World Journal of Engineering Research and Technology
WJERT
www.wjert.org
SJIF Impact Factor: 4.326

EXTRACTION OF TEXT FROM COMPLEX IMAGES FOR MULTIMEDIA APPLICATION

Harshitha Urs T R, Chaitra B S.* and Santhosh B.

Dayanand Sagar College of Engineering, Bangalore-78, Karnataka, India.

Article Received on 16/04/2017 Article Revised on 07/05/2017 Article Accepted on 28/05/2017

***Corresponding Author**
Chaitra B. S.
Dayanand Sagar College of Engineering, Bangalore-78, Karnataka, India.

ABSTRACT

Large amount of information are embedded in natural scenes which are often required to be automatically recognized and processed. For the process the automatic identification segmentation and detection of visual text entities in usual scene images are required. Scene text could be any textual part of the scene images such as street signs, name plates or text appearing on t-shirts. The extracted text can be displayed on a palm size PDA or synthesized as voice output message over the ear phone. Text in image contains useful information which can be used to fully understand images. The various methods have been proposed in the past of detection and localization of text in images and videos. The properties related to the text in an image such as color, intensity and edges are used to distinguish text regions from the background and other region within the image. In this paper we have proposed a system that detect and extract the text from the complex images using a MATLAB tool.

KEYWORDS: *Text, Images, Recognizing, Extracted.*

I. INTRODUCTION

Text extraction in images and video has been rapidly increasing since 1990s and is an significant research field in content-based information indexing and retrieval. Text Extraction from image is concerned with extracting the relevant text data from a collection of images.[1] Fast development of digital technology has resulted in digitization of all categories of materials. Many existing paper-based collections, historical manuscripts, scanned document, book covers, pamphlets, posters, pictures, painting, magazines, educational, business card,

advertisements, web pages etc are converted to images. These images present many research issues in text extraction and recognition.

There are many applications in document engineering in which automatic detection and extraction of foreground text from complex background is useful.[2] If the text is printed on a clean background then certainly OCR (optical character recognition) can detect the text regions and convert the text into ASCII form. Several commercially available OCR products are able to perform this. However those products results in low identification accuracy when the text is printed against shaded and/or complex background.

In many applications the automatic localization of text within a natural image is an important problem. Once identified, the text can be analyzed, recognized, and interpreted. Many objects such as tree branches or electrical wires easily disturb the text in the image this leads to the confusion for text by existing OCR algorithms. For this reason, applying OCR on an unrefined natural image is computationally exclusive and may produce incorrect results. Hence, robust and efficient methods are needed to identify the text-containing regions within natural images before performing OCR.

The remaining part of the paper contains: section 2 provides information on design considerations. Architecture of the text extraction is explained in section 3, the detailed design of the architecture is described in section 4.

II. DESIGN CONSIDERATION

The interface design, user, and appearance consideration are the three phases that are covered first followed by the tools that are used in the MATLAB.

The main consideration for the selection of images is as follows

- The input image can contain any characters from Aa to Zz and numerical.
- The text should be in English. Symbol and calligraphy is not supported.
- Image should have clarity and should not have noise due to image pixels.

The text containing images needed for the analysis are captured using the image capturing device such as Web camera / mobile camera. The available ICDAR Datasets are also considered. The image so obtained is then processed and the corresponding voice clips are played.

III. ARCHITECTURE

The system architecture of text analysis is as shown in the figure 1. It shows how character extraction and processing takes place from high resolution image. It consists of 5 steps as shown in the flow diagram.

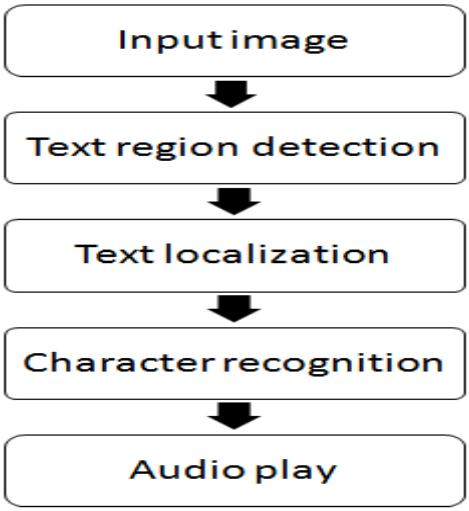

Figure 1: Architectural diagram of Text analysis from images

Original image: The graphical user interface (GUI) transforms the original image into a jpeg format (image) as shown in the figure 2. The orientation differs based on the input image position.[3]

Figure 2: Original image

Text region detection: Given the input image that is original image, the region with a possibility of text in the image is detected. This is done by preprocessing the image then crop the image exactly to text region using the algorithm. It is as shown in the figure 3.

Figure 3: Text region detected

Text localization: The Properties of the color/gray in the text region and their variation with the properties of the background region is compared to differentiate text region from its background or non-text region. The words are then separated by applying bounding box to every character.[4] The segmented image is as shown in figure 4.

Figure 4: Text located image

Character recognition

The image of the character obtained from the segmented image is compared with that of the character stored in the inbuilt character database using OCR.[5] The resulted output is displayed by using Notepad. Character recognition database is shown in figure 5.

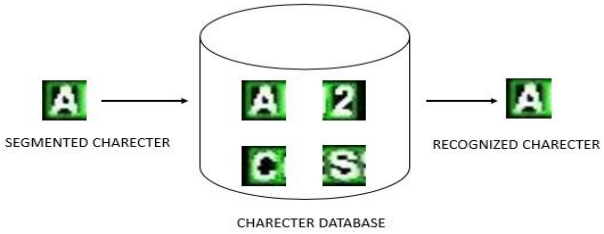

Figure 5: Character recognition database

Audio play: After displaying the character that is recognized, the corresponding audio clips are selected from the audio database and then played by using TTS (Text to speech) for figure 6.

Figure 6: Playing audio of character

The overall flow of the above steps is to perform the regular operation such as taking the input from the end-user, capturing from the camera, carrying out segmentation of text region, locating the text in the image, recognizing the letter and playing it with vocal voice.

IV. DETAILED DESIGN

Data flow models are an intuitive way of showing how data is processed by a system. At the analysis level, they should be used to model the way in which the data is processed in the existing system. The notations used in these models represent functional processing, data stores and the data movements between functions.[6] Data flow models are used to show how data flows through a sequence of processing steps. The data is transferred at each step before moving on to the next stage. These processing steps are program functions when data flow diagrams are used to explain a software design.

Level one data flow diagram for text region detection

The figure 7 represents the level one data flow diagram where the main process in level zero data flow diagram is shown to be classified into number of sub-process.

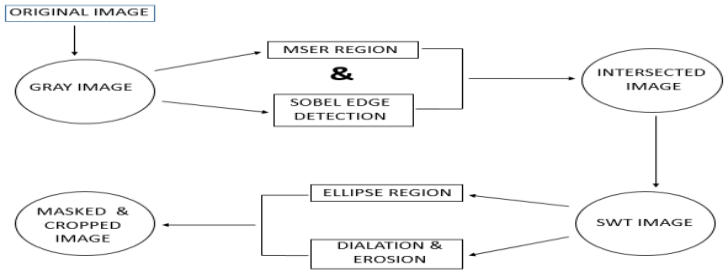

Figure 7: Level one data flow diagram for text region detection.

As per the flow diagram, we first convert the image captured into gray image. Since the original image is a RGB image and is a 3 dimensional matrix, where each pixel contains R G & B values, processing becomes complex to deal with every pixel.

So it is convenient to convert each RGB matrix to 2 dimensional gray images. Gray images have many shades of gray in between. Each gray scale matrix pixel contains values in the range 0 to 256, where 0 indicates black pixels and 1 indicates white pixels and all colors lie in between these values.[7] The RGB image converted into gray image is as shown in figure 8.

Figure 8: RGB image converted into gray image

As the intensity contrast of text to its background is typically significant and a uniform intensity or color 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.

Figure 9 shows an example where multiple letters are identified as a single MSER region.

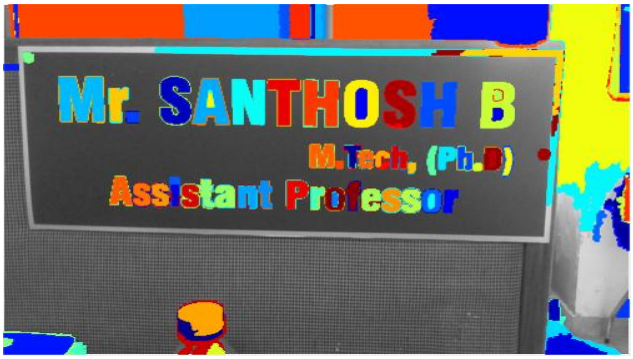

Figure 9: MSER region detected image.

To cope with blurred images we propose to combine the complimentary properties of Sobel edges and MSER. Figure 10 shows the sobel edge detected image.[8] We remove the MSER pixels outside the boundary formed by the Sobel edges. This is achieved by pruning the MSER along the gradient directions 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.

Figure 10: Sobel edge detected image

The importance of stroke width information has been emphasized in several recent studies. 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 labeled for every pixel. The processing aims to extract the stroke width at every pixel. To do this the key insight is that letters have roughly parallel sides.

To find the edges of the letters we use canny. Then, we calculate the gradient at every edge pixel. Depending on whether the text is light-on-dark or dark-on-light, the gradient will either point into letters or out of them.[9] Our algorithm depends on gradient pointing into letters, which is why it is no possible to detect light-on-dark and dark-on-light text simultaneously, but it can run twice on the same image to achieve that effect.

The output of the SWT is an image where each pixel is assigned a value equal to half of the stroke width. Figure 11 shows the stroke width transform output.

Figure 11: Stroke width transform image

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as dilation or erosion. The figure 12 shows the dilation and erosion output.

Dilation: The value of the output pixel is the maximum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to the value 1, the output pixel is set to 1.

Erosion: The value of the output pixel is the minimum value of all the pixels in the input pixel's neighborhood. In a binary image, if any of the pixels is set to 0, the output pixel is set to 0.

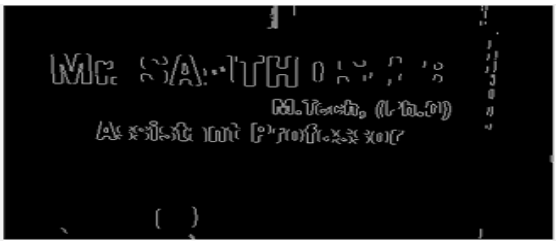

Figure 12: Dilation and erosion output image

Level two data flow diagram for text localization

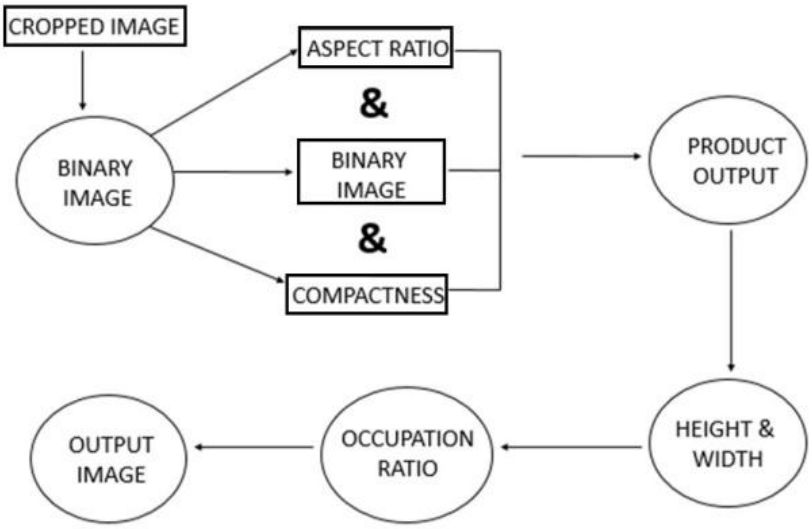

Figure 13: Level two data flow diagram for text localization.

Further the images are filtered using different properties .The properties are as follows

1) ***Aspect ratio:*** Aspect ratio is the term used to describe connected components by comparing the width to the height and expressing it in ratio form. Range is been set and the connected component in that range will be retained.

Figure 14: Aspect ratio output image

2) ***Compactness:*** Compactness is expressed in terms of area and perimeter. Compactness is defined as the ratio of area to perimeter square.

$$\text{Compactness}=\frac{area}{perimeter^2} \quad (1)$$

Figure 15: Compactness output image.

3) ***Height and width:*** We filtered using images using height and width. We assume that all the letter are of the specific height and width. Specific height and width of a connected component is retained.

4) ***Occupation ratio:*** Occupation ratio is defined as the ratio of number of connected component pixel to the bounding box area.

Formula for the occupation ratio given by

$$= \frac{number\ of\ connected\ component\ pixel}{bounding\ box\ area} \quad (2)$$

After these properties we are extracting the text in the image to the note pad. For that we use OCR. We apply bounding box to show that the characters have been recognized. The bounding box image is as shown in figure 4.

Level three data flow diagram for letter recognition

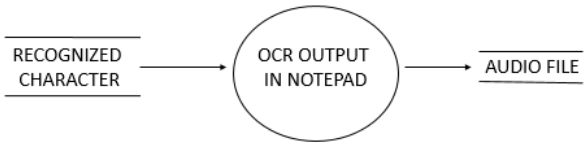

Figure 16: Level three data flow diagram for character recognition and audio play

OCR (optical character recognition): Optical character recognition (OCR) is the translation of optically scanned binary image contains text into character codes, such as ASCII.

Early versions needed to be trained with images of each character, and worked on one font at a time. Advanced systems capable of producing a high degree of recognition accuracy for

most fonts are now common. Some systems are capable of reproducing formatted output that closely approximates the original page including images, columns, and other non-textual components.

TTS (text to speech): A *text-to-speech (TTS)* system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

V. RESULT

Optical Character Recognition (OCR) is used for the translation of optically scanned binary images contain text into character codes, such as ASCII. Output of OCR is in form of a standard string and it is displayed on the notepad as shown below.

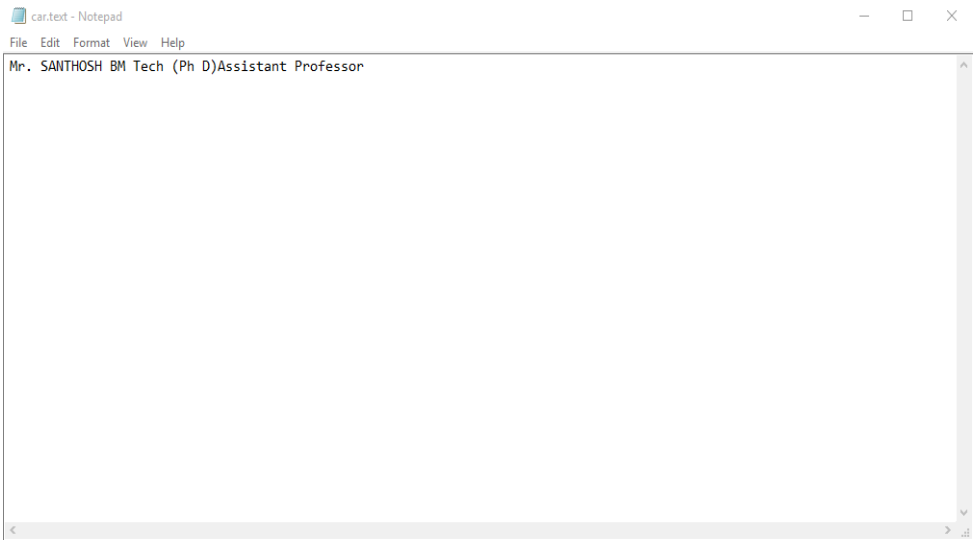

Figure 15: Recognized character displayed on the notepad

VI. CONCLUSION AND FUTURE ENHANCEMENT

In this paper there is an attempt to detect text from images and convert it into speech form. The texture feature contrast and variance have performed well in detection and segmentation of text region. The proposed methodology has produced a good result for natural scene images containing text of different size, font and alignment with varying background.

Several future enhancements could be made to the system. Inclusion of character stress recognition and punctuation marks and also inclusion of the naturalization of the voice like human expressions. Future inclusion of different kinds of text style, languages with more accuracy can also be done.

ACKNOWLEDGEMENT

The authors would like to thank to their friends and editors for their support. And also thankful to their guides, who helped them in improving the value of the paper.

REFERENCES

1. Yi-FengPan,XinwenHou,andCheng-LinLiu,SeniorMember,IEEE, "A Hybrid Approach to Detectand Localize Texts in Natural Scene Images", ieee transactions on image processing, March 2011; 20(3).
2. Wonjun Kimand ChangickKim, Member, IEEE, "A New Approach for Overlay Text Detectionand Extraction From Complex Video Scene", ieee transactions on image processing, Febavuary 2009; 18(2).
3. Sunil Kumar, Rajat Gupta, Nitin Khanna, Student Member, IEEE, SantanuChaudhury, and Shiv Dutt Joshi, "Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model", ieee transactions on image processing, August 2007; 16(8).
4. Yu Zhong, Hongjiang Zhang, and Anil K. Jain, Automatic Caption Localization in Compressed Video, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000; 22(4): 385-392.
5. D. Chen, J. Odobez, and H. Bourlard, Text Segmentation and Recognition in Complex Background Based on Markov Random Field, Proc. of International Conference on Pattern Recognition, 2002; 4: 227-230.
6. S. Antani, Reliable Extraction of Text From Video, PhD thesis, Pennsylvania State University, AUGUST 2011.
7. E. Y. Kim, K. Jung, K. Y. Jeong, and H. J. Kim, Automatic Text Region Extraction Using Cluster-based Templates, Proc. of International Conference on Advances in Pattern Recognition and Digital Techniques, 2010; 418-421.
8. J. Ohya, A. Shio, and S. Akamatsu, Recognizing Characters in Scene Images, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994; 16(2): 214-224.
9. A. R. Chowdhury, U. Bhattacharya, and S. K. Parui, "Text detection of two major indian scripts in natural scene images." Proc. of CBDAR 2011; 73-78.