

A Dynamic Web Content Reader Application for Low Vision Patients using Python

R. Seetha Kannagi¹ P. Arunkumar² D.N. Narendran³ M.Santhosh⁴ C.Sakthivel⁵

¹Assistant Professor ^{2,3,4,5}Student

^{1,2,3,4,5}Department of Computer Science & Engineering

^{1,2,3,4,5}The Kavery Engineering College, Salem, India

Abstract— With six meters highest visibility and 20 degrees maximum wide view, people who suffer from low vision are unable to see words and letters in ordinary newsprint. This fact makes the reading process becomes difficult that can disturb the learning process and slow the patient's intellectual development. Therefore, an application is needed to help them read easier. The challenges faced by visually impaired people in reading printed text in their day-to-day life are often not well understood. This project is based on a prototype which helps the user to listen to the contents of the live websites in Tamil, English, and Hindi etc. It involves the extraction of content from the websites and translates audio output in the languages mentioned above. This is done using python libraries with the concept of Natural Language Processing, Web Mining, Text-to-Speech. It is portable and easy to use thus providing a better reading experience to the visually challenged people as enjoyed by their sighted peers.
Key words: Text-to-speech; Python Libraries; Natural Language Processing; Web Mining; Visually Impaired

I. INTRODUCTION

The complexity of the existing Braille system for the visually impaired people is that it requires the text to be translated to Braille literature. Translating a book or a document into Braille literature is a complex, time consuming and more expensive process. Day-to-day information cannot be translated into Braille literature. To ease this process of reading for the visually impaired people, this prototype has been proposed. Using this prototype, text information can be converted into its equivalent audio format. This is done using python libraries with the concept of Natural Language Processing, Web Mining, Text-to-Speech. It is portable and easy to use thus providing a better reading experience to the visually challenged people as enjoyed by their sighted peers.

Remaining section describe about Existing System and Proposed System.

II. EXISTING SYSTEM

The Existing system done by integrating camera module and speakers to Raspberry Pi 3 model B which is a credit card sized single board computer. Also, two other software called Tesseract and TTS engine (Text to Speech synthesizer) is installed to the Raspbian OS. Principally, the camera captures the image and stores it as an image file with a .jpg extension. The OCR engine converts it from image file to text file by extracting the numbers and characters of the language. The file is given as an input to a python program which gives a translated speech output using Google text to speech engine.

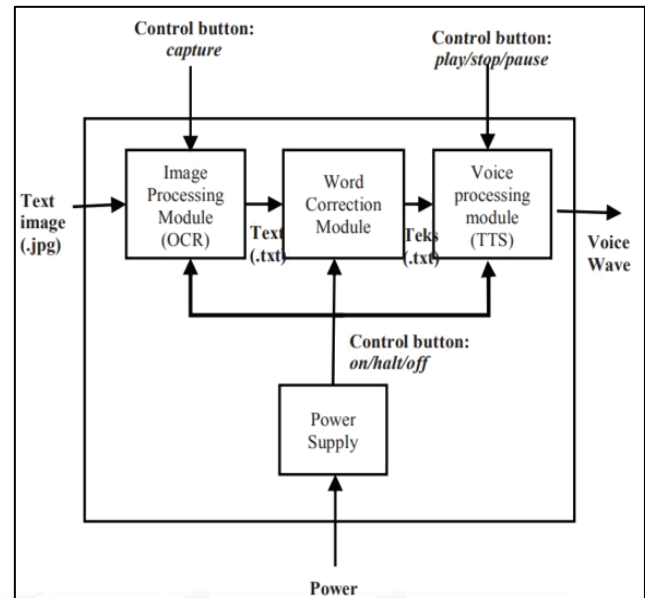


Fig. 1: Block Diagram of Text-To-Speech Device

Text-to speech device consisted of three main modules, the image processing module, word correction module, and voice processing modules. Image processing module sets the object position, focus and illumination camera, taking pictures, and converting the image into text. Word correction module makes corrections to the output image processing module to improve accuracy by matching with Indonesian dictionary. Voice processing module changes the writing into sound and process it with specific physical characteristics so that the sound can be understood.

The input image captured by the camera has a size of 5 MPI (2592 x 1944 pixels) or 215 ppi (pixels per inch). Based on the specifications of the Tesseract OCR engine, the minimum character size that can be read is 20 pixels uppercase letters. Tesseract OCR accuracy will decrease with the font size of 10pt.

The image is taken by the user via GPIO pin that are connected to the tactile key using interrupt function. Furthermore, the picture is taken by using raspistill program with sharpness mode to sharpen the image. The resulting image has a .jpg format with a resolution of 2592 x 1944 pixels.

Word correction module gets input from the image processing module in the form of text from the image processing module. Image processing module can't define truth or falsity of the word output, so that the correction module of this word, correction for whole words output from the image processing requires module. In order to improve the accuracy of the output image processing module, to design the word correction module.

The main *disadvantage* of the existing system is highly expensive. And the portability of the Device is very

difficult. The Existing system using only converts the printed content. Also, the capturing paper size is A4 only. The Device Takes more time to convert the scanned Documents. If any problem occurred in the specific part of the device, it will affect the total setup.

III. PROPOSED SYSTEM

The complexity of the existing Braille system for the visually impaired people is that it requires, the text to be translated to Braille literature. Translating a book or a document into Braille literature is a complex, time consuming and more expensive process. Day-to-day information cannot be translated into Braille literature. To ease this process of reading for the visually impaired people, this prototype has been proposed. Benefits of the visually impaired are not necessary to learn any new language. The visually impaired can easily understand because different languages converted into native languages and hearable audio output. In the Proposed system, all external devices are avoided. In the Existing System, only can convert specific languages like Tamil and English. But in the proposed system, establish multiple conversion of languages like English, Arab, Japanese, French etc. In the Proposed System, convert the dynamic web content using web mining and Natural Language Processing concepts. It is easy to use and portable. All kind of people can use this application. And content can easily understand by the visually impaired persons also other language peoples are able to understand the different languages in our native languages.

IV. DESIGN & IMPLEMENTATION

Our designed Project is called Web Reader, a simple application with the text to speech functionality. The system was developed using web mining and NLP concepts using python.

The application is divided into three modules - the main application module which includes the basic GUI components which handles the basic operations of the application such as input of parameters for conversion either via file or direct keyboard input.

The second module, Requesting the targeted website using urllib to fetch by using GET method and read the content as object (page as Object). Separate content from DOM object by using bs4 and find the targeted object.

The third module, Converting DOM object as a string. Parse string to the textblob. Finally Set output language & store in array (language conversion) and loop the string converted to audio.

Web Reader converts Dynamic web content to Audio output in native Language in the application. It provides multiple language conversion into native language. Web Reader loads the content from the website and the Language Conversion procedure starts automatically.

Web Reader contains an exceptional function that gives the user the choice of saving its already converted text to any part of the local machine in an audio format; this allows the user to copy the audio format to any of his/her audio devices, so that they can hence forth treat it as an audio book.

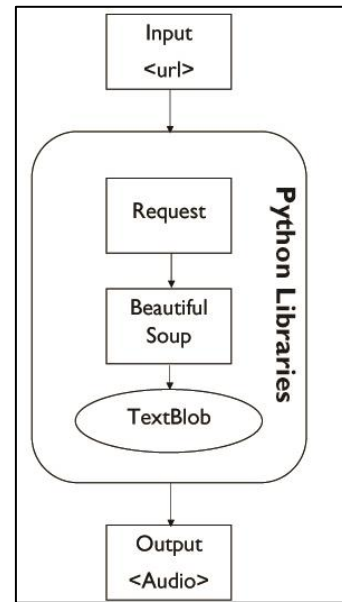


Fig. 2: Block Diagram of Dynamic Web Content Reader

The fig.2 is the block diagram of the Web Reader Application. This diagram shows the flow of the Web Reader Application. The user can give the prefixed URL input to the Web Reader Application. And the python libraries like requests, bs4, textblob are processed the URL and retrieve the specified web content from the given particular URL. And the retrieved web content is converted into the audio(.mp3) format in native language Tamil.

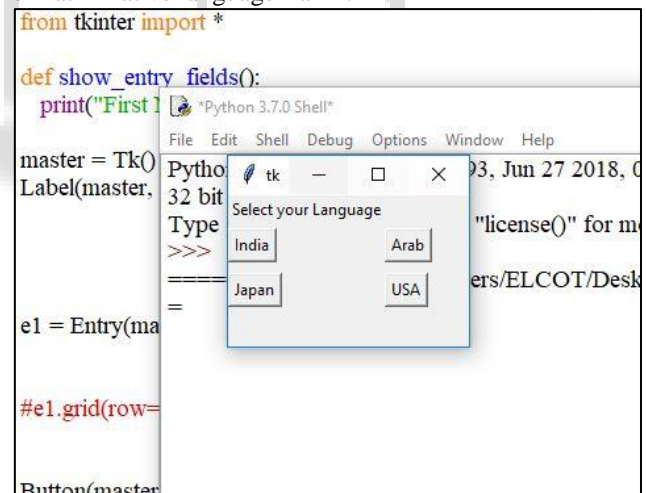


Fig. 3: sample output for Web Reader Application

The python library requests, bs4, textblob, gtts, pydub, lxml, os are used in our project. The detailed explanation of each library is followed by,

A. Requests

Requests will allow you to send HTTP/1.1 requests using Python. With it, you can add content like headers, form data, multipart files, and parameters via simple Python libraries. It also allows you to access the response data of Python in the same way.

B. Bs4

BeautifulSoup 4. BeautifulSoup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating,

searching, and modifying the parse tree. It commonly saves programmers hours or days of work

C. Textblob

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

D. gtts

gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate's text-to-speech API. Writes spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout. It features flexible pre-processing and tokenizing, as well as automatic retrieval of supported languages.

E. pydub

Manipulate audio with a simple and easy high level interface.

F. lxml

The lxml XML toolkit is a Pythonic binding for the C libraries libxml2 and libxslt. It is unique in that it combines the speed and XML feature completeness of these libraries with the simplicity of a native Python API, mostly compatible but superior to the well-known ElementTree API.

G. OS

The OS module in Python provides a way of using operating system dependent functionality. The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux.

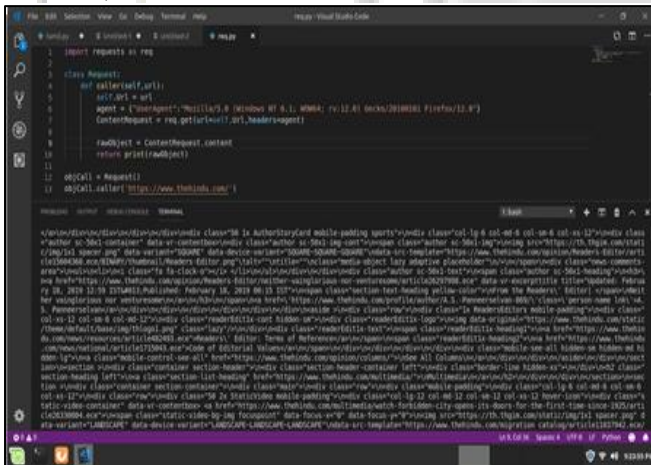


Fig. 4: sample output for requests library

The Requests library used to Requesting the targeted website using urllib to fetch by using GET method and read the content as object (page as Object). Separate content from DOM object by using bs4 and find the targeted object.

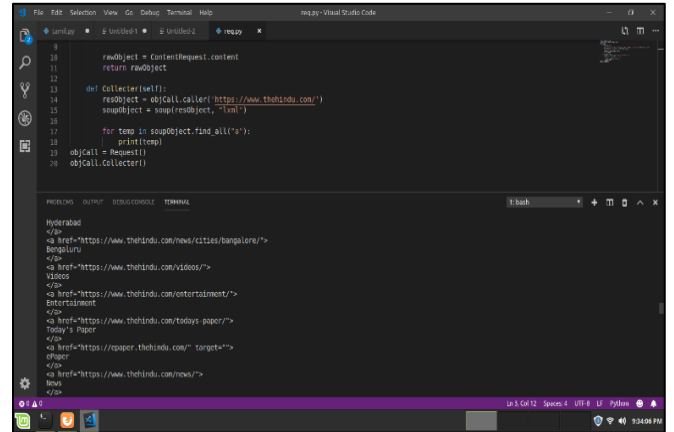


Fig. 5: sample output for bs4 library

The bs4 library is used to Converting the DOM object as a string. The converted string as given to the textblob library it's convert the text input to audio(.mp3) format output.

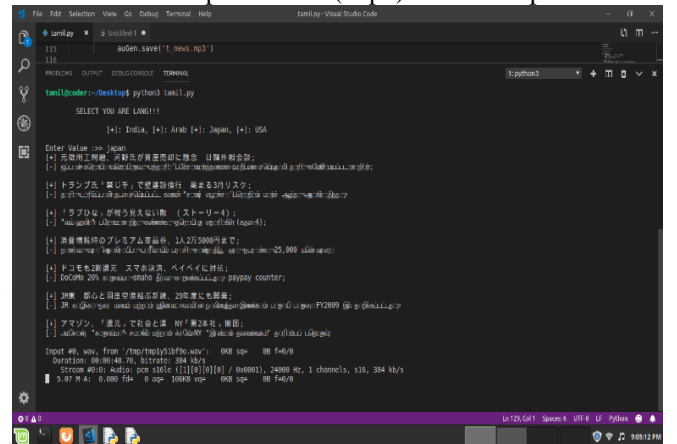


Fig. 6: sample output of the Web Reader Application

Initially four languages (India-English, USA-English, Arab, Japan) are converted into native languages and in future many language's can be converted into native language.

V. CONCLUSION

Web Reader Application is a flexible robust dynamic growing aspect of modern computer era and it is increasingly playing a more significance role in the way we interact with the system and interfaces which is based on platform independent concept. We have identified the various operations and processes involved in text to speech synthesis. We have also developed a very simple and attractive graphical user interface which allows the user to choose the language in the application. Our system interfaces with a text to speech engine developed for Arabic, Japanese, english. In future, we plan to make efforts to create engines for conversion of one language to other make text to speech technology more accessible to a wider range. Accuracy of the software is excellent in the context of its ability to work in real-life environment. We also have plans to make it a web based real-time synthesis system, so that its uses can get more expanded.

REFERENCES

- [1] Dr. I S Akila, Akshaya B, Deepthi S, Sivadharshini P, “A Text Reader for the Visually Impaired using Raspberry Pi”, International Conference on Computing Methodologies and Communication (ICCMC 2018), IEEE conference record # 42656; IEEE Xplore ISBN:978-1-5386-3452-3
- [2] Annisa Istiqomah, Arrahmah, Aulia Rahmatika, Samantha Harisa, Hasballah Zakaria, Richard Mengko, “Text to Speech Device for Patients with Low Vision”, 2015 4th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME) Bandung, November 2-3, 2015
- [3] P. S. Shetake, S. A. Patil, P. M. Jadhav, “Review of Text To Speech Conversion Methods”, International Journal of Industrial Electronics and Electrical Engineering, vol. 2(8), pp. 29-35, 2014
- [4] P. Khilari, V. P. Bhope, “A Review on Speech To Text Conversion Methods”, International Journal of Advanced Research in Computer Engineering & Technology, vol. 4(7), pp. 3067-3072, 2015
- [5] A. Joshi, D. Chabbi, M. Suman, S. Kulkarni, “Text To Speech System for Kannada Language”, International Conference on Communications and Signal Processing, 2015
- [6] H. M. Htun, T. Zin, H. M. Tun, “Text To Speech Conversion using Different Speech Synthesis”, International Journal of Scientific & Technology Research, vol. 4(7), pp. 104-108, 2015
- [7] S. Patil, M. Phonde, S. Prajapati, S. Rane, A. Lahane, “Multilingual Speech and Text Recognition and Translation using Image”, International Journal of Engineering Research & Technology, vol. 5(4), 85-87, 2016
- [8] D. B. K. Kamesh, S. Nazma, J. K. R. Sastry, S. Venkateswarlu, “Camera based Text to Speech Conversion, Obstacle and Currency Detection for Blind Persons”, Indian Journal of Science and Technology, vol 9(30), 1-5, 2016