Text Extraction from Mobile Camera Captured Images

Deepak Sharma ds5930@g.rit.edu
Department of Computer Science, Rochester Institute of Technology

Abstract—

This Paper suggests a novel approach for developing a software application for text extraction form mobile captured images. The purpose of text extraction is to translate into another language so that a person travelling at foreign places can understand, instantaneously, meaning of various sign board and available text information in his/her surroundings.)

Keywords—edge detection, labeling, segmentation

I. INTRODUCTION

This paper discuss about a novel approach inspired by the work of my predecessors[1] who have developed application for text extraction from various kind of signboards we see every day in our surrounding. This paper also discuss about limitation of suggested novel approach and previous applications developed for the similar purposes, It will also discuss ideas and approaches, which didn't worked provide expected results, while developing application.

The task of text extraction can be divided into three major steps, Pre-processing for removing unwanted information from image, noise reduction, and Contrast enhancement. Second step is to detect edges for detecting close shapes in image for labelling the various connected components followed by removing the non-text connected components using logical geomatics filters. In final step post processing carried out on text candidates by measuring their feature distance from original image and finalize the characters of the text. This application assumes that background is constant as most of the signboards have a constant background, though the text itself can be written in multiple colours and shapes. As this approach is based on edge detection it also assume that text is significantly large. In result section I will discuss the minimum size required for the text to successful recognized. Major challenge I have faced while developing this application was to find closed edges around the character, once you gain close edges around the target text element the success of extraction become very high, but with one pixel opening of edges, application may end up losing some character. An iterative algorithm for 99% edge connectivity has been suggested in this paper. In addition details of some other approaches for gaining edges connectivity without using any global morphological operation -- while ensuring the edge connectivity application has to make sure it should not end up connecting two text characters which are not connected in original image -- has been provided. The application has been tested on a dataset of 100 text images including all size of text, containing significant non-text elements and noise. Our test dataset is not restricted for road signboards.

II. PREVIOUS WORK

This application is a sequential composition of various small tasks for computer vision and image processing, we can divide this application in two major parts:

1. Text Extraction
2. Character Recognition.

There are a major research has be going on in the field of text extraction and I have found a number of generals on the topic.

While learning the techniques for text extraction from image I have perceived that there are two kind of methodologies have been used by my most of predecessors, Fist the most common for such task is to apply pre-processing followed by text region extraction and then after removing unwanted part from grayscale or binary image, sending the binary image to some OCR system (Example: [1]).

Others (Example: [3] and [4]) which were based of transforming image by wavelet or DCT transform and performing request filtering in order to distinguish the text from unwanted elements of the image. When the text and images (Unwanted) are present in a juxtaposition later approach is more efficient than former.

A ubiquitous issue with algorithms/procedure I have found that no approach work on every possible text image. And most common challenges offered by environmental factors, present in images like lighting conditions, illumination, reflection.

I have found approach [1], [3] [4], [5], [6] and [7] which are relevant to my objective of text extraction. Approach [5] has suggested boarder detection using morphological operation is sensitive to images as it is not a robust procedure, structuring element can end up connecting two characters.

Similar issue with. Approach [6] has suggested a fixed size morphological structuring element has achieving result from this approach is not possible. [7] Has proclaimed approaches for removing non-text connected components based on the

$$\left(A/(W \times L) \leq \frac{1}{4.5} \text{ or } > .95\right) \quad \text{and}$$
$$\left(\min\left(W/L, L/W\right) \leq \frac{1}{5}\right)$$

Rule W/L < 1/5 will remove character I for sure, In suggested application W/L ratio .03 is allowed, which still I haven't got 100% accuracy with 'I' character. Similarly A/(W*L)> .95 can remove character "I", " T "or "J". For character M and W sometimes area reach around 90% so we may lose M or W by A /(W*L)>.95.

III. Experiments:

Based on application suggested by [1], Initially I followed the suggested steps and developed a similar application, Based on results of the suggested approach I have changed approach for edge detection as suggested method for close body edges didn't provide close shape for my entire dataset which result into losing character, specially M, N, V and W.

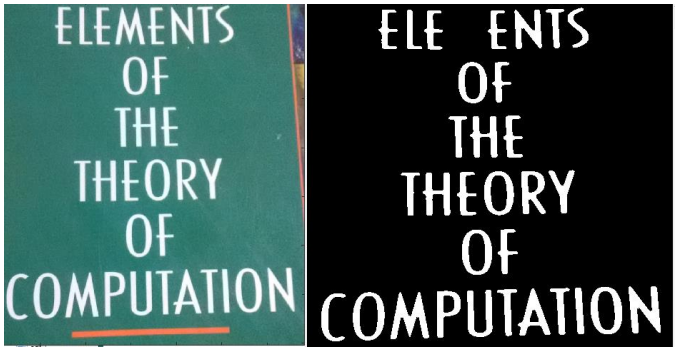


Fig. 1 result received after performing 8 Kernel filter suggested by [1]

So most part of my work has followed the skeleton of steps suggested by [1].

A. *High Level Design*

Pre-Processing: In this step task of quantization smoothing and contrast Enhancement based on entropy value was performed. Objective of this step is create ideal conditions for edge detection.

Region Sementation: In this part of the using sobel edge detector, edges was detected. Then using the suggested iterative algorithm, binarization of edges was performed. Objective of this step is creating ideal conditions for labelling so that all text elements should have a close (continuous) edge.

Post Processing: Perfroming labeling and removing non-text element. Objective of this step is creating binary image which has text and clean background.

Details of the low level Design.

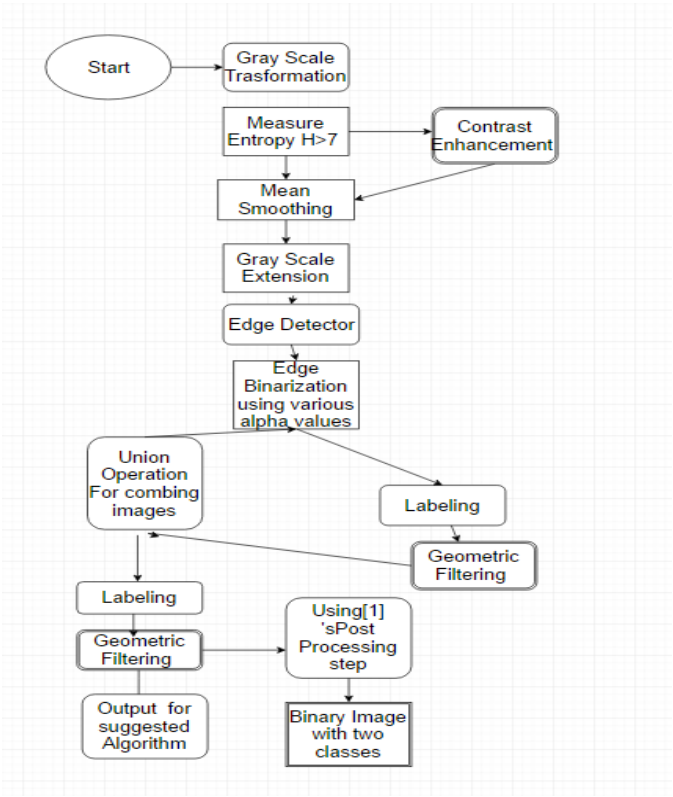


Fig2. Details of the low level design for suggested application.

1. Image Quantization/resized:- All the images with size greater than 512*512 was resized using segmentation, by performing this, performance of geometric filter was enhanced.. As too big character can be consider as a non-text elements it also enhanced the length width ratio of characters, few signboards which are large in x direction compare to y direction provide suspicious aspect ratio to geometric filter, It also reduced the computation burden of the application .

2. Grey Scale Transformation:- Image was converted to grey scale using matlab function rgb2gray. By doing so we removed unwanted information, like colors, in image.

3. Smoothing Using Mean Filter: - smoothing using mean filter before edge detection so that noise can be removed.

4. Grayscale extension: - In order improve the quality of contrast, this step has been performed. If image does not require contrast enhancement this function will leave grayscale image intact. [1] Has suggest to calculate alpha and bita values according to min and max value of index in image then using these value transform the image using below function.

$$\alpha = -min\_S \text{ and } \beta = \frac{255}{max\_S - min\_S}.$$

$$G(x,y) = (S(x,y) + \alpha) \times \beta$$

5. Entropy calculation: - Entropy of the image was calculated for measuring the monotonicity of the grey scale image. An image with low contrast will have high monotonicity. So based on the value of Entropy contrast enhancement should be perform. [1] Has suggested that if value of entropy is less than 3.8 than contrast enhancement should be performed. In my experiment I have found that images with entropy less than 6.5 to 7 should be subjected to contrast enhancement. As while experiments, for bellow image Entropy value is 4.4312 which stand in one of the lowest entropy, here contrast enhancement has given better results. But performing contrast enhancement simply based on entropy value is not a good idea, as in the figure 3 (entropy value 5.5844). Contrast enhancement distorted the 'n' character.

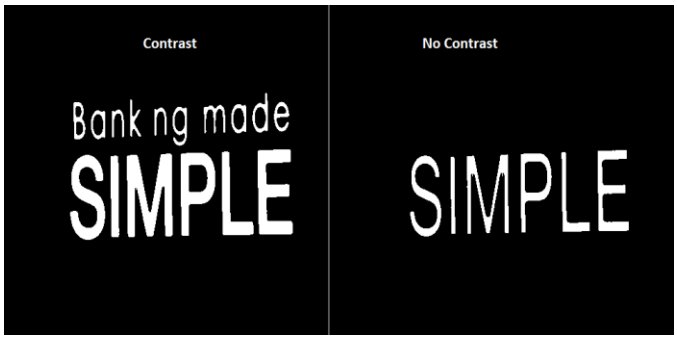


Fig 3. Output of the purposed application left with contrast enhancement right without applying contrast enhancement.

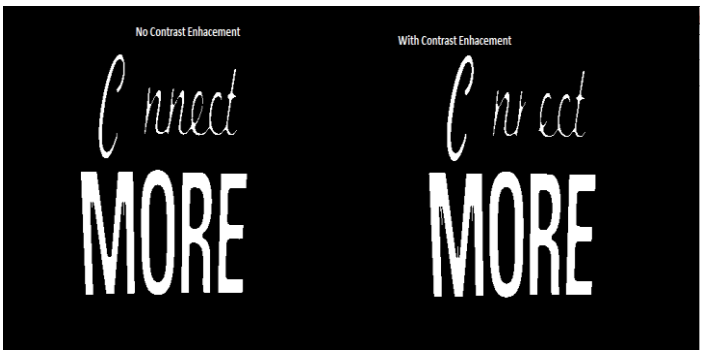


Fig 4. Output of the purposed application left with contrast enhancement right without applying contrast enhancement.

I have tested the application for 100 plus images and the average value for entropy was around 6. Generally images with white background have less entropy but they also required contrast enhancement.
Entropy of the Image can be calculated using:

$$H = -\sum \leq p_i \times \log p_i = \sum \leq p_i \times \log\left(\frac{1}{p_i}\right)$$

Equation Source: [1]

6. *Contrast Enhancement*: - Contrast enhancement plays a vital role in this application. Contrast enhancement provide a good platform for performing the task of edge banalization, without applying contrast enhancement thresholding on sobel filter output is very difficult, as in same image for a relaxed threshold thin character like 'I' get lost it mass (body), and I end up losing 'I' because algorithm considered it only an edge, on the other hand a tight threshold value does not provide bounded edges. Here nonlinear function of contrast enhancement helped to create well defined edges.

Fig 5: Using Contrast Enhancement(Intermediate result)

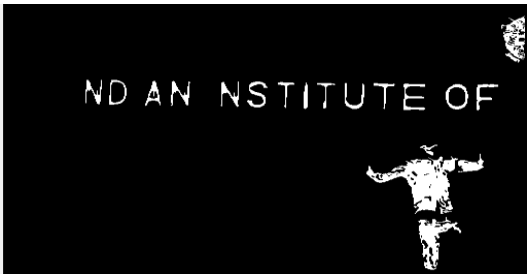


Fig 6: Without using Contrast Enhancement. (I's are missing because mass of 'I' was less and it has been originally removed by geometric filter, because it was considered as a noise in the image.)

But same time contrast enhancement can join two characters which are were originally very near to each other. This phenomena has been observed where signboards are painted by the painter, this is a bad news and disjoining the joined character will require some logical filters.

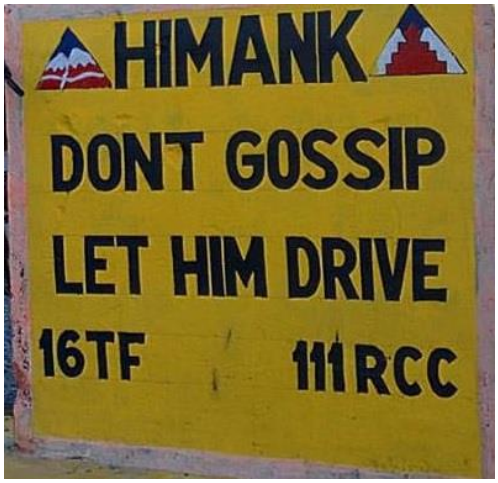


Fig7. Original Image

Fig.8 output with Contrast Enhancement

Fig 9. Output without Contrast Enhancement

To overcome this this problem one solution I tried was to do edge detection in both contrast and non-contrast image and take union of these two edges, but position of edges will be different and this approach will generate the edge between two images but same time, it's value has failed to maximize because non-contrasted image provide less strong edges in compare to contrasted image.

Fig.10 Union of Edges calculated for contrasted and non-contrasted version of image.

By multiplying from some offset like 1.7, I boosted the values of the non-contrasted edges where contrast images' edges value were less than non-cons tasted edges.

Fig 11 by boosting the non-costed image on specific location union of edges have performed well.

Contrast Enhancement has been implemented by implementing below function suggested by [2] F

$$C(x,y) = \frac{255}{1+\exp\left(\frac{aver\_T - T(x,y)}{\upsilon}\right)}$$

7. Edge Detection/Sobel Filter: - Edge detection is a crucial step in this process, as we are targeting to completely closed bodies of the words. In the purposed algorithm by [1] has suggested a novel edge detector with 8 kernels but as I discussed earlier, thresholding the output of edge detector with 2.6*mean for all images didn't worked. Which trigger to the linkages in edges and I lost the character. In order to fix this problem I tried multiple approaches. **First,** I observed that places of discontinuity were sharp angles or corners. I didn't want to apply any morphological operation because these are global method and they introduced discontinuity on some other place or can join to different character. Shape of structuring elements cannot be determine which can produce best results. In order to fix these sharp angles, I calculated Harris Corners and try to add few pixels which can fix these openings. This approach didn't give 100% results, as I had no idea how many Harris Corner I need for fix the opening as the amount of text cannot be determined.

Fig 11 2000 Harris Corner over the edges of the text.

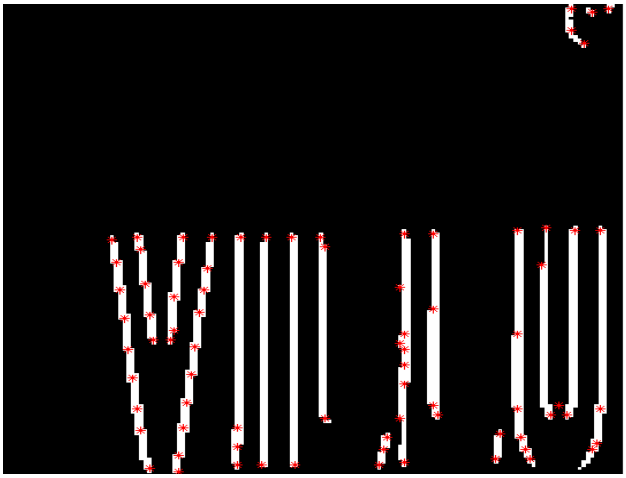


Fig 12 Harris Corners, U can be fixed but V (inner Shape) cannot be fixed by adding pixels.

Though Harris corner were on the points of discontinuity but how many and how much pixel needs to be dropped cannot be determine. In addition Harris corner were not only presented on corner but also appears on edges. To overcome this I think of **second approach**, to find corners only, I named this approach Deepaks' Corner.

Steps for calculating Deepaks' Corner:

1. Apply Sobel edge detector for calculating edge magnitude and angles.
2. Perform binarization of edges with any adequate threshold.
3. Set angles to 0 if corresponding magnitude is 0
4. In angle matrix apply a 4*4 kernel which ignore all the 0 values and calculate variance for 4*4 matrix with rest of non-zero value.
5. The resultant matrix will have only nonzero values where Conner exists.

For me this operation was slow and didn't render output in reasonable time with my limited computation power.

Final Approach was an iterative approach and I am purposing this novel method for rendering closed edges.

1. Calculate the average value of edge magnitude for the image.
2. Set alpha=2.7
3. Perform binarization of image for alpha*mean_edge_magnitude.
4. Perform labelling of the connected components
5. Perform geometric filter on output
6. Count the number of connected components, if number of CC is 1.8 times of the previous step's count, and number of CC is > 30 than abort this process.
7. union the results with previous results
8. Set alph=alpha-.2 or .3
9. Repeat step 3 if alpha is greater than .3.

This approach can give exceptionally well results if geometric filter perform well. This iterative algorithm should not be performed when number of connected components become more than 1.8 times of previous count, because when we reduce the threshold for binarization then fake edges start appearing in image, and at one point of time number of connected components will increase exponentially.

Now we will analyse the results of above suggested algorithm for below input image.

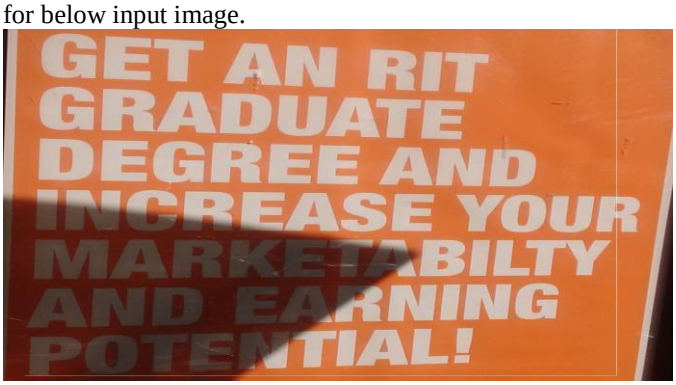


Fig 13. An input image for which number of connected components has been analysed for various threshold values.

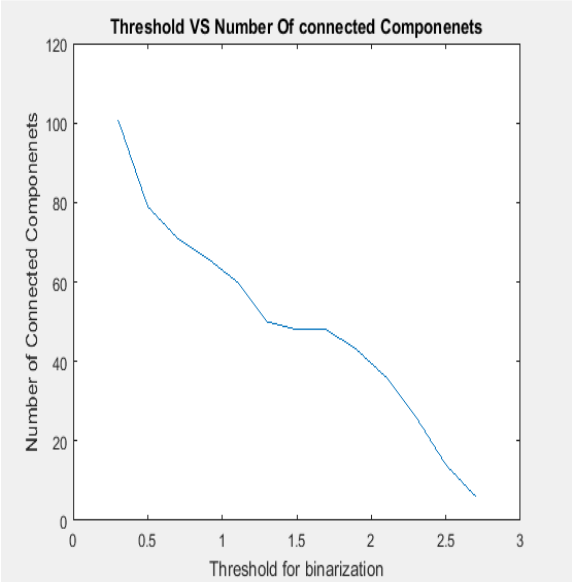


Fig. 14 Number of connected components vs Threshold value for Fig 14.

As we can see, by dropping the threshold value number of connected components have increased with varying pace. A sandal point can be observe at .5 value, as below .5 the number of connected components has increased at rapid pace. The ideal threshold value should be between 1.3 to 2.4 where we have observed 40 – 50 connected components, out of which 10 -20 can be noise. So we can synthesise and adaptive algorithms where pace of increasing number of connected components should be less than 1.2 and we can limit the number of connected components by 70 to 100. By applying this algorithm we avoided the sandal point where fake edges become prominent over real edges, we will complete our task of extracting text without losing any character, the success of algorithm depends on the efficiency of geometric filter. If geometric filter is performing statistical operations like mean area of connected components, and due to too much edginess number of connected components increased than we may end up losing information. Though I haven't tested an idea of implementing two classes geometric filter where first filter will be relaxed and after the completion of purposed algorithm other filter can executed to remove unwanted areas.

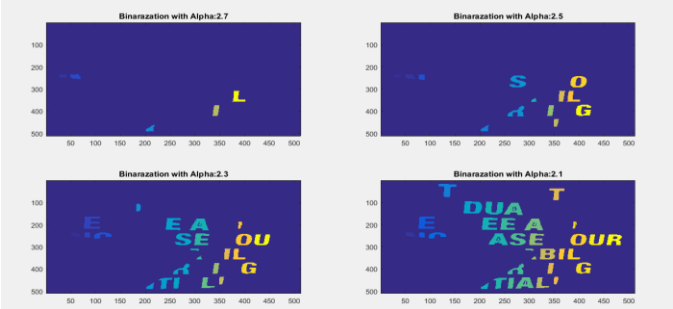


Fig. 17. Iterative output of algorithm for alpha 2.7, 2.5, 2.3 and 2.1

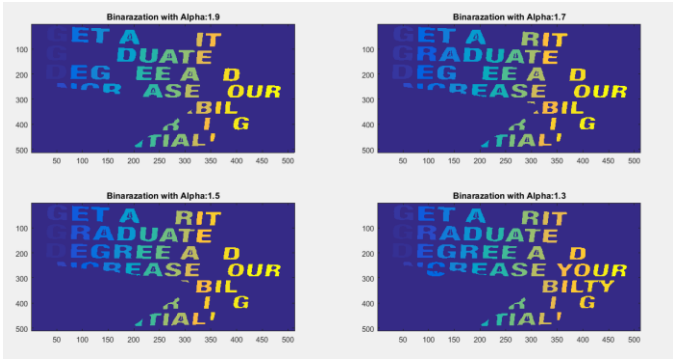


Fig.15 Iterative output of algorithm for alpha 1.9, 1.7, 1.5 and 1.3

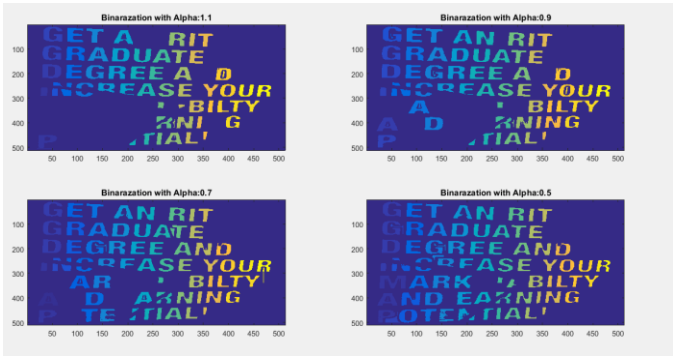


Fig. 16 Iterative output of algorithm for alpha 1.1, .9, .7 and .5

Limitations of the purposed method: The limitation of the purposed method appears when the text appearing when start getting close to each other or size of text become small than geometric remove some text or does not remove some non-text elements, which start feeding abnormal text into union method.

Fig18. Output from union operation whet text size was small or thickness was less.

8. Labelling: -using labelling we identify closed shapes. In this application using labelling we have identified various connected components present in the image. Labelling can be performed using simple union find or breathe first algorithm [2] by assuming all the pixels as nodes and there exist a path between nodes with same value. As entire graph may not be connected, there will be multiple forest. Now we will assignee different index value to each forest. In order to perform this task we have to visit all the nodes of the graph at least once. For the experiment I have used in build method 'bwlabel' of Matlab. largest connected component was considered as Background.

9. **Geometric Filtering**, similar to [1] and [3] I have applied rules for eliminating connected components which are not text. **I have relaxed multiple rules as in my understanding removing text is more harmful than keeping some noise, because any classifier can remove the noise but can't reproduced the text or lost information**. By relaxing the rules this application performed well on quantity of data. In addition the purposed application perform union operations of various connected components and rules like cc_area < max_area/10 in post filtering can remove the text itself.

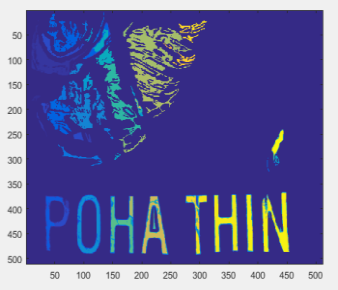


Fig. 19 Output of union operation where large unwanted connected components are visible.

Rules I used	Suggested by [1]
L > 0.8 * I(W)	L > 0.8 * I(W)
W > 0.4 * I(L)	W > 0.4 * I(L)
Mj/Mi > 30	L/H > 16
H/L > 4	H/L > 4
area > 10 * Avcs	area > 5 * Avcs
Ac < 0.05 * AvAcs	Ac < 0.2 * AvAcs
CC inside the other CC is non-text	CC inside the other CC is non-text
Text partially inside and partially outside the other text.	H > 1.8 × averageheight;
Mj*Mi<90	W > 1.8 ×
Mj<16	averagewidth
Any component within distance of 3 pixel from boundary	
Any component witch is has area less than 10.	

Where H, W Mj, Mi, AvAcg are the high, width, major axis, minor axis, average area of Connect components. I (W) and I (L) are the length and width of image.
Any connected component which fell in any of the above mentioned properties, has been removed from the image and those pixels were assigned to background label.

10. Post Processing: - union image received from suggested algorithm was again cleaned by geometric filter as by combine to connected component can become an unwanted connected component, by doing so we also removed noise introduced by union results provided by low threshold points.

Fig 20 Left Union image generated by purposed algorithm;

Fig. 21 Right After performing geometric filtering on union image.

If background is constant then for each connected component, [1] has suggest an approach for calculating relative distance from background. This distance can be

calculated using below equation revelled by author.

$$dis[i] = \frac{\left|averR_i - avrR_{bg}\right|}{averR_{bg}} + \frac{\left|averG_i - avrG_{bg}\right|}{averG_{bg}} + \frac{\left|averB_i - averB_{bg}\right|}{averB_{bg}}$$

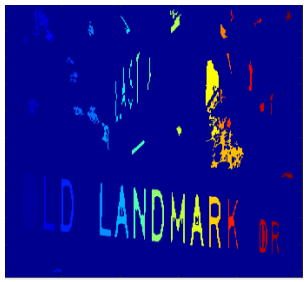
.

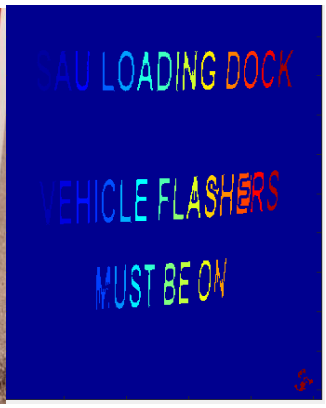
All pixels which belongs to background will have less distance from background and pixel which belongs to foreground will have relatively more distance from foreground. By doing so we will get a binary image.

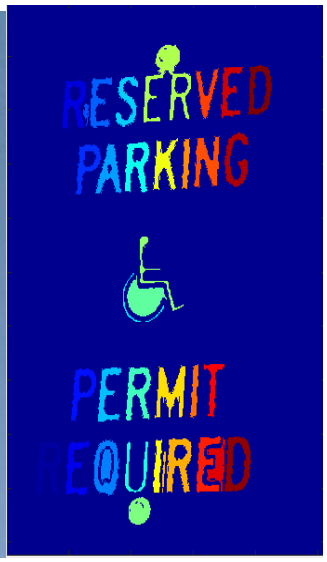
Similarly we can also construct a Mahanlobous feature matrix for each connected component and collects all the points similar from these known connected components in the original image, by doing so we will be able to recover any lost character in previous step if a character with similar feature like color is part of known text candidates connected components , But this approach introduce a lots of noise in image due noise point which join arbitrary class, in addition due to multiclass, the probability a pixel of one class can show up in other class, hence it requires sophisticated statistical filtering. I tried this approach for finding lost character but it was futile. .

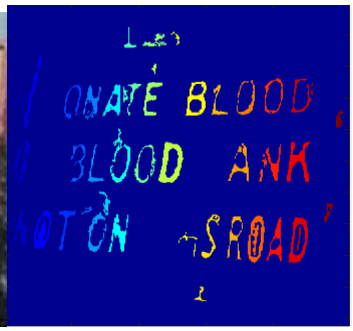
IV. RESULTS:

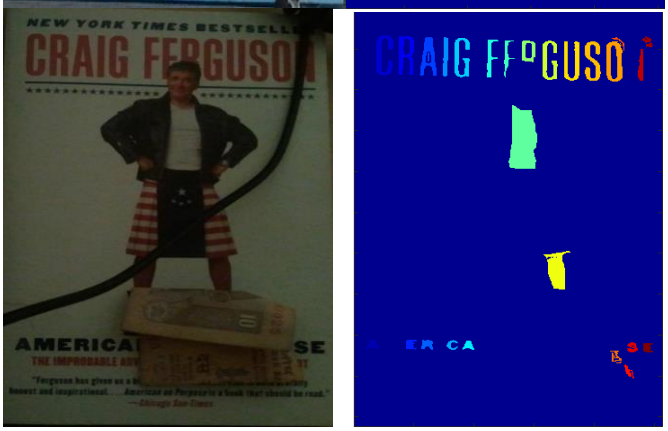
I tested the suggested approach for nearly 100 images, I tested for entire dataset of [1] and extraction results were above 92%, then I tested the application for various signboards with constant background, few characters were connected to each other as we discussed in constant enhancement step. Then I also tested the application for varying background images. I am sharing few results and for discussion.

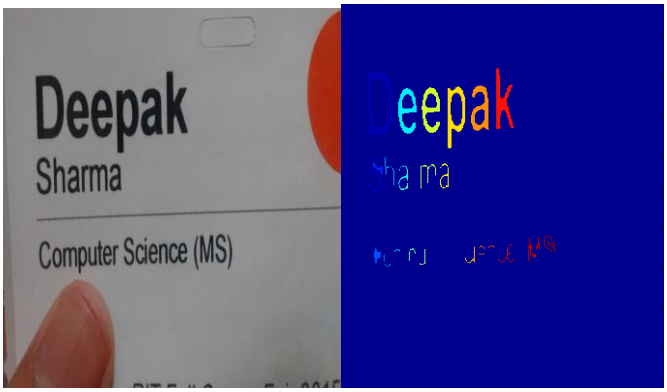












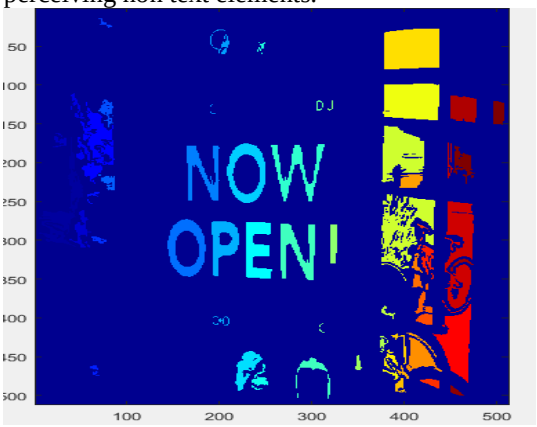
If the background is uniform or text is thick and close to camera, than this application can render correct output. In the "technology – scholarship" example the text was not thick so it didn't recognize I and due change in light it has failed to reorganize 'N' properly. In the process of collecting union of results this algorithm is prone to collect noise as in 'Sona Mysore Rice' example O has collected noise inside but now we have segmented the character and on independent character we can perform morphological cleaning to remove these noise. 'Way to Kala Pather' example can be cleaned by developing a separate filter.

There is no mass between the edges of the image because this application had down sampled the image before running the algorithm. So image should be somewhat near to the camera, this algorithm is heavy in computation so when I tried to run it on original size, my machine went out of memory. In order to fix it I have to develop a new application which can do region extraction.

Some text can't be extracted due to stylish writing.

V. Future Work:

There is a need for developing more geometric filtering for perceiving non text elements.



In order to perceive signboard at distance, I have to extract the text area/signboard from it, I tried to do this with Hough transform but I didn't found 4 points where two vertical and two horizontal lines meets.

VI. Conclusion:

In order to get edges, this application has introduced noise for multiple cases, by setting the value of alpha too low, this generated many unwanted connected components and geometric filter didn't remove them all. But as far as my understanding is guiding me, we can detect the value to threshold for which flood of edges will be generated. For strengthening my idea and understanding I checked the histogram distribution of edges, for multiple images it is a normal distribution, further I noticed that images with high contrast have single strong sandal point in their histogram and images with low contrast have multiple such points. One or multiple there exist such point and identification of such points can give us best output.

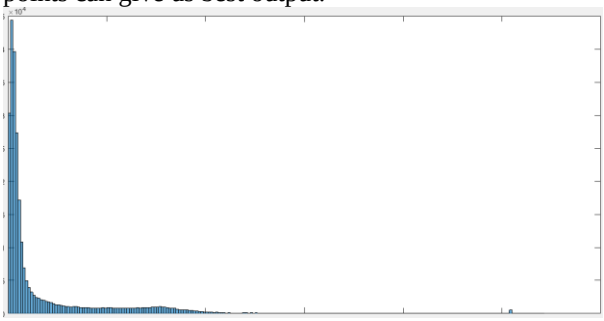


Fig. Histogram distribution of edge magnitude for a sharp image.

VII. References

[1] Yi Zhang* and Kok Kiong Tan Text extraction from images captured via mobile and digital Department of Electrical and Computer Engineering, National University of Singapore, in Int. J. Computational Vision and Robotics, Vol. 1, No. 1, 2009

[2] http://blogs.mathworks.com/steve/2007/05/25/connected-component-labeling-part-6/

[3] Intelligent Text Detection and Extraction from Natural Scene Images Department of Digital Media Design,Asia University, Taichung, Taiwan, R.O.C.Rong-Chi ChangE-mail: roger@asia.edu.tw

[4] TEXT EXTRACTION ALGORITHM UNDER BACKGROUND IMAGE USING WAVELET TRANSFORMS XIAO-WEI ZHANG, XIONG-BO ZHENG, ZHI-JUAN WENG School of Science, Harbin Engineering University, Harbin, 150001, china E-MAIL: zhangxiaowei@hrbeu.edu.cn

[5] Text Region Extraction from Low Resolution Natural Scene Images using Texture Features S. A. Angadi Department of Computer Science & Engineering Basaveshwar Engineering College Bagalkot, Karnataka, India M. M. Kodabagi Department of Computer Science & Engineering Basaveshwar Engineering College Bagalkot, Karnataka, India

[6] Devanagari and Bangla Text Extraction from Natural Scene Images U. Bhattacharya, S. K. Parui and S. Mondal Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata – 108, India {ujjwal, swapan, srikanta_t}@isical.ac.in}

[7] Text Detection and Removal from Image using Inpainting with Smoothing Priyanka Deelip Wagh Dept. of Computer Engineering SES's R. C. Patel Institute of Technology Shirpur (MH), India. D. R. Patil Dept. of Computer Engineering SES's R. C. Patel Institute of Technology Shirpur(MH).

[8] Morphological Text Extraction from Images Yassin M. Y. Hasan and Lina J. Karam

[9] Matlab help for implemnting Harris cornner.