IJARSCT

ISSN (Online) 2581-9429

International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)

Impact Factor: 7.301

Volume 3, Issue 2, April 2023

Text Reader for Visually Impaired Person using Image Processing/Open-CV

Prof. Sunita Chavan[1], Rutika Wandhekar[2], Shraddha Tole[3], Aishwarya Chaukate[4], Jaishree Kargane[5]

Professor, Department of Information Technology[1]

Students, Department of Information Technology[2,3,4,5]

Smt. Kashibai Navale College of Engineering, Pune, Maharashtra, India

Abstract: *The main issue that visually impaired individuals confront these days is that they are unable to do text recognition on their own, forcing them to rely on others for day-to-day tasks such as reading newspapers, writing letters, referring to books, and so on. This issue may erode their confidence because they are unable to cope on their own. The project's ultimate goal is to assist visually challenged persons with text recognition. This goal is accomplished by creating a module that converts text into speech and speaks it through the provided headphone/speaker. The code is written in Python after importing pytesseract and gtts . For character recognition, this project employs the concept of image processing and the OCR approach.*

Keywords: Image Processing, OCR, Tesseract , LSTM, gTTS

I. INTRODUCTION

The project's ultimate goal is to assist visually challenged persons in recognizing text. When a printed text is displayed in front of the web cam, it must capture the image, extract the text from the image, and read out the text via computer audio or headphone. The code is written in Python after importing pytesseract and gtts. For character recognition, this project employs the concept of image processing and the OCR approach. The main issue that visually impaired persons have these days is that they are unable to do text recognition on their own, forcing them to rely on others for day-to-day tasks such as reading newspapers, writing letters, and referring to websites., referring books etc.. . This problem may reduce their confidence as they could not withstand independently. The project's ultimate goal is to assist visually challenged persons with text recognition. This goal is accomplished by creating a module that converts text into speech and speaks it through the provided headphone/speaker. The image is captured using the system's webcam, and the text is extracted using the built-in application. The text is then identified for words and spoken out loud using headphones or the system's audio. The Python programming language offers PIL (Python Imaging Library), which is used to do simple image operations such as creating thumbnails, resizing, rotating, and converting between different file formats.

II. REVIEW ON LITERATURE

OC based facilitator for the visually challenged. The paper encouraged us to do this project. From this paper we got to know that there are many people who are facing the BVI problem. Also this paper gave us brief idea about OCR technology and the implementation details which were very useful.[2] We found this as reference and have tried to approach in a efficient way. Smart Reader for Visually Impaired People Using Raspberry Pi: This paper propose that how to convert image into text and text into audio. Also this system give complete information about hardware and software implementation for blind reader.

Raspberry Pi Based Reader for Blind People:

The software Implementation and programming along with the details of ocr engine were very useful from this paper. This paper gave the detail information about which engines to be used for image to text conversion , and text to speech.“OCR based automatic book reader for the visually impaired using Raspberry PI”–This paper provided the case study and from this paper we learn to build a system on English language, and we were able to think that in other language can also be done, which we put it in advancement[3]. The system accepts a page of printed text with English

DOI: 10.48175/IJARSCT-9146

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 7.301
Volume 3, Issue 2, April 2023

numerals, scans it into a digital document which is then subjected to skew correction, segmentation, before feature extraction to perform classification. Once classified, the text is read out by a text to speech conversion unit. An innovative, efficient and real-time cost beneficial technique that enables user to hear the contents of text images instead of reading through them as been introduced.[6] It combines the concept of Optical Character Recognition (OCR) and Text to Speech Synthesiser (TTS) in Raspberry pi. Text Image using Raspberry Pi". Optical Character recognition is used to digitize and reproduce texts that have been produced with non computerized system. Digitizing texts also helps reduce storage space[7].

Design and implementation of Automatic Scene text detection and recognition system for visually impaired people has been discussed. Combining different techniques for Text detection and extraction results into accurate and better system than using single technique for overall system. Text recognition is successfully performed using pattern matching technique. After successful recognition, text is converted into audio output. A prototype system to read printed text and hand held objects for assisting the blind people is proposed.[9] To extract text regions from complex backgrounds, novel text localization algorithm based on models of stroke orientation and edge distributions is adopted. An image to speech conversion technique using Raspberry Pi was implemented. Output has been tested using different samples. The algorithm successfully processes the image and reads it out clearly.

III. EXISTING SYSTEM

In the running world there is a growing demand for the users to convert the printed documents in to electronic documents for maintaining the security of their data. Hence the basic OCR system was invented to convert the data available on papers in to computer process able documents, So that the documents can be editable and reusable. The existing system/the previous system of OCR on a grid infrastructure is mostly based on expensive and complex hardware setup. This leads to cost of system and uses become limited based on availability and affordability of blind person.

This prototype system is to read printed text on hand- held objects for assisting blind persons. In order to solve the common aiming problem for blind users, we have proposed a motion-based method to detect the object of interest, while the blind user simply shakes the object for a couple of seconds. The automatic ROI detection and text localization algorithms were independently evaluated as unit tests to ensure effectiveness and robustness of the whole system.

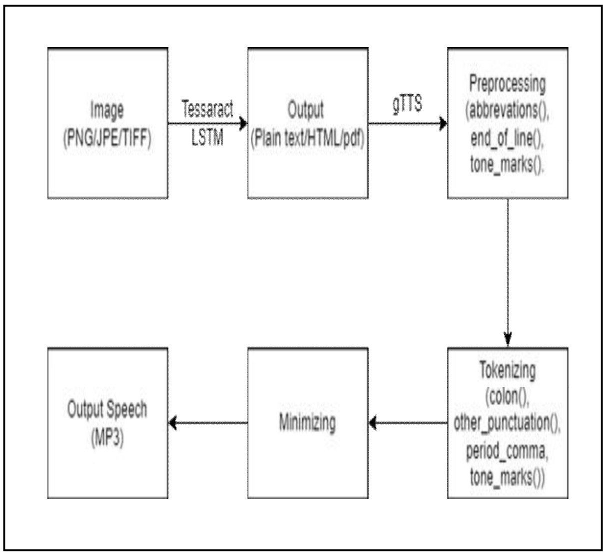

Fig 1. Existing System

A Raspberry Pi board, an ultrasonic sensor and a webcam to recognize the text in the scene. In this, the webcam is focussed on the scene. A video streaming is obtained, from which the images are captured frame by frame. We subsequently evaluated this prototype system of assistive text reading using images of hand-held objects captured by ten blind users in person. Two calibrations were applied to prepare for the system test. First, we instructed blind users to place hand- held object within the camera view. Since it is difficult for blind users to aim their held objects, we employed a camera with a reasonably wide angle.

Then the images are refined in order to eliminate any noise that is present in it. A feature called segmentation is used in order to separate each character from other in the text. Graphical details such as icons or logos, if any, are eliminated. Each obtained character is compared with the datasets that are created as a part of the Tesseract library.

The Tesseract OCR is the most efficient algorithm available that checks for the obtained character in ten dimensions. Once, the character is recognized, it must be made available as an audio output. For this, we use a software called festival. The festival is used to provide the audio output for the recognized character. Apart from these features, an extra feature is added, that enables the blind to know the type of object that he/she interacts with. (a menu, newspaper and the like). An ultrasonic sensor is included as a part of the project, that makes the project obtain characters only within a particular distance.

IV. METHODOLOGY

The suggested system is a software module that accepts input from the user extracts the text content using the code produced, converts the text to speech, and reads it out using the headphone/speaker. Our System can read multiple languages e.g. Hindi , English, Russian etc.This project eliminates the use of the Raspberry Pi board, which is regarded as one of the most significant advantages of the suggested system board. Human communication is primarily conducted through speech and text. To access the information in a text, a person must have vision. Those with low vision, on the other hand, can gather information through voice. This study presents a camera-based assistive text reading system to assist visually impaired people in reading text on captured images.

The proposed idea entails extracting text from a scanned image using Tesseract Optical Character Recognition (OCR), reading the image with the open cv2 provided by the Python library, and converting the text to speech using gtts (Google Text To Speech), a process that allows visually impaired people to read the text. This is a prototype helping visually impaired persons to recognise products in the real world by extracting text from images and converting it to audio. The proposed solution is carried out simply installing software, making it more portable and less expensive. Those who are blind or visually handicapped can use optical character recognition (OCR) equipment to scan printed text and have it read in synthetic speech or saved to a computer file. OCR technology is comprised of three components: scanning, recognition, and text reading. The data we collect or generate is generally raw data, which means it cannot be used directly in applications for a variety of reasons. As a result, we must first examine it, then do the Necessary processing, and then use it modeling and analysis.

4.1 Image Processing

Image processing library primarily focused on real- time computer vision with applications in a variety of disciplines such as 2D and 3D feature toolkits, facial and gesture recognition, human-computer interaction, mobile robotics, object identification, and others. The open CV2 library is used for image processing. PIL is used to do simple image operations such as thumbnail creation, resizing, rotating, and converting between different file formats. The image is loaded directly using the Image class's open () function. This produces an image object including the image's pixel data as well as image information.. The image's format property will report the image format (e.g., png, jpeg), the mode will report the pixel channel format (e.g., CMYK or RGB), and the size will report the image dimensions in pixels (e.g., 400*260). The show() function will display the image in the default application of the operating system. Pillow is one of the most popular and widely used Python libraries for image processing. Pillow is a modified version of the Python Image Library (PIL) that provides a variety of basic and advanced image editing capabilities. It's also the foundation for basic image support in other Python libraries like SciPy and Matplotlib.

4.2 OCR Technique

The electronic or mechanical translation of images of typed, handwritten, or printed text into machine-encoded text, whether from a scanned document, a snapshot of a document, a scene-photo, or subtitle text superimposed on an image, is known as optical character recognition (OCR). It is responsible for extracting text from image files and saving it in a text file. We process the photos and convert them to text here. Now we have the text as a string variable, we can do whatever we want with it.

The OCR engine or OCR software works by using the following steps:

1. **Image acquisition**: A scanner reads documents and converts them to binary data. The OCR software analyzes the scanned image and classifies the areas as background and the dark areas as text.
2. **Preprocessing**: The OCR software first cleans the image and remove errors to prepare it for reading. Seven steps to perform image pre-processing for OC:

- **Normalization:** This process changes the range of pixel intensity values. The purpose of performing normalization is to bring image to range that is normal to sense. OpenCV uses normalize () function for the image normalization.

```
norm_img = np.zeros((img.shape[0], img.shape[1]))
img = cv2.normalize(img, norm_img, 0, 255, cv2.NORM_MINMAX)
```

- **Skew Correction:** While scanning or taking a picture of any document, it is possible that the scanned or captured image might be slightly skewed sometimes. For the better performance of the OCR, it is good to determine the skewness in image and correct it.
- **Image Scaling:** To achieve a better performance of OCR , the image should have more than 300 PPI(pixel per inch). So, if the image size is less than 300PPI, we need to increase it. We can use the pillow library for this.
- **Noise Removal:** This step removes the small dots/patches which have high intensity compare to the rest of the image for smoothing of the image. OpenCV's fast NI means Denoising Coloured function can do that easily.

```
def remove_noise(image):
return cv2.fastNlMeansDenoisingColoured(image , None, 10, 10, 7,15)
```

- **Thinning and Skeletonization:** This step is performed for the handwritten text, as different writers use different stroke widths to write. This step makes the width of strokes uniform. This can be done in OpenCV
  - img = cv2.imread('j.png',0) kernel = np.ones((5,5),np.unit8)
  - erosion = cv2.erode(img kernel, iterations = 1
- **Gray Scale Image:** This process converts an image from other color spaces to shades of Gray. The color varies between complete black and complete white. OpenCV's cvtColor() function perform this task very easily.

```
def get_grayscale(image):
return cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
```

- **Thresholding or Binarization:** This step converts any colored image into a binary image that contains only two colors black and white. It is done by fixing a threshold (normally half of the pixel range 0-255, i.e., 127.The pixel value having greater than the threshold is converted into a white pixel else into a black pixel. To determine the threshold value according to the image Otsu's Binarization and Adaptive Binarization can be a better choice. In OpenCV, this can be done as given.

```
def thresholding(image):
returncv2.threshold(image,0,255,cv2.THRESH_BINARY+c 2. THRESH_OTSU)[1]
```

3. **Text Recognition:** The two main types of OCR algorithms or software processes that an OCR software uses for text recognition are called pattern matching and feature extraction.

- **Pattern matching**: Pattern matching works by isolating a character image, called glyph, and comparing it with a similarly stored glyph. Pattern recognition works only if the stored glyph has a similar font and scale to the input glyph. This method works well with scanned images of documents that have been typed in a known font.
- **Feature extraction**: Feature extraction breaks down or decompose the glyphs into feature such as lines, closed loops, line direction, and line intersections. It then uses these features to find the best match or the nearest neighbor among its various stored glyphs.

4. **Postprocessing**: Post-processing reduce the number of errors. Post-processing corrects one sentence at the time. ⅄ In Post-processing we want to replace the input sequence of characters with another sequence of characters that is graphically similar and form the likeable sentence of the give language.

General form of model:

P (O, S) = P (O | S) * P(S)

O – output of the OCR system

S – candidate sequence of character

P (O | S) – probability, that the sequence S will be recognize as O by OCR - corresponds to optical similarity between O and S

– usually denoted as error model

P (S) – probability of S – corresponds to the likeableness of the sequence S – this quantity should have greater value for well- formed sentences - denoted as language model.

5. **Google Text To Speech Converter:** gTTS(Google Text -to-Speech), a python library and command-line interface for interacting with Google Translate's text – to – speech API. In python, there are numerous Interface for converting text to speech. The Google Text to Speech API, often known as the gTTS API, is one of these APIs. gTTS is a simple utility that turns the text entered into audio that may be saved as an mp3 file. English, Hindi, Tamil, French, German, and many other languages are supported via he gTTS API. The speech can be delivered in either of two audio speed: fast or slow. But, as of the most recent update, changing the voice of the generated audio is no longer available.

Module gTTS (gtts.gTTS):

An interface to Google Translate's Text-to -Speech API.

Parameters

[a]text (string) -The text to be read.

[b]tld (string - Top-level domain for the Google Translate host,

i.e https://translate.google.<tld>. Different Google domains can produce different localized 'accents' for a given language. This is also useful when google.com might be blocked within a network but a local or different Google host (e.g. google.cn) is not. Default is com.

[c] Lang (string, optional) – The language (IETF language tag) to read the text in. Default is en.

[d]Slow (bool, optional) – Reads text more slowly. Defaults to False.

[e]lang_check(bool, optional) – Strictly enforce an existing lang, to catch a language error early. If set to True, a ValueError is raised if lang doesn't exit. Setting lang_check to False skips Web requests (to validate language) and therefore speeds up instanciation. Default is True.

[f]Pre_processor_funcs(list) – A list of zero or more functions that are called to transform (Pre_process) text before tokenizing

PRE-PROCESSOR

Function that takes text and return text. Its goal is to modify text (for example correcting pronounciation), and/or to prepare text for proper tokenization (for example ensuring spacing after certain character).You can pass a list of any function to **gtts.tts.gTTS's** pre_processor_funcs attribute to act as pre-processor (as long as it takes a sitting and returns a string).

By default, **gtts.tts.gTTS** takes a list of the following pre- processors, applied in order:

pre-processors.tone_marks, pre-processors.end_of_line, pre-processors.abbreviations, pre-processors.word_sub

TOKENIZER

Function that takes text and returns it split into a list of tokens(strings). In the gTTS context, its goal is to cut the text into smaller segments that do not exceed the maximum character size allowed for each TTS API request, while making the speech sound natural and continuous. It does so by splitting text where speech would naturaly pause (for example on" .") while handling where it should not (for example on"10.5" or "U,S,A").Such rules are called tokenizer cases, which it takes a list of. By default, **gTTS** takes

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
Impact Factor: 7.301
Volume 3, Issue 2, April 2023

The **gtts.tokenizer.core.Tokenizer's gtts.tokenizer.core**. **Tokenizer.**run(), initialized with default tokenizer cases: Tokenizer ([tokenizer_cases.tone_marks, tokenizer_cases.period_comma, tokenizer_cases.other_punctuation]).run
Tokenizer case: Function that defines one of the specific cases used by **gtts.tokenizer.core.Tokenizer**. More specefically, it returns a regex object that describes what to look for a particular case.**gtts.tokenizer.core.Tokenizer** then creates its main regex pattern by joining all tokenizer cases with"|".

MINIMIZING

The Google Translate text-to-speech API accepts a maximum of 100 characters.If after tokenization any of the tokens is larger than 100 characters, it will be split in two:
[a]On the last space character that is closest to, but before the 100th character;
[b]Between the 100th and 101st character if there's no space.

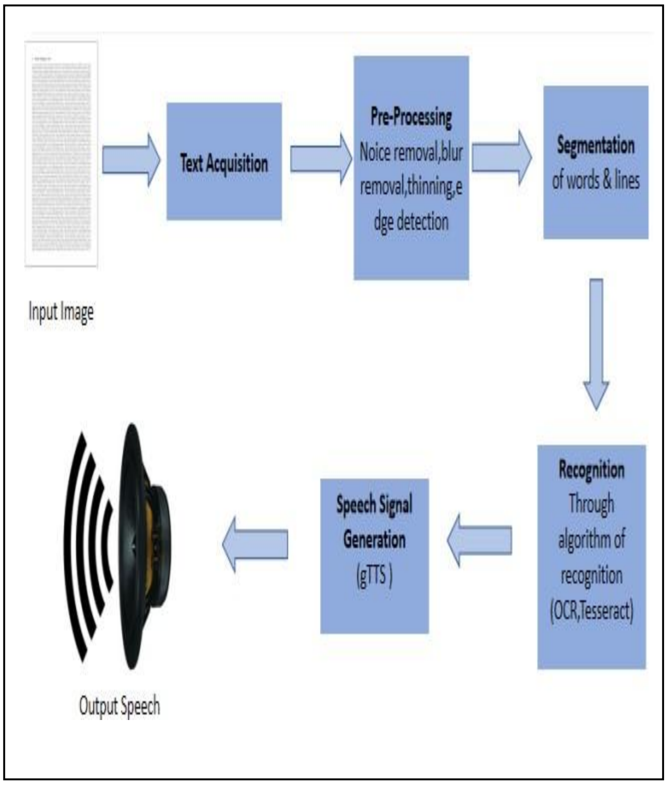

Fig 1. Overall Architecture Diagram

V. CONCLUSION

As a result, the suggested system's ultimate goal has been met. This technology can translate speech to text and serve as a text reader for the visually handicapped. The text is displayed in front of the system's webcam or in front of the integrated camera. The Image Processing technique is used to examine the acquired image. OCR (Optical Character Recognition) separates and identifies the words in the image to recognise the characters. Consequently, the words acquired are transformed to speech using GTTS (google text to speech converter). Lastly, the collected text is read out via the speaker or headphones. As a result, visually handicapped persons benefit from the environment.

REFERENCES

[1]. Bindu Philip and r. d. Sudhaker Samuel 2009 "Human machine interface- a smart ocr for the visually challenged" International journal of recent trends in engineering, vol no.3, November.

[2]. K Nirmala Kumari, Meghana Reddy J [2016]. Image Text to Speech Conversion Using OCR Technique in Raspberry Pi. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering Vol. 5, Issue 5, May 2016.

[3]. V. Ajantha devi, dr. Santhosh baboo "Embedded optical character recognition on tamil text image using raspberry pi" international journal of computer science trends and technology (ijcst)" volume 2 issue 4, jul-aug 2014

[4]. Jaiprakash verma, khushali desai "Image to sound conversion" International journal of advance research.

[5]. R. Mithe, S. Indalkar and N. Divekar. " Optical Character Recognition" International Journal of Recent Technology and Engineering (IJRTE)", ISSN: 2277- 3878,Volume-2, Issue-1,March 2013.

[6]. Character Detection and Recognition System for Visually Impaired People by Akhilesh A. Panchal, Shrugal Varde, M.S. Panse .

[7]. Giudice, N. A., & Legge, G. E, Blind navigation and the role of technology. In A. Helal, M. Mokhtari & B. Abdulrazak (Eds.), Engineering handbook of smart technology for aging, disability, and independence (pp. 479- 500): John Wiley & Sons.

[8]. Sunil Kumar, Rajat Gupta , Nitin Khanna, Santanu Chaudhury and Shiv Dutt Joshi, Text Extraction and Document Image Segmentation Using Matched Wavelets and MRF Model, IEEE Transactions on Image Processing (Volume: 16 , Issue: 8 , Aug. 2007) 2117 - 2128.

[9]. Ray Kurzweil K Reader Mobile User Guide, knfb Reading Technology Inc. (2008). [Online]. Available: http://www.knfbReading.com

[10]. Ms.Athira Panicker Smart Shopping assistant label reading system with voice output for blind using raspberry pi, Ms.Anupama Pandey, Ms.Vrunal Patil YTIET, University of Mumbai ISSN: 2278 - 1323 International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol. 5, Issue 10, Oct 2016 2553 www.ijarcet.org.

[11]. Raspberry pi 3b,Optical Character Recognition (OCR), Text to speech (TTS), Pi-camera, Speaker, Headphone.

[12]. Gurav, Mallapa D., et al."B-LIGHT:A Reading aid for the Blind People using OCR and OpenCV. "International Journal of Scientific Research Engineering &Technology(IJSRET),ISSN(2017).

[13]. Goel, Anush, et al. "Raspberry Pi Based Reader for Blind People." International Research Journal of Engineering and Technology 5.6.

[14]. Chaudhari, Harshada. "Raspberry Pi technology: a review." International Journal of Innovative and Emerging Research in Engineering 2.3.

[15]. V. Ajantha devi, dr. Santhosh baboo "Embedded optical character recognition on tamil text image using raspberry pi" international journal of computer science trends and technology (ijcst) volume 2 issue 4, jul-aug 2014.