Text Extraction from Images for Speech Synthesis

Utkarsha D. Raikar[1] Alka P. Tank[2] Varsha A. Redekar[3] Bhagyashree N. Sawant[4]
[1,2,3,4]K. J. Somaiya College of Engineering, Mumbai, Maharashtra, India

Abstract— We usually come across images that have some text written on them. These may include images of certain signboards that have the name of the destination, or text written on certain billboards. Also, some T-shirts may have text written on them. Other examples include menu cards, business cards, resumes, book covers, etc. The text may have a single font or multiple fonts, the font size may vary character wise or word wise. Also, the text can be written in multiple colors and the background colors can also vary. There may be certain objects in the image other than the text. The aim is to handle all these factors and extract only the text from the image. The text then needs to be converted into speech, which will be the output of the implementation. We currently have certain applications and software's that can read out text from a PDF document. Most such documents have a plain white background and black-colored text. Most images do not come under the category of such plain images. The proposed system deals with those images that have the above mentioned factors and complexity. The proposed system can be useful in multiple applications, most important of which being robot eyes. It can also be used in software's that scan resumes and business cards to shortlist people based on certain keywords found in these documents. Coupled with an image acquisition module, the proposed system can be useful to blind people as well.

Key words: Speech Synthesis, Text Extraction

I. INTRODUCTION

The aim is to identify and extract text from images and read aloud the words detected. The images may range from simple to complex. The complex ones generally have letters and words written in multiple fonts, multiple sizes, varying colors, varying contrast between the foreground and background, and certain components and objects other than the text. All these issues are handled to extract only the text from the image.

A certain amount of pre- processing is done on the image. After that, the connected components are identified and bounding boxes are drawn around them. At this point, bounding boxes will be drawn around the non-text entities as well. This is followed by a MATLAB code to remove the bounding boxes around such connected components and retain those around the textual part only. The font size, color information and contrast parameters are handled in this code. After this step, we have the textual part of the image which is fed into the Optical Character Recognition (OCR) engine. The output is received into a simple text file. This text file is given as input to the speech APIs and finally the words are read out. A major issue to be handled here is dealing with multiple fonts. The OCR engine is trained to handle multiple fonts. The fonts majorly dealt with in the implementation are the AR Berkley font, the Harlow Solid Italic font and the Times New Roman font. The work can be further extended to handle a wider range of fonts and even handwritten text.

II. LITERATURE SURVEY

Text extraction from images is a very challenging approach. There are various proposed methodologies available for extracting the text from images. Each method differs based on the characteristics of an image like text of different fonts, different size, orientation and its background (simple or complex). Some of the proposed approaches are described below.

Saluja et al. [1] have presented a text extraction from colored images approach in which first a colored image is converted to grayscale image and then noise is removed for further processing of the image. Secondly, text detection is done using connected component labeling and generating bounding boxes around them. Several features like height, inter-character distance, aspect ratio and occupation ratio are used to separate the text regions. Finally, the non-textual regions are removed by eliminating the other regions of the bounding box.

Pan et al. [2] proposed a hybrid text detection approach by using region-based and connected component based methods. Their system consists of three stages – pre-processing, connected component analysis and text grouping. In pre-processing, a text region detector is used to detect text and generates the candidate text components. In connected component analysis, a conditional random field model is used to filter out non-text regions. At the last stage, a learning based minimum spanning tree algorithm is used for grouping the text into lines or words.

Sumathi et al. [3] have discussed about the various schemes proposed for text extraction from images. They have provided the comparison of several existing methods proposed by various researchers according to different criteria – document text images, scene text images, caption text images and heterogeneous text images. They have also provided the performance analysis of each algorithm proposed by the respective authors and also the benefits of these algorithms. Each technique has its own advantages and it depends on the application which algorithm is required to be used.

Gllavata et al. [4] have presented the text extraction method as several tasks- text detection, text localization, character segmentation and binarization and text recognition. Text detection identifies the textual parts of an image. Text localization merges the text regions and locates the position of it. Character segmentation will separate the text from background and then the binary image is created of it. This binary image is given as an input to Optical character recognition (OCR) for recognizing the characters and then it generates a text file.

III. PROPOSED SYSTEM

The proposed system works in a sequential manner. The entire system is divided into five different steps. First is the image pre-processing, second is connected component, third is text localization and the output of all this steps given as an input to the fourth step that is OCR. Final step is text to speech which reads out the text using Windows SAPI. The image preprocessing step includes transformation of colored image and adaptive thresholding. Second step deals with connected component detection and bounding box formation. Next is text localization that contains detection of overlapping bounding boxes, removal of irrelevant bounding boxes. The output of above steps is given to the OCR and then to the text to speech step to read out the text using Windows SAPI.

First module of system is text extraction and second is speech synthesis. Text extraction deals with segmenting the image into textual part and non-textual part where textual part contains actual content of image.

A. *Pre-processing:*

Initially the system accepts colored image. It is very complex to handle colored images because variation in intensity of colored pixel is high. Sometimes color information does not help us to identify edges and other features.

1) *Gray Scale Translation:*

Colors images are formed by using three color channels, each of the channels represent the level value. Gray scale image gives us the result of measuring the intensity of light at each pixel in a single channel. The color to gray scale translation is done by using rgb2gray() function.

2) *Adaptive Thresholding:*

The earliest simple method for classifying image foreground and background pixel is the global thresholding technique. This technique is good for images which are well separated with foreground and background intensities. But this technique is not applicable if there is large variation in background intensity hence adaptive thresholding is used to produce better results. An adaptive thresholding algorithm separates foreground from the background with non-uniform illumination. Here we are using adaptivethreshold() function to binarize an image. The function calculates threshold value in regions surrounding each pixel. Each threshold value is the weighted mean of difference between local neighborhood and offset value. It helps to get better result in extracting text from its background

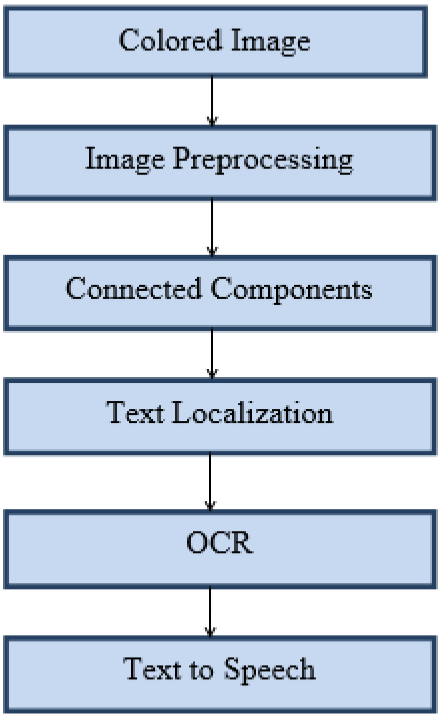


Fig. 1: Proposed System

B. *Connected Components:*

In connected components labeling, an image is scanned pixel by pixel, from top to bottom and left to right and then pixels are grouped into components fixed on pixel connectivity. That is each connected component contains similar pixel intensity values. Once all connected components have been determined, each group is labeled with some values. The labeling of connected components works on graylevel or binary images. For forming connected components use 4 or 8 connectivity. If we assume there is 4 connectivity used while labeling component having one of the pixel is ‘p’, then the labeling of ‘p’ occurs as follows:

- Assign new label to ‘p’, if all neighbors are 0, else
- If one of the neighbor with value 1, then assign its value as label to ‘p’, else
- If more than one of the neighbors having value 1, then assign one from the labels to ‘p’ and equivalent all the neighbors.

1) *Bounding Box Formation:*

Bounding boxes are generated around the connected components which are identified. In non-textual area connected components are also identified. Bounding boxes are the rectangles containing same group of connected components. In the entire image bounding boxes are generated for textual and non-textual region. So, to eliminate non-textual region, text localization is done.

C. *Text Localization:*

Bounding boxes that are formed on connected components contain both textual and non-textual regions. The non-textual region is eliminated by following steps:

1) *Bounding Box Area:*

If the area of the bounding box is greater than ¾ the size of the image then it is eliminated because a character or a word does not occupy such a large space. Also, if the area of the bounding box is less than 50 pixels then it is eliminated because OCR cannot recognize characters of smaller font size.

2) Overlap Ratio:

Two bounding boxes that are text will not overlap each other. The Overlap Ratio decides the elimination of bounding boxes. The overlap ratio is calculated as follows:

Overlap Ratio= Intersection Area/Union Area

Union Area is obtained by adding the two bounding box areas and subtracting their Intersection Area. If the Overlap Ratio is greater than 0.1, then both the bounding boxes are eliminated. This threshold value was taken into consideration by experimenting on various images.

3) Internal Bounding Boxes

Textual Bounding Box contains less than 3 internal boxes e.g. 8 or B, whereas, non-textual bounding box can contain more than 3 internal boxes. If a bounding box contains less than 3 internal boxes then all internal boxes are eliminated otherwise the external bounding box is eliminated.

B1.x>B2.x & B1.x+B1.width-1<B2.x+B2.width-1
B1.y>B2.y & B1.y+B1.height-1<B2.y+B2.height-1

If both the statements are true then B1 is an internal box.

4) Aspect Ratio:

The bounding box that is too long as compared to its width or too wide as compared to its height is a non-textual region. The minimum of the two ratios (width/height ratio, height/width ratio) is taken as Aspect Ratio. If the Aspect Ratio of the bounding box is less than 0.22, then it is eliminated.

5) Occupy Ratio:

The bounding box with very few white pixels does not contain text because for any character there are as many white pixels as there are black pixels. Some characters like ‘l’, ‘1’, ‘I’ or ‘i’ have more white pixels. Occupy Ratio is defined as the ratio of number of white pixels to total number of pixels in the bounding box. After experimenting with different values a threshold was obtained. If the Occupy Ratio is less than 0.2, then the bounding box is eliminated.

6) Bounding Box Alignment:

Textual Bounding boxes lie on the same line. E.g. ‘o’ and ‘n’ are on the same line whereas, ‘y’ and ‘o’ lie on different lines. The minimum height of the bounding boxes (i.e. ‘o’) is considered as the maximum difference allowed between the two lines.

Textual Bounding box have uniform height. E.g. ‘o’ and ‘n’ have the same height whereas, ‘h’ and ‘o’ have different heights. The minimum height of the bounding boxes (i.e. ‘o’) is considered as the maximum difference allowed in their heights.

The inter-character distance between two characters is uniform. If the inter-character distance between two bounding boxes is greater than 20 pixels then they are eliminated.

If the neighboring bounding box is on the same line, having equal height and uniform inter-character distance then the bounding box is not eliminated.

7) Average Height:

The average height of the bounding boxes is calculated. If the height of a certain bounding box differs than the average height to a large extent, then the bounding box is eliminated.

After performing 7 text localization steps, all pixels are made black except the contents of the remaining bounding boxes which contain text. The image containing only text is sent to the OCR for Recognition.

D. OCR:

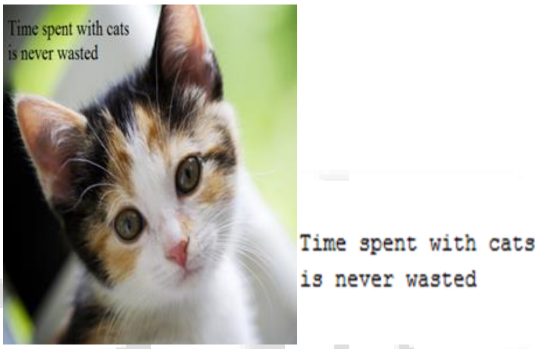
The steps to train the OCR engine for a particular font can be briefly explained as follows:-

- Obtain a document with all the characters and numbers written in a particular font to be trained in the pdf format.
- By executing commands on the Command Prompt, the .box file and tiff files are obtained.
- The box file is then loaded into the jTessBoxEditor and manipulated to correctly identify all characters.
- Again, after executing commands on the Command Prompt, the tesseract is executed.
- The mf training, shape clustering, and cn training is done.
- A .trained data filed is obtained which is saved into Tesseract OCR engine.
- The newly trained font can now be recognized.

E. Text to Speech:

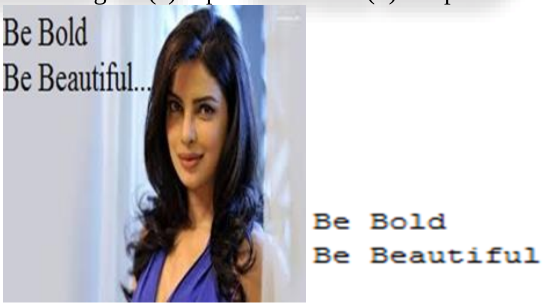
The output of OCR is in the form of speech. The text to speech function reads the text string by calling the Microsoft’s Windows Speech API.

IV. RESULTS



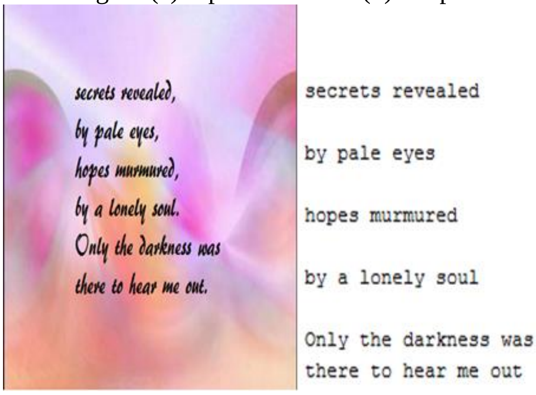
```
Time spent with cats
is never wasted
```

Fig. 2: (a) Input (b) Output



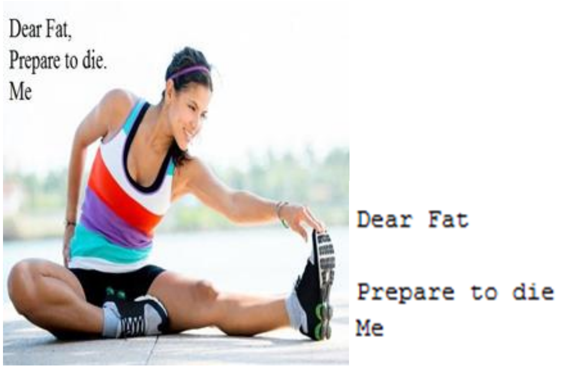
```
Be Bold
Be Beautiful
```

Fig. 3: (a) Input (b) Output



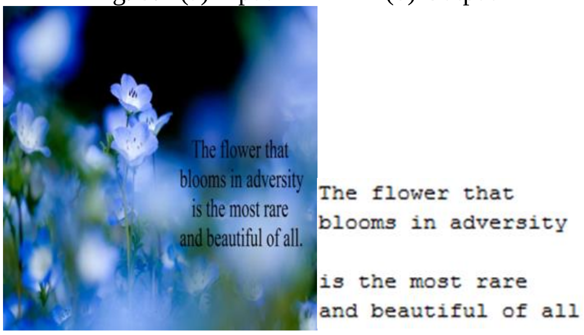
```
secrets revealed

by pale eyes

hopes murmured

by a lonely soul

Only the darkness was
there to hear me out
```

Fig. 4: (a) Input (b) Output



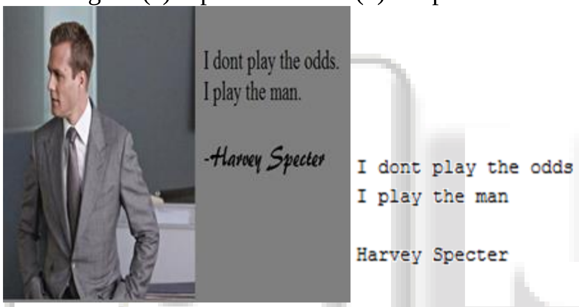
```
Dear Fat

Prepare to die
Me
```

Fig. 5: (a) Input (b) Output



```
The flower that
blooms in adversity

is the most rare
and beautiful of all
```

Fig. 6: (a) Input (b) Output



```
I dont play the odds
I play the man

Harvey Specter
```

Fig. 7: (a) Input (b) Output

V. Conclusion and Future Work

A connected-component approach is used to detect text in an image. The proposed system works on 1) Images having low background-foreground contrast 2) Images with different font size 3) Images where background doesn't interfere with text. 4) Images where background slightly interferes with text. 5) Images with different fonts viz. Times New Roman, Calibri, Arial, Microsoft Sans Serif, Harlow solid italic and AR Berkley

Although the methodology used gives good results, further improvements are needed. The system can be extended to support more fonts by training OCR. Also, future work can be carried out on how to detect text in complex images. The system can be customized/extended as per various applications in robot eye, menu card reader, business card information retriever, vehicle number detection, resume screening, blind-people aid etc.

Acknowledgement

We would like to thank K.J. Somaiya College of Engineering for allowing us to undertake this project. Also, we would like to thank Prof. Uday Joshi for guiding us throughout our project, without whom this challenging task would have been impossible.

References

[1] Shivani Saluja, Tushar Patnaik, Tanvi Jain, "Text Extraction and Non Text Removal from Colored Images", IJCA, Volume 44- No.22, April 2012.

[2] Yi-Feng Pan, Xinwen Hou and Cheng-Lin Liu, "A Hybrid Approach to Detect and Localize Texts in Natural Scene Images", IEEE Transactions on image processing, Vol. 20, No. 3, March 2011.

[3] C.P. Sumathi, T. Santhanam and G. Gayathri Devi, "A Survey on various Approaches of Text Extraction in Images", IJCSES Vol. 3, No. 4, August 2012.

[4] Julinda Gllavata, Ralph Ewerth and Bernd Freisleben, "A Text Detection, Localization and Segmentation System for OCR in Images", Proceedings of IEEE sixth International Symposium, pp. 310-317, December 2004.

[5] Jisha Gopinath, Aravind S, Pooja Chandran, Saranya S S, "Text to Speech Conversion System using OCR", IJETAE, Vol. 5, Issue 1, January 2015.

[6] JaiprakashVerma, Khushali Desai, Barkha Gupta, "Image to Sound Conversion", IJARCSMS, Vol. 1, Issue 6, November 2013.

[7] NitiSyal, Naresh Kumar Garg, "A Study of Text Localization Algorithms for Complex Images", IJIRCCE, Vol. 2, Issue 4, April 2014.

[8] Xiaoqing Liu and Jagath Samarabandu, "Multiscale Edge-based Text Extraction from Complex Images", IEEE International Conference, pp. 1721-1724, July 2006.

[9] Lukas Neumann and Jiri Matas, "A method for text localization and recognition in real-world images", 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010.

[10] Keechul Jung, Kwang In Kim, Anil K. Jain, "Text Information Extraction in Images and Video: A Survey", Pattern Recognition, pp. 977-997, Vol. 37, Issue 5, May 2004.