



Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

An efficient ROI detection algorithm for Bangla text extraction and recognition from natural scene images

Rashedul Islam, Md. Rafiqul Islam, Kamrul Hasan Talukder

Computer Science & Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

ARTICLE INFO

Article history:

Received 19 June 2021

Revised 25 January 2022

Accepted 3 February 2022

Available online 2 March 2022

Keywords:

HOG

SVM

Connected component

Vertical projection

Filtering

Character recognition

ABSTRACT

This research work plays a significant role in finding information from the scene images to fulfill the demand of real life applications like detection of license plate, navigation of robot and helping the visually impaired persons. Here, a new algorithm has been proposed and applied on the scene images to extract Region of Interest (ROI). All the Bangla words are then separated from a sentence by analyzing and applying the Connected Component (CC) method along with bounding box technology. Another new algorithm has been proposed and applied to apart and bring-out Bangla characters from the Bangla words. This algorithm works by the method of vertical scanning of the images of Bangla words. Finally, the extracted characters are recognized by using the Support Vector Machine (SVM) as a classifier which works with Histogram of Oriented Gradient (HOG) features. There are 500 scene images with variations in colors, writing styles and orientations in our designed database. The proposed algorithm yields the accuracy 92.70% and 93.23% in extraction of ROI and character respectively. In the recognition of Bangla characters (digits, Basic characters, and joined characters), the average accuracy is 99.16%. The recognition accuracy of Bangla characters using Convolutional Neural Network (CNN) is also calculated and the obtained result is 83.52%.

© 2022 The Authors. Published by Elsevier B.V. on behalf of King Saud University. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

Images of poster, banner, road sign, bill board, license plate, etc. are considered as natural scene image. Such images contain many useful textual information and these types of information play a significant role in today's knowledge based economy and so perception of information has evolved to a mandatory factor (Francis and Sreenath, 2017). The retrieved information from scene images may also play vital role in text based image indexing, conversion of text to speech, text mining (Bouakkaz et al., 2018), robotics, recognition of license plate (Zhu et al., 2016; Zhang and Zhao, 2013) etc. Researchers have proposed many approaches to identify and locate texts in the scene images for specific applications including page segmentation (Jain and Zhong, 1996; Tang et al., 1996), address block location (Yu et al., 1997), license plate

location (Cui and Huang, 1997), and content-based image/video indexing (Zhang et al., 1994; Shim et al., 1998). To design a standard method that can extract textual information is still a difficult issue as there are variations in different parameters of Bangla text like writing styles, size, color, and alignment of font, light intensity, blurry image, noise, etc. Some objects present in scene images like doors, borders, windows, leaves, etc., create false positives.

Text understanding from the natural scene images includes two major chores that are text detection and text recognition (Francis and Sreenath, 2019). According to Sun et al., 2015, the major groups of the existing text detection methods are: sliding window-based methods (Chen and Yuille, 2004; Wang et al., 2011), CC based methods (Neumann and Matas, 2013), and hybrid methods (Pan et al., 2011; Neumann and Matas, 2013). The Maximally Stable Extremal Region (MSER) based process secured the first position in both the International Conference on Document Analysis and Recognition (ICDAR) 2011 and ICDAR 2013 competitions (Shahab et al., 2011; Karatza et al., 2013). This method is under the CC based method. Though the task of scene text detection is successfully done by the MSER method, it produces huge amount of false positives and can be claimed of causing decrease of precision and *f-measure*. The weakness of the proposed algorithm (Bhattacharya et al., 2009) is that, the algorithm fails to detect

E-mail addresses: rashedcse98@ku.ac.bd (R. Islam), dmri1978@ku.ac.bd (M.R. Islam), khtalukder@ku.ac.bd (K.H. Talukder)
Peer review under responsibility of King Saud University.



Production and hosting by Elsevier

<https://doi.org/10.1016/j.jksuci.2022.02.001>

1319-1578/© 2022 The Authors. Published by Elsevier B.V. on behalf of King Saud University.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

curved text if the size of the text is not sufficient large. The limitation of the work (Ghoshal et al., 2011) is that, it can work only with highlighted texts. The weakness of the recent work Moyeen et al. (2013) is that, it can't recognize the combined character properly. The weakness of the proposed approach of Islam et al. (2016) for text detection is that the method was confined on detection of text regions only. There was no indication to extract characters from the words. Another weakness is the use of less number of input images. Another recent work on detection of Bangla text from scene images is performed by Islam et al. (2017). The proposed approach produced more false positives which influences the performance of the proposed method. The weakness of the proposed algorithm of Dey et al. (2017) is that the precision of the algorithm is 52% and this method didn't propose any specific features for eliminating false positives. Sen et al. (2022) described end-to-end scene text recognition system. They have used matra removal technique for isolation of character from the words. But main drawback of matra removal technique is that if matra is removed, some character may be changed to another character specially in Bangla language. We have discussed about this problem in Section 4 subSection 4.1.5. So, this is the weakness of this method.

Considering the above mentioned complexities of text detection and localization, a new algorithm is proposed and applied to detect ROI from the scene images. In this algorithm, at first the regions without any text are filter-out on the basis of the adaptive threshold values T_h and T_v . T_h and T_v are calculated for Horizontal Projection Profile (HPP) and Vertical Projection Profile (VPP) respectively. Further non-text regions are removed from the scene images using some distinguishable characteristics of text region: area, aspect ratio, solidity etc. These are the morphological features of text regions whose range is based on heuristic rules (Epshtein et al., 2010; Ghanei and Faez, 2017; Tian et al., 2017; Yao et al., 2012). Then the words are localized by bounding box technology and separation of words is performed by analysis of CCs. Since the characters in Bangla text are linked by a upper-line called 'matra', so a new algorithm (algorithm-2) is proposed and applied to take apart them (characters) from the words. The width and height of the extracted characters are set to 32 pixels and stored them in a database which contains Bangla digits, characters and joined letters. From this database training and test data sets are prepared. Features are extracted from training and test sets using HOG method. The features which are extracted from the training set are used to train the SVM classifier. Finally, the accuracy of recognition is shown by generating a confusion matrix. The CNN has been used for recognition of Bangla characters and a comparison is made between the proposed method and the CNN approach. The originality and contribution of this work are given below.

1. A new pre-processing technique is introduced to filter unwanted regions from the scene images.
2. A new system architecture has been designed to accomplish the research work. The experimental results are better in comparison with the existing methods.
3. To extract ROI from the scene images, a new algorithm is proposed.
4. Another algorithm has been proposed to extract Bangla characters from the ROI.
5. A database of different categories of scene images containing Bangla texts has been created and the related experiments have been done using the proposed architecture and algorithms.
6. A database of Bangla characters is created where there are 500 images for every Bangla digits, basic characters and some joined characters. In the database there are a total of 31000 images of Bangla characters.
7. An experiment using the CNN has been performed and it is proved that the proposed method is a better approach.

The remaining sections are arranged by the following way. Some similar researches on natural scene images are described in Section 2. Section 3 represents remarkable features of Bangla text. In Section 4, the proposed method is discussed. Experimental results are focused in Section 5. Section 6 gives conclusions of the paper.

2. Related Works

The researchers face great challenge in detecting text from the scene images and for this reason, different kinds of methods have been introduced. In (Unar et al., 2018), different combinations of CC, texture, stroke and edge based method have been discussed to identify and locate text regions from scene images. In (Yu et al., 2016; Silva and Ciarelli, 2016; Lee and Cho, 2010), edges of text characters are detected by the method of detection of edge.

The texture based method combines histogram of oriented gradient (HOG) and local binary pattern (LBP) (Soni et al., 2019; Bai et al., 2013). Discrete cosine transform (DCT) and wavelet transform are also in the group of texture based method. To detect dense characters, these methods play vital role. But their performance is not good when text alignment is different in the same image.

Stroke width of text components plays an important role to apart the non-text components from the text. Stroke width based method is used to segment the text components (Yi and Tian, 2011; Zhang et al., 2017; Fabrizio and Seidowsky, 2016). In this method, the obtained results become poor for the case where the background of the image is complex.

CC based method (Bai et al., 2013; Epshtein et al., 2010; Fabrizio and Seidowsky, 2016) is one of the important methods to group small CCs into large one using bottom up approach. In this method, CCs are extracted using edge detection and color clustering methods. To get a good result, some heuristic rules like area, aspect ratio, etc. are necessary. Recently invented method of text detection like MSER (Matas and Chum, 2004) is based on the CC based method. The MSER methods are sensitive to blurriness presents in an image (Li and Lu, 2012; Matas and Chum, 2004). Research is going on in the field of extraction of Bangla text from the scene images (Moyeen et al., 2013).

Two parameters of CC as height and standard deviation can separate CCs from scene images (Bhattacharya et al., 2009). This method can recognize Bangla text from scene images. The method was applied on 100 images. Precision and recall rate achieved by this method is 68.8% and 71.2% respectively. To extract Bangla text from scene images, research was carried using the morphological operation (Ghoshal et al., 2011). The limitation of the method was that it could work only with highlighted texts only. Segmentation of CC and text area detection is performed by the algorithm.

Another approach (Moyeen et al., 2013) is proposed for detection and recognition of Bangla text from natural scene images for mobile application. The authors have used several techniques like efficient binarization, proper filtering, 'matra' detection and modified CC analysis. For recognition purpose, they have used google's OCR engine 'tesseract'. They achieved 90% recognition rate where the image contained only text. the proposed method was limited to the images of highlighted text only.

There is no recent research work using deep learning related to our approach. There are some recent works only on the detection of English text from scene images using the deep learning method. Some of such types of recent works have been discussed below. Most of the recent research work on scene text detection and localization are conducted by deep network models. Most importantly, deep learning approaches are free from the task of designing and testing low level handcrafted features, which give an acceleration to a work. The overall process of text detection is being simplified

by deep learning by reducing the number of laborious steps and time-complexity. Moreover, deep learning based methods produce significant improvement over traditional approaches on public data sets (Khan et al., 2021).

Recent development of deep learning based approaches towards text detection from scene images demonstrates its high performance in complex environments, leading to effectiveness and robustness into the problem. Several deep learning based methods have been reported so far to tackle with extremely diverse scene texts (Wang et al., 2019; Zhang et al., 2019; Xue et al., 2019; Kobchaisawat et al., 2020; He et al., 2020 and Ma et al., 2020). According to our analysis, state of the art deep learning approaches are broadly categorized into four groups, (1) regression-based methods, (2) segmentation based methods, (3) hybrid methods, and (4) end to end text spotting. The regression based method can detect text regions by convolving rectangle or quadrilateral text boxes in multiple directions over the entire image (Gao et al., 2019; Liao et al., 2018). In segmentation based methods, text regions are segmented based on text intrinsic information obtained from the scene images (Yang et al., 2018; Tang and Wu, 2017; Qin et al., 2019). In this approach, computationally expensive post-processing techniques are needed for efficient text component extraction from segmented regions. The hybrid methods which are basically a combination of regression-based and segmentation-based methods able to detect texts more accurately (He et al., 2017; Zhong et al., 2019; Lin et al., 2017). The end to end text spotting methods combine both text detection and recognition for accurate text spotting in scene images (Busta et al., 2017; Li et al., 2017; Sun et al., 2018; Liao et al., 2019; Qiao et al., 2020).

3. Distinguished Features of Bangla Text

Bangla text has some distinguishable characteristics than English text. In Bangla language, texts are written using different types of modifiers like vowel and consonant modifiers. These modifiers can be used in any side of a Bangla letter. Use of curve line above a character is another feature of Bangla language. There are few Bangla letters that have curve line above them. The only one unique property of Bangla text is the use of headline or 'matra'. It is located at the top position of a character. 32 Bangla characters have full 'matra' and 8 characters have half 'matra' and 10 characters do not have any 'matra'. A Bangla text may be cloven into three regions or zones like upper, middle and the lower region. The base line is the imaginary line that divides the lower region from the middle region. All the zones of Bangla texts are shown in Fig. 1.

Like English language, there is no upper case and lower case letter in Bangla language. As the Bangla texts have the above mentioned features, it has become a challenging task to detect and recognize them properly. Fig. 2 shows all of the Bangla numerals, Basic characters and some compound characters.

4. Proposed Method

Character extraction and recognition are the two major phases of the proposed method. Fig. 3 shows various phases of the proposed method which gives a clear concept about the method. A brief description of Fig. 3 is stated below.

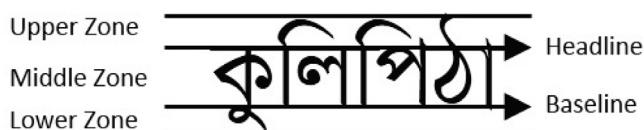


Fig. 1. All the zones of a Bangla text.

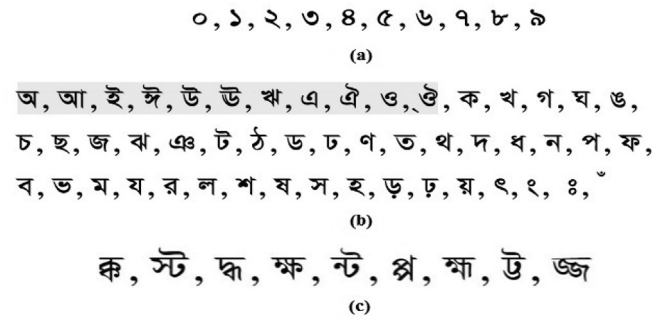


Fig. 2. Bangla character set (a) Digits (b) Vowels and consonants (c) Compound characters.

At first scene image is captured by digital camera and then some preprocessing measures (convert to gray image, contrast enhancement and convert to binary image) are taken. Then ROI is extracted automatically from the binary image. To extract ROI, we have implemented a new algorithm (algorithm-1). Bangla words are detected and segmented from the ROI by analyzing CCs of the ROI. Here each of the Bangla words is treated as CC since the words are connected by 'matra' or headline.

Another algorithm (algorithm-2) has been used to extract individual Bangla characters from each of the Bangla words. In this stage CC analysis technique is used to localize and isolate the Bangla characters. To get actual Bangla characters, we have adopted filtering technique to remove false positives. Finally Bangla characters are stored as character data set. The obtained character data set is divided into two sets (training and test sets). For character recognition, HOG features are extracted from both the training and test sets. Multiclass SVM is used to train the classifier with the obtained features. Obtained features from training set and that from test set are then matched and a confusion matrix is produced. The confusion matrix helps to calculate accuracy of recognition.

The illustration of various phases of Fig. 3 is given in the next sections.

4.1. Character Extraction

Character extraction is composed of the following steps.

4.1.1. Preparation of the database of scene images

Here it can be stated that, the first image data set is provided by the proposed method. The main sources of images are posters, banners, billboards, signboards etc. containing Bangla texts. As the images are taken from different locations of Bangladesh, this data-set is considered as a unique one. There are variations in the intensities of the images as the images are captured in different times of a day having variations in lighting conditions. Second, a database is essential to evaluate the performance of existing methods. Their performance has been evaluated concretely to design actual system. These results play a vital role to assess the efficiency of the extraction and recognition process of Bangla text as there is lack of availability of benchmark data set.

The research works with scene images have some sorts of complexities which are very common such as noise, uneven illumination, complex backgrounds, presence of multilingual texts etc. These are the main difficulties to develop a comprehensive scene text detection and localization method. During text detection, most of the challenges we faced are discussed below.

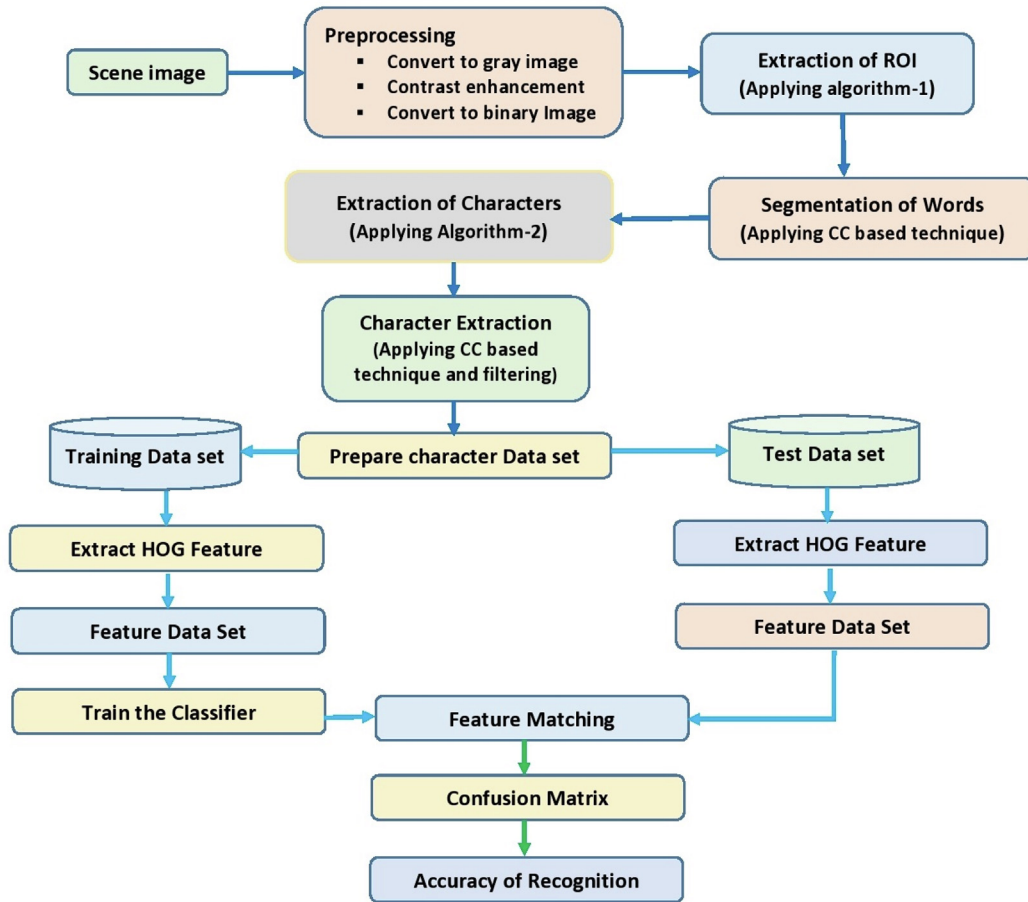


Fig. 3. The proposed method with various phases.

- Varied nature of text in scene image: Text in scene images shows more variation and inconsistency in terms of style, orientation, layout and other factors. Moreover, natural scene images bring more complexity in the task of text detection due to irregular pattern.
- Complex background: Backgrounds of scene images are often very complex and at the same time unpredictable to localize the texts in the images.
- Erroneous detection: In scene images, sometimes objects look like as texts and occlusion may lead to confusion and erroneous detection due to noise and uneven lighting effects.
- Problem in image acquisition: Resolution of the captured image may be low and there may be blurred text in the captured images due to improper capturing, distance of lens from the target, shooting angle etc.

Way of collection of images: To collect the images, mainly we used the camera of android phone. Some of the images were collected from the internet. While capturing the images using camera of mobile phone we maintained the distance of about 1-2 meter from the object. The capturing angle were in between 0 to 30 degree. The images were captured in different times of a day like in the morning, at noon, in the evening and at night. A few images of our database is shown in Fig. 4.

4.1.2. Preprocessing

The preprocessing section is divided into the following two sub-sections. **Transform into Gray image** As the source images are color image, they have to be converted to gray scale image to prepare



Fig. 4. Some of the captured images of our database.

them as usable images for the succeeding stages of the proposed method. The process of transforming a color or RGB image into a gray scale image is expressed by Eq. 1.

$$Gr.I = 0.2999 * r + 0.587 * g + 0.114 * b \quad (1)$$

Where r, g and b represent red, green and blue respectively. **Adjustment of contrast** It may not possible to extract texts from a low contrast image. So, to overcome the problem, special measure has been taken applying the process of contrast limited adaptive histogram equalization (CLAHE) with the value of standard deviation equal to 0.01. In this process, new gray value is set by replacing each gray value in the low contrast image. Fig. 5 shows the effect of enhancement technique on a gray scale image. **Convert to monochrome image** The gray scale image is transformed into a monochrome image by applying Otsu's thresholding method. In this method, the pixels are divided into two classes as foreground and background by using



Fig. 5. Gray image (a) before (b) after applying CLAHE method.

the single intensity threshold which is provided by the algorithm. As shown in (2),

$$\sigma_v^2(t) = v_0(t)\sigma_0^2(t) + v_1(t)\sigma_1^2(t) \quad (2)$$

Here, the probabilities of the two classes are indicated by the weights v_0 and v_1 , and they are detached from each other by a threshold t . σ_0^2 and σ_1^2 are the variances of the two class.

On the basis of the generated threshold value (T), the gray image is converted to monochrome image using Eq. (3).

$$r(x,y) = \begin{cases} 1, & \text{if } g(x,y) \geq T \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Where, $r(x,y)$ and $g(x,y)$ represent threshold and gray scale image pixel at (x, y) respectively.

Here, the pixels are classified as background or 0 if the gray level values of the pixels are below the threshold. In the same way the pixels are classified as foreground or 1 if the gray level values of the pixels are greater than or equal to the threshold value.

4.1.3. Extraction of ROI

To extract ROI from scene images, A new algorithm (Algorithm 1) has been introduced.

Algorithm 1

1. Input a color image, create a Gaussian kernel of default size 3×3

Pseudo-code:

```

READ RGB image 'rgbimg'
SET rgbimg to I;
CALL rgb2gray with I RETURNING G; //G is a gray image
CALL graythresh with G RETURNING gt; // gt is Otsu's
threshold
CALL imbinarize with G and gt RETURNING Ibin; //Ibin is the
Binary image of the color image I
// Direction filters
Assign [-1 -1 -1; 2 2 2; -1 -1 -1] to k_0; // 0 degree Assign [-1 -1
2;-1 2 -1; 2 -1 -1] to k_45; // 45 degree
Assign [-1 2 -1;-1 2 -1;-1 2 -1] to k_90; // 90 degree
Assign [2 -1 -1;-1 2 -1;-1 -1 2] to k_135; // 135 degree
Assign k_0 to K[1]
Assign k_45 to K[2]
Assign k_90 to K[3]
Assign k_135 to K[4]
CALL fspecial with Gaussian RETURNING h // h is a Gaussian
kernel

```

2. Create a Gaussian pyramid Pseudo-code:

```

SET I to img
SET img to PD{1} // PD{} is the Gaussian pyramid

```

Algorithm (continued)

Algorithm 1

```

FOR i = 2 to 4 DO
CALL imfilter with img, h RETURNING img
ENDFOR
3. Detect edges in 0°, 45°, 90° and 135° orientations by
convolving 4 directional kernels.
Pseudo-code:
FOR X=1 to 4
FOR y=1 to
CALL imfilter with pdx, Ky RETURNING Cx,y // Cx,y is the
image convolved by 4 directional kernels.
ENDFOR
ENDFOR
FOR x= 1 to 4
ADD C{1,x}, C{2,x}, C{3,x} and C{4,x} and SET the result to resp
{x} // Response of all directional filters ENDFOR
4. Creation of a feature map
Pseudo-code:
CALL imadd with (resp1+resp3) and (resp2+resp4)
RETURNING Total // Total is the total of all directional
response
SET resp3 to e_strng //e_strng is the strong edge of the input
image
CALL strel with 'line', 3, 0, RETURNING SE //SE is a structuring
element
CALL imdilate with e_strng and SE RETURNING D // D is the
dilated image
CALL strel with 'line', 15, 90 RETURNING SE2 //SE2 is the line
type structuring element
CALL imclose with D and SE2 RETURNING i_close //i_close is
the image after applying morphological closing operation
on D
SUBTRACT D from i_close and SET to wek_edge
ADD e_strng and wek_edge and SET to v_edge/v_edge is
vertical edge obtained by adding strong edge and weak
edge
CALL bwmmorph with v_edge, 'thin' and 'inf' RETURNING
thnd_img/thnd_img is the image obtained after applying
morphological thinning operation on v_edge CALL bwlabel
with thnd_img and 4 RETURNING L and N //L is label no
and N is number of CC
CALL regionprops with L, 'all' RETURNING prop // prop is the
property of the connected regions
CALL strel with 'line', 3, 90 RETURNING SE3
CALL imdilate with thnd_img, SE3 RETURNING cand // cand is
candidate
text regions
CALL immultiply with cand, Total RETURNING R // R is refined
image
CALL imdilate with R, (CALL strel with 'square', 4) RETURNING
R_img //R_img is the image obtained after dilating R by
square shaped structuring element
MULTIPLY R_img and total and SET to F_map //F_map is the
feature map of the input image
5. Dilating the feature map to cluster text regions together
Pseudo-code:
CALL imdilate with F_map, 'square', 4 RETURNING DF //DF is
the dilated feature map.
CALL bwlabel with DF, 4 and RETURNING Lab and Num
Lab is label no
and Num is number of CC
CALL regionprops with Lab, 'all' RETURNING R //R is the

```

Algorithm (continued)

Algorithm 1

```

property of all connected regions
FOR i=1 TO length of R
SET R(i).Area to A
IF A < MaxArea/20 THEN CALL bwareaopen with DF, A
RETURNING DF
ENDIF
ENDFOR
6. Apply coarse level filtering to eliminate nontext regions
Pseudo-code:
FOR i=1 to length R
IF (Major_axis_length of R(i)) (Minor_axis_length of R(i)) > 5
SET 0 to all pixels of R(i)
ENDIF
ENDFOR
FOR i=1 to length R
SET R(i).solidity to Sd //Sd is the solidity of R(i)
IF Sd<0.6
SET 0 to all pixels of R(i)
ENDIF
ENDFOR
7. Calculation of horizontal & vertical projection pixels and extract text region
Pseudo-code:
SET DF to new
CALL sum with new,1 RETURNING S1 //S1 is number of pixel in each column
CALL sum with new,2 RETURNING S2 // S2 is n umber of pixel in each row
CALCULATE Vt=mean(S1,1)+mean(S2)20 // Vt is vertical threshold
FOR l= 1 to LENGTH(S1)
IF S1(i)>TY THEN
SET S1(i)=0
ENDIF
ENDFOR
INIT V_Edge=zeros(size(new,1))
FOR J=1 to size(V_Edge,1)
FOR i=1 to length(S1)
IF S1(i)=0 THEN
V_Edge(j,i)=0
ENDIF
ENDFOR
ENDFOR
CALCULATE Vh=mean(S2)20 //Vh is horizontal threshold
FOR x=1TO LENGTH(S2)
IF S2(x)<Tx THEN
SET S2(x)=0
ENDIF
ENDFOR
INIT h_Edge=zeros(size(new,1))
FOR X=1 to size(H_edge,1)
FOR y=1 to length(S2)
IF S2=0 THEN
SET H_edge(x,y)=0
ELSE
SET H_edge(y,x)=new(y,x)
ENDIF
ENDFOR
ENDFOR
CALL imadd with H_edge,V_edge RETURNING Total_edge
CALL immultiply with NOT lbin, medfilt RETURNING Final
DISPLAY Final // Final is an image with Extracted ROI

```

The details of Algorithm 1 is given below.

- **Step-1:** In this step, a Gaussian kernel of size 3×3 is created by using the following functions as shown in (4) and (5).

$$h_g(c_1, c_2) = e^{-\frac{(c_1^2 + c_2^2)}{2\sigma^2}} \quad (4)$$

$$h(c_1, c_2) = \sum_{c_1} \sum_{c_2} h_g \quad (5)$$

Where c_1 is the number of rows and c_2 is the number of columns, in our case $c_1 = c_2 = 3$, and the value of (standard deviation), $\sigma = 0.5$. We denote this kernel as G_h . Fig. 6 shows the Gaussian filter and respective kernel values with size 3×3 .

- **Step-2:** In this step, Gaussian pyramid is constructed and each images in the Gaussian pyramid are convolved by the Gaussian kernel G_h as mentioned in (6).

$$M_p = G_h \star M_B \quad (6)$$

Where, M_B is the input image and \star is convolution operator. The size of M_p is $(2^n + 1) \times (2^n + 1)$. The images of are the copies of the image in different scales. The input image is M_B and the output images are R_0, R_1, R_2 and R_3 , where the size of R_i is $(2^{n-i} + 1) \times (2^{n-i} + 1)$.

- **Step-3:** In this step, four directional filters are created to detect edges in horizontal, vertical and two diagonal directions. Let $\theta(x, y)$ be a direction at any location (x, y) of an image G . The gradient ∇G can be defined as shown in (7).

$$\nabla G = \text{grad}[G(x, y)] \equiv \begin{bmatrix} g_x(x, y) \\ g_y(x, y) \end{bmatrix} = \begin{bmatrix} \frac{\partial G(x, y)}{\partial x} \\ \frac{\partial G(x, y)}{\partial y} \end{bmatrix} \quad (7)$$

The direction of the gradient vector at the point (x, y) can be defined as expressed in (8).

$$\theta(x, y) = \tan^{-1} \left[\frac{G_y(x, y)}{G_x(x, y)} \right] \quad (8)$$

Angles are measured in the counter clockwise direction based on x-axis. Fig. 7 illustrates the orientation of an edge at point (x, y) . Fig. 7(a) shows a small part of an image which is being zoomed that contains segment of an edge. Each square represents a pixel and the value of each shaded pixel is 1 and that of each white pixel is 0. From the figure it is observed that at point (x, y) , the direction of an edge and gradient vector are perpendicular to each other. Fig. 7(b) illustrates the process of calculating derivatives in x and y directions.

Let the directional kernels are represented as W_0, W_1, W_2 and W_3 , which are created by edge direction $0^\circ, 45^\circ, 90^\circ$, and 135° respectively.

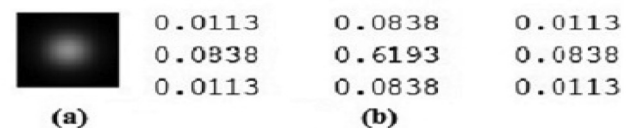


Fig. 6. (a) Gaussian filter (b) Corresponding kernel values.

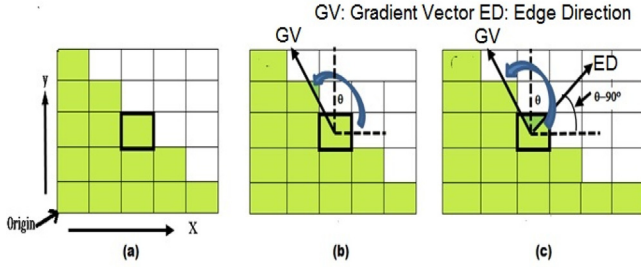


Fig. 7. Determine strength and direction of an edge at a point using the gradient.

- **Step-4:** This step corresponds to convolve each image of the pyramid by each direction kernel. By convolving each of the images of the pyramid with each direction filter will yield 16 images as follows: $F_k = W_i \star R_j$, where $1 \leq k \leq i \times j$, $1 \leq i \leq 4$, for each value of i , $1 \leq j \leq 4$.
- **Step-5:** A feature map of the input image is created by performing a series of morphological operation as stated below.

1. All the 16 images stored in F_k are re-sized to original image size as follows:
 $F_{k2} = \text{resize}(F_k(i,j) \text{ as } (2^n + 1) \times (2^n + 1))$, where $1 \leq i \leq 4$, for each value of i , $1 \leq j \leq 4$.
2. for $1 \leq m \leq 4$ compute

$$t(m) = \sum_{i=1}^4 F_{k2}(i, m) \quad (9)$$

$$T = \sum_{i=1}^4 t(i) \quad (10)$$

3. Strong edges are created by the way as shown in (11)

$$E_{strong} = t(3) \quad (11)$$

4. Dilate E_{strong} by a structuring element B to get the dilated image as shown in (12)

$$E_d = E_{strong} \oplus B = \bigcup_{e \in E_{strong}} B_e \quad (12)$$

Where, the size of the structuring element B is 1×3 . The union is used as neighborhood operator. The images of Fig. 8(a) and 8(b) represent the original image and structuring element. Fig. 8(c) shows the effect of dilation operation.

5. Closing of the image obtained from (12) by structuring element B_1 is performed by the way as shown in (13).

$$E_{close} = E_d \bullet B_1 = (E_d \oplus B_1) \ominus B_1 \quad (13)$$

Here, the symbols \bullet , \oplus and \ominus indicate morphological closing, dilation and erosion operation respectively. B_1 is the line type

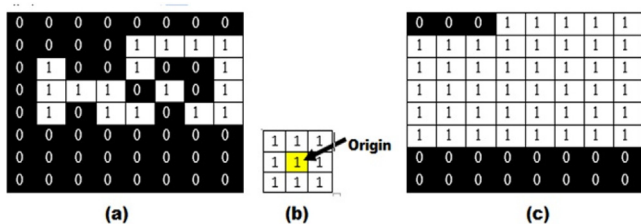


Fig. 8. (a) Original image (b) Structuring element (c) Effect of dilation.

structuring element with angle 90° . Closing of a binary image is performed by dilating the image followed by erosion it by the same structuring element.

6. An image containing vertical (90°) edges are constructed by the way as shown in (14) and (15)

$$E_{weak} = E_{close} - E_d \quad (14)$$

$$E_{90} = E_{strong} + E_{weak} \quad (15)$$

7. Skeleton of the Image created by (15) is constructed by applying thinning operation on the image. In mathematical morphology, thinning of a set S by structuring element R, can be defined with respect to hit-or-miss transform as expressed in (16).

$$\begin{aligned} \text{Thin}(S, R) &= S - (S \otimes R) \\ &= S \cap (S \otimes R)^c \end{aligned} \quad (16)$$

Where, the symbol \otimes indicates hit-or-miss transformation and the superscript notation c indicates complement. A symmetric thinning can be expressed by a sequence of structuring element: $\{R\} = \{R_1, R_2, R_3, \dots, R_n\}$

Where R_i is the rotated form of R_{i-1} . Using the above concept, the thinning can be defined by a series of structuring element as shown in (17)

$$S \otimes \{R\} = ((\dots((S \otimes R_1) \otimes R_2) \dots) \otimes R_n) \quad (17)$$

According to (17), the image E_{90} has been thinned by a structuring element S is expressed by the way as shown in (18).

$$\begin{aligned} E_{thin} &= E_{90} \otimes \{S\} \\ &= ((\dots((E_{90} \otimes S_1) \otimes S_2) \dots) \otimes S_n) \end{aligned} \quad (18)$$

Here, size of S is 3×3 and $n=8$.

8. Dilate E_{thin} by line type structuring element B_2 of size 10 with angle 90° to get the candidate regions as shown in (19).

$$E_{candidate} = E_{thin} \oplus B_2 \quad (19)$$

9. The process of obtaining refined image is shown in (20) and (21).

$$E_{refined} = E_{candidate} * T \quad (20)$$

$$E_{ref} = E_{refined} \oplus B_3 \quad (21)$$

Here, B_3 is a square shaped structuring element of size 5×5 .

10. A feature map is created by the following procedure as shown in (22)–(25).

$$\text{Target1} = t(1) \wedge t(3) \quad (22)$$

Here, the symbol \wedge indicates logical AND operator.

$$\text{Target2} = t(2) \wedge t(4) \quad (23)$$

$$FF = \text{Target1} + \text{Target2} \quad (24)$$

$$F_{map} = E_{ref} \wedge FF \quad (25)$$

- **Step-6:** In this step, the resultant image F_{map} is dilated to bring the expected text regions close to each other. All the characters present in a scene image appear very near to each other. So, a group of text pixel is formed here and a dilation operation helps to cluster all the text pixels together. It also eliminate the pixels those are not close to the candidate text region. The dilation operation enhance the ROI using fixed size structuring element. In this algorithm, a 10×10 size structuring element is used to enhance the probable text regions.

- **Step-7:** This step corresponds to removing non-text regions. Use of coarse level filtering helps to remove the non-text areas. Some common objects like doors, windows, fences, bricks, leaves etc. which look like text may appear in a scene image. These objects may be the cause of degradation of efficiency of the algorithm as they are treated as false positives. So, to eliminate these objects from the image, coarse level filtering has been used which is based on morphological property like area, aspect ratio and solidity. For coarse level filtering, three vectors for three parameters are taken such as V_A is a vector of areas, V_R is a vector of all aspect ratios, and V_S is a vector of solidities of all candidates regions. Let there are k candidate regions, so the vectors can be defined as follows:

$V_A = (A_1, A_2, A_3, \dots, A_k)$ where A_i is the area of the i^{th} region.

$V_R = (R_1, R_2, R_3, \dots, R_k)$ where $R_i = \frac{LM_i}{LN_i}$, LM_i is length of the major axis and LN_i is length of the minor axis of the i^{th} region.

$V_S = (S_1, S_2, S_3, \dots, S_k)$, if H_i be a region of convex hull that covered the region then $S_i = \frac{A_i}{H_i}$ is called the solidity of i^{th} region.

Next we set optimum value of each parameter as, $opt_A = \frac{maxA}{20}$, $opt_R > 5$, and $opt_S < 0.25$, where $max_A = \max(V_A)$ is the maximum value in the vector, V_A . These optimum values are selected by performing experiment on different types of scene images from our image database. Next, the filtering is done as follows. For $1 \leq i \leq k$ if $(A_i < opt_A \text{ or } R_i > opt_R \text{ or } S_i < opt_S)$ then A_i is removed from V_A . Let the output of this step is C_F .

- **Step 8:** In this step, horizontal and vertical thresholds are used to eliminate the non-text regions.

Let $P_{i1}, P_{i2}, P_{i3}, \dots, P_{im}$ pixel values of i^{th} row for the input image obtained from step 7 of size $m \times n$ and HP_i and VP_j represent the sums of the values (1s) of all white pixels in the i^{th} row and j^{th} column respectively. Now compute

1. For $1 \leq i \leq m$ compute $HP_i = \sum_{j=1}^n P_{ij}$ if $P_{ij} = 1$ (sum of all white pixels for each row).

2. For $1 \leq j \leq n$ compute $VP_j = \sum_{i=1}^m P_{ij}$ if $P_{ij} = 1$ (sum of all white pixels for each column).

Let, $HP = (HP_1, HP_2, HP_3, \dots, HP_m)$ and

$VP = (VP_1, VP_2, VP_3, \dots, VP_n)$

Where HP and VP denote the horizontal and vertical project profiles respectively.

3. Compute $T_h = \frac{mean(HP)}{20}$ and

$T_v = \frac{mean(VP)}{10}$

4. To eliminate possible non-text regions from the output binary image (C_F) of step 7, the adaptive threshold value T_h and T_v are calculated for horizontal and vertical projections respectively. The value of T_h and T_v are selected to eliminate regions with long edges in the horizontal and vertical directions respectively. The operations are shown below. If $(HP_i < T_h)$ set $HP_i = 0$, for $1 \leq i \leq m$. If $(VP_j < T_v)$ set $VP_j = 0$, for $1 \leq j \leq n$. Let the output of this step is H_V .

- **Step 9:** In this step the output image of step-8 (H_V) is multiplied with the binary image (B_M) to have only the extracted characters in the image as shown in (26).

$$T_R = H_V \times B_M \quad (26)$$

Here, the symbol \times represent multiplication.

The resultant output image from the algorithm is shown in Fig. 9 along with the input image.



Fig. 9. (a) Input color image (b) Extracted ROI.

Table 1

Performance measure of ROI Extraction.

Image Type	No. of Images	Pr (%)	RR (%)	F1-score (%)	Accuracy (%)
Banner	270	95.24	98.24	96.32	93.70
Poster	149	94.94	97.38	95.69	92.59
Signboard	39	93.52	97.46	95.12	91.53
License Plate	42	90.16	97.07	93.38	88.88

4.1.4. Localization of Words

Two or more Bangla characters connected by 'matra' or headline make a Bangla word. In order to segment out the words from the image, bounding box technology and CC analysis is applied to select all the CCs. Here, Bangla words are regarded as CCs. To label the CCs of a binary image, following steps have been followed.

1. The process of labeling starts from the first pixel P of the image. Set $L=1$ as current label.
2. If P is a foreground pixel and it is not already labeled, give it the current label and add it as the first element in a queue, then go to step-3. If it is a background pixel or it was already labelled, repeat step-2 for the next pixel in the image.
3. Pop out an element from the queue, and look at its neighbors. If a neighbor is a foreground pixel and is not already labeled, give it the current label and add it to the queue. Repeat step-3 until there are no more elements in the queue.
4. Go to step-2 for the next pixel in the image and increment current label by 1.

Here, 4 connectivity is used in the proposed method to label all the CCs. By applying 4 connectivity along with bounding box technology, all the connected regions have been marked by yellow color bounding box as shown in Fig. 10 (c).

4.1.5. Character extraction

Since the characters in Bangla text are linked by a upper-line called 'matra', it is not so easy to separate characters from a word. As a result, it has become a challenging task for the researchers to retrieve characters from scene images. The removal of the headline

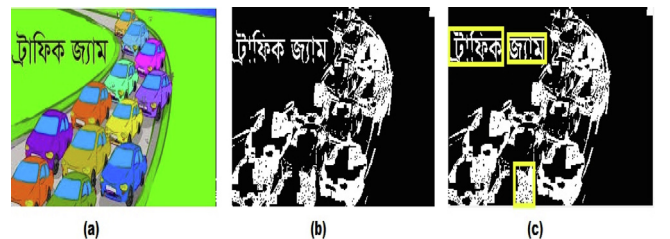


Fig. 10. (a) Input color image (b) Monochrome image (c) Identified Bangla words.

Table 1 shows the result of extraction of ROI from different categories of scene images with different types of performance measure metrics like precision (Pr), Recall Rate (RR), F1-score and Accuracy.

or 'matra' is followed by the existing approaches to apart the characters from the words. But it is a matter of fact that once the 'matra' is removed from a word, meaning of some character will be changed. Examples of some Bangla characters are shown in Fig. 11.

Considering the above problem, following algorithm (Algorithm 2) is proposed to apart the characters from the Bangla words without removing the 'matra'.

Algorithm 2

1. Separate all the characters from each of the bangla words. Pseudo-code:

```
CALL bwlabel with Final, RETURNING L and N // L is Label
number and N is number of connected components
CALL regionprops with L, RETURNING prop //prop is the
property of all the connected regions
SET n1_index=1, count=0
INIT xy(1)=n1_index
FOR n=1 to size(prop,1)
IF prop(n).solidity<0.8
IF 300>=prop(n).area<=2000
SET n1_index=n
INCREMENT count by 1
CALL rectangle with (position, prop(n).BoundingBox,'EdgeCo
lor','y','LineWidth',3)
SET xy(count)=n1_index
ENDIF
ENDFOR
FOR i=1 to COUNT
SET M to 1000, p=0, total=0
CALL find with (L=xy(i)) RETURNING row and col
SET final(min(row) to max(row) and min(col) to max(col)) to
n1
ASSIGN size(n1) to r1 and c1 //r1 is row and c1 is column
CALL sum with n1,1 RETURNING colsum // colsum is the
summation pixels of all the columns
FOR i=1 to length(colsum)
IF (colsum(i) < M
Assign colsum(i) to M
ENDIF
ENDFOR
FOR i=2 to length(colsum)-2
IF(colsum(i) < =M+5
FOR j=1 to size(n1,1)
SET n1(j,i)=0
ENDFOR
ENDIF
ENDFOR
```

2. Extract Bangla characters individually

Pseudo-code:

```
CALL bwlabel with n1 RETURNING Lab, N
CALL regionprops with Lab, 'BoundingBox' RETURNING prop
FOR q=1 to N
IF Lab=q
SET r1=rownumber, c1=column number
INIT n2=n1(min(r1) to max(r1), min(c1) to max(c1)) //n1 is
the extracted Bangla words
INIT h=height of n2 and w=width of n2
IF w>h THEN
CALL resize with n2, [32,32] RETURNING n2 // n2 is the
extracted characters
WRITE n2 to disk
ENDIF
ENDIF
ENDFOR
```

Original Character	After removing the 'matra'	Looks like the following character
হা	হ	হ (Ha)
ষ	ষ	ষ (Six)
ঢা	ঢ	ঢ (Dha)
ত	ত	ত (Three)

Fig. 11. Effects of removing matra.

In this algorithm, we have to input all the CCs of the Binary image. Let V is a binary image and such that $V(b, d) = V(b', d') = p$, where $p = \{0, 1\}$. The pixels (b, d) and (b', d') are connected with regard to value p if the pixel sequence $(b, d) = (b_0, d_0), (b_1, d_1), \dots, (b_n, d_n) = (b', d')$ in which $V(b_i, d_i) = p$; $i = 0, 1, \dots, n$, and (b_i, d_i) are the neighbors of (b_{i-1}, d_{i-1}) for each $i = 1, 2, \dots, n$. The sequence of pixels $(b_0, d_0), (b_1, d_1), \dots, (b_n, d_n)$ creates a connected path from (b, d) to (b', d') . A pixel set C is a connected component of value p where each pixel has a value p in such a way that each pixel pair in the set is connected with regard to p .

Detail description of steps 1 to 3 of Algorithm 2 is given below.

Determine minimum value: In order to determine the minimum value, vertical scanning of the binary image $C1$ is performed. Next, compute the number of 1 or white pixels in each column and find a minimum value min among all the columns of the image.

Separate the characters: In a binary image of a Bangla word with white foreground and black background, there must be a horizontal line or 'matra' that connects almost all the characters. There is a slide gap between two characters. If number of 1 is counted using



Fig. 12. Separating zones of Bangla words.

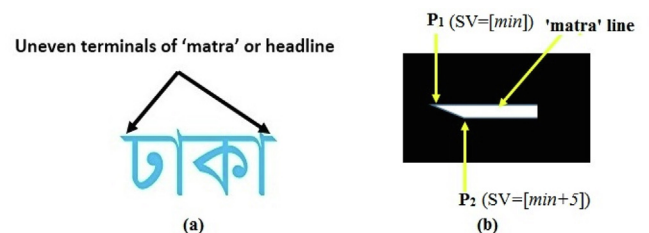


Fig. 13. (a) Two ends of a 'matra' line (b) Calculation of SV.

vertical scanning process, summation will be minimum in these locations. Because, in these locations minimum number of 1 will be obtained along the thickness of the 'matra' line. So, these locations can be treated as the separating zone. In the rest of the columns, the summation of white pixels are more than the min . Fig. 12 shows the separating zones of Bangla words. (See Fig. 13)

To separate each of the characters from the binary image, following conditions have been set. for any column if the value of $SV=[min, min+5]$, the entire column value will be set to zero. Here, the optimal settings of the parameter value SV is used. The reason of selecting the maximum value of SV as $min+5$ is, in some Bangla words two ends of a 'matra' line are not flat. So, in the very first column of 'matra' line there is only 1 pixel and gradually it increases in the next successive columns. According to Fig. 13(b), at point P_1 , the value of $SV=[min]$ and it will increase up to the point P_2 . But from the point P_2 of the 'matra' line the value of $SV=[min+5]$ at the separating zones of Bangla words.

So, in the first column, value of min will be 1 but this is not the actual thickness of the 'matra'. So, to work with the full thickness of 'matra' to find out the separating zones, the value of $SV= min +5$ is set to successfully separate all the characters from the words. Fig. 14 shows the effect of using the parameter values and from this figure it can be noticed that the optimal parameter setting of SV is $SV=[min, min+5]$, which helps to separate all the characters from the words successfully. Each stage of the proposed system is shown in Fig. 15.

4.2. Character Recognition

Character recognition is the last phase of this research work. In this phase, two data set named as training and testing are prepared

from the main database. Then the classifier is trained with the training data set. Finally accuracy of recognition is calculated. So, it is evident that the recognition phase is composed of the training and the test phases. The following subsections illustrate these two phases.

4.2.1. Training Phase

This phase is important for any recognition scheme. Because if the classifier is properly trained by the extracted features of the input image, will help to increase the accuracy of recognition. Different steps of character recognition are stated below.

1. **Preparing Training and Test Dataset:** Training dataset is prepared as an essential object of the training phase of character recognition system. The designed method works with specific number of images those have been selected randomly from 62 sets of Bangla characters. for each of the Bangla character, there are 500 sample images have been extracted from the scene images. among 500 images, 70% or 350 is selected as number of training samples for each Character. So, a total of 21,700 (350×62) images have been selected to built up the training data set. At the same time 30% or 150 images for each character is selected for test data and a total of 9300 (150×62) images have been considered as test data set.
2. **Feature Extraction:** In this step, feature sets are extracted from the input image. Extracted Features plays a vital role to measure performance of any character recognition system. In the proposed method, HOG feature is extracted from the Bangla characters. In this feature, the Sobel mask is used to calculate H and H^T as horizontal and vertical component respectively of the gradient.








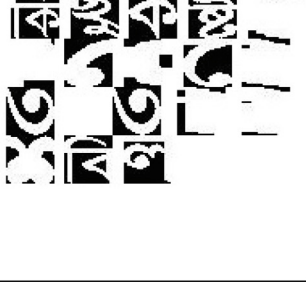
Text Localization	Extracted Characters		
	Sum<=minimum+1	Sum<=minimum+3	Sum<=minimum+5
			
Results	Precision:25% Recall: 100% F-Score: 40% Accuracy: 8.33%	Precision: 100% Recall: 88.95% F-Score: 94% Accuracy: 100%	Precision:100% Recall: 100% F-Score: 100% Accuracy: 100%
			
Results	Precision:90.91% Recall: 100% F-Score: 95% Accuracy: 90.91%	Precision: 90% Recall: 90% F-Score: 90% Accuracy: 90%	Precision:100% Recall: 91.67% F-Score: 95.65% Accuracy: 100%

Fig. 14. Effects of different parameter values.

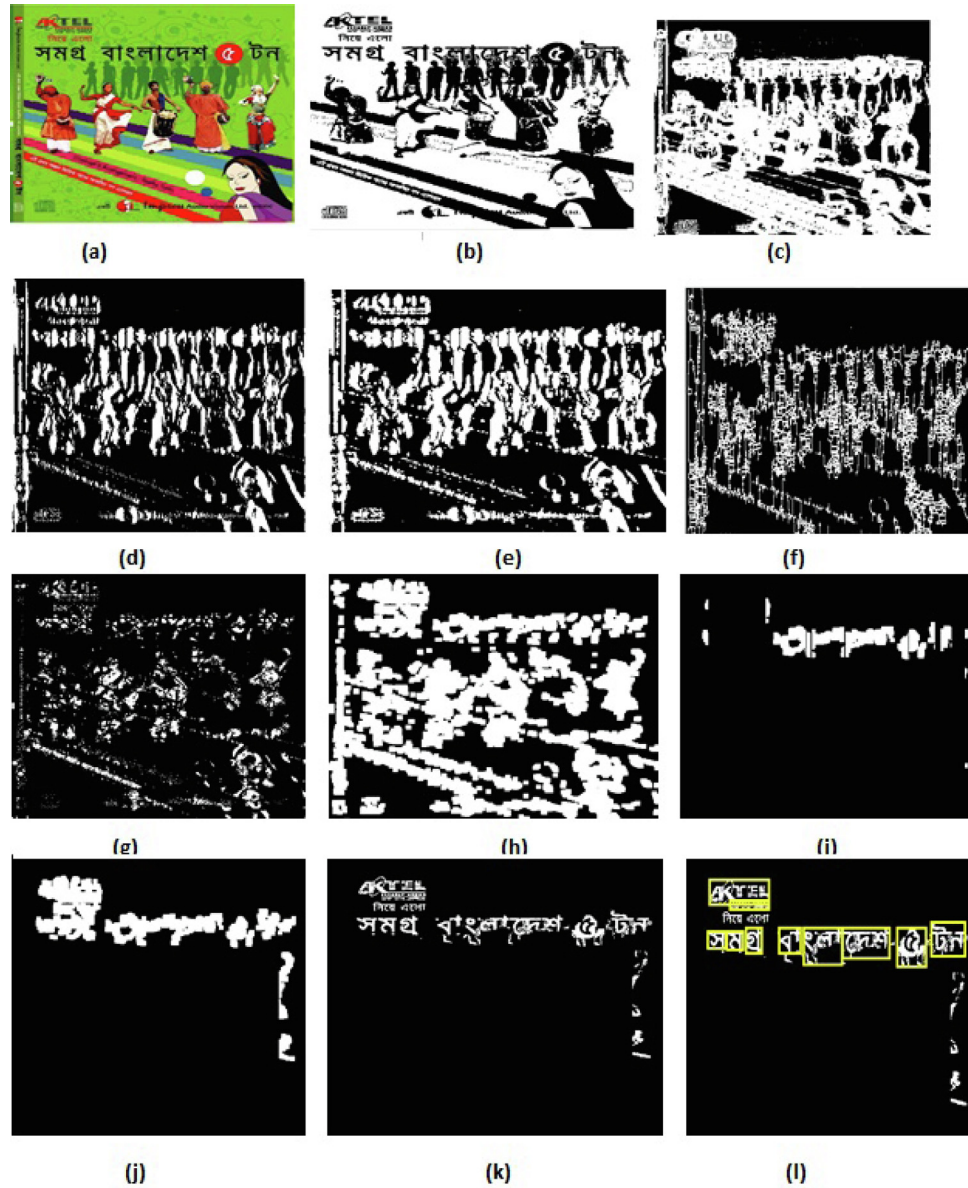


Fig. 15. Pictorial view of different stages of the proposed method. (a) Original image (b) Binary Image (c) Response of all direction filters (d) Strong edge (e) Dilated strong edge (f) Candidate edge (g) Feature map (h) Dilated feature map (i) Vertical projection pixel (j) Horizontal projection pixel (k) Extracted ROI (l) Text localization.

Let, the source image is I . The horizontal and vertical derivative approximation at each point is contained by the two images V_x and V_y respectively. The computations can be shown in (27) and (28).

$$V_x(x, y) = (H * I)(x, y) \quad (27)$$

$$V_y(x, y) = H^T * I(x, y) \quad (28)$$

The technique to calculate the gradient is shown in (29).

$$V(x, y) = \sqrt{V_x^2(x, y) + V_y^2(x, y)} \quad (29)$$

Eq. (30) shows the process of calculation of the angle of gradient.

$$\alpha(x, y) = \tan^{-1} \frac{V_x(x, y)}{V_y(x, y)} \quad (30)$$

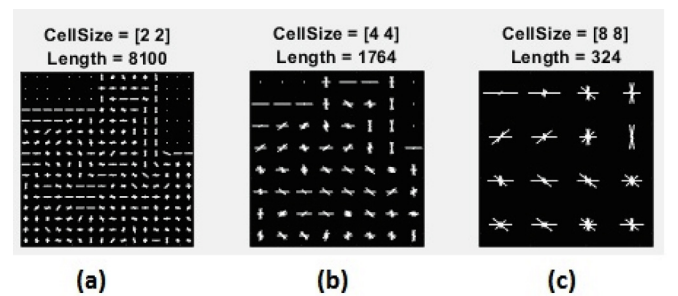


Fig. 16. Various sizes of blocks in the HOG features of a Bangla character (a) 2×2 (b) 4×4 (c) 8×8 .

The process of calculation of HOG features is shown in (31).

$$\delta = \begin{cases} V(x, y), & \text{if } \alpha(x, y) \in \text{bin}_k \\ 0, & \text{otherwise} \end{cases} \quad (31)$$

Here, the bins of the histogram for all the blocks are mentioned by bin_k . The extracted HOG feature of a Bangla character is shown in Fig. 16.

4.2.2. Test Phase

In test phase, features are generated from each of the test images of the test data set. The obtained features are used to make a class prediction on the basis of the trained classifier. Finally, the accuracy of recognition is computed by comparing the actual labels and the predicted labels of test data set.

5. Analysis of Results

The main two phases of the proposed method on which the experiment was performed are:

- Extraction of characters and
- Recognition of extracted characters.

There are 500 scene images of banners, posters, billboards and license plates in our developed database. These images contain Bangla texts in different colors, styles, sizes, and orientations. All the experiments are done in the environment which is based on MATLAB and these are stated below.

5.1. Character Extraction

The results of character extraction are analyzed by the four metrics like *precision*, *recall*, *F1-score*, and *accuracy* based on the parameters TP, TN, FP and FN. The definitions of TP, TN, FP and FN are same as given by Islam et al., 2020.

5.2. a) Precision (Pr):

The process of calculation of precision is shown in (32).

$$Pr = \frac{TP}{TP + FP} \quad (32)$$

5.3. b) Recall Rate (RR):

The process of calculation of recall rate is presented in (33).

$$RR = \frac{TP}{TP + FN} \quad (33)$$

5.4. c) Accuracy:

Eq. (34) shows the process of calculation of accuracy.

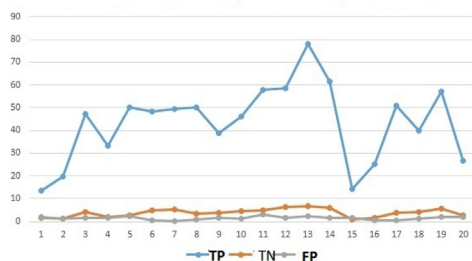


Fig. 17. Relationship among TP, TN, and FP.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (34)$$

The experiment was performed on 500 scene images and the values of TP, FP, and TN have been obtained for each of the images. These 500 values are divided into 20 groups and for each group, their average values have been taken to draw the line graph which is shown in Fig. 17.

5.5. d) F1-score:

F1-Score is calculated on the basis of the values of *Pr* and *RR* as shown in (35).

$$F1 - score = \frac{2 * Pr * RR}{Pr + RR} \quad (35)$$

Accuracy and *F1-score* are the two important metrics used to measure the performance of any classifier. But for the uneven class distribution, *F1-score* better than the *accuracy*. When there is a little difference between the values of false positives and false negatives, *accuracy* works well. But both the *Precision* and *Recall* are taken into account when there is more difference between the values of false positives and false negatives. The relationship between *Pr*, *RR*, and *F1-score* is shown in Fig. 18.

In the proposed method Bangla characters have been extracted from different types of scene images and the *accuracy* of extraction along with *Pr*, *RR* and *F1-score* is shown in Table 2.

5.6. Calculation of the accuracy of recognition

The efficiency of this research work depends on this step as this is the target step. Considering this into account, the proposed method must be compared with the existing methods. So, the accuracy of character recognition has been calculated by using CNN method.

5.6.1. Character recognition using CNN

Among different kinds of machine learning algorithms, CNN is one of the most popular one. In CNN method, the task of classification is performed by a model which can extract features directly from images, video, text, or sound. So, CNNs are useful to classify images and it does not require manual feature extraction. It is known to all that many hidden layers are involved in designing a typical CNN. These layer and their respective parameter values along with training options those have been used to design the network are discussed below.

- Input Layer:** The size of the input image must be specified in this layer. For this experiment, each of the extracted Bangla characters is fixed by a size of 32-by-32-by-1. Here, 32 indicates the height and width of the image and 1 indicates the channel size for binary image.

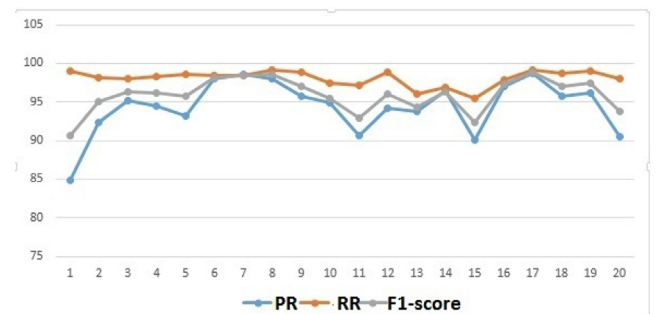


Fig. 18. Line graph showing Pr, RR, and F1-score.

Table 2

Results of character extraction.

Image type	No. of images	Pr (%)	RR (%)	F1-score (%)	Accuracy (%)
Banner	270	93.92	98.39	95.77	93.27
Poster	149	95.61	97.88	96.42	94.23
Signboard	39	93.20	97.88	95.26	92.28
License Plate	42	92.52	96.79	94.39	90.32

2. **Convolutional Layer:** This layer works with the three parameters such as filter size, number of filter and 'padding' name value pair determined by number of filter. The neurons are connected to the same region of the input. Padding is added to the input feature map by the parameter 'Padding'. In the experiment the following hyper-parameters have been used for the function convlayer (3, 16, 'Padding', 1).

Here, Filter size = [3 × 3], number of filters = 16, and Padding size = [1 1 1 1].

3. **Batch Normalization Layer:** The speed of the training network is enhanced by this layer. CNN is sensitive to initialization and the sensitivity is reduced by the Batch Normalization Layer.

4. **ReLU Layer:** The Relu is the abbreviated form of the rectified linear unit. The piece-wise linear function is Relu which transforms the input directly to output if the input is positive, otherwise, output will be zero and it is shown in Eq. (36).

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x < 0 \end{cases} \quad (36)$$

5. **Pooling Layer:** The presence of features in different regions of feature map is summarized by the pooling layer. In this context, two types of pooling like maximum and average pooling are used. To get maximum value from a patch of a feature map, the max pooling layer is used. Following hyper-parameters are used for this layer.

Ps=[2 × 2], Std=[2 × 2], Padmode='manual', Padsz=[0 0 0 0]

6. **Fully Connected Layer:** In this layer, the features are combined to classify the images. In our designed CNN architecture, 62 has been set as the parameter value of the fully connected layer for the classification of 62 Bangla characters (digits, basic characters and compound characters).

The above mentioned parameters are set to train the CNN with the training data set. Then a class prediction is made by classifying the test data set with the trained network. Finally, the predicted labels and actual labels of the test data set are compared to calculate the accuracy of recognition.

The obtained accuracy from the CNN is 83.52%.

5.6.2. Recognition of Character

The following three consecutive steps have been followed to compute the accuracy of recognition using the proposed method.

5.7. (a) Training

In this step, the training dataset of Bangla digits, most of the Bangla basic characters and some joined letters are used. Then the SVM classifier is trained by the extracted feature sets. The binary classes ($k = 2$) is separated by the SVM with a maximized margin criterion (Yao et al., 2012). All the real world problems may not be confined with only two classes rather it may require to deal with more than two classes. Here comes the concept of multi class SVM which has also been used here.

Digits	0	1	2	3	4	5	6	7	8	9
0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00

Fig. 19. Results of recognition of Bangla digits.

5.8. (b) Testing

Features are extracted from the test data set which are used to make a class of prediction using the trained SVM classifier and finally predicted labels of all the characters of the test data set are obtained. To calculate recognition accuracy a confusion matrix is formed by the predicted labels and original labels of the test data set. Fig. 19 shows such a confusion matrix.

To get the confusion matrix, 70% digits for training and 30% digits for testing are used, i.e. among 5000 digits, 3500 digits were taken for training and 1500 digits were taken for testing and on the basis of Fig. 19, 100% accuracy in recognition of Bangla digits is achieved.

To calculate the recognition of Bangla digits, basic characters and some joined letters altogether, training is performed on 70%, 60%, and 50% characters. The obtained accuracy for the above three categories of characters are presented in Table 3.

5.8.1. Comparison of text extraction and recognition accuracy

In Table 4, for comparisons of text extraction, the methods that are only related to text extraction are considered. The work of (Moyeen et al., 2013 and Ghoshal et al., 2017) are related to both

Table 3

Comparison of the accuracy of text extraction with the existing methods.

No. of Char.	No. of Images for each char.	Total No. of images	% of Tr. img.	Accuracy of Recog.(%)
62	500	31000	70	99.44
			60	99.07
			50	98.99

Table 4

Comparison of the accuracy of text extraction with the existing methods.

Method	No. of Images	Pr(%)	RR(%)	F1-score (%)
Proposed	500	94.25	98.06	95.80
Moyeen et al., 2013	400	100.00	80.50	89.19
Bhattacharya et al., 2009	100	68.8	71.20	69.97
Ghoshal et al., 2011	100	68.15	69.96	69.04
Ghoshal et al., 2017	250	70.70	73.30	71.98
Gillavata et al., 2003	326	83.90	88.70	86.23

Table 5

Recognition accuracy of the proposed method and existing methods.

Methods	No. of images	No. of Characters	Accuracy of Recognition(%)
Proposed	500	31000	99.44
CNN	500	31000	83.52
Moyeen et al., 2013	400	2672*	73.25
Ghoshal et al., 2011	100	7500	94.47
Ghoshal et al., 2017	250	7100	92.00

*No. of words



Fig. 20. The Images where the proposed algorithm can not work properly.

text extraction and character recognition. So, in Table 5 for comparison of recognition, the work of (Moyeen et al., 2013; Ghoshal et al., 2017) and the other different methods that are related to character recognition have been taken. From Tables 4 and 5 it is apparent that the proposed method is the best one in comparison with other method in case of recognition and extraction of Bangla characters.

There are two specific cases where the algorithm will fail or may not work properly. First one is concerned with character extraction. In this case, if the right side of a character is connected with the left side of its succeeding character they will not be extracted individually by the algorithm. In the other case, the algorithm will fail to extract characters from round or curved shaped word. But there are very few images of such type of Bangla text. Fig. 20 shows some of the images of such type of texts.

6. Conclusions

The task of Bangla character extraction and that of recognition is accomplished successfully by the proposed method. The developed database of Bangla character has been used in this research work. Observing the experimental results, it is clearly seen that the proposed method is better than the existing related methods. In spite of variations in the scene images like size, background, contrast, illumination, camera angle etc., the proposed approach works well to detect and extract Bangla characters. A new algorithm is proposed and applied to apart the characters of a Bangla word from each other. The obtained accuracy of character extraction is 93.23%. In case of character recognition, multiclass SVM is used taking the HOG features as input. The accuracy of recognition is calculated by implementing and using the CNN. The obtained accuracy of recognition from the CNN is 83.52%. The achieved recognition accuracy of the proposed method on an average for 70%, 60% and 50% training images of all categories of characters is 99.16%. This result is better than other methods. The performance of the proposed method is measured using *F1-score*. In the case of similar distribution of classes, the *accuracy* can be used. But *F1-score* is used as a better metric for the case where the class distribution is imbalanced. So, the proposed method is evaluated observing the obtained values of *F1-score*. We have a plan to increase the number of scene images in our database which in turns will exaggerate the database by increasing the number of characters. This approach can be tested as language independent method and enhanced it to resolve the encountered problems.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Bouakkaz, M., Quinten, Y., Loudcher, S., Fournie, P., 2018. Efficiently mining frequent itemsets applied for textual aggregation. *Appl. Intelligence* 48 (4), 1013–1019.
Bhattacharya, U., Parui, S.K., Mondal, S., 2009. Devanagari and bangla text extraction from natural scene images. *Proc. Int. Conf. Docu. Anal. Recognit.*, 26–29

Busta, M., Neumann, L., Matas, J., 2017. Deep text spotter: an end-to-end trainable scene text localization and recognition framework. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2204–2212.
Chen, X., Yuille, A., 2004. Detecting and reading the text in natural scenes. In: *Proceedings of IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 366–373.
Cui, Y., Huang, Q., 1997. Character extraction of license plates from video. In: *Proceedings of IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Juan, Puerto Rico, pp. 502–507.
Epshtein, B., Ofek, E., Wexler, Y., 2010. Detecting text in natural scenes with stroke width transform. In: *Proceedings of IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 2963–2970.
Dey, S., Shivakumara, P., Raghunandan, K.S., et al., 2017. Script Independent Approach for Multi-Oriented Text Detection in Scene Image. *Neurocomputing*. <https://doi.org/10.1016/j.neucom.2017.02.061>.
Fabrizio, J., Seidowsky, R., et al., 2016. Text catcher: a method to detect curved and challenging text in natural scenes. *Int. J. Docu. Anal. Recognit. (IJRAR)* 19 (2), 99–117.
Francis, L.M., and Sreenath, N., 2017. TEDLESS - Text detection using least-square SVM from natural scene. *Journal of King Saud University - Computer and Information Sciences*, Vol. 32, Issue 3, pp. 287–299, ISSN 1319–1578, doi: 10.1016/j.jksuci.2017.09.001. (<https://www.sciencedirect.com/science/article/pii/S131915781730126X>).
Francis, L.M., Sreenath, N., 2019. Robust scene text recognition: Using manifold regularized Twin-Support Vector Machine. *Journal of King Saud University - Computer and Information Sciences*, ISSN 1319–1578, <https://doi.org/10.1016/j.jksuci.2019.01.013>. (<https://www.sciencedirect.com/science/article/pii/S1319157818309509>).
Gao, J., Wang, Q., Yuan, Y., 2019. Convolutional regression network for multi-oriented text detection. *IEEE Access* 7, 96424–96433.
Ghanei, S., Faez, K., 2017. A robust approach for scene text localization using rule-based confidence map and grouping. In: *Proceedings of J. Pattern Recog. Artif. Intell.*, vol. 31, no. 2, pp. 1753002(31 pages).
Ghoshal, R., Roy, A., et al., 2011. Headline based text extraction from outdoor images. In: *Proceedings of Int. Conf. Pattern Recognit. Mach. Intell., LNCS*, vol. 6744, pp. 446–451.
Ghoshal, R., Roy, A., Dhara, B.C., Parui, S.K., 2017. Recognition of Bangla text from outdoor images using decision tree model. *Proceedings of J. Know.-based Intell. Engg. Sys.* 21 (1), 29–38.
Gilavata, J., Ewerth, R., Freisleben, B., 2003. A Robust Algorithm for Text Detection in Images. In: *Proceedings of 3rd Int. Sympos. Image Signal Process. Anal.*, pp. 611–616.
He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., Li, X., 2017. Single shot text detector with regional attention. In: *IEEE international conference on computer vision*, pp. 3047–3055.
He, W., Zhang, X.Y., Yin, F., Luo, Z., Ogier, J.M., Liu, C.L., 2020. Real time multiscale scene text detection with scale-based region proposal network. *Pattern Recognit* 98, 107026.
Islam, R., Islam, M.R., Talukder, K.H., 2016. An Approach To Extract Text Regions From Scene Image. In: *IEEE International Conference on Computing, Analytics and Security Trends (CAST)*, College of Engineering Pune Dec 19–21, India, pp. 138–143.
Islam, R., Islam, M.R., Talukder, K.H., 2017. Rule Based Filtering Approach for Detection and Localization of Bangla Text from Scene Images. In: *IEEE International Conference on Recent Trends in Electrical, Electronics and Computing Technologies (ICRTEECT)*, pp. 139–144. <https://doi.org/10.1109/ICRTEECT.2017.16>.
Islam, R., Islam, M.R., Talukder, K.H., et al., 2020. An efficient method for extraction and recognition of bangla characters from vehicle license plates. *Multimed Tools Appl (MTAP)* 79 (27), 20107–20132.
Jain, A.K., Zhong, Y., 1996. Page segmentation using texture analysis. *Pattern Recognition* 29 (5), 743–770.
Karatzas, D., Shafait, S., Uchidad, S., Iwamuraetal, M., 2013. ICDAR 2013 Robust Reading Competition. *Proceedings of ICDAR 2013*, 1484–1493.
Khan, T., Sarkar, R., Mollah, A.F., 2021. Deep learning approaches to scene text detection: a comprehensive review. *Artif. Intell. Rev.* 54, 3239–3298.
Kobchaisawat, T., Chalidabhongse, T.H., Satoh, S.I., 2020. Scene text detection with polygon offsetting and border augmentation. *Electronics* 9 (1), 117.
Liao, M., Zhu, Z., Shi, B., Xia, G.S., Bai, X., 2018. Rotation-sensitive regression for oriented scene text detection. In: *IEEE conference on computer vision and pattern recognition*, pp. 5909–5918.
Lee, S., Cho, M.S., et al., 2010. Scene text extraction with edge constraint and text co linearity. In: *20th IEEE Int Conf. Pattern Recognit. (ICPR)*, pp. 3983–3986.
Li, Y., Lu, H., 2012. Scene text detection via stroke width. In: *Proceedings of IEEE Int. Conf. Pattern Recognit. (ICPR)*, pp. 681–684.
Li, H., Wang, P., Shen, C., 2017. Towards end-to-end text spotting with convolutional recurrent neural networks. In: *Proceedings of the IEEE international conference on computer vision*, pp. 5238–5246.
Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017. Feature pyramid networks for object detection. In: *IEEE conference on computer vision and pattern recognition*, pp. 2117–2125.
Liao, M., Lyu, P., He, M., Yao, C., Wu, W., Bai, X., 2019. Mask text spotter: an end-to-end trainable neural network for spotting text with arbitrary shapes. In: *IEEE transactions on pattern analysis and machine intelligence*. <https://doi.org/10.1109/tpami.2019.2937086>.

- Matas, J., Chum, O., et al., 2004. Robust wide-baseline stereo from maximally stable extremal regions. *Image Vis. Comput.* 22 (10), 761–767.
- Ma, C., Sun, L., Zhong, Z., Huo, Q., 2020. ReLa Text: exploiting visual relationships for arbitrary-shaped scene text detection with graph convolutional networks. In: arXiv :2003.06999.
- Moyeen, M.A., Alam, K.M.R., Awal, M.A., 2013. Bangla Text Extraction from Natural Scene Images for Mobile Applications. *Journal of Elect. Engg.*, The Institution of Engineers, Bangladesh, Vol. EE 39, No. I & II.
- Neumann, L., Matas, J., 2013. On combining multiple segmentations in scene text recognition. In: *International Proceedings of ICDAR*, pp. 523–527.
- Shahab, A., Shafait, F., Dengel, A., 2011. ICDAR 2011 robust reading competition challenge 2: reading scene images. *Proceedings of ICDAR 2011*, 1491–1496.
- Qin, H., Zhang, H., Wang, H., Yan, Y., Zhang, M., Zhao, W., 2019. An algorithm for scene text detection using multibox and semantic segmentation. *Appl. Sci.* 9 (6), 1054.
- Qiao, L., Tang, S., Cheng, Z., Xu, Y., Niu, Y., Pu, S., Wu, F., 2020. Text perception: towards end to end arbitrary shaped text spotting. In: arXiv :2002.06820.
- Shim, J.C., Dorai, C., Bolle, R., 1998. Automatic text extraction from video for content-based annotation and retrieval. In: *Proceedings of Int. Conf. Pattern Recognit.*, Vol. 1, Australia, pp. 618–620.
- Silva, D.A., Ciarelli, P.M., 2016. Edge detection and confidence map applied to identify textual elements in the image.
- Soni, R., Kumar, B., Chand, S., 2019. Text detection and localization in natural scene images based on text awareness score. *Applied Intelligence* 49, 1376–1405.
- Sen P., Das A., Sahu N., 2022. End-to-End Scene Text Recognition System for Devanagari and Bengali Text. In: Vasant P., Zelinka I., Weber GW. (eds) *Intelligent Computing & Optimization, ICO 2021. Lecture Notes in Networks and Systems*, vol 371. Springer, Cham. doi: 10.1007/978-3-030-93247-3_34.
- Sun, L., Huo, Q., Jia, W., Chen, K., 2015. A robust approach for text detection from natural scene images. *Pattern Recognition* 48, 2906–292.
- Sun, Y., Zhang, C., Huang, Z., Liu, J., Han, J., Ding, E., 2018. Textnet: irregular text reading from images with an end-to-end trainable network. In: *Proceedings of the Asian conference on computer vision*, pp. 83–99.
- Tang, Y.Y., Lee, S.W., Suen, C.Y., 1996. Automatic document processing: a survey. *Pattern Recognition* 29 (12), 1931–1952.
- Tang, Y., Wu, X., 2017. Scene text detection and segmentation based on cascaded convolution neural networks. *IEEE Trans Image Process* 26 (3), 1509–1520.
- Tian, C., Xia, Y., Zhang, X., Gao, X., 2017. Natural scene text detection with mc-mr candidate extraction and coarse-to-fine filtering. *Neuro computing* 260, 112–122.
- Unar, S., Hussain, A., Shaikh, M., et al., 2018. A study on text detection and localization techniques for natural scene images. *International Journal of Comput. Science and Net. Security(IJCSNS)* 18 (1), 99–111.
- Wang, K., Babenko, B., Belongie, S., 2011. End-to-end scene text recognition. In: *Proceedings of IEEE Int. Conf. Comput. Vis. (ICCV)*, pp. 1457–1464.
- Wang, W., Lu, T., Yu, G., Shen, C., 2019. Efficient and accurate arbitrary shaped text detection with pixel aggregation network. In: *Proceedings of the IEEE international conference on computer vision*, pp. 8440–8449.
- Xue, C., Lu, S., Zhang, W., 2019. MSR: multi-scale shape regression for scene text detection. In: arXiv:1901.02596.
- Yang, Q., Cheng, M., Zhou, W., Chen, Y., Qiu, M., Lin, W., Chu, W., 2018. Incep text: a new inception-text module with deformable psroi pooling for multi-oriented scene text detection. In: arXiv :1805.01167.
- Yao, C., Bai, X., Liu, W. et al., 2012. Detecting texts of arbitrary orientations in natural images. In: *Proceedings of IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 1083–1090.
- Yi, C., Tian, Y., 2011. Text string detection from natural scenes by structure-based partition and grouping. *IEEE Trans. Image Process.* 20 (9), 2594–2605.
- Yu, B., Jain, A.K., Mohiuddin, M., 1997. Address block location on complex mail pieces. In: *Proceedings of Int. Conf. Document Analysis and Recognit.*, Ulm, Germany, pp. 897–901.
- Yu, C., Song, Y., Zhang, Y., 2016. Scene text localization using edge analysis and feature pool. *Neuro computing* 175, 652–661.
- Zhang, H., Zhao, K., et al., 2013. Text extraction from natural scene image: A survey. *Neuro computing* 122, 310–323.
- Zhang, H.J., Gong, Y., Smoliar, S.W., Tan, S.Y., 1994. Automatic parsing of news video. In: *Proceedings of IEEE Int. Conf. Multimedia Computing and Systems Boston*, pp. 45–54.
- Zhang, G., Kai, H., Zhang, B., 2017. A natural scene text extraction method based on the maximum stable extremal region and stroke width transform. *Journal of Xi'an Jiaotong University* 1, 21.
- Zhong, Z., Sun, L., Huo, Q., 2019. An anchor-free region proposal network for Faster R-CNN based text detection approaches. *Int. J. Doc. Anal. Recognit.* 22 (3), 315–327.
- Zhu, Y., Yao, C., Bai, X., 2016. Scene text detection and recognition: Recent advances and future trends. *Front Comput. Science* 10 (1), 19–36.
- Zhang, C., Liang, B., Huang, Z., En, M., Han, J., Ding, E., Ding, X., 2019. Look more than once: an accurate detector for text of arbitrary shapes. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 10552–10561.