IV. Proposed Work

Our proposed work of text detection consists of three phases: 1) Edge detection include two steps: (1.1) Convert RGB (3 channel) image into grey (1 channel) image. (1.2) Apply canny edge detection method (1.3) Edge extraction using vertical and horizontal projection by Sobel projection. 2) MSER region detection.3) Apply geometric properties like aspect ratio, occupation, stroke width variance.4) Character region detection and bounding box generation.4.1) find stroke width ratio.4.2) Binarize the image.4.3) Dilate image and find contours.

*A. Edge detection*

In this work, first original RGB image is converted into grey image for easy processing. Apply canny edge detection to get edge strokes. The edge is extracted by vertical and horizontal projection by sobel operation. We can see the vertical and horizonal soble projection in below figures.



Fig.1 Sobel X-axis (horizontal) projection



Fig.2 Sobel Y-axis (vertical) projection

*1) Canny edge detection method:*

Canny edge operator identifies edges in an image. Also, to extract useful structural information canny operator is required. It finds the intensity gradient of the image. Edges computed as follow:

Edge\_Gradient (G) =

Where Gx value for the 1st derivation in the horizontal direction and vertical direction (Gy).

*2) Edge extraction using sobel operator:*

There are many techniques are there too extract edges but, in this work, we have used sobel technique. Sobel operator produce more perfect edges in different orientation. Also, we have calculated X and Y gradient using sobel operator.

*B. MSER region detection*

By using MSER technique we have detected maximum stable regions. It retrieves a number of co-variant regions. In this technique, image I is a set of {0….255} number.

I indicate as: I : D ⊂ Z2 {0…255}

Region Q denotes continuous subset of D. Extremal region is denoted as Q subset of D as follow:



(Above statement is indicating for maximum intensity region)

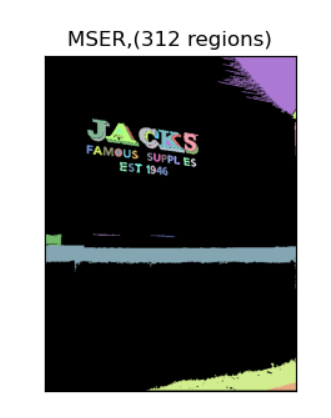


Fig.3 MSER regions shows in different color

*C. Character region detection*

In this last phase, stroke widths are calculated, and resultant image is binarized. Binarization is the process of finding threshold value and assign them as 1 or 0 (background or text object). After binarization, image is dilated. Dilation is used to join the broke part of text. It connects the edges into cluster. Also, it reduces the noise and increases the white region in the image.



Fig.4 Binarized Image

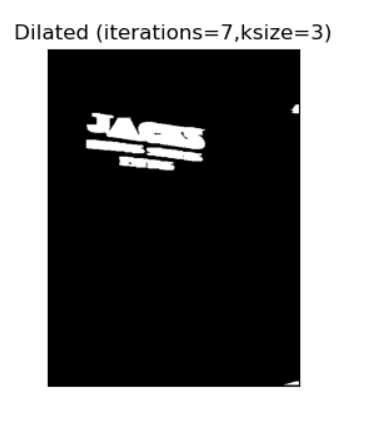


Fig.5 Dilated image with kernel size=3 and iteration=7

Dilated image is given as input to contour method to find the points that have same intensity. Also, it joins all the points along with boundary of an image. This contour method is useful to produce bounding box around the interest of area (i.e. text region).

Fig.6 Image with bounding boxes around text

*D. Algorithm*

Algorithm: Text detection using MSER and stroke width

Step 1: Convert the input RGB image in gray image.

Step 2: Apply the Sobel projection on grey image.

Step 3: Compute maximally stable extremal regions (MSER) of RGB image.

Step 4: Apply geometric features like aspect ratio, compactness on MSER regions.

Step5: Find stroke width variance ratio of grey image on which Sobel projection is applied.

Step 6: Binarized the grey image.

Step 7: Dilate the binarized image and find contours in dilated image.

Step 8: Draw the bounding boxes around contoured area where text is present.

Step 9: Finally, bounding box generated around text.

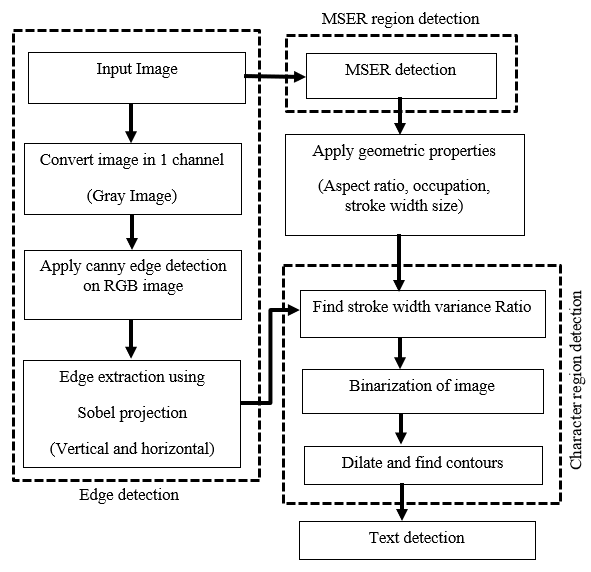
*E. Flowchart of proposed work*

Fig. 7 Flow chart of text detection

V. Results

To check the result and performance of text detection, we used ICDAR 2003 dataset (Robust Reading and text locating dataset). Also, we have captured some real time scenario images. In the figures, figure (a) is input image which is taken from ICDAR 2003 dataset, figure (b) is image with dilation processes is done on image to get area of interest (i.e. text region) and figure (c) shows final output of text detection, text is detected and bounding boxes generated around text. Figure (d) and (e) are captured in real time and final output is shown. Figure (f) is showing some failed text detection samples.



Fig.8 Input image

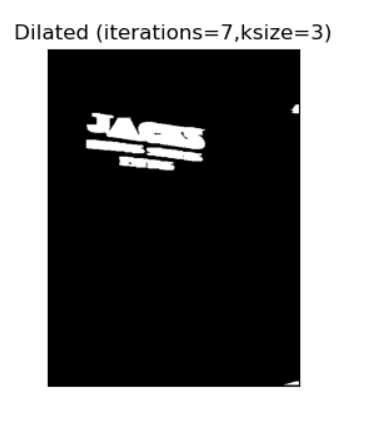


Fig.9 Interest of area region with dilation



Fig.9 Final output with bounding boxes



Fig.10 Realtime scene image captured by camera



Fig.11 Realtime scene image captured by camera

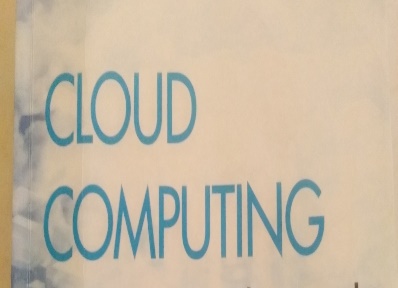
 

Fig.12 Failed text detection samples