|  |
| --- |
| **PROJECT REPORT**  **ON** |
| **CARLISLE - THE SMART READER** |
| **SUBMITTED BY**  SHEENA (04001012015)  GUNJAN (06301012015)  PRACHI (06401012015)  MONIKA (06501012015)  MUDITA (07501012015)  **INDIRA GANDHI DELHI TECHNICAL UNIVERSITY FOR WOMEN** |
| **ACADEMIC YEAR 2017-2018**  **PROJECT GUIDE**  Prof. SRN Reddy |
|  |
|  |

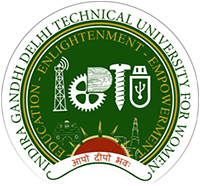
|  |
| --- |
|  |

**ACKNOWLEDGEMENT**

We are really greatful because we managed to complete our project titled **‘CARLISLE - THE SMART READER’** within the given time under the guidance of **Sir Bramha Tripathi** and **Ma’am Ritu Gupta**. This project cannot be completed without the effort and the co-operation of the group members : Sheena Mattu, Gunjan, Prachi, Monika and Mudita Nimje. We also sincerely thank our lecturer **Prof. SRN Reddy** for the guidance and encouragement in finishing this project and also for teaching us this course.

**INDIRA GANDHI TECHNICAL UNIVERSITY FOR WOMEN**

**KASHMERE GATE, DELHI-06**



**CERTIFICATE OF THE GUIDE**

This is to certify that project entitled ‘CARLISLE – THE SMART READER’ is a bonafide work of Sheena Mattu (04001012015), Gunjan (06301012015), Prachi (06401012015), Monika (06501012015) and Mudita Nimje (07501012015) carried out in partial fulfilment for the award of degree of Bachelor of Technology (Computer Science and Engineering) under my guidance. This project work is original and not submitted earlier for the award of any degree/diploma or associate ship of any other University/Institution.

**Place :** Delhi **Signature of the Guide**

**Date :**  (Prof. S.R.N. Reddy)

**ABSTRACT**

This report proposes a novel implementation of an Optical Character Recognition (OCR) based smart reader. There is a need for a portable text reader that is affordable, portable and readily available to the community. We propose a camera based framework built on the Raspberry Pi, integrated with Image processing algorithms, OCR and Text-to-Speech (TTS) synthesis module. The camera module is used to capture an image of the printed text, and the image is then subject to preprocessing before being fed into the OCR. The preprocessing stage includes binarization, de-noising, deskewing, segmentation and feature extraction. This report addresses the integration of a complete Text Read-out system. The OCR used in this project is Pi Tesseract and the TTS employed is Espeak.

**INDEX**

1. Introduction
2. System Design
3. Working Principle
4. System Hardware Components
5. System Software Design
6. Project Features
7. Implementation
8. Testing
9. Scope and Future Work
10. Conclusion
11. References

**INTRODUCTION**

OCR is the electronic conversion of images into machine encoded text. It provides alphanumeric recognition of printed or handwritten characters.

This project addresses a complete text read out system built on an embedded framework. The proposed idea uses the principle of a camera based assistive device implemented on a Raspberry Pi 3 board. The integrated system consists of a camera module / web camera, Tesseract Optical Character Recognition Engine, Pico Text to Speech Engine, Speakers/ Headphones and Computer Vision software for image processing. The camera is an input device which feeds the required image for digitization.The processed image serves as an input to the OCR, which digitizes the image and performs character recognition. Finally, the Text to Speech engine reads out the text to the user.

This project will be a smart text image-to-audio converter system which will perform the conversion in real time. This product aims to design an efficient system for aiding the teachers and students. While aiding the user in reading and daily life necessities, the system will remain efficient and easy to use.

The Smart Reader is also useful for the Visaully Impaired People. According to the World Health Organization (WHO), around 285 million people around the world are estimated to be visually impaired, out of which 90% live in developing countries. Thus there is a pressing need to develop a reader that is affordable to the low income sections of the society.

**SYSTEM DESIGN**

**WORKING PRINCIPLE :**

The image of a printed text is captured through the Raspberry Pi camera module. This image is subject to pre-processing which includes correcting skew angles, sharpening of image, thresholding and segmentation. The processed image is sent to the TTS synthesizer.

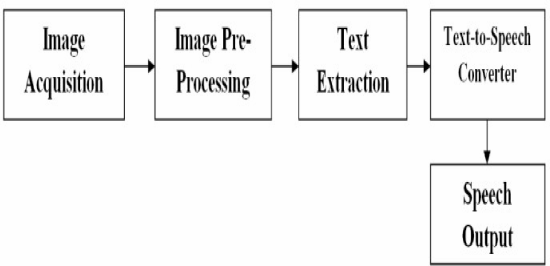


Fig 1. Working Principle

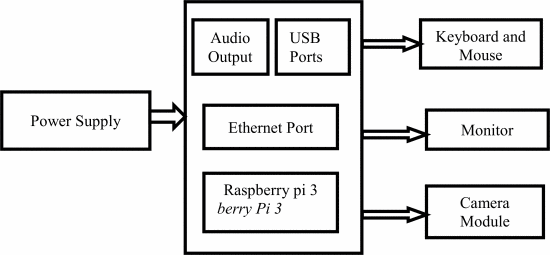


Fig 2. System Hardware Architecture

**SYSTEM HARDWARE COMPONENTS :**

1. **Raspberry Pi 3** : The Raspberry Pi is a credit card sized single-board computer or an SOC (System On Chip in which CPU (Central Processing Unit), GPU (Graphics Processing Unit), USB controller, RAM and so on, is compressed down into one tidy package) developped by Raspberry Pi Foundation. The advantage of having an SoC is that it reduces the cost and increases the performance.

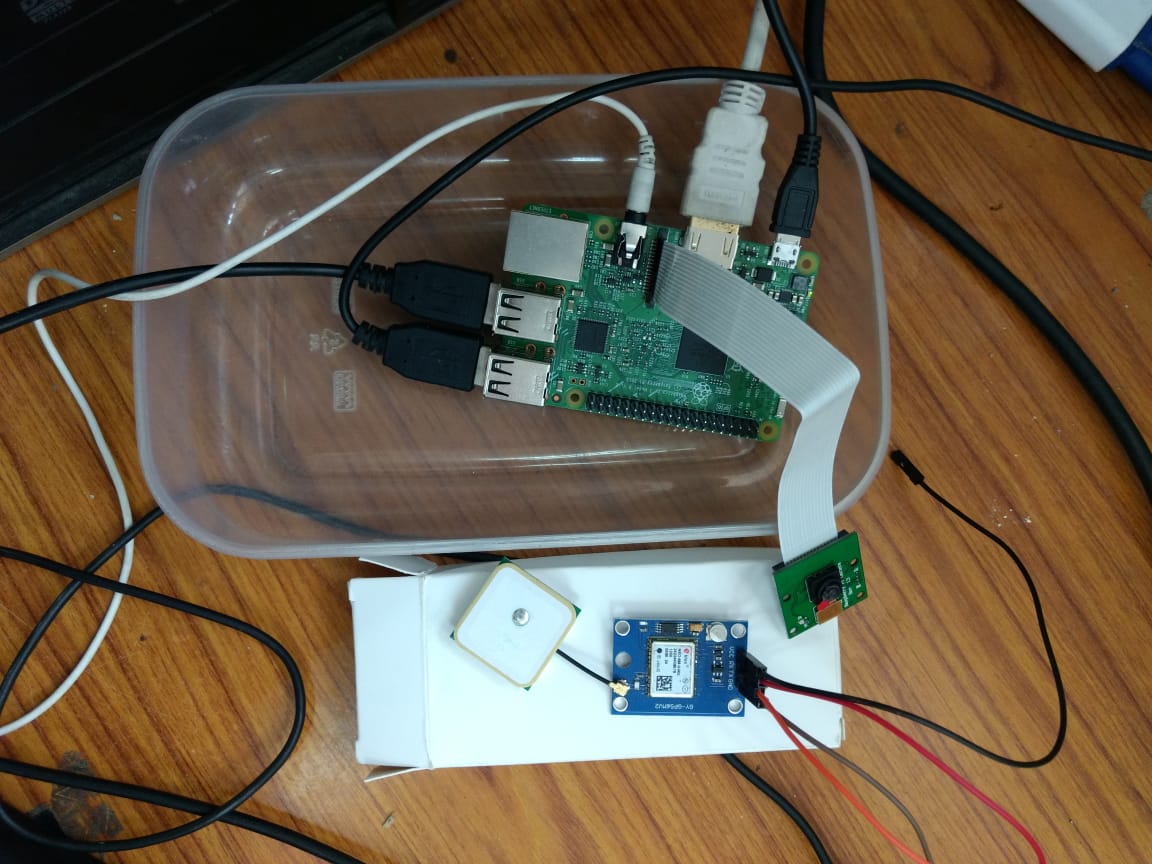


Fig 3 : SoC (Raspberry Pi)

Specifications: Monitor, Keyboard and speakers are connected to the GPIO pins of the Raspberry Pi. The Pi has 26 GPIO pins

1. Camera interface (CSI), Display Interface(DSI)
2. MicroSD card slot

### Raspberry Pi Camera Module : Raspberry Pi Camera is used to capture images. The camera plugs directly into the Camera Serial Interface (CSI) connector on the Raspberry Pi. It's able to deliver clear 5MP resolution image.

### images

Fig 4. RPi Camera

Specifications:

### 5MP (2592×1944 pixels)

### Omni vision 5647 sensor in a fixed focus module Camera Module.

### 1.2 GHz 64 bit quad core ARMv8 CPU

### 802.11n Wireless LAN

### Bluetooth 4.1

### Audio Device : Speaker, or headphones, in the smart reader is used as the output device for audio. It is used for giving instructions about the directions or knowledge about the things that the user sees. There is a standard 3.5mm jack for audio out to an amplifier.

**SYSTEM SOFTWARE DESIGN :**

1. **Operating System (Raspbian (Debian))** : Raspbian is a [Debian](https://en.wikipedia.org/wiki/Debian" \o "Debian)-based [computer operating system](https://en.wikipedia.org/wiki/Operating_system) for [Raspberry Pi](https://en.wikipedia.org/wiki/Raspberry_Pi). There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie.Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012. Raspbian uses PIXEL, Pi Improved Xwindows Environment, Lightweight as its main desktop environment as of the latest update. It is composed of a modified [LXDE](https://en.wikipedia.org/wiki/LXDE) desktop environment and the [Openbox](https://en.wikipedia.org/wiki/Openbox" \o "Openbox) stacking window manager with a new theme and few other changes.
2. **Language (Python 2.7)** : Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level programming language](https://en.wikipedia.org/wiki/High-level_programming_language) for [general-purpose programming](https://en.wikipedia.org/wiki/General-purpose_programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python has a design philosophy that emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability), and a [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)) that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) notably using [significant whitespace](https://en.wikipedia.org/wiki/Significant_whitespace). It provides constructs that enable clear programming on both small and large scales.
3. **Test-To-Speech Converter (Espeak)** : A text-to-speech (TTS) system converts normal language text into speech; other systems render [symbolic linguistic representations](https://en.wikipedia.org/wiki/Symbolic_linguistic_representation)like [phonetic transcriptions](https://en.wikipedia.org/wiki/Phonetic_transcription) into speech.Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a [database](https://en.wikipedia.org/wiki/Database). Systems differ in the size of the stored speech units; a system that stores [phones](https://en.wikipedia.org/wiki/Phone_(phonetics)) or [diphones](https://en.wikipedia.org/wiki/Diphone" \o "Diphone) 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](https://en.wikipedia.org/wiki/Vocal_tract) and other human voice characteristics to create a completely "synthetic" voice output.
4. **OCR (Pi Tesseract)** : A good Optical Character Recognition (OCR) can be used to convert an image of a document to text.Tesseract is an optical character recognition engine for various operating systems. It is free software, released under the Apache License, Version 2.0, and development has been sponsored by Google since 2006. In 2006 Tesseract was considered one of the most accurate open-source OCR engines then available.

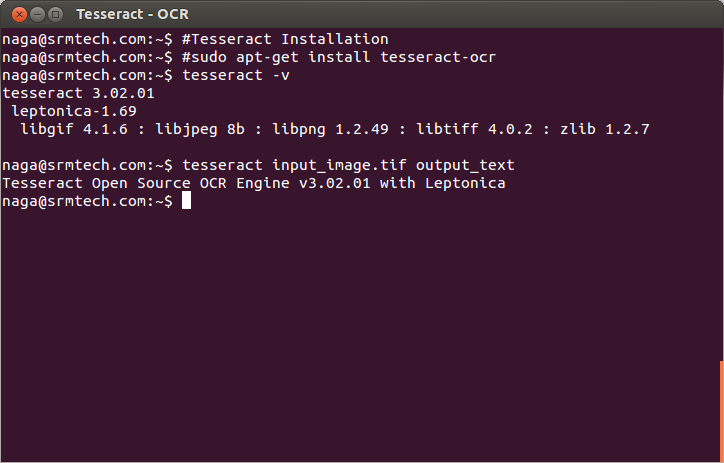


Fig 5 : Tesseract OCR

5**. GUI (TKinter) :** Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

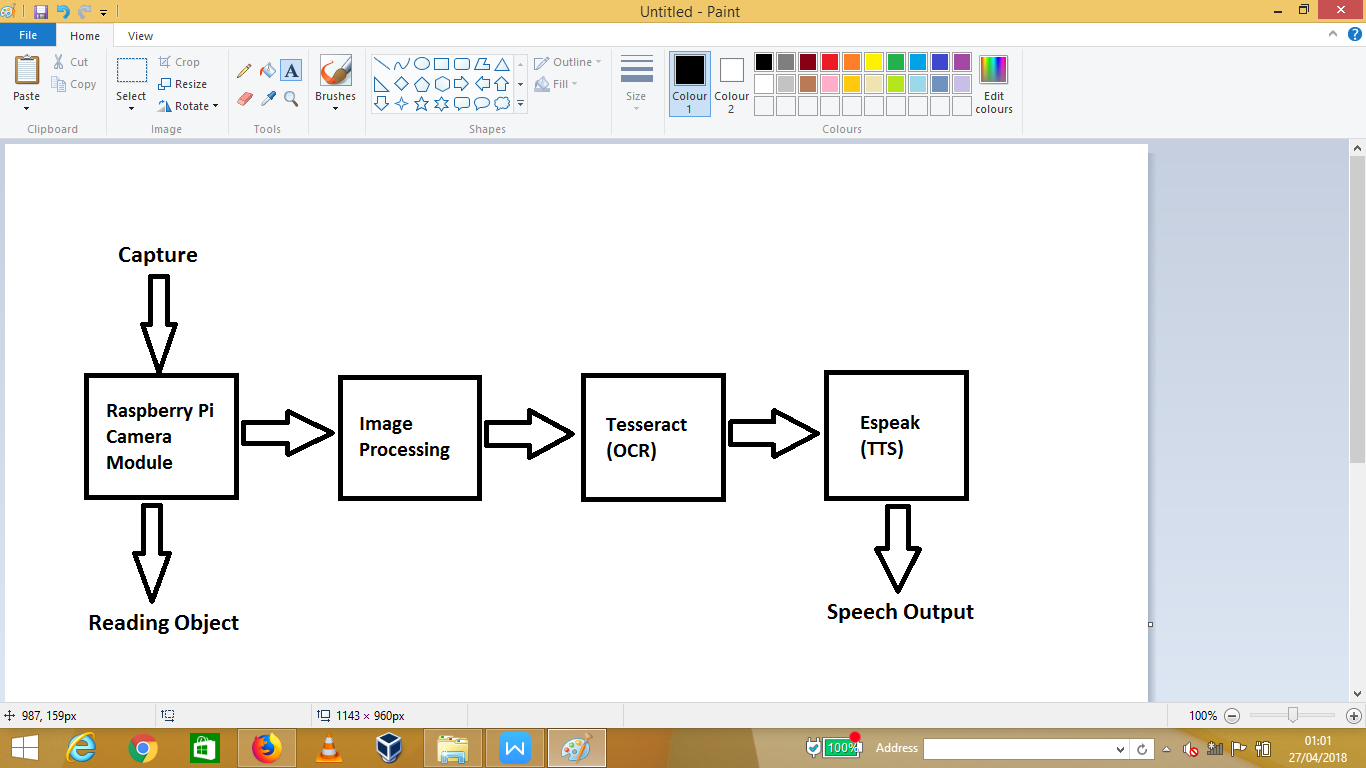


Fig 6. System Software Design

**PROJECT FEATURES**

1. **Portable Monitor, TV or desktop monitor & read along :**

The Monitor is used as an output source. It enlarges the captured image or text and follow along as it reads.

1. **Audio and visual prompts :** The second most important output media is the Audio. The captured image is converted into audio file.
2. **Attach headphone for privacy :** Though speakers are a part of the product the product, but there are some cases when the user may want to listen to the audio in private. Then in that case, the headphones can be plugged in for privacy.
3. **Capture and save modes :** The product provides the facility of capturing the text or image and saving it in the memory for future references.
4. **Natural sounding male or female voice :** Through this product, the user can experience natural human voice.
5. **User friendly GUI :** A GUI (Graphical User Interface) is used in the product which consists of all the functionalities such as read aloud, get summary, read keywords and look up online.
6. **Read aloud facility :** The captured image is read aloud to the user(s).
7. **Get Summary facility :** The Smart Reader has a unique feature of extracting the summary of the text that is captured and presenting it to the user(s).
8. **Read Keywords facility** : It reads the keywords from the text, looks up the dictionary for the meanings and prompts it to the user(s).
9. **Look up online facility :** There is an additional feature in the product which provides the user(s) with the articles related to the text we are referring.

**IMPLEMENTATION**

The image that is to be converted is saved at a location on the system. The path where the image is saved is copied in the code wherever required.

OCR (Optical Character Recognition) PROCESSING :

* + - 1. An OCR scanning system has to understand and be able to read many different kinds of image formats: JPEG, PNG, GIF, BMP, TIFF (one page and multi page) and PDF are a few that OCR can read.
      2. Next, it detects the most important image features that include the resolution and text. OCR scans look for some predefined range of font sizes and page colors so the image is typically edited before it’s processed when it is necessary. If an image is off center or has a lot of noise (lots of black or white spots) the OCR program can improve image quality with some quick tweaks.
      3. Most OCR scanning algorithms require an image with only black and white. This is called a bi-tonal image, and it is applied to images that have color or are gray. The process is called **binarization** and it’s very important because if it’s skipped than an image will come out incorrectly.
      4. Detecting and removing lines is another thing OCR scanning will take care of. This is a required step to improve the analysis of a page layout. This step will recognize quality for things such as underlined text and tables on the page.
      5. The next step is called **zoning** or page layout analysis. The OCR system will detect the position and type of important areas on the page. Detecting text lines and words. This can be difficult due to difference in font sizes and the amount of space between words.
      6. Analysis of broken characters and combining them. Sometimes, some characters will be broken in several parts or characters might be touching, this is necessary to detect and correct the position of each character.
      7. The most important part of any OCR scanning is the **recognition of characters**. Each character has to be converted so it’s legible. If there’s an uncertain character within an image, the algorithm will produce more than one choice of character and the finalized character can be selected later. Some OCR scanners have dictionary support to help improve recognition quality, certain characters can look very similar and the dictionary can help make the decision about what the character should be in context with the characters around it.
      8. After all of this is said and done, the OCR algorithm will save the end results to the selected output file type which could be a searchable PDF, DOC, RTF, or TXT. It’s very important to save the original page in its original format as well.

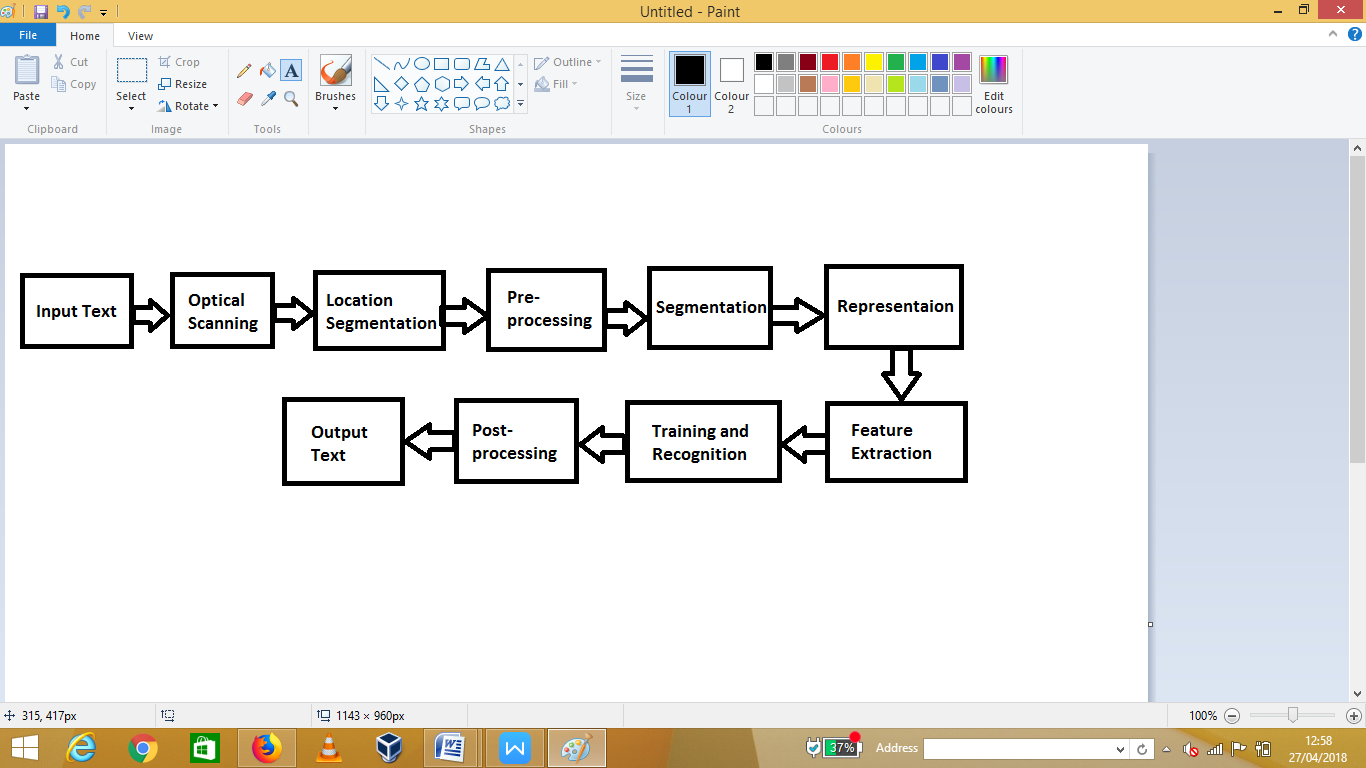


Fig 7. OCR Processing Stages

TEXT TO SPEECH CONVERSION :

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read any text aloud. The block diagram given below explains the same :

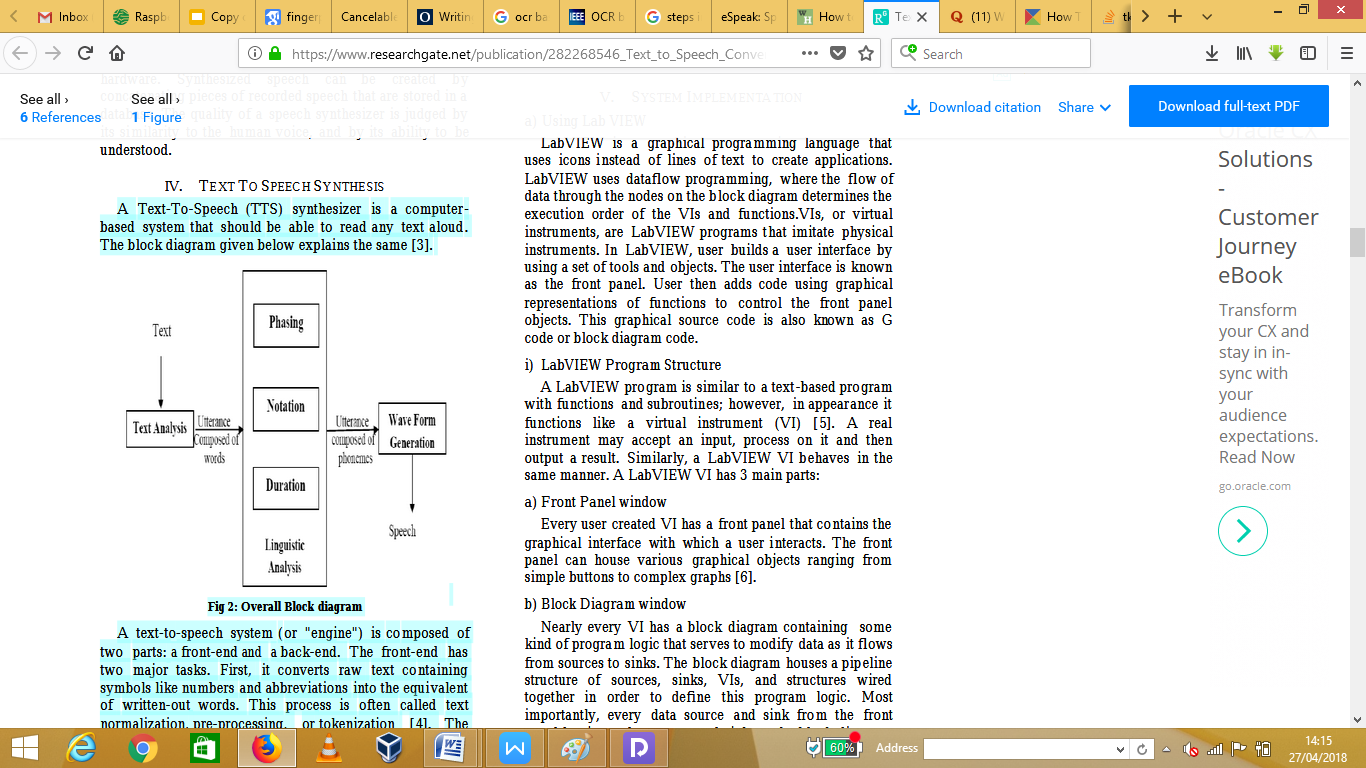


Fig 8. TTS Overall Block Diagram

A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text **normalization**, **pre-processing**, or **tokenization**. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called **text-to-phoneme conversion**. The back-end often referred to as the synthesizer—then converts the symbolic linguistic representation into sound. In certain systems, this part includes the computation of the target prosody (pitch contour, phoneme durations), which is then imposed on the output speech.

GUI DESIGNING :

The GUI (Graphical User Interface) handles user input and output. GUIs often use a form of OO programming which we call event-driven: the program responds to **events*,*** which are actions that a user takes. The GUI is designed using Tkinter. The functioanlities included in the GUI are :

1. Read Aloud functionality
2. Get Summary functionality
3. Read Keywords functionality
4. Look Up Online unctionality

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

1. Import the Tkinter module.
2. Create the GUI application main window.
3. Add one or more of the above-mentioned widgets to the GUI application.
4. Enter the main event loop to take action against each event triggered by the user.

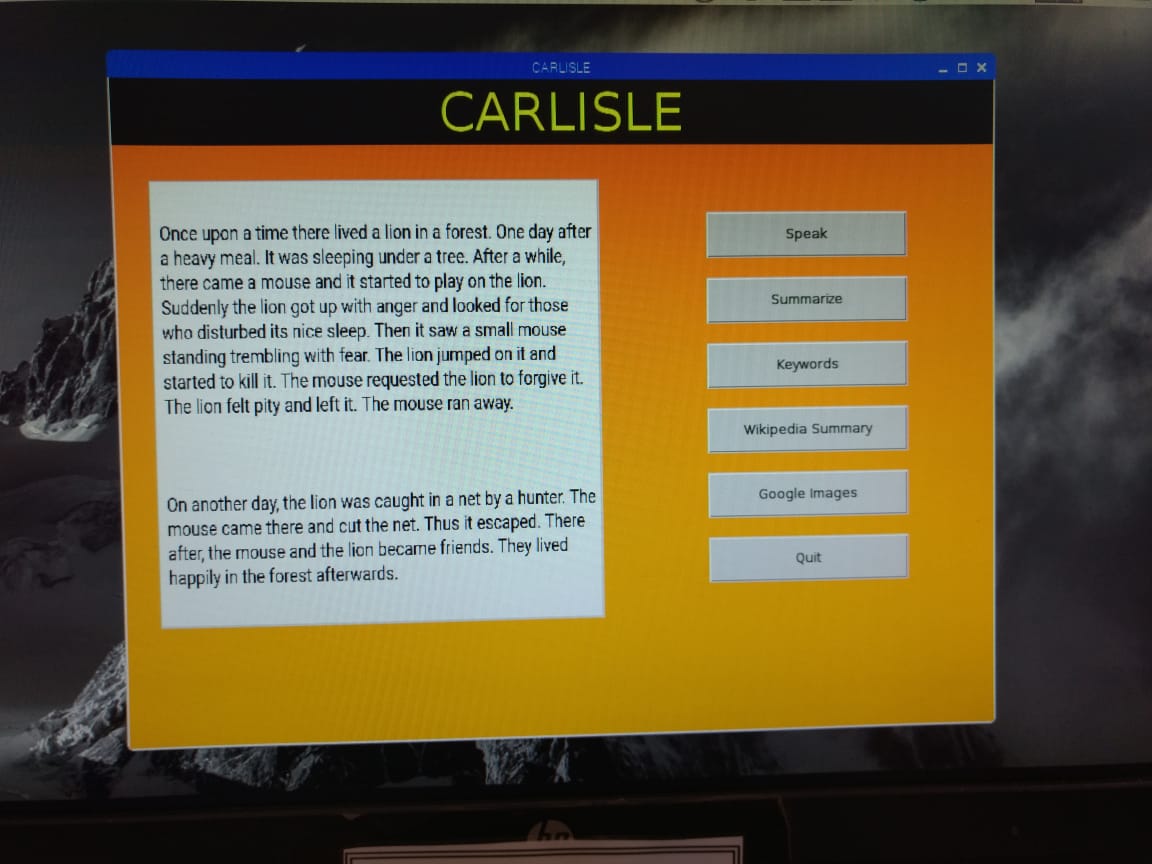


Fig 9. CARLISLE GUI

**TESTING**

SPEAK : This option reads the text which is there on the screen as shown in fig 10.

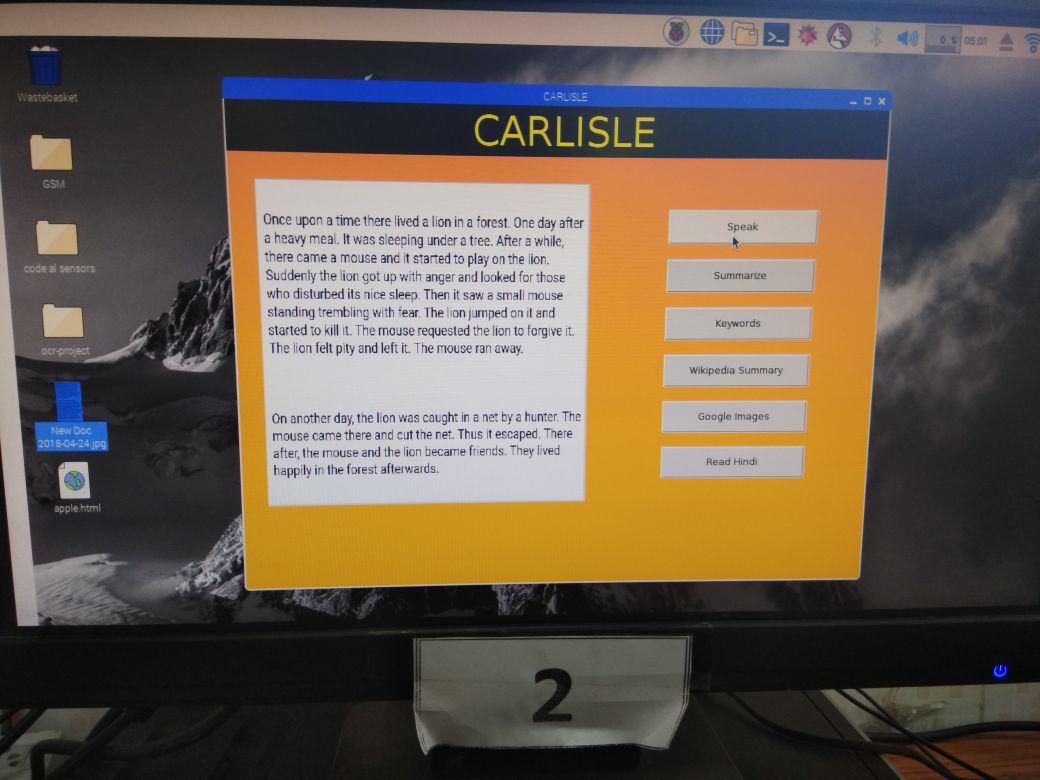


Fig 10.

SUMMARIZE : This option gives the summary of each and every line from the text provided as shown in fig 11.

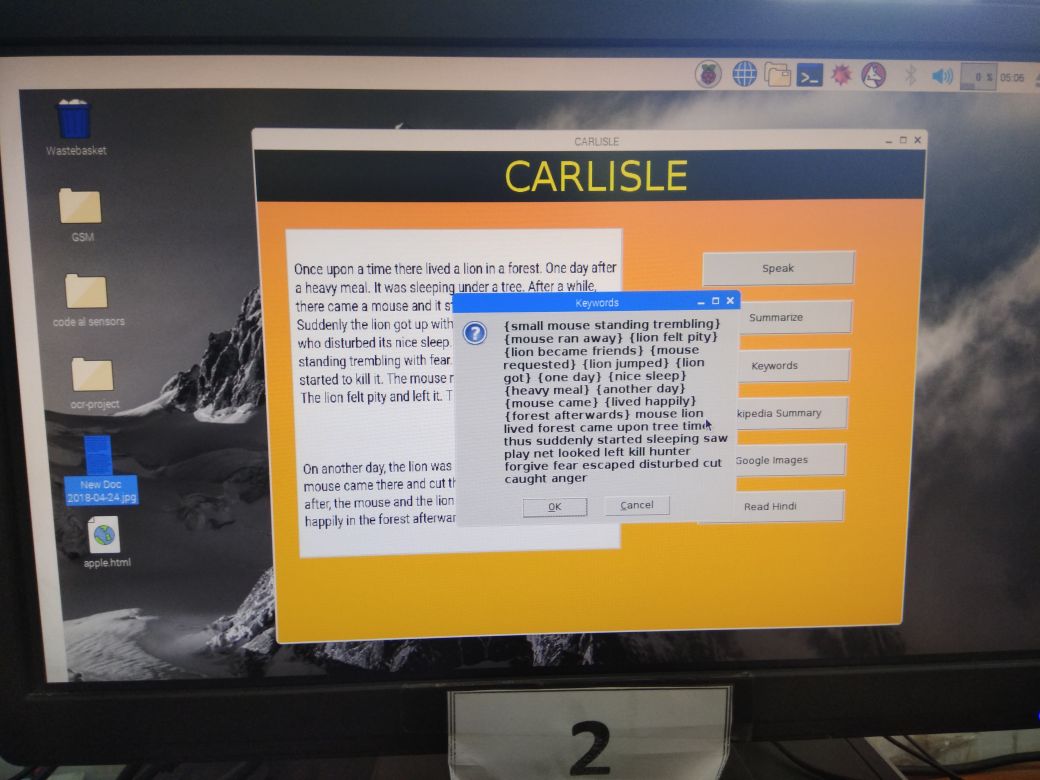


Fig 11.

We can also vary the summary size. Suppose we want the summary of just the two lines as shown in fig 12. and we’ll get the summary accordingly as shown in fig 13.

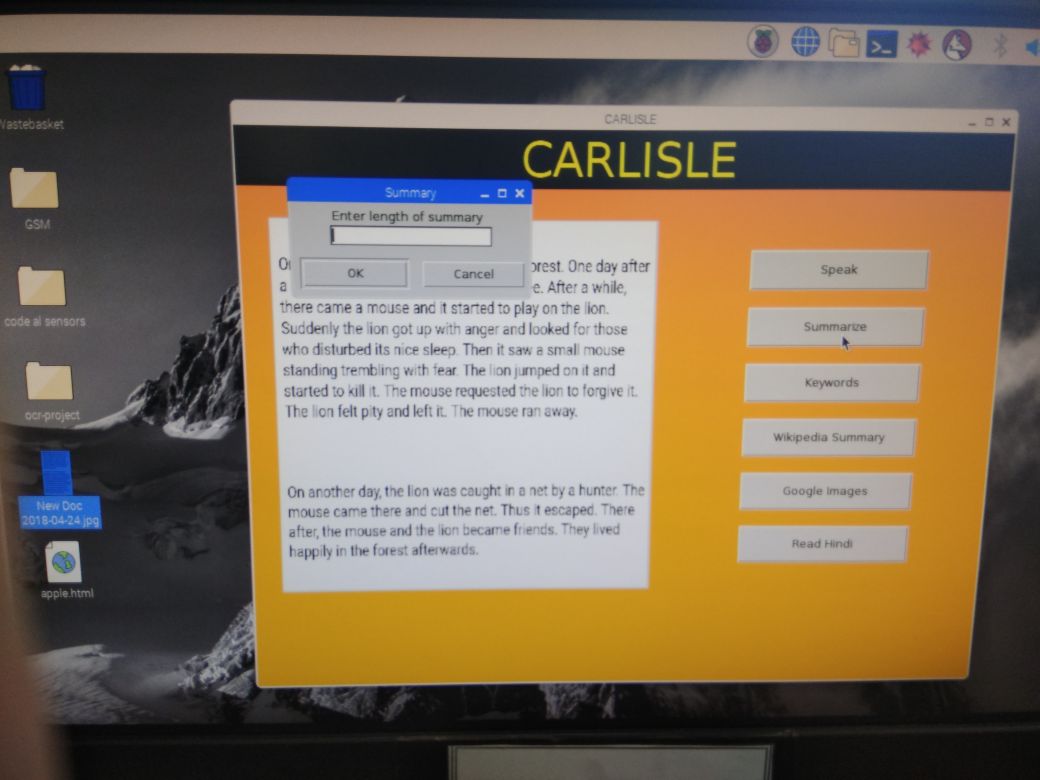


Fig 12.

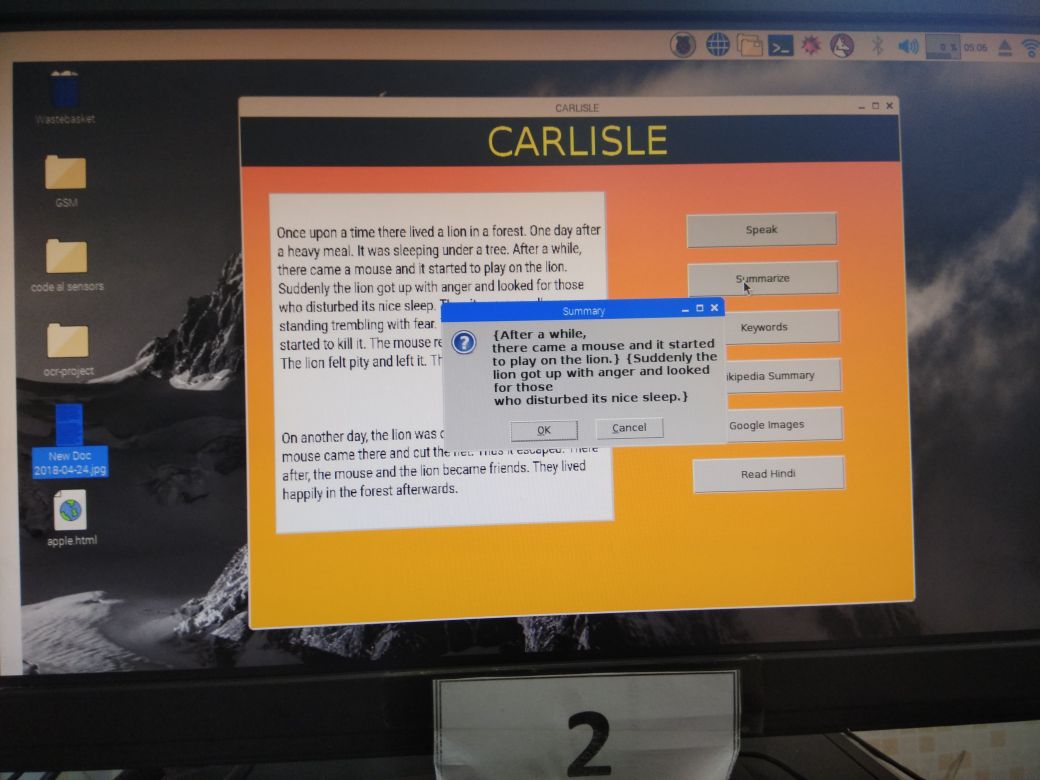


Fig 13.

KEYWORDS : This option displays the important keywords from the text.

WIKIPEDIA SUMMARY : This option asks for a keyword, searches the web and gives the Wikipedia article related to that keyword as shown in fig 14. Suppose the keyword entered is ‘LION’.

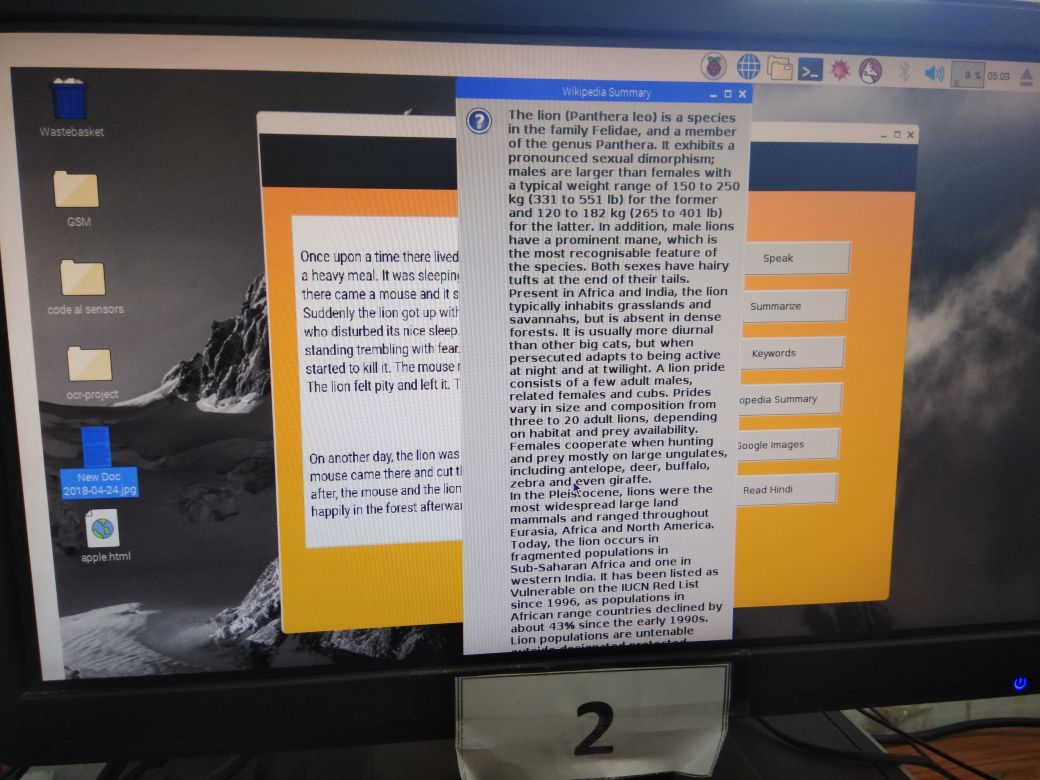


Fig 14.

GOOGLE IMAGES : This option lets the user to retrieve an image from the web for the keyword that the user will enter as shown in fig 15.

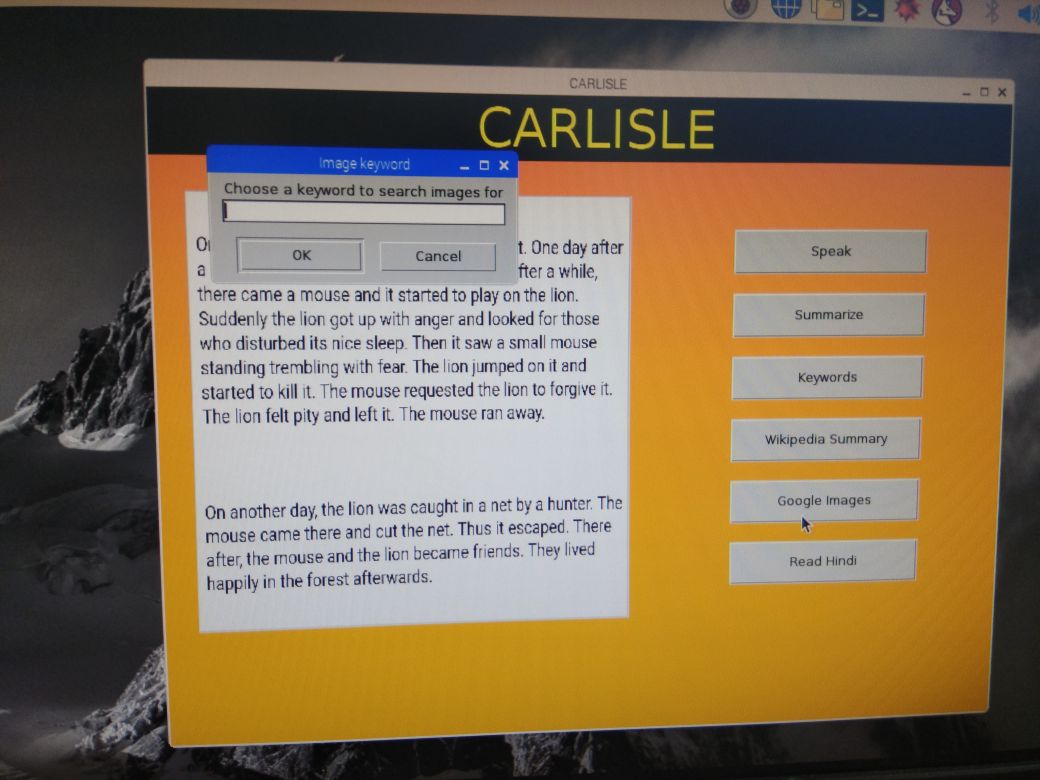


Fig 15.

READ HINDI : This Smart Reader has a unique functionality of reading hindi text. This option will read the text displayed in the box as shown in fig 16.

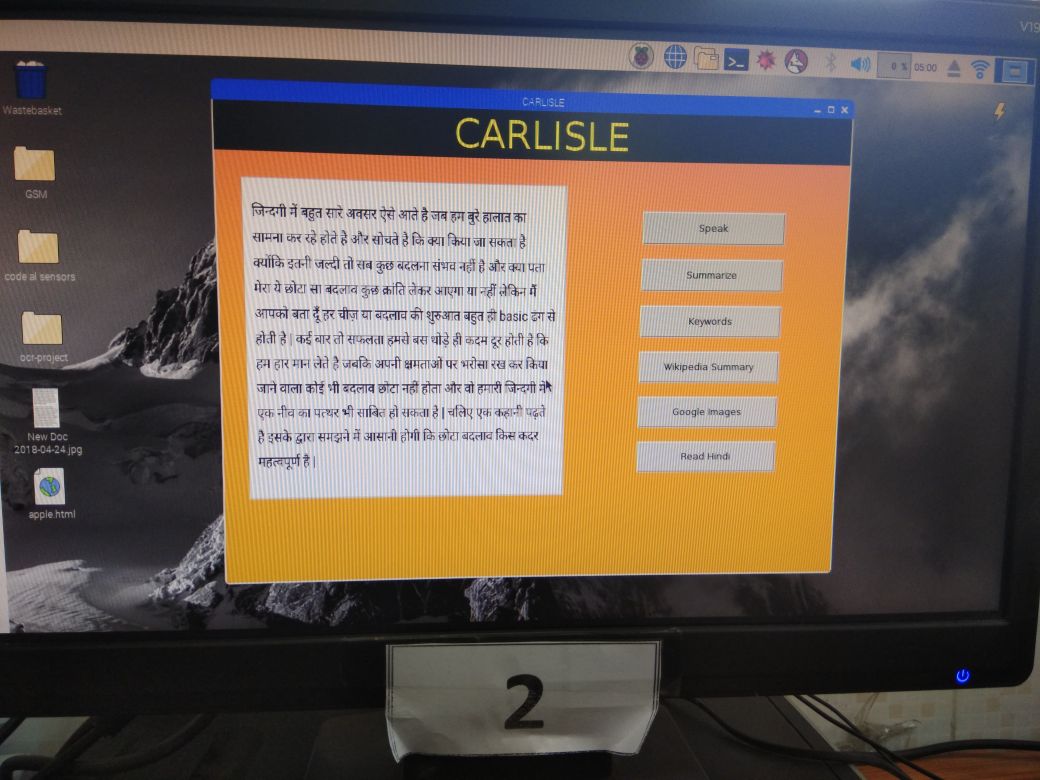


Fig 16.

**SCOPE AND FUTURE WORK**

The Scope of this project is promising. Optical Character Recognition finds its applications in Education, Online Retailing, Medecine, etc. The proposed system works well for English language. Our aim for future work is to extend the same functionality for other Indian regional languages such as Hindi, Tamil, Kannada, Telegu, etc. Optical character recognition finds its use in the following applications

1. Captcha
2. Optical Music Recognition
3. Handwriting Detection
4. Invoice Imaging
5. Automatic Number Detection

**CONCLUSION**

This paper proposes a novel implementation of a smart OCR based reader for teachers, students as well as the visually impaired. This project has been implemented on an embedded platform, and uses various technologies such as optical character recognition, image processing and text to speech engines. The major goal of this project was to provide an affordable hand held device to the under-represented sections of the society.

**REFERENCES**

1. <https://ieeexplore.ieee.org/document/8284628/>
2. <https://en.wikipedia.org/wiki/Optical_character_recognition>
3. <http://www.robopgmr.com/?p=4486>
4. <https://en.wikipedia.org/wiki/Tesseract_(software)>
5. https://en.wikipedia.org/wiki/Tesseract\_(software)#/media/File:Tesseract\_v3.02.png
6. https://www.geeksforgeeks.org/python-gui-tkinter/
7. <http://www.scansmart-usa.com/how-it-works-optical-character-recognition/>
8. http://python-textbok.readthedocs.io/en/1.0/Introduction\_to\_GUI\_Programming.html
9. <http://www.tutorialspoint.com/python/python_gui_programming.htm>
10. T. Dutoit, "High quality text-to-speech synthesis: a comparison of four candidate algorithms," Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on, vol.i, no., pp.I/565-I/568 vol.1, 19-22 Apr 1994.
11. B.M. Sagar, Shobha G, R. P. Kumar, “OCR for printed Kannada text

to machine editable format using database approach” WSEAS

Transactions on Computers Volume 7, Pages 766-769, 6 June 2008.

1. <http://www.voicerss.org/tts/>
2. http://www.comsys.net/technology/speechframe/text-to-speech-

tts.html

1. Image Acquisition and Processing with LabVIEW, Christopher G

Relf, CRC Press, 2004.

1. <http://www.rspublication.com/ijst/aug%2013/6.pdf>
2. Implementing Optical Character Recognition on the Android Operating System for Business Cards Sonia Bhaskar, Nicholas Lavassar, Scott Green EE 368 Digital Image Processing.
3. J. Liang, et. al. “Geometric Rectification of Camera-captured Document Images,” IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 591-605, July 2006.
4. G. Zhu and D. Doermann. “Logo Matching for Document Image

Retrieval,” International Conference on Document Analysis and

Recognition (ICDAR 2009), pp. 606-610, 2009.