www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 5 May 2015, Page No. 11797-11803

Text Extraction From Images With Edge-Enhanced Mser And Hardware Interfacing Using Arduino

[1]G. Nagaraju, [2]Dr. P. V. Ramaraju, [3]P.M.M.P.Sandeep, [3] SK.Mansoor Nawaz, [3]S.Kiran Bhargav, [3]M.Swaroop kiran
Asst. Professor Department of ECE
SRKR Engineering College
Bhimavaram, A.P., India
bhanu.raj.nikhil@gmail.com
Professor, Department of ECE
SRKR Engineering College
Bhimavaram, A.P., India
pvrraju50@gmail.com
B.Tech Students, Department of ECE
SRKR Engineering College
Bhimavaram, A.P., India

Abstract: Detecting ***text in natural images is an important prerequisite. Text Characters and Strings provide valuable information for many applications, extracting text directly from natural scene images or videos is a challenging task because of diverse text patterns and various background interferences. In this paper an attempt is made to extract text in Natural Scene image. This text detection algorithm employs edge-Enhanced Maximally Stable Extremal Regions as basic letter candidates. These candidates are then filtered using geometric and stroke width information to exclude non-text objects. Letters are paired to identify text lines, which are subsequently separated into words. The coding has implemented in MATLAB R2014a and the qualitative analysis is done. Based on the simplicity of eliminating cluttered background. we followed other techniques in extracting text easily such as color thresholding app and Region of interest (ROI).According to the statements in text extracted from natural images the controlling of electrical motors can also be done. This can be explained by hardware implementation of text extraction application using ARDUINO kit.***

Keywords: Edge Enhanced MSER, Color Thresholding, Region of interest (ROI), Text extraction, Image processing.

1. INTRODUCTION

Mobile visual search has gained popular interest with the increasing availability of high performance, low cost camera phones. In recent years, visual search systems have been developed for applications such as product recognition and landmark recognition. In these systems, local image features are extracted from images taken with a camera-phone and are matched to a large database using visual word indexing technique. Although current visual search technologies have reached a certain level of maturity, they have largely ignored a class of informative features often observed in images: text. In fact, text is particularly interesting because it provides contextual clues for the object appearing inside an image. Given the vast number of text-based search engines, retrieving an image using the embedded text offers an

efficient supplement to the visual search systems. As an essential prerequisite for text-based image search, text within images has to be robustly located. However, this is a challenging task due to the wide variety of text appearances, such as variations in font and style, geometric and photometric distortions, partial occlusions, and different lighting conditions. Text detection has been considered in many recent studies and numerous methods are reported in
the literature These techniques can be classified into two categories: texture-based and connected component (CC)-based.

Texture-based approaches view text as a special texture that is distinguishable from the background. Typically, features are extracted over a certain region and a classifier (trained using machine learning techniques or by heuristics) is employed to identify the existence of text. As opposed to texture-based methods, the CC-based approach extracts regions from the image and uses geometric constraints to rule out non-text candidates. An adaptive binarization method to find CCs. Text lines are then formed by linking the CCs based on geometric properties. Recently, Epstein et al proposed using the CCs in a stroke width trans-formed image, which is generated by shooting rays from edge pixels along the gradient direction. Shivakumara et al. extract CCs by performing K-means clustering in the Fourier-Laplacian domain, and eliminate false positives by using text straightness and edge density.

In this work, a novel CC-based text detection algorithm is proposed, which employs Maximally Stable Extremal Regions (MSER) as our basic letter candidates. Despite their favourable proper-ties, MSER has been reported to be sensitive to image blur. To allow for detecting small letters in images of limited resolution, the complimentary properties of canny edges and MSER are combined in our edge-enhanced MSER. Further we propose to generate the stroke width transform image of these regions using the distance transform to efficiently obtain more reliable results. The geometric as well as stroke width information are then applied to perform filtering and pairing of CCs. Finally, letters are clustered into lines and additional checks are performed to eliminate false positives. The overall process of the text detection is illustrated in Fig. 1. In comparison to previous text detection approaches, our algorithm offers the following major advantages. First, the edge-enhanced MSER detected in the query image can be used to extract feature descriptors like for visual search. Hence our text detection can be combined with visual search systems without further computational load to detect interest regions. Further, our system provides a reliable binarization for the detected text, which can be passed to OCR for text recognition. Finally, the proposed algorithm is simple and efficient. MSER as well as the distance transform can be very efficiently computed and determining the stroke width only requires a lookup table (Section 2.3).

(a) Detected MSER (b) Text candidates (c) Detected text

Fig. (1). Extracting text from a natural image. (a): Detected MSER for dark objects on bright background. (b): After geometric and stroke width filtering, text candidates are pair wise grouped to form text lines. The text lines are shown by the red lines. (c): Text line verification rejects false positives and the detected text is highlighted by the blue box.

2. TEXT DETECTION ALGORITHM

The flowchart of text detection algorithm is shown in Fig.2. At the input of the system, the image intensities are linearly adjusted to enhance the contrast. Subsequently, MSER regions are efficiently extracted from the image and enhanced using canny edges obtained from the original gray-scale image (Section 2.1). As a next step, the resulting CCs are filtered using geometric constraints on properties like aspect ratio and number of holes (Section 2.2). The stroke width information is robustly computed using a distance trans-form (Section 2.3) and objects with high variation in stroke width are rejected. Text candidates are grouped pair wise and form text lines. Finally, words within a text line are separated, giving segmented word patches at the output of our system.

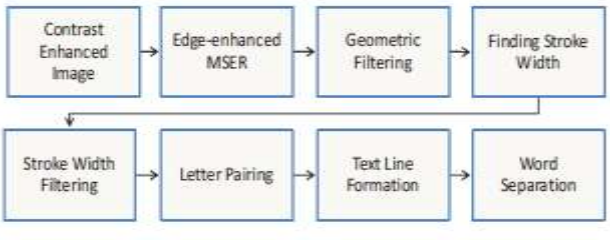


Fig (2): System flow chart

2.1. Edge-enhanced MSER

As the intensity contrast of text to its background is typically significant and a uniform intensity or colour within every letter can be assumed, MSER is a natural choice for text detection. While MSER has been identified as one of the best region detectors due to its robustness against view point, scale, and lighting changes, it is sensitive to image blur. Thus, small letters cannot be detected or distinguished in case of motion or defocus blur by applying plain MSER to images of limited resolution. Fig. 3(a) shows an example where multiple letters are identified as a single MSER region. To cope with blurred images we propose to combine the complimentary properties of canny edges and MSER. The outline of extremal regions can be enhanced by applying the precisely located but not necessarily connected canny edges. As shown in Fig.3 (a), we remove the MSER pixels outside the boundary formed by the canny edges. This is achieved by pruning the MSER along the gradient directions (indicated by the blue arrows) computed from the original gray-scale image. Since the type of the letter (bright or dark) is known during the MSER detection stage, the gradient directions can be adapted to guarantee that they point towards the background. Fig.3 (b) shows the edge-enhanced MSER, which provides a significantly improved representation of the text where individual letters are separated. This not only improves the performance of geometric filtering (Section 2.2), but also increases the repeatability of MSER based feature matching under different image blur conditions.

(a)

(b)

Fig. (3). Edge-enhanced MSER. (a) Detected MSER for blurred text. Canny edges are shown in red lines and the blue arrows indicate gradient directions. (b) MSER after pruning along the gradient

2.2. Geometric Filtering

With the extraction of edge-enhanced MSER, we obtain a binary image where the foreground CCs are considered as letter candidates. As in most state-of-the-art text detection systems, we perform a set of simple and flexible geometric checks on each CCs to filter out non-text objects. First of all, very large and very small objects are rejected. Then, since most letters have aspect ratio being close to 1, we reject CCs with very large and very small aspect ratio. A conservative threshold on the aspect ratio is selected to make sure that some elongated letters such as 'i' and 'l' are not discarded. Lastly, we eliminate objects which contain a large number of holes, because CCs with many holes are unlikely to be letter candidates.

2.3. Finding Stroke Width by Distance Transform

The importance of stroke width information has been emphasized in several recent studies. Motivated by Epstein's work on the Stroke Width Transform (SWT) , we develop an image operator to transform the binary image into its stroke width image. The stroke width image is of the same resolution as the original image, with the stroke width value labelled for every pixel. We deter-mine the stroke width using a novel approach based on the distance transform, which differs drastically from the SWT proposed in. Epstein's SWT forms CCs by shooting rays from the edge pixels along the gradient, and only keeps the rays if they are terminated by another edge pixel having the opposite gradient direction. This method does not work well when the opposite stroke edges are not parallel. Consequently, the stroke width CCs formed by the SWT often have undesirable holes appearing in curved strokes or stroke joints.

In contrast to the SWT, this proposed method guarantees that the SW information is provided at every pixel of the original CC with any stroke shape. In our algorithm, the Euclidean distance transform is applied to label each foreground pixel with the distance to its nearest background pixel. As can be seen in Fig.4a, the ridge values of the distance map correspond to half the width of the stroke. Then, we propagate the stroke width information from the ridge to the boundary of the object, along the "downhill" direction. The stroke width image is shown in Fig.4(b). This method bypasses the need to locate ridge pixels by iteratively propagating the stroke width information, starting from the maximum value to the minimum value of the distance map algorithm.

The output of the algorithm is an image where each pixel is assigned a value equal to half of the stroke width. Assuming that the stroke width of characters has a low variation, we exclude CCs with a large standard deviation. The rejection criterion is *std/mean* > 0.5, which is invariant to scale change.

a) Distance map b) Stroke width image

Fig (4): Finding the distance information (a) The distance transformed image (b) Stroke width image is formed by propagating the stroke width information from the ridge to the boundary. The numbers label half of the stroke width.

2.4. Text Line Formation and Word Separation

Text lines are important cues for the existence of text, as text almost always appear in the form of straight lines or slight curves. To detect these lines, we first pair wise group the letter candidates using the following rules. As letters belonging to the same text line are assumed to have similar stroke width and character height, two letter candidates are paired if the ratio of their stroke width medians is lower than 1.5 and their height ratio is lower than 2 (taking upper and lower case letters into account). Additionally, two CCs should not be paired if they are very distant.

Subsequently, text lines are formed based on clusters of pair wise connected letter candidates. A straight line is fitted to the centroid of pairs of letter candidates within each cluster and the line that intersects with the largest number of text candidates is accepted. The process is iterated until all text candidates have been assigned to a line, or if there are less than three candidates available within the cluster. A line is declared to be a text line if it contains three or more text objects.

We filter out improbable text lines by two additional validation steps. As shown in Fig. 1b, a false text line is formed along the repetitive windows. Repeating structures such as windows and bricks are commonly seen in urban images, resulting a large number of false positives. This can be avoided by applying template matching among the letter candidates. A text line is rejected if a significant portion of the objects are repetitive. Also, based on the observation that most letters have low solidity (proportion of the object pixels in the convex hull), a text line is rejected if most of the objects within that line have a very large solidity.

As a final step, text lines are split into individual words by classifying the inter letter distances into two classes: the character spacing and the word spacing. We calculate the distance between the vertical projections of each character along the text line and perform a two class classification using the Otsu's method

3. STEPS FOR EXTRACTION OF TEXT IN NATURAL SCENE IMAGES

3.1. Using Edge Enhanced MSER

Step 1: Load image

Load the image. The text can be rotated in plane, but significant out of plane rotations may require additional pre-processing.

Step 2: **Detect MSER Regions**

Since text characters usually have consistent color, we begin by finding regions of similar intensities in the image using the MSER region detector

Step 3: **Use Canny Edge Detector to Further Segment the Text**

Since written text is typically placed on clear background, it tends to produce high response to edge detection. Furthermore, an intersection of MSER regions with the edges is going to produce regions that are even more likely to belong to text. Note that the original MSER regions in MSER Mask still contain pixels that are not part of the text. We can use the edge mask together with edge gradients to eliminate those regions. Grow the edges outward by using image gradients around edge locations is computed using HelperGrowEdges helper function.

Step 4: **Filter Character Candidates Using Connected Component Analysis**

Some of the remaining connected components can now be removed by using their region properties. The thresholds used below may vary for different fonts, image sizes, or languages

Step 5: **Filter Character Candidates Using the Stroke Width Image**

Another useful discriminator for text in images is the variation in stroke width within each text candidate. Characters in most languages have a similar stroke width or thickness throughout. It is therefore useful to remove regions where the stroke width exhibits too much variation.

The stroke width image is computed using the helperStrokeWidth helper function. Note that most non-text regions show a large variation in stroke width. These can now be filtered using the coefficient of stroke width variation.

Step 6: **Determine Bounding Boxes Enclosing Text Regions**

To compute a bounding box of the text region, we will first merge the individual characters into a single connected component. This can be accomplished using morphological closing followed by opening to clean up any outliers.

Step 7: Perform Optical Character Recognition on Text Region

The segmentation of text from a cluttered scene can greatly improve OCR results. Since our algorithm already produced a well segmented text region, we can use the binary text mask to improve the accuracy of the recognition results.

3.2. Using Colour Thresholding APP

Step 1: Open Colour Thresholder App

The Color Thresholder App is opened in MATLAB R2014a in applications section.

Step 2: Load image

Load the color contrasted image to the app from the workspace.

Step 3.Select the Required Color Space

The Colour space is selected from available color spaces such as RGB, HSV, YCbCr, L*A*B

Step 4: Adjust Values

We have to adjust the colour thresholding values until the background is eliminated and only text is highlighted.

Step 5: Export image/function:

Now the Thresholded image is exported to workspace

We can also export function of this image and used for future

Step 6: Perform Optical Character Recognition on Text Region

The segmentation of text from a cluttered scene can greatly improve OCR results. Since our app already produced a well segmented text region, we can use the binary text mask to improve the accuracy of the recognition results.

3.3. Using Region of Interest (ROI):

Step 1: Load image

Load the image from figures

Step 2: Select Region

Drag the Cursor at the image where the text is located

Step 3: Perform Optical Character Recognition on Text Region

The segmentation of text from a cluttered scene can greatly improve OCR results, Since we selected the text from cluttered background to improve accuracy of recognition results.

4. RESULTS

4.1 Using Edge Enhanced MSER:

The input image, extracted text image and the extracted text output are shown in Fig 5.

(a) **(b)**

Output:

ans =

LE»

HANDICAPPED
PARKING
SPECIAL PLATE
REQUIRED
UNAUTHORIZED
VEHICLES
MAY BE TOWED
AT OWNERS
EXPFNSE

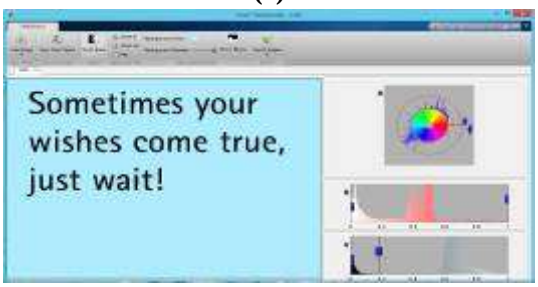
(c)

Fig5 (a) Input Image (b) Highlighted text after performing all the steps (C) Extracted text.

4.2.Using color Thresholder app:

The input image, extracted text image and the extracted text output are shown in Fig 6.

(a)



(b)

Output:

(c)

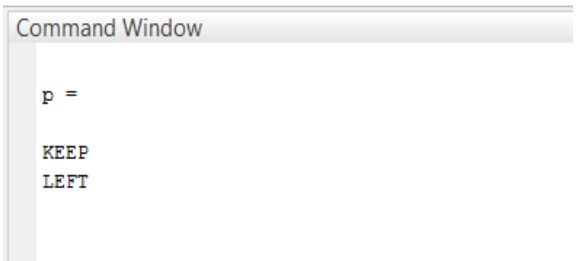
Fig6 (a) Input Image (b) Highlighted text after performing all the steps (C) Extracted text.

4.3 Using ROI

The input image, extracted text output are shown in Fig (7)

(a)

Output:



(b)

Fig7 (a) Input Image with selected text region (b) Extracted text.

5. TEXT EXTRACTION APPLICATION BY HARDWARE INTERFACING USING ARDUINO

5.1 APPLICATION:

Direction control and Speed control of DC motor with the text extracted from MATLAB by using Arduino Microcontroller. Step by step procedure is shown in fig (8)

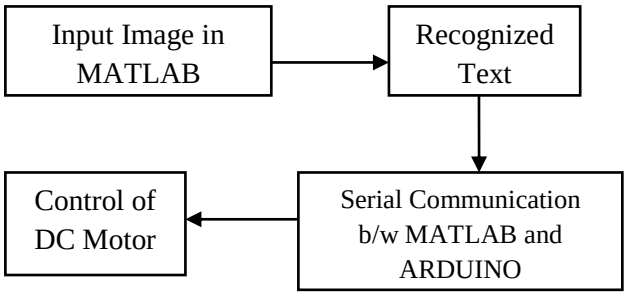


Fig (8)

5.2 Apparatus Required:

1. Arduino Microcontroller

2. IC L293D

3.9V battery

4. USB cable

5. DC Motor

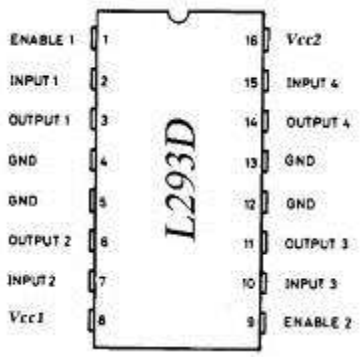
6. MATLAB 2014a

5.3 Theory:

5.3.1 Arduino: Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.). The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

5.3.2. IC L293D:

L293D is a dual h-bridge motor control motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. This higher current signal is used to drive the motors. L293D contains two inbuilt H-bridge driver circuits. In its common mode of operation, two DC motors can be driven simultaneously, both in forward and reverse direction. The motor operations of two motors can be controlled by input logic at pins 2 & 7 and 10 & 15. Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively. Enable pins 1 and 9 (corresponding to the two motors) must be high for motors to start operating. When an enable input is high, the associated driver gets enabled. As a result, the outputs become active and work in phase with their inputs. Similarly, when the enable input is low, that driver is disabled, and their outputs are off and in the high-impedance state.



5.4.Procedure:

Step1: Connections

1. Connect the DC motor with IC L293D at output pins of IC i.e., 3rd and 6th pins

2. Connect Remaining grounds pins and enable pins with Arduino.

3. Input pins of IC i.e, 2nd and 7th are connected to arduino digital output pins 9th and 10th.

4. 9V battery is connected as supply to Breadboard.

All these connections are shown in fig (9)

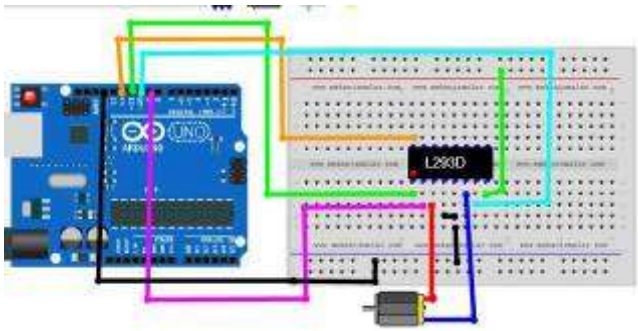


Fig (9)

Step2: Coding

1. Code the Arduino to control dc motor and communicate serially with MATLAB and upload to Arduino board

2. Code the MATLAB with Text recognition Algorithm and Serial communication with arduino.

Step3: Interfacing

1. Interface Arduino and MATLAB with USB cable

5.5.Results:

After Connecting Arduino and MATLAB with USB Cable

Run the MATLAB code and select the text region in the image then the Recognized Text in the image will communicate serially with Arduino via USB cable and this text will Control the DC motor direction and speed.

6. CONCLUSION

In this work, a novel text detection algorithm is proposed, which employs Maximally Stable Extremal regions as basic letter candidates. To overcome the sensitivity of MSER with respect to image blur and to detect even very small letters, an edge-enhanced MSER is developed which exploits the complimentary properties of MSER and Canny edges. Further, a novel image is presented operator to accurately determine the stroke width of binary CCs. This method has demonstrated state-of-the-art performance for localizing text in natural images. The detected text is binarized letter patches, which can be directly used for text recognition purposes.

Additionally, our system can be efficiently combined with visual search systems by sharing MSER as interest regions. And it is easy to use the extracted text to control different Mechanical and Electrical systems. And the extracted text is also very useful for blind people by converting the extracted text into speech. In future, this can be applicable for region language text extraction from images. With this any type of text will be extracted from any image can be happened.

7. REFERENCES

1. Digital Image Processing Using MATLAB by Rafael C.Gonzalez, Richard E.Woods,Stevan L.Eddins.

2. Chen,Huizhong,et al. "Robust Text Detection in Natural Images with Edge Enhanced Maximally Stable Extermal Regions." IEEE 2011.

3.J. canny- Pattern Analysis and Machine Intelligence, IEEE …, 1986 - ieeexplore.ieee.org

4. M. Donoser,H- Bischof-Computer Vision and Pattern …, 2006 - ieeexplore.ieee.org

5. R. Smith. *An Overview of the Tesseract OCR Engine*, Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007) Vol 2 (2007).

6. www.instructables.com.

7. Dr.P.V.Rama raju, G.Nagaraju, V.Rajasekhar, "Degraded document image enhancement in spatial domain using adaptive contrasting and thresholding", September 2014, International Journal of Multi disciplinary Educational research, volume 3. Issue 9(2), pp 318-331.

AUTHORS

Dr. P.V. Rama Raju

Presently working as a Professor at the Department of Electronics and Communication Engineering, S.R.K.R. Engg College, AP, India.

His research interests include Biomedical-Signal Processing, Signal Processing, VLSI Design and Microwave Anechoic Chambers Design. He is author of several research studies published in national and international journals and conference proceedings.

G. Nagaraju

Presently working as Asst Professor at the Department of Electronics and Communication Engineering, S.R.K.R. Engg College, AP, India.

He received the B.Tech degree from S.R.K.R. Engineering College, Bhimavaram in 2002, and M. Tech degree in Computer electronics specialization from Govt. College of engg, Pune university in 2004. His current research interests include Image processing, digital security systems, Biomedical-Signal Processing, Signal Processing, and VLSI Design.

P. M. M. P. Sandeep

Currently pursuing B.E in Electronics and Communication Engineering from S.R.K.R. Engineering College, A.P., India.

SK. Mansoor Nawaz

Currently pursuing B.E in Electronics and Communication Engineering from S.R.K.R. Engineering College, A.P., India.

S. Kiran Bhargav

Currently pursuing B.E in Electronics and Communication Engineering from S.R.K.R. Engineering College, A.P., India.