the value of m is selected to be 100 for the data labeling tool.

TABLE II: Label error rate (err) and constrained average labeled count (\bar{l}) for voting threshold value (W). The vote threshold value needed to confirm an image’s label is W . The average needed voting count is represented by the constrained

average labeled count (\bar{l}) for different values of W . Likewise, the label error rate(err) which is in percentage, is used to show error frequency for each final labeling.

W	err (%)	\bar{l}
2	0.0010	2.69
3	0.0006	4.45
4	0.0005	6.14

In Table II, the label error rate(err) and constrained average labeled count (\bar{l}) are shown. The result shows that the error rates are low for all the values of W . The difference in error values for values of W is small. But increasing the value of W adds extra validation time and steps for the image labeling which slows down the data collection process. The value of W is set to 2 for the data labeling tool.

Additionally, a survey on the tool is conducted on the “people factor” (Laitenberger & Dreyer, 1998). It covers the usefulness and ease of use of the software. The definitions are given by Davis (1989). The scale items (SI) are given in Table III. In the survey, Likert Scaling is used to collect the rating value of the SI used (Laitenberger & Dreyer, 1998) which includes seven responses: Extremely likely (7), quite likely (6), slightly likely (5), neither (4), slightly unlikely (3), quite unlikely (2), extremely unlikely (1). The numerical values in the bracket correspond to the rating of the responses

TABLE III: Scale Items (SI) of the usefulness and the ease-of-use concept (Laitenberger & Dreyer,1998)

Usefulness
U1 Using OBDCT would enable me to work quickly in my academic/professional work. (Quick)
U2 Using OBDCT would improve my academic/professional work performance. (Performance)
U3 Using OBDCT would improve academic/professional work productivity. (Productivity)
U4 Using OBDCT would make academic/professional work easier. (Makes the job easier)
U5 Using OBDCT would enhance the effectiveness of academic/professional work. (Effectiveness)
U6 I would find OBDCT useful in my academic/professional work (Usefulness)
Ease of Use
E1 Learning to operate OBDCT is easy for me. (Easy to learn)
E2 My interaction with OBDCT is clear and understandable. (Clear and Understandable)
E3 I would find it easy to get OBDCT to do what I want to do. (Controllable)
E4 OBDCT somehow helps to enrich Intellectual skills. (Skill)
E5 It is easy to remember how to perform tasks using OBDCT. (Remember) E6 The tool is easy to use. (Easy to use)

TABLE IV: Mean and standard deviation of rating for SI

SI	U1	U2	U3	U4	U5	U6	E1	E2	E3	E4	E5	E6
\bar{r}	5.60	5.75	5.72	5.85	5.72	6.12	6.30	6.17	5.87	6.04	5.96	6.48
$\sigma(r)$	0.82	0.85	1.03	0.89	0.86	0.82	0.56	0.72	0.87	1.02	0.82	0.51

In Table IV, the average value (\bar{r}) and standard deviation ($\sigma(r)$) of SI are shown. For ease of use, participants find the tool is quick, improves work performance, and is easy to use with a mean rating above 6. The controlling ability of the tool and the ease of remembering is rated with a mean value of 5.86 and 5.96 which are also nearer to the rating value of 6. It is also found that there is a slight disagreement among the participants on E5(Skill) with a higher standard deviation ($\sigma(r)$). Participants find the usefulness of the tools with mean ratings above 5.50. But there is higher disagreement among the participants over the productivity of the tool.

B. Result of Segmentation

The OBDCT incorporates the segmentation approach to collect the data for labeling. Here the result of the segmentation process is discussed. According to our expectation of the segmentation algorithm, the measured character segmentation error rate is 2.8% and the word level segmentation error rate is 0.2%. The main reason for the character level segmentation error is due to characters such as ঞ, ঐ which are sometimes aligned above/with the *matra/headline* line. An extra skew correction for the line after thinning process gave 0.3% less error overhead on character-level segmentation. Therefore, another reason for the character segmentation error is due to the inner baseline error. Since modifiers exist on the upper portion of the baseline, the algorithm sometimes does not segment those modifiers. Therefore ট, ঙ, ঞ, উ, উ, etc. are segmented into two parts because of *matra/headline* deletion.

V. Conclusion and Future Work

This paper is focused on a tool that helps the construction of a dataset for Bangla OCR. As the usage of Bangla is increasing day by day, it must have sufficient technological resources for easier digital use of Bangla and to make it more well-

known and accessible to all kinds of people. Here a segmentation approach is proposed for the Bangla OCR system and an intelligent suggestion-based data labeling tool is proposed for OCR image data labeling. The design of the tool keeps human-computer interaction (HCI) standards.

Researchers have been trying to provide a more accurate segmentation approach concerning Bangla characters for years. Nevertheless, none of them came out entirely successful. Also, the standard OCR corpora does not exist for Bangla. This work is a contribution to the effort of data collection for Bangla OCR. The tool described above can be used for different image-labeling tasks. The tool is being used for further data collection at the time of authoring this paper. It is expected that further work can be done on a better segmentation approach based on using the method described here. Also, the design of the tool can be modified for better user experience and compatibility with other image data labeling works. While the proposed method may not solve all the issues in the existing systems on segmentation and character recognition, we expect that it will open doors on problem-solving strategies for future research in this field.

References

- Ahmed S, & Kashem, MA. 2013. Enhancing the character segmentation accuracy of bangla ocr using bpnn. *International Journal of Science and Research (IJSR) ISSN (Online)*,2319–7064.
- Ahmed S, Sakib, A.N., Ishtiaque Mahmud, M., Belali H., & Rahman, S. 2012. The anatomy of bangla ocr system for printed texts using backpropagation neural network. *Global Journal of Computer Science and Technology*.

- Akter N, Hossain, S., Islam, M. T., & Sarwar, H. 2008. An algorithm for segmenting modifiers from bangla text. In *2008 11th international conference on computer and information technology* (pp. 177–182).
- Alam M. M., & Kashem, M. A. 2010. A complete bangla ocr system for printed characters. *JCIT*, 1(01), 30–35.
- Alginahi Y.M. 2013. A survey on Arabic character segmentation. *International Journal on Document Analysis and Recognition (IJ DAR)*, 16(2), 105–126.
- Ali M.B.H. 1996. Background noise detection and cleaning in document images. In *Proceedings of 13th international conference on pattern recognition* (Vol.3, pp. 758–762).
- Bag S, & Harit, G. 2011. An improved contour-based thinning method for character images. *Pattern Recognition Letters*, 32(14), 1836–1842.
- Bairagi, P.P., & Dulal, G. 2018. Optical character recognition for hindi. *Int. Res. J. Eng. Technol.(IRJET)*, 5(5).
- Bansal V., & Sinha, M. 2001. A complete ocr for printed hindi text in devanagari script. In *Proceedings of sixth international conference on document analysis and recognition* (pp. 0800–0800).
- Bass L., & John, B.E. 2003. Linking usability to software architecture patterns through general scenarios. *Journal of Systems and Software*, 66(3),187–197.
- Bass L., John, B.E., & Kates, J. 2001. *Achieving usability through software architecture* (Tech. Rep.). CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST.
- Bevan N. 2001. International standards for hci and usability. *International journal of human-computer studies*, 55(4), 533–552.
- Bishnu, A., & Chaudhuri, B. 1999. Segmentation of bangla handwritten text into characters by recursive contour following. In *Proceedings of the fifth international conference on document analysis and recognition. icdar'99 (cat. no. pr00318)* (pp. 402–405).
- Borpuzari N, & Mahanta, A.K. 2021. A framework for preprocessing, recognizing and distributed proof reading of assamese printed text. In *2021 international conference on innovative computing, intelligent communication and smart electrical systems (ices)* (pp. 1–7).
- Braun G. J., & Fairchild, M. D. 1999. Image lightness rescaling using sigmoidal contrast enhancement functions. *Journal of Electronic Imaging*, 8(4), 380–393.
- Chaudhuri A., Mandaviya, K., Badelia, P., & Ghosh, S.K. 2016. *Optical character recognition systems for different languages with soft computing*. Springer Publishing Company, Incorporated.
- Chaudhuri A., Mandaviya, K., Badelia, P., & Ghosh, S. K. 2017. Optical character recognition systems for gujrati language. In *Optical character recognition systems for different languages with soft computing* (pp. 217–240). Springer.
- Chaudhuri B., & Pal, U. 1997a. An ocr system to read two Indian language scripts: Bangla and devnagari (hindi). In *Proceedings of the fourth international conference on document analysis and recognition* (Vol.2,pp.1011–1015).
- Chaudhuri B., & Pal, U. 1997b. Skew angle detection of digitized indian script documents. *IEEE Transactions on pattern analysis and machine intelligence*, 19(2),182–186.
- Chaudhuri B., & Pal, U. 1998. A complete printed bangla ocr system. *Pattern recognition*, 31(5), 531–549.
- Constantine LL., & Lockwood, L. A. D. 1999. *Software for use: A practical guide to the models and methods of usage-centered design*. USA: ACM Press/Addison-Wesley Publishing Co.
- Das N., Acharya, K., Sarkar, R., Basu, S., Kundu, M., & Nasipuri, M. 2014. A benchmark image database of isolated bangla handwritten compound characters. *International Journal on Document Analysis and Recognition (IJ DAR)*, 17(4),413–431.

- Davis F. D. 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319–340.
- Farahmand A., Sarrafzadeh, A., & Shanbehzadeh, J. 2013. Document image noises and removal methods. In *Proceedings of the international multiconference of engineers and computer scientists* (Vol.1).
- Gamkrelidze T. V., & Ivanov, V. V. 1990. The early history of indo-european languages. *Scientific American*, 262(3),110–117.
- Garain U., & Chaudhuri, B.B. 2002. Segmentation of touching characters in printed devnagari and bangla scripts using fuzzy multifactorial analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(4), 449–459.
- Garg N., & Garg, N. 2013. Binarization techniques used for grey scale images. *International Journal of Computer Applications*, 71(1), 8–11.
- Garg N. K., Kaur, L., & Jindal, M. K. 2010. A new method for line segmentation of handwritten hindi text. In *2010 seventh international conference on information technology: new generations* (pp. 392–397).
- Ghosh S., Bora, P. K., Das, S., & Chaudhuri, B. 2012. Development of an assamese ocr using bangla ocr. In *Proceeding of the workshop on document analysis and recognition* (pp.68–73).
- Govindaraju V., & Setlur, S. 2009. *Guide to ocr for indic scripts*. Springer.
- Hakro D., Awan, S., Memon, M., AAMUR, A., & MOJAI, G. 2015. Interactive thinning for segmentation-based and segmentation-freesindhi ocr. *Sindh University Research Journal-SURJ (Science Series)*, 47(3).
- Hore A., & Ziou, D. 2010. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition* (pp. 2366–2369).
- Huehnergard J. 2017. Arabic in its semitic context. In *Arabic in context* (pp.3–34). Brill.
- Jawahar C., Kumar, M. P., & Kiran, S. R. 2003. A bilingual ocr for hindi-telugu documents and its applications. In *Seventh international conference on document analysis and recognition, 2003. proceedings.* (pp.408–412).
- Juristo N., Moreno, A., & Sánchez, M. 2003. Architectural sensitive usability patterns. In *ICSE workshop bridging the gaps between usability and software development*.
- Juristo N., Moreno, A., & Sanchez, M. 2003. Deliverable d. 3.4. techniques, patterns and styles for architecture-level usability improvement. *ESPRIT project (IST-2001-32298)*
<http://www.ls.fi.upm.es/status/results/deliverables.html>.
- Juristo N., Moreno, A., & Sanchez-Segura, M.-I. 2007. Guidelines for eliciting usability functionalities. *IEEE Transactions on Software Engineering*, 33(11), 744-758.
- Khan N., Ghosh, R., Migovich, M., Johnson, A., Witherow, A., Taylor, C., Sarkar, N. 2022. A data collection and annotation tool for asynchronous multimodal data during human-computer interactions. In Q. Gao & J. Zhou (Eds.), *Human aspects of it for the aged population. design, interaction and technology acceptance* (pp.201–211). Cham: Springer International Publishing.
- Khosravi H., & Kabir, E. 2009. A blackboard approach towards integrated Farsi ocr system. *International Journal of Document Analysis and Recognition (IJDAR)*, 12(1), 21–32.
- Khurshid K., Siddiqi, I., Faure, C., & Vincent, N. 2009. Comparison of niblack inspired binarization methods for ancient documents. In *Document recognition and retrieval xvi* (Vol.7247, pp. 267–275).
- Kim S., Mankoff, J., & Paulos, E. 2013. Sensr:evaluatingaflexibleframeworkforauthoringmobiledata-collection tools for citizen science. In *Proceedings of the 2013 conference on computer supported cooperative work* (pp. 1453–1462).

- Kumar P. P., Bhagvati C., Negi A., Agarwal, A., & Deekshatulu, B. L. 2011. Towards improving the accuracy of telugu ocr systems. In *2011 international conference on document analysis and recognition* (pp. 910–914).
- Laitenberger O., & Dreyer, H. M. 1998. Evaluating the usefulness and the ease of use of a web-based inspection data collection tool. In *Proceedings fifth international software metrics symposium. metrics (cat. no. 98tb100262)* (pp. 122–132).
- Mahmud J. U., Raihan, M. F., & Rahman, C. M. 2003. A complete ocr system for continuous bengali characters. In *Tencon 2003. conference on convergent technologies for asia-pacific region* (Vol.4, pp. 1372–1376).
- Mahmud S.M., Shahrier, N., Hossain, A.D., Chowdhury, M.T.M., & Sattar, M.A. 2003. An efficient segmentation scheme for the recognition of printed bangla characters. In *Proceedings of iccit* (pp.283–286).
- Märgner V., & El Abed, H. 2012. *Guide to ocr for arabic scripts*. Springer.
- Mathew M., Singh, A. K., & Jawahar, C. 2016. Multilingual ocr for indic scripts. In *2016 12th iapr workshop on document analysis systems (das)* (pp. 186–191).
- Mehran R., Pirsiavash, H., & Razzazi, F. 2005. A front-end ocr for omni-font persian/arabic cursive printed documents. In *Digital image computing: Techniques and applications (dicta'05)* (pp. 56–56).
- Negi A., Bhagvati, C., & Krishna, B. 2001. An ocr system for telugu. In *Proceedings of sixth international conference on document analysis and recognition* (pp. 1110–1114).
- Nielsen J. 1994. *Usability engineering*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Omee F.Y., Himel, S.S., Bikas, M., & Naser, A. 2012. A complete workflow for development of bangla ocr. *arXiv preprint cybernetics*, 9(1),62–66.
- Otsu, N. 1979. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1),62–66.
- Ozok, A. A. 2007. Survey design and implementation in hci. In *The human-computer interaction handbook* (pp. 1177–1196). CRC Press.
- Pal U., & Chaudhuri, B. 1994. Ocr in bangla: an indo-bangladeshi language. In *Proceedings of the 12th iapr international conference on pattern recognition, vol. 3-conference c: Signal processing (cat. no. 94ch3440-5)* (Vol.2, pp. 269–273).
- Pavlidis T. 1993. Threshold selection using second derivatives of the gray scale image. In *Proceedings of 2nd international conference on document analysis and recognition (icdar'93)* (pp. 274–277).
- Pizer S. M., Johnston, R. E., Ericksen, J. P., Yankaskas, B. C., & Muller, K. E. 1990. Contrast-limited adaptive histogram equalization: speed and effectiveness. In *[1990] proceedings of the first conference on visualization in biomedical computing* (pp. 337–338).
- Rownak A.F., Rabby, M.F., Ismail, S., & Islam, M.S. 2016. An efficient way for segmentation of bangla characters in printed document using curved scanning. In *2016 5th international conference on informatics, electronics and vision (iciev)* (pp.938–943).
- Sara U., Akter, M., & Uddin, M.S. 2019. Image quality assessment through fsim, ssim, mse and psnr - a comparative study. *Journal of Computer and Communications*, 7(3), 8–18.
- Sattar M. A., Mahmud, K., Arafat, H., & Zaman, A. N. U. 2007. Segmenting bangla text for optical recognition. In *2007 10th international conference on computer and information technology* (pp. 1–6).
- Sauvola, J., Seppanen, T., Haapakoski, S., & Pietikainen, M. 1997. Adaptive document binarization. In *Proceedings of the fourth international conference on document analysis and recognition* (Vol. 1, pp. 147–152).

- Sezan M.I.1990.A peak detection algorithm and its application to histogram-based image data reduction. *Computer vision, graphics, and image processing*, 49(1),36–51.
- Singh B., Chand, V., Mittal, A., Ghosh, D., et al.2012. A comparative study of different approaches of noise removal for document images. In *Proceedings of the international conference on soft computing for problem solving (socpros 2011) december 20-22, 2011* (pp. 847–854).
- Singh P.K., Sarkar, R., Das, N., Basu, S., Kundu, M., & Nasipuri, M. 2018. Benchmark databases of handwritten bangla-roman and devanagari-roman mixed-script document images. *Multimedia Tools and Applications*, 77(7), 8441–8473.
- Solihin Y., & Leedham, C. 1997. Noise and background removal from handwriting images. In *Proceedings intelligent information systems. iis'97* (pp. 366–370).
- Srinivas B. A., Agarwal, A., & Rao, C. R. 2008. An overview of ocr research in indian scripts. *IJCSES*, 2(2), 141–153.
- Tidwell J. 1999. *The case for hci design patterns*. Retrieved from <https://www.mit.edu/jtidwell/commongroundonefile.html>
- Tidwell, J. 2005. *Patterns for effective interaction design*. O'Reilly.
- Vassilvitskii S., & Arthur, D. 2006. k-means++:The advantages of careful seeding. In *Proceedings of the eighteenth annual acm-siam symposium on discrete algorithms* (pp.1027–1035).
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4), 600–612.
- Welie, M. v. 2007. *A pattern library for interaction design*. Retrieved from <http://www.welie.com/>
- Yadav D., Sánchez-Cuadrado, S., & Morato, J. 2013. Optical character recognition for hindi language using a neural-network approach. *Journal of Information Processing Systems*, 9(1),117–140.
- Zahan, T., Iqbal, M. Z., Selim, M. R., & Rahman, M. S. 2018. Connected component analysis based two zone approach for bangla character segmentation. In *2018 international conference on bangla speech and language processing (icbslp)* (pp. 1–4).
- Zhu Y., & Huang, C. 2012. An improved median filtering algorithm for image noise reduction. *Physics Procedia*, 25, 609–616.
- Zimmerman J., Forlizzi, J., & Evenson, S. 2007. Research through design as a method for interaction design research in hci. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 493–502).