Recognition of Bangla text from Scene Images through Perspective Correction

Ranjit Ghoshal
St. Thomas' College of Engineering and Technology
Kolkata, India
ranjit.ghoshal@rediffmail

Anandarup Roy
CVPR Unit
Indian Statistical Institute, India
roy.anandarup@gmail.com

Swapan Kr. Parui
CVPR Unit
Indian Statistical Institute, India
swapan@isical.ac.in

Abstract—**This article proposes a scheme for automatic extraction and recognition of Bangla text from natural scene images. An image, when captured by a digital camera may have perspective distortion. Before extracting text symbols, this distortion is corrected using Homography transform. For text extraction, headlines are detected using morphology. First, the components attached or close to the detected headlines, are separated. Further, by applying certain shape and position based conditions we could distinguish text and non-text. Afterwards, by removing the headline we partition the text into two different zones. For recognition purpose, the local chain code histograms of input character are used as features. Finally, separate Multilayer perceptrons (MLPs) are used to recognize text symbols reside in different zones. The classifiers are trained using about 7500 samples of 53 classes. We tested our algorithm on 100 scene images.**

I. INTRODUCTION

Automatic recognition of text portions in a natural scene image is useful to blind and foreigners with language barrier. Such a recognition methodology should also employ an extraction of text portions from the scene images. Moreover, segmentation of such text portions has a crucial impact on document processing, content based image retrieval, robotics and intelligent transport systems. With the growing popularity of various image-capturing devices such as digital cameras, mobile phones, PDAs etc, digital images are nowadays easily available. Extraction and recognition of texts from scene images captured by such devices is a challenging problem now a day.

There have been several studies on text segmentation in the last few years. Wu et. al. [1] used a local threshold method to segment texts from gray image blocks containing texts. Assuming texts in images and videos are always colorful, Tsai et. al. [2] developed a threshold method using intensity and saturation features to segment texts in color document images. Lienhart et. al. [3] and Sobottka et. al. [4] used color clustering algorithm for text segmentation. In recent years, Jung et al. [5] employed a multi-layer perceptron classifier to discriminate between text and non-text pixels. A sliding window scans the whole image and serves as the input to a neural network. High probability areas inside a probability map are considered as candidate text regions. Wavelet transform has also been applied for text segmentation. In this context Gllavata et al. [6] considered wavelet transform and K-means based texture analysis for text detection. Saoi et al. [7] improved the method of Gllavata et al. [6] and applied wavelet transform to all of R, G and B channels of input color image separately. More recently Bhattacharya et. al. [8] proposed a scheme based on analysis of connected components (*CCs*) for extraction of Devanagari and Bangla texts from camera captured scene images. Also a few criteria for robust filtering of text components were studied.

Camera based image processing throws many challenges to document analysis research [9]. Images acquired through cameras may suffer from projective distortion, heterogeneity of light and lens distortion. The perspective distortion can be understood as a projective transformation on a planar surface. Perspective distortion does not preserve distance between points also not the angle. Hence, perspective image becomes a barrier for text segmentation. A perspective correction method was studied in [10].

The recognition issue also draws interest of many scientists. Let us take a short review considering works in Bangla language. An earlier work was by Choudhuri and Pal [11] in 1998. That work refers to a design of complete OCR system for printed Bangla documents. Besides, more recently, Pal and Choudhuri [12] presented a good survey on Indian script character recognition. However all the efforts are still done using scanned documents. To the best of our knowledge no current work exists that deals with Bangla text recognition from natural scene images.

Here, a color perspective image is the input. We assume at least two non-parallel lines are present inside the image. We could find such lines in a banner or hoarding image. With these two lines we can detect the perspective distortion. These lines are found by applying Hough transform and some other criteria. Further, we correct the perspective distortion by computing the Homography matrix [10] from a set of points on these two lines. A unique feature of Bangla scripts is the presence of a headline. After perspective correction, the headlines are detected using morphology on skeleton image.

978-1-61284-861-7/11/$26.00 ©2011 IEEE

The headline detection eventually leads us to the zone detection. Here, we could define two zones at either sides of the headline. The middle-lower zone contains most of the Bangla text symbols. The upper zone contains a few specific symbols, hence easy for recognition.

Afterwards, the *CCs* attached with these headlines are separated. These separated components may contain both text and non-text. We apply certain shape and position based conditions in order to distinguish between text and non-text. However, some text portions that are not connected with headline will not be separated by the above procedure. To cope this we increase the area of the bounding box of each *CC* to a specified limit. The portions that lie inside this modified bounding box are now studied against the previous criteria to obtain text. By removing the headline we are able to separate the *CCs* each of which represents a text symbol. Finally, we create a dataset of almost 7500 samples of 53 classes. For recognition MLP is used, one for each zone. Our recognition results reveal satisfactory performance.

Concerning the dataset, to the best of our knowledge, no benchmark dataset of scene images consisting Bangla text is available. The present study is based on a set of 100 outdoor images of signboards, banners and hoardings. Each image is taken with a digital camera with varying degree perspective distortion.

II. Perspective Correction

The scene images are captured through a digital camera. Size of an input scene image varies depending upon the resolution of the digital camera. Usually, this resolution is 1 MP or more. Initially, we down-sample the input image by an integral factor so that its size is reduced to the nearest of 0.25 MP. Next, it is converted to 8-bit grayscale image [8]. After obtaining a grayscale image, edges of the image are now extracted. For this purpose the Sobel edge detector is used. The Sobel method finds edges using the Sobel approximation to the derivative. It returns the edges at those points where the gradient of the image is the maximum. We concern the edges only, instead of the original image, since to find the lines in an image, the edge image is sufficient. After obtaining Sobel edges, we apply the Hough transform on the edge image. The Hough transform is used to find the lines inside the image. The basic method consists of mapping points in Cartesian space (x, y) to sinusoidal curves in (ρ, θ) space via the transformation

$$\rho = x\cos\theta + y\sin\theta.$$

Each time a sinusoidal curve intersects another at a particular value of ρ and θ the likelihood that a line corresponding to that (ρ, θ) coordinate value is present in the original image, increases. An accumulator array *Acc* is used to count the number of intersections at various ρ and θ values. This accumulator *Acc* is further used to obtain four dominating lines (termed Hough lines) according to the *Acc* value. Here, for perspective correction purpose we consider the horizontal/nearly horizontal lines only. Among these lines, the Hough lines correspond to the four highest scores of *Acc*. The four dominating lines are shown in Fig. 1(a).

(a) (b) (c)

Figure 1. (a) Initial four Hough lines, (b) final two refined Hough lines for "Kolkata Police" and (c) The four points forming a rectangle on "Kolkata Police" image.

If all the lines become parallel, we could conclude that the image has no perspective distortion. The parallel lines do not carry any information about perspective distortion and hence the first task is to find two most dominant Hough lines, which are not parallel. Note, from Hough transform we only obtain the equation of a Hough line. Originally the image may contain several small line segments forming a Hough line. All these line segments are detected by testing the collinearity of the points of such segments. After finding the line segments, the points are tested to see if they lie on a single line. Fig. 1(b) presents the two Hough lines after joining small line segments.

Now, consider the top-most and the bottom-most lines with respect to the image. Let, the lines be L_1 and L_2. Without loss of generality, let us denote the smaller line by L_2. We find the end points (x_1, y_1) and (x_2, y_2) of L_1 which are inside the image. Let the line L_3 through (x_1, y_1) and parallel to Y-axis intersects L_2 at (x_3, y_3). Again, the line L_4 though (x_2, y_2) and parallel to Y-axis intersects L_2 at (x_4, y_4). Let L_3 be larger than L_4. Note, due to the perspective present in the image, the two lines always differ in length. We collect the four points (x_1, y_1), (x_2, y_1), (x_3, y_3) and (x_2, y_3) respectively at the corner of a rectangle. On the other hand if L_4 becomes larger than L_3, we have the set of points (x_1, y_2), (x_2, y_2), (x_1, y_4) and (x_4, y_4) respectively. In Fig. 1(c) we present the four points forming a rectangle.

With these four non-collinear points, the corresponding points of the perspective corrected image are calculated. Here, since the larger line (i.e., L_1) is not extended up to the boundary of the image, the points (x_1, y_1) and (x_2, y_2) will be situated within the image. Ideally, we may need the intersection points of L_1 and the image boundary. Doing so will form a large rectangle to cope the perspective. However, finding the points with our strategy also forms a rectangle large enough to correct the perspective distortion. Instead of choosing the longer line L_1 we may also take the shorter line L_2 during formation of the perpendiculars L_3 and L_4. However, the formed rectangle will have a smaller area than the previous. Hence, this rectangle may not correct the perspective distortion well enough compared to the previous rectangle. So, a good strategy is to consider the longer line L_1. Let for each point

(x_i, y_i) of the perspective image, corresponding point of the corrected image is (x_i', y_i'). Though these points seem to be sufficient for getting the corrected position of the pixels, an implementation problem may occur. During real implementation, we define the corrected image matrix being the same size as the perspective image matrix. Initially corrected image matrix is empty. We map (x_i, y_i) of the perspective image to (x_i', y_i') of the corrected image. Now we put the grayscale value of (x_i, y_i) to (x_i', y_i'). In this course, we should round off the values of x_i' and y_i' to produce a valid position inside the correct image matrix. In effect, one or more pixel positions collide and we missed some pixel positions in the corrected image. This problem however could be solved using the following simple technique. The Homography matrix *H* is inverted first. Now, for each corrected image pixel (x_i', y_i') we could obtain the corresponding pixel (x_i, y_i) inside the perspective image using inverted Homography matrix. Finally the grayscale value of (x_i, y_i) is put to (x_i', y_i'). Note, now we will not miss any pixels of the corrected image.

III. HEADLINE DETECTION AND TEXT COMPONENT SEPARATION

The headline is an important feature of Bangla script. Most of the Bangla characters are connected by a horizontal line termed as the headline. Hence, in order to separate out the Bangla characters/symbols we should detect and remove the headline that joins them. Here, we apply mathematical morphology operation to obtain the lines inside the image. Further, applying some filtering we are able to detect possible candidate headline. The procedure works as follows. First, the grayscale image is converted to a binary image using Otsu binarization method [13]. Initially, from the binary image we separate the *CCs*. On each *CC* the skeleton image is constructed by morphological operation. Let us denote the skeleton image by *A*. Now, we take a linear structuring element *B*. The erosion of image *A* by *B*, denoted by *A-B*, is defined as the set of all pixels $P \in A$ such that if *B* is placed on *A* with its center at *P*, *B* is entirely contained in *A*. The dilation operation, on the other hand, is in some sense dual of Erosion. For each pixel $P \in A$, consider the placement *B(P)* of the structuring element *B* with its center at *P*. Then Dilation of image *A* by structuring element *B*, denoted by *A+B*, is defined as the union of such placements *B(P)* for all $P \in A$. Finally, opening of *A* by the *B* is defined as *(A-B)+B*.

It is evident that opening of an image *A* with a linear structuring element *B* can effectively identify the horizontal line segments present in a *CC*. However, a suitable choice of the length of this structuring element is crucial for processing in the latter stage and we empirically selected its length as 21 for the present problem. The effect of skeletonisation removes pixels on the boundaries of the image but does not allow components to break apart. We do not need all the pixels to find a line. After skeletonisation only the required pixels remain and thus a less number of lines detected. However, if no such lines are detected after morphology on skeleton image, we need to apply morphology on the actual *CC* image. Note, we may encounter a component small enough that we could not find any horizontal line. Such components remain as it is. One may argue, instead of using morphology we could use the Hough transform to obtain the horizontal lines. However, Hough transform generally consume more time than the morphology based approach. We need to apply Hough transform to each of the *CCs*. Since we may have a large number of components, Hough transform seems to slow down the process.

Now, we have a set of lines over the image. We next apply some simple criteria to separate out the possible headline. Let a detected line be denoted by L_i. Let H_u and H_l be the heights of the portions of the corresponding *CCs* lie at the upper and at the lower half to the line L_i. Bangla characters mostly lie at the lower portion of the headline, only a small portion of it may resides at the upper of the headline. So, L_i becomes a candidate headline if $H_u < H_l$. Thus, at the next step we sort out all the lines L_i for which $H_u < H_l$. All these lines are candidates for a headline. Here, we select the longest line among these lines to be the headline.

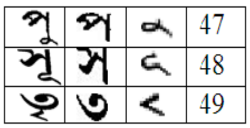
A. Separation of headline attached text component

After finding the headlines, the components attached with these headlines are separated. These components may include text as well as non-text. However, we observe certain characteristic of the text portions. These observations can be used to seperate out text components. Below we describe the conditions that are used to identify text. All the *CCs* are subjected to these conditions in the same sequence as given. One important note here, is, the thresholds specified in one condition are found after performing the previous conditions. Thresholds may differ if the sequence is altered.

- *Elongatedness ratio:* For elongatedness ratio (ER) we use the measure designed by Roy et. al. [14]. Empirically it is found that a component with ER value greater than is a text part.
- *Number of holes:* Usually text like patterns contain less number of holes than that of non-text patterns. Using the Euler number we calculate the number of holes inside a component. Found empirically, a text component has less than 9 holes.
- *Aspect Ratio:* The aspect ratio of a non-text component is either very small or very large. We found by experiments, that the aspect ratio of a non-text component become less than 0.3 or greater than 2.0.
- *Object to background pixels ratio (r):* Due to the elongated nature of Bangla or Devanagari texts only a few object pixels fall inside text bounding box. On the other hand, elongated non-texts are usually straight lines, so, contribute enough object pixels. We observe $0.3 \le r \le 3$ could identify text components.

After the above procedure, all the headline attached text components are separated.

(a)



(b)

Figure. 2. (a) All text symbols in different zones. (b) The lower zone text symbols separated from middle-lower zone.

B. Separation of headline unattached text component

As discussed above, we successfully separate headline attached text portions. However, in Bangla some text components do not meet the headline. Now we consider such components. These components, though not connected, must be close to one/more of already detected text components. Then if we increase the area of the bounding box enough, the possible text components may lie inside it. With this view, we increase the width of the bounding box by its height and the height by an empirical threshold. Now, the components inside this modified bounding box are now subjected to the text identification conditions (Section III-A).

C. Headline removal and text symbol separation

In Bangla script, the headline connects most of the characters in a word. Thus, in order to separate out the characters, we need to remove the headline successfully. Most of the characters, however, are joined with the headline itself. If the headline is removed totally these characters become separated. So, we select and remove only that portion of the headline that left intact by any character. In this course, let us examine all the pixels over the detected Headlines. Let such a pixel p_i (the ith pixel) belong to a component CC_j. Let v_i be the vertical run length of p_i at the ith position considering only the corresponding component CC_j. We experimentally set the thickness of the headline to be H (say). Now, if $v_i \leq H$, we can conclude that p_i is a pixel solely from the headline and thus removed.

After removing the headline, several small components may exist in the image. Since, the actual Bangla characters are thick enough, we may remove the components with smaller thickness. Let h_i and v_i be the horizontal and the vertical run lengths of a pixel p_i at the ith position of a component CC_j. We next compute the minimum of h_i and v_i and further constitute a set $MIN_j = \{m_i \mid m_i = \min(h_i, v_i), \forall i\}$. Thus MIN_j denotes the set of all the minimum run lengths considering all the pixels of CC_j. The thickness T_j of the component CC_j is defined to be the maximum element of the set MIN_j. We have set a predefined tolerance threshold (T_H say) for component thickness. If $T_j < T_H$ we could remove the corresponding component CC_j.

In previous studies Chaudhuri et. al. [11], the headline is used to partition a Bangla word into three zones, namely, the upper, the middle and the lower zones. The upper zone is the partition, above the headline. However, to obtain the lower zone, one may need to identify the "baseline". This adds additional works. Instead, we drop detecting lower zone and proceed with the upper and the middle-lower (combining middle and lower) zone at the lower portion of the headline. Finally, a dataset of almost 7500 text components with 53 classes is created. A small sample set consisting of a few text component images from the present dataset is shown in Fig. 2.

IV. Bangla Text Recognition

The binary text symbol image is now resized to predefined dimension. Afterwards, median filter is applied to remove certain perturbations. Note, the middle-lower zone can be decomposed into two. Here, we consider recognition of only upper and the middle zone text symbols. In this context we have to refine out the middle zone text from the middle-lower zone text symbols. Parui et. al. [15] described a method to filter out and recognize lower zone symbols from middle-lower zone text. Using this method we separate and recognize all the lower zone text symbols (47-49). Hence, only middle zone (1-46) and upper zone (50-53) symbols remain.

Here, chain code histogram features [16] of an input character image have been computed from its contour representation. The skeletal [16] representation is not considered here, since, this representation is often affected by the presence of hair(s), removal of which is usually difficult. With the histogram features, we now apply MLP separately to each zone, for recognition.

V. Results and Discussion

We obtained results based on 100 test images captured by a SONY DCR-SR85E handy cam used in still mode (1.0 MP). All the images are down-sampled to in 576×432 pixels. The images are focused on Bangla text present in outdoor images. Certain amount of perspective distortion is imposed to the images, during capturing.

A. Results of perspective correction

This section presents results obtained from various steps up to perspective correction. First, consider the “Panio jal” image. The original color image is presented in Fig. 3(a). Fig. 3(b) shows the four points forming a rectangle on this image, and the perspective corrected image is shown in Fig. 3(c). Here, we observe that perspective is almost successfully corrected.

Now, consider Fig. 4(a) presenting "Kolkata Police" image. The perspective corrected image is shown in Fig. 4(b). Here, also our algorithm works correctly. The next "Nabadiganta" image (Fig. 5(a)) contains other objects than text symbols. Also, the text is heavily distorted. Yet, the perspective is almost successfully corrected, as shown in Fig. 5(b).

B. Text extraction results

This section describes the results after applying our text extraction algorithms. Here, the text portions are white in color. Consider "Kolkata Police" (Fig. 4). We describe the steps towards text symbol extraction. After removing sufficiently small and large *CC*, the result is shown in Fig. 4(c). The detected headlines are then shown in Fig. 4(d). All *CCs*, attached with headlines are presented in Fig. 4(e). The first two operations (ER and number of holes) have no effect on this image. The result of aspect ratio testing is shown in (Fig. 4(f)). Note, we can filter out the non-text surrounding the text portion. The next another one operation (Object to background pixels ratio) have no effect on this image. Next, the headline unattached symbols are identified (Fig. 4(g)).
We may notice, the symbols absent in Fig. 4(f) are now identified successfully. Fig. 4(h) shows symbols after removing the headlines. Yet, some small non-text are still present. Finally, such components can be removed using the thickness measures ((Fig. 4(i)).
Next, consider the "Nabadiganta" image. The extracted text portions are shown in (Fig. 5(c)). We can successfully remove the headline to get the text symbols (Fig. 5(d)).
Now, let us consider another scene image "Panio jal" shown in Fig. 6 (a). Here, text portions are surrounded by a rectangle similar to text color. The extracted text symbols are shown in Fig. 6(b). Finally, after removing headline, the image is shown in Fig. 6(c).
In order to assess the text extraction and headline removal algorithm we now take the "axis bank" image (Fig. 7(a)). This image has no perspective distortion; instead a certain amount

(a) (b) (c)

Figure 3. (a) Input "Panio jal" image, (b) The four points forming a rectangle on "Panio jal" image and (c) after perspective correction.

of skew is present. The extracted text is shown in Fig. 7(b). The headline removal procedure performs satisfactory considering this case also, according to Fig. 7(c), (d).
Now, take a case where the perspective detection fails. Consider the "masla" image (Fig. 8(a)). This image has varying perspective in a sense that different parts of it have different perspective distortion. The Hough transform can detect nonparallel lines inside the image (Fig. 8(b)). This varying perspective is not a common scenario. Present method is unable to correct such distortions.

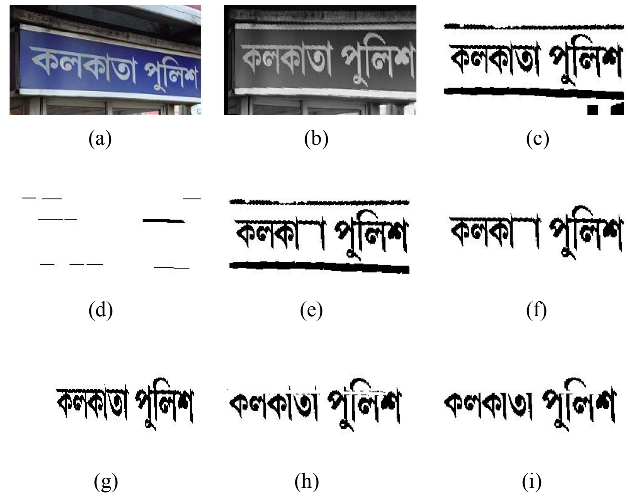
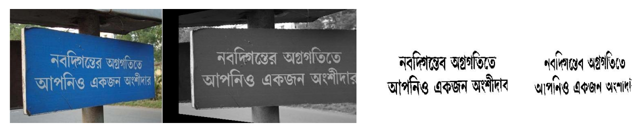


Figure 4. (a) Input "Kolkata Police" image, (b) after perspective correction, (c) binary image after removing small and large CC, (d) after detecting headlines, (e) all CCs attached with headlines, (f) after performing aspect ratio, (g) final text portions, (h) after removing headline, and (i) after removing smaller thickness CC.

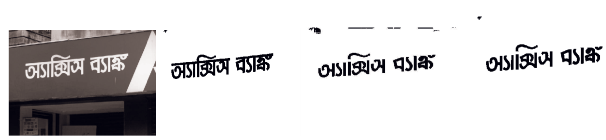


(a) (b) (c) (d)

Figure 5. (a) Input "Nabadiganta" image, (b) after correcting perspective, (c) after extracting text portion and (d) after removing the headline components.

(a) (b) (c)

Figure 6. (a) Input "Panio jal" image. (b) The extracted text symbols and (c) after removing the headline components.



(a) (b) (c) (d)

Figure 7. (a) The "axis bank" skewed image,(b) extracted Bangla text, (b) the image after headline removal and (c) after removal of small thickness CC.

(a) (b)

Figure 8. (a) The "masla" image, (b) lines detected by Hough transform.

TABLE I. RECOGNITION ACCURACIES (RA) OF THE MIDDLE ZONE TEXT SYMBOLS (IN %)

Class #	RA	Class #	RA	Class #	RA	Class #	RA	Class #	RA	Class #	RA	Class #	RA	Class #	RA
1	86.3	7	99.1	13	80.7	19	89.2	25	72.1	31	100	37	73.2	43	95.3
2	82.3	8	87.5	14	76.1	20	100	26	95.7	32	65.5	38	70.1	44	92.5
3	90.1	9	96.1	15	100	21	97.5	27	88.4	33	79.1	39	100	45	100
4	75.5	10	96.7	16	95.3	22	100	28	72.3	34	82.4	40	78.3	46	94.2
5	100	11	82.3	17	82.6	23	68.7	29	97.5	35	98.5	41	94.7		
6	93.2	12	75.4	18	84.7	24	83.5	30	99.3	36	91.7	42	100		

Further, we here do not consider images having both perspective distortion and skew. That situation is more general and seldom happens in real. Those cases can be handled after an enhancement in the current framework.

When both perspective distortion and skew are present, one should carefully choose the order of correction. The skew may be corrected first, then the perspective, or the vise-versa. Both the orders may incur some errors. Nevertheless, handling such situations may be an interesting research problem.

C. Text recognition results

We now apply the work of [15] to separate out and recognize all lower zone symbols from middle-lower zone. Besides, the middle and upper zone symbols are recognized separately, using MLP. We have a total of about 600 upper zone symbols and 4900 middle zone symbols in our training set.

The Table I presents the recognition results of the middle zone text symbols. Table II displays the recognition accuracies at lower and upper zones. We could notice satisfactory recognition for all classes.

VI. SUMMERY AN FUTURE SCOPE

This article provides an automatic extraction and recognition of visual Bangla text entities embedded in outdoor natural scene images through perspective correction. After extracting the Bangla text symbols we remove headline(s) to divide the text symbols into upper and middle-lower zones. After that, we have created a database of all possible Bangla text symbols. For recognition purpose, the database is divided into two parts. The first part contains upper zone and the second part, middle-lower zone text symbols. Lower zone text symbols are separated and recognized using [15]. The classification is done by MLP classifier. The proposed method is robust with respect to font, printed scanned documents and natural scene Bangla text images. Future study may aim towards combining machine-learning tools to improve recognition.

TABLE II. RECOGNITION ACCURACIES OF LOWER AND UPPER ZONE TEXT SYMBOLS

Lower zone class	Recognition Accuracy	Upper zone class	Recognition Accuracy
47	98.43	50	98.31
48	96.57	51	96.43
49	97.21	52	97.65
		53	98.41

REFERENCES

[1] V. Wu, R. Manmatha, and E. M. Riseman, “Textfinder: An automatic system to detect and recognize text in images,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.21, no.11, pp. 1224--1229, 1999.

[2] C.Tsai and H.Lee, “Binarization of color document images via luminance and saturation color features,” IEEE Trans. on Image Processing, vol.11, no.4, pp. 434--451, 2002.

[3] R.Lienhart and F.Stuber, “Automatic text recognition in digital videos,” in Proc. of the SPIE (Image and Video Processing IV 2666), 1996, pp. 180--188.

[4] K.Sobottka, H.Bunke, and H.Kronenberg, “Identification of text on colored book and journal covers,” in Proc. of the International Conference on Document Analysis and Recognition, 1999, pp. 57--63.

[5] K.Jung, I.K. Kim, T.Kurata, M.Kourogi, and H.J. Han, “Text scanner with text detection technology on image sequences,” in Proc. of Int. Conf. on Pattern Recognition, vol.3, 2002, pp. 473--476.

[6] J.Gllavata, R.Ewerth, and B.Freisleben, “Text detection in images based on unsupervised classification of high frequency wavelet coefficients,” in Proc. of Int. Conf. on Pattern Recognition, vol.1, 2004, pp. 425--428.

[7] T.Saoi, H.Goto, and H.Kobayashi, “Text detection in color scene images based on unsupervised clustering of multihannel wavelet features,” in Proc. of Int. Conf. on Doc. Anal. and Recog., 2005, pp. 690-694.

[8] U.Bhattacharya, S.K. Parui, and S.Mondal, “Devanagari and bangla text extraction from natural scene images,” in Proc. of the Int. Conf. on Document Analysis and Recognition, 2009, pp. 171--175.

[9] D.Doerman, J.Liang, and H.Li, “Progress in camera-based document image analysis,” in Proc. of Int. Conf. on Doc. Anal. and Recog., 2003, pp. 606--616.

[10] R.Hartley and A.Zisserman, Multiple View Geometry in Computer Vision, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[11] B.B. Chaudhuri and U.Pal, “A complete printed bangla ocr system,” Pattern Recognition, vol.31, pp. 531--549, 1998.

[12] U.Pal and B.B. Chaudhuri, “Indian script character recognition: A survey,” Pattern Recognition, vol.37, pp. 1887--1899, 2004.

[13] N.Otsu, “A threshold selection method from gray-level histograms,” IEEE Trans. System Man and Cybernetics, vol.9, no.1, pp. 377--393, 1979.

[14] A.Roy, S.K. Parui, A.Paul, and U.Roy, “A color based image segmentation and its application to text segmentation,” in Proc. of Ind. Conf. on Computer Vision, Graphics \& Image Processing., 2008, pp. 313--319.

[15] S.K. Parui, U.Bhattacharya, A.Datta, and B.Shaw, “A database of handwritten bangla vowel modifiers and a scheme for their detection and recognition,” in proceedings of the Third Workshop on Computer Vision Graphics and ImageProcessing., 2006, pp. 204--209.

[16] U.Bhattacharya, M.Shridhar, and S.K. Parui, “On recognition of handwritten bangla characters,” in Proc. ICVGIP, 2006, pp. 817-828.