# 

# IMAGE TO AUDIO CONVERSION FOR BLIND

# A PROJECT REPORT

# Submitted By

# 

# PRADEEP K [REGISTER NO: 211418104194]

**SAI KASI VISWANATH M [REGISTER NO: 211418104152]**

**KODIGALA SRIABHILASH [REGISTER NO: 211418104128]**

***in partial fulfillment for the award of the degree***

**of**

## BACHELOR OF ENGINEERING

## IN

**COMPUTER SCIENCE AND ENGINEERING**



## PANIMALAR ENGINEERING COLLEGE

**(**An Autonomous Institution, Affiliated to Anna University, Chennai**)**

**MAY 2022**

## PANIMALAR ENGINEERING COLLEGE

**(**An Autonomous Institution, Affiliated to Anna University, Chennai**)**

## BONAFIDE CERTIFICATE

Certified that this project report **“IMAGE TO AUDIO CONVERSION FOR BLIND PEOPLE”** is the Bonafide work of **PRADEEP K [211418104194], SAI KASI VISWANATH M [211418104152], KODIGALA SRIABHILASH [211418104128]** who carried out the project work under my supervision.

#### SIGNATURE SIGNATURE

**Dr.S.MURUGAVALLI,M.E, Ph.D , MS.C.VIJAYALAKSHMI,M.Tech**

**PROFESSOR ASSISTANT PROFESSOR,**

**HEAD OF THE DEPARTMENT SUPERVISOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALARENGINEERINGCOLLEGE, NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above mentioned students were examined in End Semester project Viva-Voice held on........................…

**INTERNAL EXAMINER EXTERNAL EXAMINER**

## DECLARATION BY THE STUDENT

We **Pradeep k [211418104194], Sai kasi Viswanath M [211418104152], Kodigala Sri Abhilash [211418104128]** hereby declare that this project report titled “IMAGE TO AUDIO CONVERSION FOR BLIND PEOPLE”, under the guidance of C.Vijayalakshmi the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**ACKNOWLEDGEMENT**

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A.,Ph.**D**.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHIKUMAR**

**B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,**

for the support extended throughout the project.

We would like to thank my **Project Guide** –**MS. C**.**Vijayalakshmi** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**PRADEEP K [211418104194]**

**SAI KASI VISWANATH M [211418104152]**

**KODIGALA SRIABHILASH [211418104128]**

# ABSTRACT

Many times, the main problem in communication is language bias between the communicators. This paper is based on a prototype that helps users to hear the contents of the text images in the desired language. It involves extraction of text from the image and converting the text to translated speech in the user desired language. This relieves the travelers as they can use this device to hear the English text in their own desired language. It can also be used by the visually impaired. This device helps users to hear the images being read in their desired language.The proposed system will help the user to take a picture and then the image will be scanned and the application will read the text written in English language. The data obtained is then converted to voice, and blind people will get the visual of what the text reads. Thus, the output will be given in speech format. The purpose of delivering the output in the form of voice/speech is to serve the information that is present on the document to the visually impaired. We use Natural Language Processing in the proposed system for better performance and accuracy. The complete system is built using GUI with higher accuracy.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE NO** |
|  | ABSTRACT | **iii** |
|  | LIST OF FIGURES | **vi** |
|  | LIST OF ABBREVIATIONS | **vii** |
| **1.** | **INTRODUCTION** | **1** |
| 1.1 INTRODUCTION | **1** |
| **2.** | **LITERATURE SURVEY** | **5** |
| **3.** | **SYSTEM ANALYSIS** | **12** |
| 3.1 EXISTING SYSTEM | **13** |
| 3.2 PROPOSED SYSTEM | **13** |
| * 1. REQUIREMENT SPECIFICATION   3.4 ABOUT LANGUAGE -PYTHON | **14**  **14** |
| 3.5 PLATFORM SPECIFICATION– | **18** |
| ANACONDA |  |
| **4.** | **SYSTEM DESIGN** | **23** |
| 4.1 SYSTEM ARCHITECTURE | **24** |
| 4.2 USE-CASE DIAGRAM | **25** |
| 4.3 ACTIVITY DIAGRAM | **26** |
| 4.4 SEQUENCE DIAGRAM | **27** |
| 4.5 DATA FLOW DIAGRAM 0 | **27** |
| * 1. DATA FLOW DIAGRAM 1   2. STATE DIAGRAM | **28**  **29** |
| 4.8 ER DIAGRAM | **30** |
| **5.** | **MODULE DESCRIPTION** | **31** |
| 5.1 DATA COLLECTING AND  CREATING GUI | **32** |
| 5.2 NLP TOOLS AND APPROCHES USED IN THE MODEL | **32** |
| 5.3 PROCESSING OUTPUT  5.4 MODULE SPLITUP  5.5 ALGORITHMS | **34**  **35**  **36** |
| **6.** | **TESTING**  6.1 SYSTEM TESTING | **41**  **42** |
| 6.2.TESTING TECHNIQUES | **45** |
| 6.3 SOFTWARE TESTING STRATEGIES | **46** |
| **7** | **CONCLUSION**  7.1 CONCLUSION | **49**  **50** |
| 7.2 FUTURE ENHANCEMENT | **50** |
| **8** | **APPENDIX 1** | **51** |
| **9** | **APPENDIX 2** | **57** |
| **10** | **REFERENCES** | **61** |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| 4.1 | SYSTEM ARCHITECTURE | **24** |
| 4.2 | USE-CASE DIAGRAM | **25** |
| 4.3 | ACTIVITY DIAGRAM | **26** |
| 4.4 | SEQUENCE DIAGRAM | **27** |
| 4.5 | DATA FLOW DIAGRAM 0 | **27** |
| 4.6 | DATA FLOW DIAGRAM 1 | **28** |
| 4.7 | STATE DIAGRAM | **29** |
| 4.8 | ER DIAGRAM | **30** |
| 7.1 | INPUT SCREEN | **57** |
| 7.2 | TEXT CONVERTED | **58** |
| 7.3 | AUDIO CONVERSION | **59** |
| 7.4 | AUDIO OUTPUT | **60** |

**LIST OF ACRONYMS AND ABBREVIATIONS**

**NLP -** Natural Language Processing **NER -** Named Entity Recognition **OCR -** Optical character Recognition **CNN -** Convolution Neural Network **HDS -** Human Detection System

## 

## CHAPTER 1

## INTRODUCTION

## CHAPTER 1

## INTRODUCTION

#### 1.1 Introduction

Every year, the number of visually impaired people grows as a result of eye disorders, ageing, traffic accidents, and other factors. Because reading is such an important part of human life (text can be found in newspapers, commercial products, signboards, computer screens, and so on), visually impaired people encounter numerous challenges. Visually impaired people can use mobile applications to assist them in reading text. The goal of our study is to see if a visually impaired person can acquire auditory information regarding printed text, text boards, scene text, hoardings, and traffic signboard instructions.

To attain the best results and reach state-of-the-art accuracy for the particular challenge, many techniques have been created for picture to audio conversion for blind persons. Existing methods based on have produced impressive results for text to audio conversion, but obtaining speech with high accuracy remains a challenge. The current system is incapable of extracting text from photos and of providing an audio output for the image to be read. It can only transform text inputs into speech, making it ineffective for usage in real-world applications to assist blind people. The current model employs CNN, which is computationally intensive. They necessitate a large training data set and a lengthy preparation period.

We propose a model for the image to audio conversion for blind people using machine learning techniques. In this technique provides a camera-based assistive text reading system allowing visually impaired (partly or completely blind) people to read text from

labels and product packaging in their daily lives. This paper presents a method for defining the area of interest (ROI) in which objects can be segregated from cluttered backgrounds that is both efficient and effective. To obtain text information, this ROI extracts the text localization and recognition.

A printed copy of a book is the usual means of reading it. Books came in handy because it was impossible to carry a physical copy of a book everywhere. Both printed and electronic books have an impact on the eyes. Visual impairment is one of humanity's most significant constraints, particularly in this day and age where information is obtained by reading a large number of books (electronic and paper based).

The study is based on the Android platform, as well as machine learning. For blind persons, reading text from text pictures and text boards is a difficult effort. Visual impairment is one of humanity's most significant limitations, especially in this day and age, when information is increasingly delivered by text messages (both electronic and paper-based) rather than voice.

According to surveys, many find living in today's world to be extremely difficult. The writers have created a new device called the Blind Book Reader System to assist them. When blind people are in need, it's an easy remedy to get them to read books and newspapers.

In this research, an image-to-audio-description (I2AD) task is presented to produce audio descriptions from photos in order to help visually impaired people better perceive the information around them.Text-to-speech (TTS) systems convert written text into speech signals. The conversion of Devanagari text to speech for Marathi printed text is done in this study. Two techniques are used to obtain the needed output: Optical Character Recognition and Pattern Recognition (OCR).

The authors talk about how automated Audio Captioning is a cross-modal task that generates natural language summaries of the sound occurrences in audio clips. However, there hasn't been any research into grounding the actual sound events in the given audio based on the caption. This work includes an Audio-Grounding dataset 1 that shows how sound events correspond to the captions provided in Audiocaps.

In the proposed technology is cost-effective and assists visually challenged individuals in hearing text. The fundamental idea behind this project is to employ optical character recognition to convert text letters into audio signals.In a model for converting natural Bengali language to text is presented in this research. The suggested model necessitates the use of the open-source Sphinx 4 framework, which is built in Java and provides the necessary procedural coding tools for developing an acoustic model for a bespoke language such as Bengali.

Our key goal was to make sure that the system had received appropriate word-by-word training from a variety of people so that it could recognise new speakers fluently.As cloud computing adds new capabilities and services, it also brings new obstacles, such as cost, complexity, and integration issues. Traditional cloud architecture has been employed by several academics for their applications, such as text-to-speech (TTS).

CHAPTER 2

LITERATURE SURVEY

**CHAPTER 2**

**LITERATURE SURVEY**

#### TITLE 1 :

Image to Audio Conversion using Portable Camera January 2018 Journal of Electrical & Electronic Systems 07(03)DOI:10.4172/2332-0796.1000268

#### AUTHORS:

Baskaran S;Lakshmi D

#### DESCRIPTION:

This method proposes a camera based assistive text reading for the visually impaired persons (partially or completely blind) to read the text in the label and from the products packaging in their daily lives. This proposes an efficient and effective based method to define the region of interest (ROI) in which objects can be isolated from the cluttered backgrounds. This ROI extracts the text localization and recognition to acquire text information

#### TITLE 2:

Image To Speech Conversion Website

#### AUTHORS:

Sanket Munot 1 , Akshay Patil 2 , Utkarsha Kandale 3 , Prof. Supriya S Ambarkar 4

#### DESCRIPTION:

The traditional way of reading a book is through a printed copy. It’s not possible to carry a physical copy of a book everywhere, that’s when books came in handy. Printed books as well as e-books affect the eyes. Visual impairment is one of the biggest limitations for humanity, especially in this day and age when information is received by reading a lot of books (electronic and paper based).

#### TITLE 3:

IMAGE TEXT TO VOICE CONVERTER WITH SENTIMENT ANALYSIS

#### AUTHORS:

1Prof. Vivek Pandey, 2Shani Kumar Maurya, 3Sonam Gupta, 4ShaikhUsman Gani

#### DESCRIPTION:

We proposed a text to speech converter for blind people with sentiment analysis. The paper is based on Android domain along with machine learning. Reading text from text images and text boards is a difficult task for blind people. Visual disability is one of the biggest limitations for humanity, especially in this day and age when information is communicated a lot by text messages (electronic and paper based) rather than voice.

#### TITLE 4:

A Novel Approach for Blind - Image to Audio Conversion in Regional Language

#### AUTHORS:

B. Hemalatha, B. Karthik, S. Balaji, G. Vijayalakshmi & Rabindra Nath Shaw

#### DESCRIPTION:

In today’s world visually challenged people are 6.6 trillion in number and it keeps on increasing every year. Through surveys it is found that in today’s life it’s been very challenging for them to live. To help them we have designed a new product called Blind Book reader system. It’s a simple solution to make the blind people read books, newspapers when they are in need.

#### TITLE 5:

Audio description from image by modal translation network

#### AUTHORS:

HailongNingab;Xiangtao Zheng;YuanYuan;Xiaoqiang Luo

#### DESCRIPTION:

Audio is the main form for the visually impaired to obtain information. In reality, all kinds of visual data always exist, but audio data does not exist in many cases. In order to help the visually impaired people to better perceive the information around them, an image-to-audio-description (I2AD) task is proposed to generate audio descriptions from images in this paper.

#### TITLE 6:

Devanagari Printed Text to Speech Conversion using OCR

#### AUTHORS:

Nidhi Kalidas Sawant; Sangam Borkar

#### DESCRIPTION:

Text to speech is very much useful in today's world. People use this technology in their day to day work but couldn't realize it. A text-to-speech (TTS) system transforms common language text into speech signals. In this paper Devanagari text to speech conversion is done for Marathi printed text. To obtain the required output the two techniques are implemented that are Optical Character Recognition (OCR)

#### TITLE 7:

Text-to-Audio Grounding: Building Correspondence Between Captions and Sound Events

#### AUTHORS:

Xuenan Xu; Heinrich Dinkel; Mengyue Wu; Kai Yu

#### DESCRIPTION:

Automated Audio Captioning is a cross-modal task, generating natural language descriptions to summarize the audio clips’ sound events. However, grounding the actual sound events in the given audio based on its corresponding caption has not been investigated. This paper contributes an Audio-Grounding dataset 1 ,

which provides the correspondence between sound events and the captions provided in Audiocaps.

#### TITLE 8:

OCR Based Image Text to Speech Conversion Using MATLAB

#### AUTHORS:

Sneha.C. Madre; S.B. Gundre

#### DESCRIPTION:

There are millions of blind people in the world who are visually impaired. Disability to read has a large impact on the life of visually impaired people. The Proposed system is cost-efficient and helps the visually impaired person to hear the text. The main idea of this project is optical Character recognition which is used to convert text characters into the audio signal.

#### TITLE 9:

A Real Time Speech to Text Conversion Technique for Bengali Language

#### AUTHORS:

Abdullah Umar Nasib; Humayun Kabir; Ruhan Ahmed; Jia Uddin

#### DESCRIPTION:

This paper presents a model to convert natural Bengali language to text. The proposed model requires the usage of the open sourced framework Sphinx 4 which is written in Java and provides the required procedural coding tools to

develop an acoustic model for a custom language like Bengali. Our main objective was to ensure that the system was adequately trained on a word by word basis from various speakers so that it could recognize new speakers fluently.

#### TITLE 10:

A Text-To-Speech Web Service-Based Application Using Serverless Cloud Computing

#### AUTHORS:

Imran Ghani; Andrew Daniel; Bransyn Luther

#### DESCRIPTION:

As cloud computing continues to introduce new features and services, it also introduces some challenges like cost, complexity, and integration issues. Some researchers have used traditional cloud architecture for their applications, like text-to-speech (TTS), using cloud architectu**re.**

## CHAPTER 3

## SYSTEM ANALYSIS

## CHAPTER 3

## SYSTEM ANALYSIS

* 1. **EXISTING SYSTEM**

Many systems have been developed for the image to audio conversion for blind people to get maximum results and achieve state-of-the-art accuracy for the given problem.The existing methods based on have achieved remarkable results for the text to audio, but it is still difficult to obtain speech with high accuracy.The existing system cannot detect the text from images and can neither provide an audio output for the image to be read.It can only convert the text inputs into speech and thus, isn’t very efficient to be used in real-world applications for helping blind people. In the existing model, CNN is used which is computationally expensive.They require a huge training data set, and Its preprocessing time is quite high .The existing system often needs manual initialization, is not running in real-time, or does not work for images with many manuscripts, while also being sensitive to other factors.

## PROPOSED SYSTEM

We propose a model for the image to audio conversion for blind people using machine learning techniques.Input will be given in the form of an image, which will be processed so that the texts in the image would get converted to audio, to be received as the output of the system.The proposed system has shown excellent performance with a high accuracy rate and a much higher speed up rate as compared to the previously used state-of-the-art methods.The proposed framework is not only much faster than the previous work but also maintains competitive accuracy with the state-of-the-art human detection system.It is independent of the user's physical interaction. It has very precise measurements and permits for high deployment and authentication.Use of our framework is

not limited to a single field of application and is useful for many more real- world applications. Our proposed model for the image to audio conversion is beneficial to the world for advanced applications in helping visually impaired people.

## REQUIREMENT SPECIFICATION

* + 1. **HARDWARE REQUIREMENTS**

Processor : Pentium Dual Core 2.00GHZ

Hard disk : 120 GB

RAM : 2GB (minimum)

Keyboard : 110 keys enhanced

## SOFTWARE REQUIREMENTS

Operating system : Windows7 (with service pack 1), 8, 8.1 and 10 IDE : Anaconda

Language : Python

## ABOUT LANGUAGE– PYTHON

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are

available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy,that a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables,evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

### What can Python do?

* Python can be used on a server to create web applications.
* Python can be used alongside software to create workflows.
* Python can connect to database systems. It can also read and modify files.
* Python can be used to handle big data and perform complex mathematics.
* Python can be used for rapid prototyping, or for production-ready software development.

### Why Python?

* Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
* Python has a simple syntax similar to the English language.
* Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
* Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
* Python can be treated in a procedural way, an object-oriented way or a functional way.

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. A source level debugger allows inspection of local and global variables The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python web site

There is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

#### Python modules and packages

Suppose you have developed a very large application that includes many modules. As the number of modules grows, it becomes difficult to keep track of them all if they are dumped into one location. This is particularly so if they have similar names or functionality. You might wish for a means of grouping and organizing them. Packages allow for a hierarchical structuring of the module namespace using dot notation. In the same way that modules help avoid collisions between global variable names, packages help avoid collisions between module names. Creating a package is quite straightforward, since it makes use of the operating system’s inherent hierarchical file structure.

If a file named init .py is present in a package directory, it is invoked when the package or a module in the package is imported. This can be used for execution of package initialization code, such as initialization of package-level data.

Instead, Python follows this convention: if the init .py file in the package directory contains a list named all , it is taken to be a list of modules that should be imported when the statement from <package\_name> import \* is encountered.

Using import \* still isn’t considered terrific form, any more for packages than

for modules. But this facility at least gives the creator of the package some control over what happens when import \* is specified. (In fact, it provides the capability to disallow it entirely, simply by declining to define all at all. As you have seen, the default behavior for packages is to import nothing.)

Note that this does not make the module contents directly accessible to the caller. Each module has its own private symbol table, which serves as the global symbol table for all objects defined in the module. Thus, a module creates a separate namespace, as already noted.The statement import <module\_name> only places <module\_name> in the caller’s symbol table.

The objects that are defined in the module remain in the module’s private symbol table.From the caller, objects in the module are only accessible when prefixed with <module\_name> via dot notation. An alternate form of the import statement allows individual objects from the module to be imported directly into the caller’s symbol table

## PLATFORM SPECIFICATION– ANACONDA

Anaconda is an open-source package manager for Python and R. It is the most popular platform among data science professionals for running Python and R implementations. There are over 300 libraries in data science, so having a robust distribution system for them is a must for any professional in this field.Anaconda simplifies package deployment and management. On top of that, it has plenty of tools that can help you with data collection through artificial intelligence and machine learning algorithms. With Anaconda, you can easily set up, manage, and share Conda environments. Moreover, you can deploy any required project and also you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

You can deploy any required project with a few clicks when you’re using Anaconda.There are many advantages to using Anaconda and the following are the most prominent ones among them: Anaconda is free and open-source. This means you can use it without spending any money. In the data science sector, Anaconda is an industry staple. It is open-source too, which has made it widely popular. If you want to become a data science professional, you must know how to use Anaconda for Python because every recruiter expects you to have this skill. It is a must-have for data science.

It has more than 1500 Python and R data science packages, so you don’t face any compatibility issues while collaborating with others. For example, suppose your colleague sends you a project which requires packages called A and B but you only have package A. Without having package B, you wouldn’t be able to run the project. Anaconda mitigates the chances of such errors. You can easily collaborate on projects without worrying about any compatibility issues.It gives you a seamless environment that simplifies deploying projects. You can deploy any project with just a few clicks and commands while managing the rest. Anaconda has a thriving community of data scientists and machine learning professionals who use it regularly. If you encounter an issue, chances are, the community has already answered the same. On the other hand, you can also ask people in the community about the issues you face there, it’s a very helpful community ready to help new learners. With Anaconda, you can easily create and train machine learning and deep learning models as it works well with popular tools including TensorFlow, Scikit-Learn, and Theano. You can create visualizations by using Bokeh, Holoviews, Matplotlib, and Datashader while using Anaconda.

#### How to Use Anaconda for Python

Now that we have discussed all the basics in our Python Anaconda tutorial, let’s discuss some fundamental commands you can use to start using this package

Listing All Environments

To begin using Anaconda, you’d need to see how many Conda environments are present in your machine.

conda env list

It will list all the available Conda environments in your machine. Creating a New Environment

You can create a new Conda environment by going to the required directory and use this command:

conda create -n <your\_environment\_name>

You can replace <your\_environment\_name> with the name of your environment. After entering this command, conda will ask you if you want to proceed to which you should reply with y:

proceed ([y])/n)?

On the other hand, if you want to create an environment with a particular version of Python, you should use the following command:

conda create -n <your\_environment\_name> python=3.6

Similarly, if you want to create an environment with a particular package, you can use the following command:

conda create -n <your\_environment\_name>pack\_name

Here, you can replace pack\_name with the name of the package you want to use.

If you have a .yml file, you can use the following command to create a new Conda environment based on that file:

conda env create -n <your\_environment\_name> -f <file\_name>.yml

We have also discussed how you can export an existing Conda environment to a

.yml file later in this article.

#### Activating an Environment

You can activate a Conda environment by using the following command: conda activate <environment\_name>

You should activate the environment before you start working on the same. Also, replace the term <environment\_name> with the environment name you want to activate. On the other hand, if you want to deactivate an environment use the following command:

conda deactivate

#### Installing Packages in an Environment

Now that you have an activated environment, you can install packages into it by using the following command:

conda install <pack\_name>

Replace the term <pack\_name> with the name of the package you want to install in your Conda environment while using this command.

#### Updating Packages in an Environment

If you want to update the packages present in a particular Conda environment, you should use the following command:

conda update

The above command will update all the packages present in the environment. However, if you want to update a package to a certain version, you will need to use the following command:

conda install <package\_name>=<version>

#### Exporting an Environment Configuration

Suppose you want to share your project with someone else (colleague, friend, etc.). While you can share the directory on Github, it would have many Python packages, making the transfer process very challenging. Instead of that, you can

create an environment configuration .yml file and share it with that person packages, making the transfer process very challenging. Instead of that, you can

Now, they can create an environment like your one by using the .yml file.

For exporting the environment to the .yml file, you’ll first have to activate the same and run the following

command:conda env export ><file\_name>.yml

The person you want to share the environment with only has to use the exported file by using the ‘Creating a New Environment’ command we shared before.

#### Removing a Package from an Environment

If you want to uninstall a package from a specific Conda environment, use the following command:

conda remove -n <env\_name><package\_name>

On the other hand, if you want to uninstall a package from an activated environment, you’d have to use the following command:

conda remove <package\_name>

#### Deleting an Environment

Sometimes, you don’t need to add a new environment but remove one. In such cases, you must know how to delete a Conda environment, which you can do so by using the following command:

conda env remove –name <env\_name>

The above command would delete the Conda environment right away.

**CHAPTER 4**

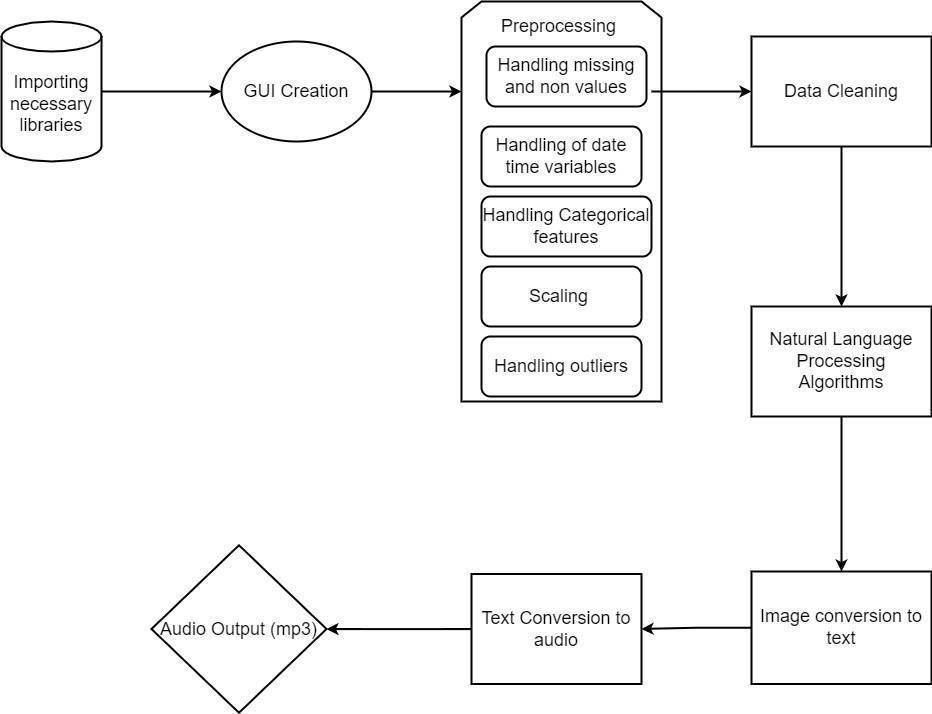
**SYSTEM DESIGN**

## CHAPTER 4

**SYSTEM DESIGN**

## SYSTEM ARCHITECTURE

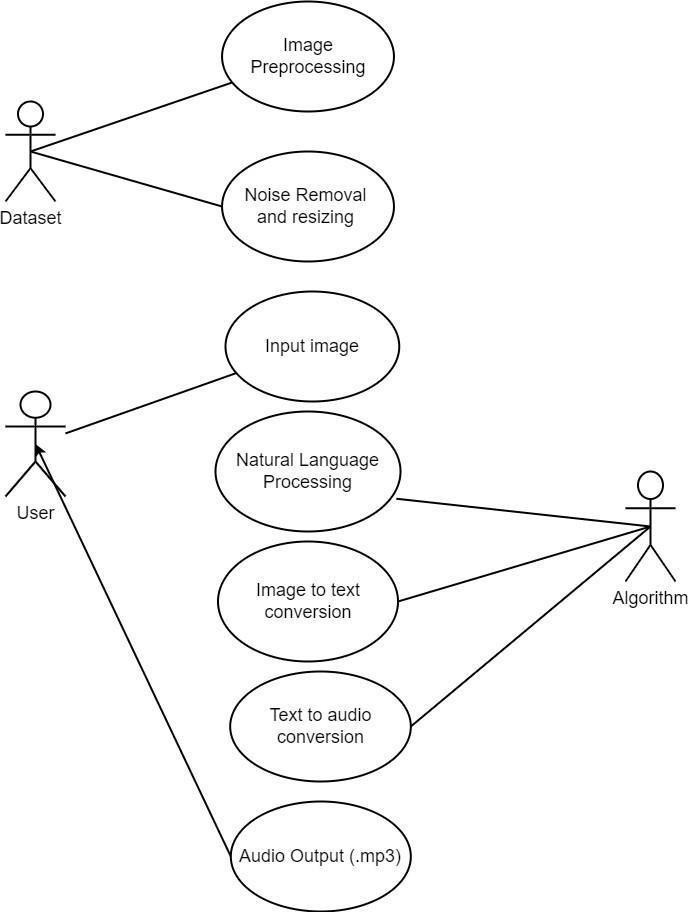
This diagram is nothing but a simple description of all the entities that have been incorporated into the system. The diagram represents the relations between each of them and involves a sequence of decision-making processes and steps. You can simply call it a visual or the whole process and its implementation. All functional correspondences are explained in this diagram.



**Fig 4.1 – Architecture Diagram**

## USE-CASE DIAGRAM

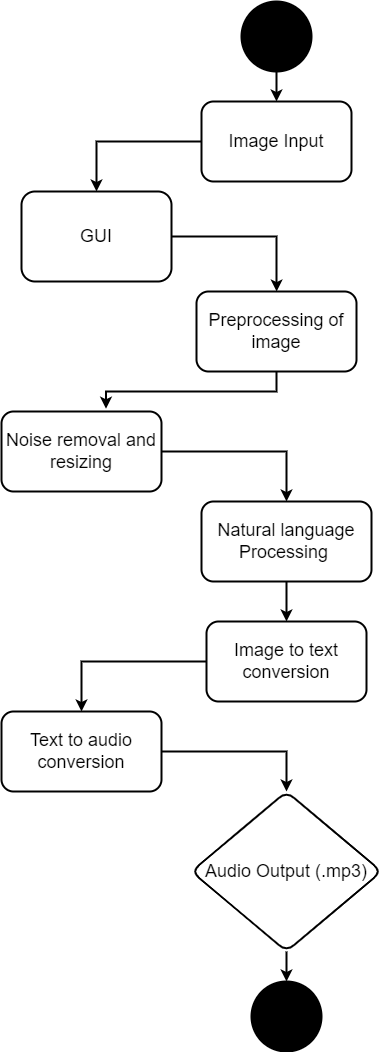
A use case diagram is a standard diagram that shows all interactions between the user, dataset, and algorithm used. It is developed in the early stages of the process.



**Fig 4.2 – Use-Case Diagram**

## ACTIVITY DIAGRAM

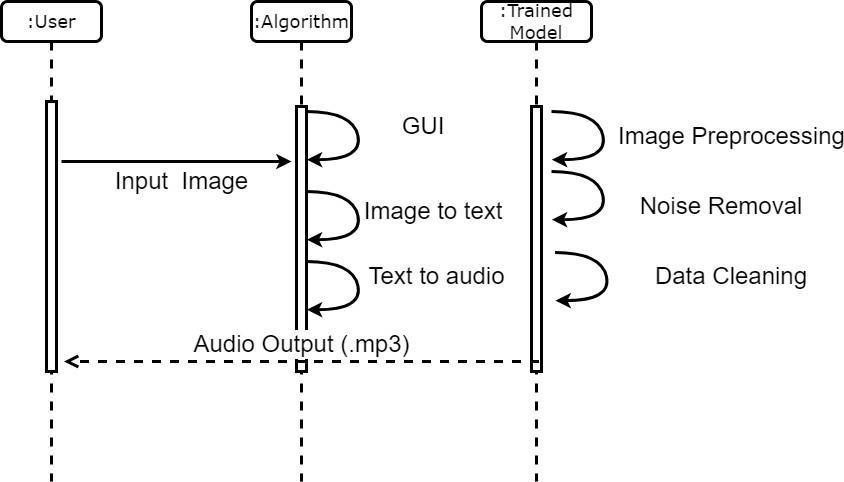
In simple terms, a diagram that represents the order of all activities is called the activity diagram. It shows the workflow between different activities that take place in the whole process. However, these are not exactly flowcharts but are similar.



**Fig 4.3 – Activity Diagram**

## SEQUENCE DIAGRAM

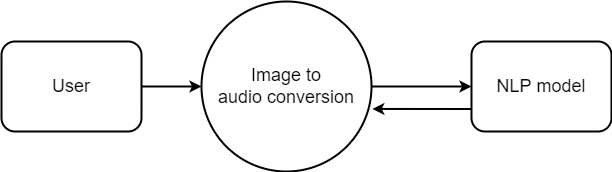
These are other kinds of interaction-based diagrams that show how all the operations are carried out. They capture the context of collaborations between objects and processes.



**Fig 4.4 – Sequence Diagram**

## DATA FLOW DIAGRAM LEVEL 0

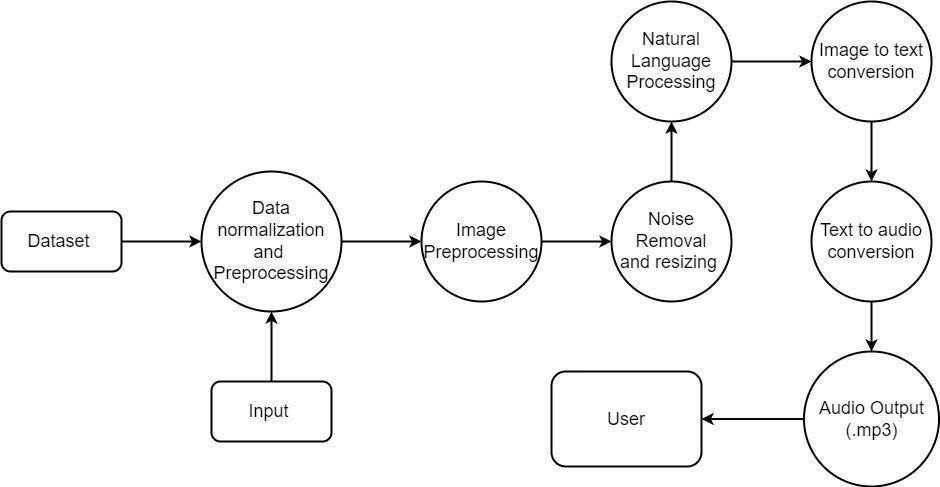
This is basically a contextual diagram, also referred to as a “context diagram”. It only represents the top level or the 0 Level in the whole process.it gives an abstraction kind of view and shows the whole system as a single process and its relationship to externalities.



**Fig 4.5 – Data Flow Diagram 0**

## DATA FLOW DIAGRAM LEVEL 1

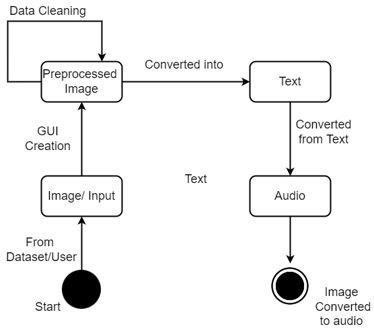
Level DFDs represent the complete system as a single process. it notates every process and sub-process that comes together in a sequence to form the complete system. This along with ) and 2-level data flow diagrams comprise the “fundamental system model”.



**Fig 4.6 – Data Flow Diagram**

* 1. **State Diagram**

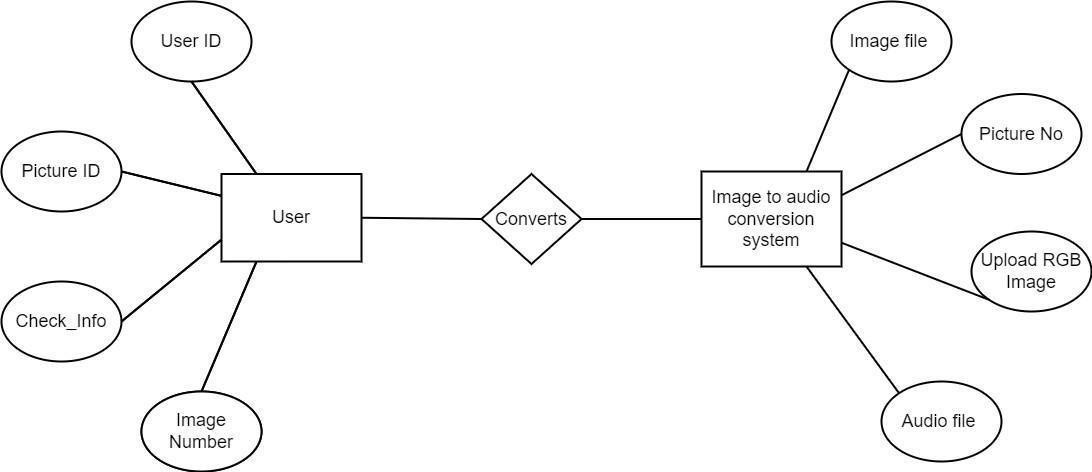
A state diagram is used to represent the condition of the system or part of the system at finite instances of time. It's a behavioral diagram and it represents the behavior using finite state transitions. State diagrams are also referred to as State machines and State-chart Diagrams



**Fig 4.7 – State Diagram**

## ER DIAGRAM

ER stands for Entity Relationship. These diagrams display the relationship of entities that are used and stored in the database. They explain the structure of the whole process. these diagrams can be made using three basic concepts, attributed, relationships, and entities.



**Fig 4.8– ER Diagram**

**CHAPTER 5**

**MODULE DESCRIPTION**

## CHAPTER 5

## MODULE DESCRIPTION

#### Data Collection and Creating GUI

The first step is data collection where libraries are imported from source websites. The dataset is of high-quality images with high resolution for better accuracy and precision. The complete system is built using GUI with higher accuracy. We use Natural Language Processing in the proposed system for better performance and accuracy.

#### Natural Language Processing

Natural language processing (NLP) is a subject of computer science— specifically, a branch of artificial intelligence (AI)—concerning the ability of computers to understand text and spoken words in the same manner that humans can.NLP blends statistical, machine learning, and deep learning models with computational linguistics—rule-based modelling of human language. These technologies work together to allow computers to process human language as a way of text or speech data and 'understand' its full meaning, including the speaker's or writer's intent and sentiment.

#### NLP Tools and Approaches Used In The Model

The Natural Language Toolkit and Python (NLTK)

For tackling specific NLP tasks, the Python programming language provides a wide range of tools and libraries. The Natural Language Toolkit, or NLTK, is an open source collection of libraries, tools, and educational resources for developing NLP programmes.

The NLTK offers libraries for many of the above-mentioned NLP tasks, as well as libraries for subtasks like sentence parsing, word segmentation, stemming and lemmatization (word-trimming methods), and tokenization (for breaking phrases, sentences, paragraphs and passages into tokens that help the computer better understand the text). It also offers libraries for developing capabilities like semantic reasoning, which allows users to draw logical inferences based on information retrieved from text.

#### Machine and Deep Learning Techniques

Statistical natural language processing (NLP) uses computer algorithms with machine learning and deep learning methods to effectively extract, classify, and label parts of text and speech input, and then assign a statistical likelihood to each probable meaning. Deep learning models and learning approaches based on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) now allow NLP systems to 'learn' as they go, extracting ever more accurate meaning from massive amounts of unstructured, unlabeled text and voice input.

Finally, we obtain high quality images in the .PNG format that are pre- processed to be converted into text systematically and efficiently.

1. Conversion of Image To Text

A bilateral filter is used to remove noise. This allows us to identify and extract only the text-containing regions while removing the unnecessary backdrop. The text is then transmitted to a second algorithm, which is in charge of editing it and recovering sentences where possible.

Finally, using the Text-to-Speech method, the result is turned to text. This is done so that OCR can turn the image to text quickly. It converts the pre- processed image, which is in .png form, to a .txt file.

1. Conversion of Text To Audio

The speech processing module is the final module. The.txt file is converted to an audio output. For the visually challenged user, the speech synthesizer translates the final material into audio output. This method artificially generates human speech. The module is responsible for converting the converted text into an audio format. It is possible to implement it in both hardware and software.

This is done using the AdaBoost Algorithm. Let us see how this works:

#### Processing Output

The first model is created, and the algorithm records faults from the first model. The improperly classified record is fed into the next model as input. This operation is repeated until the requirement is satisfied. As you can see in the diagram, 'n' models were created by using the errors from the preceding model. Boosting works in this manner. Individual models, known as decision trees, are models 1, 2, 3,..., N. The principle behind all boosting models is the same. With AdaBoost, the algorithm only makes a node with two leaves, known as Stump.

In AdaBoost, the order of the stumps is crucial. The first stump's mistake has an impact on how other stumps are constructed.

Coding the AdaBoost algorithm in Python is simple and involves about 3-4 lines. The AdaBoost classifier from the sci-kit learn library must be imported. Any dataset should be divided into train and test before implementing AdaBoost. The training data is ready to train the AdaBoost model after partitioning the data into train and test. This data contains both the input and output. Following the training data, our system will attempt to predict the outcome on the test data. Only the inputs make up the test data. The model has no idea what the result of the test data will be. By comparing the actual output of the test data with the output predicted by the model, accuracy may be determined.

Finally, the txt. Input dataset is transformed into Mp3 file for use by visually impaired people.

## MODULE SPLT UP

* + - Import of Basic Libraries And Creating GUI
    - Conversion of Image To Text
    - Conversion of Text To Audio

#### Import of Basic Libraries and Creating GUI

* + - * The first step is data collection where libraries are imported from source websites.
      * The dataset is of High quality images with high resolution for better accuracy and precision.
      * We use natural Language Processing in the proposed system for better performance and accuracy.
      * The complete system is built using GUI with higher accuracy.
      * Output of the module is that we obtain high quality images in the .PNG format which will be pre-processed to be converted into text accurately and efficiently

#### Conversion of image to text

* Noise removal is done using a bilateral filter. This enables us to detect and extract only that region that contains text and removes the unwanted background.
* Then the text is forwarded to a second algorithm in charge of correcting the text and recovering sentences when possible. Finally, the result is converted to text using the Text-to-Speech algorithm.
* This is done to allow the OCR to efficiently convert the image to text.
* It converts the pre-processed image, which is in .Png form, to a .txt file.

#### Conversion of Text to Audio (mp3)

* + - * The final module is the voice processing module. It converts the .txt file to an audio output.
      * The speech synthesizer converts the final content into audio output to the visually impaired user. This system produces human speech artificially.
      * The module performs the task of conversion of the transformed text to the audible form. It can be implemented in either hardware or software form.
      * Output of the module is The text is extracted and converted into an audio output in the mp3 format.

# Algorithms

### Adaboost

* + - Natural Language Processing
    1. **ADABOOST**
       - It combines multiple classifiers to increase the accuracy of classifiers. AdaBoost is an iterative ensemble method.
       - The basic concept behind Adaboost is to set the weights of classifiers and train the data sample in each iteration such that it ensures accurate predictions of unusual observations.
       - Any machine learning algorithm can be used as a base classifier if it accepts weights on the training set.
       - It assigns the weight to the trained classifier in each iteration according to the accuracy of the classifier.
       - The more accurate classifier will get high weight.This process iterate until the complete training data fits without any error or until reached to the specified maximum number of estimators.
       - To classify, perform a "vote" across all of the learning algorithms you built.

The AdaBoost algorithm involves using very short (one-level) decision trees as weak learners that are added sequentially to the ensemble. Each subsequent model attempts to correct the predictions made by the model before it in the sequence. This is achieved by weighing the training dataset to put more focus on training examples on which prior models made prediction errors.

AdaBoost combines the predictions from short one-level decision trees, called decision stumps, although other algorithms can also be used. Decision stump algorithms are used as the AdaBoost algorithm seeks to use many weak models and correct their predictions by adding additional weak models. The contribution of each model to the ensemble prediction is weighted based on the performance of the model on the training dataset.

The training algorithm involves starting with one decision tree, finding those examples in the training dataset that were misclassified, and adding more weight to those examples. Another tree is trained on the same data, although now weighted by the misclassification errors. This process is repeated until a desired number of trees are added.

The algorithm was developed for classification and involves combining the predictions made by all decision trees in the ensemble. A similar approach was also developed for regression problems where predictions are made by using the average of the decision trees. The contribution of each model to the ensemble prediction is weighted based on the performance of the model on the training dataset.

AdaBoost can be challenging to configure as the algorithm as many key hyperparameters that influence the behavior of the model on training data and the hyperparameters interact with each other. As such, it is a good practice to use a search process to discover a configuration of the model hyperparameters that works well or best for a given predictive modeling problem. Popular search processes include a random search and a grid search.

In this section we will look at grid searching common ranges for the key hyperparameters for the AdaBoost algorithm that you can use as starting point for your own projects. This can be achieving using the Grid Search CV class and specifying a dictionary that maps model hyperparameter names to the values to search. A similar approach was also developed for regression problems where predictions are made by using the average of the decision trees. The contribution of each model to the ensemble prediction is weighted based on the performance of the model on the training dataset.

In this case, we will grid search two key hyperparameters for AdaBoost: the number of trees used in the ensemble and the learning rate. We will use a range of popular well performing values for each hyperparameter. Each configuration combination will be evaluated using repeated k-fold cross-validation and configurations will be compared using the mean score, in this case, classification accuracy.

## Natural Language Processing

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software.

**Step #1:** Sentence Segmentation: Breaking the piece of text in various sentences.

**Step #2**: Word Tokenization: Breaking the sentence into individual words known as tokens

**Step #3**: Predicting Parts of Speech for each token: Predicting whether the word is a noun, verb, adjective, adverb, pronoun, etc.

**Step #4:** Identifying stop words

**Step #5:** Dependency Parsing: This means finding out the relationship between the words in the sentence and how they are related to each other.

**Step #6:** Named Entity Recognition(NER): Kinds of objects that a typical NER system can tag: People’s names. Company names. Geographical locations Product names. Date and time. Amount of money.

Natural Language Toolkit, is a Python package that you can use for NLP. A lot of the data that you could be analyzing is unstructured data and contains human-readable text.This project has been built around Raspberry Pi processor board. It is controlling the peripherals like Camera and speaker which act as an interface between the system and the user. Optical Character Recognition or OCR is implemented in this project to reco gnize characters which are then read out by the system through a speaker.

The camera is mounted on a stand in such a position that if a paper is placed in front of camera, it captures a full view of the paper into the system. Also, when the camera takes the snapshot of the paper, it is ensured that there are good lighting conditions.

A lot of the data that you could be analyzing is unstructured data and contains human-readable text. Before you can analyze that data programmatically, you first need to preprocess it. In this tutorial, you’ll take your first look at the kinds of text preprocessing tasks you can do with NLTK so that you’ll be ready to apply them in future projects. You’ll also see how to do some basic text analysis and create visualizations.

By tokenizing, you can conveniently split up text by word or by sentence. This will allow you to work with smaller pieces of text that are still relatively coherent and meaningful even outside of the context of the rest of the text. It’s your first step in turning unstructured data into structured data, which is easier to analyze.

Stemming is a text processing task in which you reduce words to their [root](https://simple.wikipedia.org/wiki/Root_(linguistics)), which is the core part of a word. For example, the words “helping” and “helper” share the root “help.” Stemming allows you to zero in on the basic meaning of a word rather than all the details of how it’s being used. NLTK has [more than one stemmer](http://www.nltk.org/howto/stem.html), but you’ll be using the [Porter stemmer](https://www.nltk.org/_modules/nltk/stem/porter.html).

**CHAPTER 6**

**TESTING**

## CHAPTER 6

**TESTING**

#### Testing Methodologies

There are many different types of testing methods or techniques used as part of the software testing methodology. Some of the important testing methodologies are:

#### SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise, the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as a walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

* Static analysis is used to investigate the structural properties of the Source code.
* Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

#### Unit Testing

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

#### Functional Tests

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

* Performance Test
* Stress Test
* Structure Test

#### Performance Test

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

#### Stress Test

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

#### Structure Test

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White- Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

* Exercise all logical decisions on their true or false sides.
* Execute all loops at their boundaries and within their operational bounds.
* Exercise internal data structures to assure their validity.
* Checking attributes for their correctness.
* Handling end of file conditions, I/O errors, buffer problems, and textual errors in the output information

#### Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules that makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach evolved from the unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that interface dispenses can be easily found and corrected.

The major error that was faced during the project is a linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

#### TESTING TECHNIQUES / TESTING STRATEGIES

Testing is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet

undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected

before live operation commences. It verifies that the whole set of programs hangs together. System testing requires a test consisting of several key activities and steps for runninga program, string, and system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise, the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that has a probability of finding a yet undiscovered error. A successful test is one that uncovers a yet undiscovered error. Any engineering product can be tested in one of the two ways:

#### White-box testing

This testing is also called Glass box testing. In this testing, by knowing the specific functions that a product has been designed to perform tests can be conducted that demonstrate each function is fully operational and at the same time search for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases.

Basis path testing is white box testing.

* Flow graph notation
* Cyclometric complexity
* Deriving test cases
* Graph matrices Control

#### Black box testing

In this testing by knowing the internal operation of a product, a test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black-box test case design are:

* + Graph-based testing methods
  + Equivalence partitioning
  + Boundary value analysis
  + Comparison testing

#### SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason, a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

* Testing begins at the module level and works “outward” toward the integration of the entire computer-based system.
* Different testing techniques are appropriate at different points in time.
* The developer of the software and an independent test group conducts testing.
* Testing and Debugging are different activities but debugging must be acc
* accommodated in any testing strategy.

#### Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with it. Individual modules, which are highly prone to interface errors, should

not be assumed to work instantly when we put them together. The problem, of course, is “putting them together”- interfacing. There may be the chances of data loss across another’s sub-functions when combined may not produce the desired major function; individually acceptable impressions may be magnified to unacceptable levels; global data structures can present problems.

#### Program Testing

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax errors. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items, and invalid combinations. Since the compiler s will not deduct logical errors, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or an arithmetic expression. The condition testing method focuses on testing each condition in the program the purpose the of condition test is to deduct not only errors in the condition of a program but also other errors in the program.

#### Security Testing

Security testing attempts to verify the protection mechanisms built into a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attacks must also be tested for invulnerability from rear attacks. During security, the tester places the role of the individual who desires to penetrate the system.

#### Validation Testing

At the culmination of integration testing, the software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing

can be defined in many ways, but a simple definition is that validation succeeds when the software functions in a manner that is reasonably expected by the customer. Software validation is achieved through a series of black-box tests that demonstrate conformity with requirements. After the validation test has been conducted, one of two conditions exists.

* The function or performance characteristics conform to specifications and are accepted.
* A validation from the specification is uncovered and a deficiency is created.

Deviation or errors discovered at this step in this project are corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

#### User Acceptance Testing

User acceptance of the system is a key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective systems and users at the time of developing and making changes whenever required. This is done in regard to the following points.

* Input screen design.
* Output screendesign

.

## CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENTS**

## CHAPTER 7

**CONCLUSION AND FUTURE ENHANCEMENTS**

#### CONCLUSION

There is a need for some application to help the visually impaired people and old people to detect the text in order to identify the medicine and be able to read other written work.

Thus, we propose a model for the image to audio conversion for blind people using machine learning techniques.The purpose of delivering the output in form of voice/speech is to serve the information that is present on the document to the visually impaired.The proposed system will help the user to take a picture or scan the document presented with the user using the phone’s camera, the image will be scanned and the application will read the text written in English language and convert the output in speech format. The system is low-cost, lightweight, simple, and easy to use.

#### FUTURE ENHANCEMENTS

IOT device can be used and the programs can be implemented by adding IOT module and thus the accuracy can be improved

## APPENDIX 1

#### CODING

#importing the Libraries from gtts import gTTS import PIL

import gtts

import pytesseract

from tkinter import filedialog from tkinter import \*

from PIL import Image,ImageTk import pyperclip

print("All Libraries Imported") #defining the Window window = Tk() window.geometry('1280x832') window.resizable(0, 0)

window.title("Text Recognition System for Visually Impaired") image=Image.open("space-background-6.jpg") photo=ImageTk.PhotoImage(image) lab=Label(image=photo,bg='#8fb5c2')

lab.pack()

#Defining the Labels

message = Label(window, text="Text Recognition System for Visually Impaired" ,bg="#000000" ,fg="white" ,width=50 ,height=3,font=('Helvetica', 35, 'italic bold '))

message.place(x=60, y=10)

message = Label(window, text="" ,bg="white" ,fg="black" ,width=50

,height=12, activebackground = "yellow" ,font=('Helvetica', 20 , ' bold ')) message.place(x=180, y=170)

lbl2 = Label(window, text="Enter the text :",width=16 ,fg="white"

,bg="#00008B" ,height=2 ,font=('Helvetica', 20 , ' bold ')) # #000000 lbl2.place(x=90, y=490)

txt2 = Entry(window,width=40 ,bg="white" ,fg="black",font=('Helvetica', 20 , ' bold '))

txt2.place(x=330, y=505)

def Picture\_to\_Text():

window.filename = filedialog.askopenfilename() img= PIL.Image.open(window.filename)

result= pytesseract.image\_to\_string(img)

res = "\*\*\*Successfully Converted to Text\*\*\*\n" + result pyperclip.copy(res)

message.configure(text= res) if(result==""):

res = "Oops, No Text Recognized" message.configure(text= res)

def Picture\_to\_Speech():

window.filename = filedialog.askopenfilename() img= PIL.Image.open(window.filename)

result= pytesseract.image\_to\_string(img) if(result==""):

res = "Oops, No Text Recognized" message.configure(text= res)

res= gTTS(result) #Internet required window.filename = filedialog.asksaveasfilename() res.save(window.filename+ '.mp3')

res = "Audio Saved Successfully" message.configure(text= res)

def Text\_to\_Speech(): textInp= (txt2.get()) res= gTTS(textInp)

window.filename = filedialog.asksaveasfilename() res.save(window.filename+ '.mp3')

res = "Audio Saved Successfully" message.configure(text= res)

#Defining the buttons

pictext = Button(window, text="Picture\_to\_Text", command=Picture\_to\_Text

,fg="red" ,bg="white" ,width=20 ,height=3, activebackground = "yellow"

,font=('Helvetica', 15 , ' bold ')) pictext.place(x=1000, y=170)

picspeech = Button(window, text="Picture\_to\_Speech", command=Picture\_to\_Speech ,fg="red" ,bg="white" ,width=20 ,height=3, activebackground = "yellow" ,font=('Helvetica', 15 , ' bold ')) picspeech.place(x=1000, y=270)

textspeech = Button(window, text="Text\_to\_Speech", command=Text\_to\_Speech ,fg="red" ,bg="white" ,width=20 ,height=3, activebackground = "yellow" ,font=('Helvetica', 15 , ' bold ')) textspeech.place(x=1000, y=370)

quitWindow = Button(window, text="QUIT", command=window.destroy

,fg="red" ,bg="white" ,width=17 ,height=2, activebackground = "yellow"

,font=('Helvetica', 15 , ' bold ')) quitWindow.place(x=1005, y=500)

window.mainloop() from PIL import Image

import math, wave, array, sys, getopt

def start(inputfile, outputfile, duration): im = Image.open(inputfile)

width, height = im.size rgb\_im = im.convert('RGB')

durationSeconds = float(duration) tmpData = []

maxFreq = 0

data = array.array('h') sampleRate = 44100

channels = 1

dataSize = 2

numSamples = int(sampleRate \* durationSeconds) samplesPerPixel = math.floor(numSamples / width)

C = 20000 / height

for x in range(numSamples): rez = 0

pixel\_x = int(x / samplesPerPixel) if pixel\_x >= width:

pixel\_x = width -1

for y in range(height):

r, g, b = rgb\_im.getpixel((pixel\_x, y)) s = r + g + b

volume = s \* 100 / 765

if volume == 0: continue

freq = int(C \* (height - y + 1))

rez += getData(volume, freq, sampleRate, x)

tmpData.append(rez) if abs(rez) > maxFreq:

maxFreq = abs(rez)

for i in range(len(tmpData)): data.append(32767 \* tmpData[i] / maxFreq)

f = wave.open(outputfile, 'w')

f.setparams((channels, dataSize, sampleRate, numSamples, "NONE", "Uncompressed"))

f.writeframes(data.tostring())

f.close()

def getData(volume, freq, sampleRate, index):

return int(volume \* math.sin(freq \* math.pi \* 2 \* index /sampleRate))

if \_name\_ == '\_main\_': inputfile = '' outputfile = '' duration = ''

try:

opts, args = getopt.getopt(sys.argv[1:], "hi:o:t:") except getopt.GetoptError:

print 'imgencode.py -i <input\_picture> -o <output.wav> -t

<duration\_seconds>' sys.exit(2)

for opt, arg in opts: if opt == '-h':

print 'imgencode.py -i <input\_picture> -o <output.wav> -t

<duration\_seconds>'

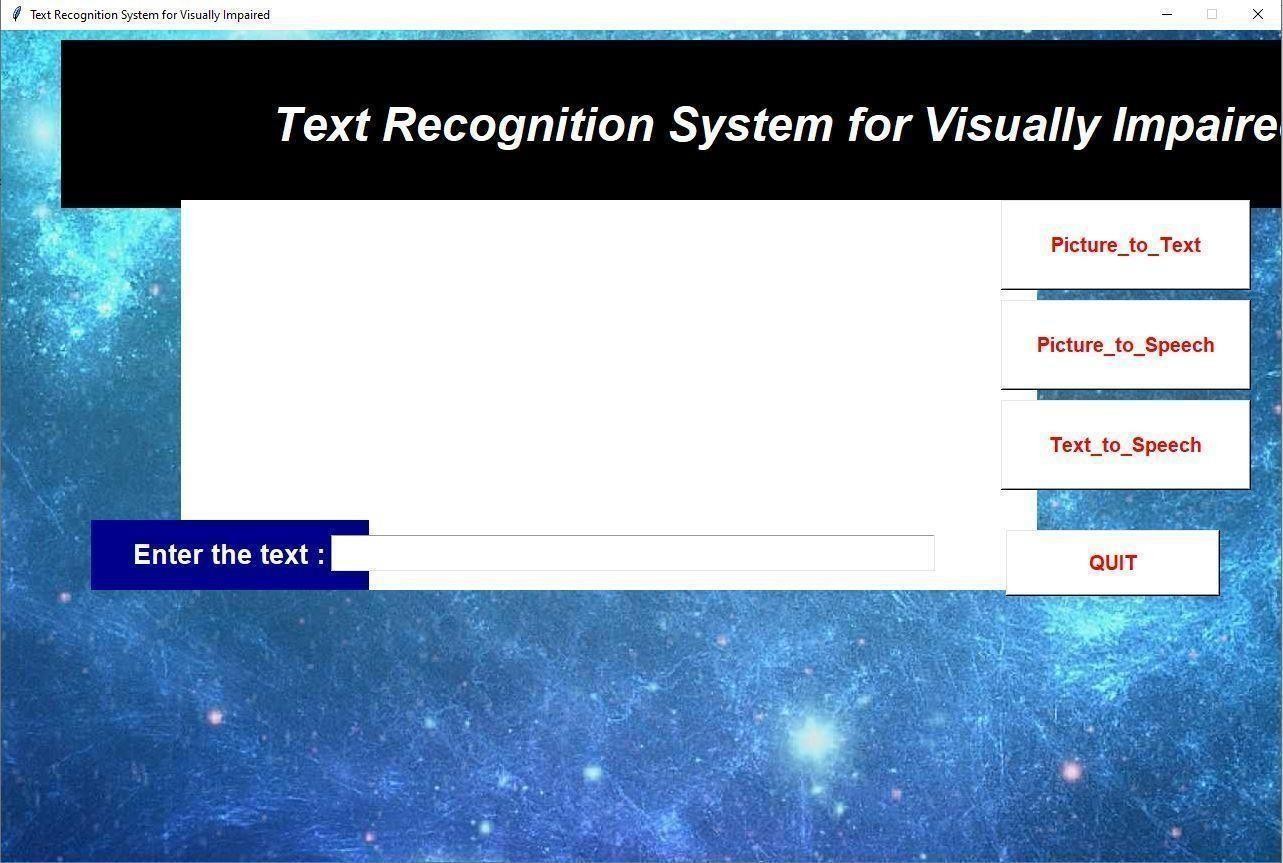
sys.exit() elif opt == "-i":

inputfile = arg elif opt == "-o":

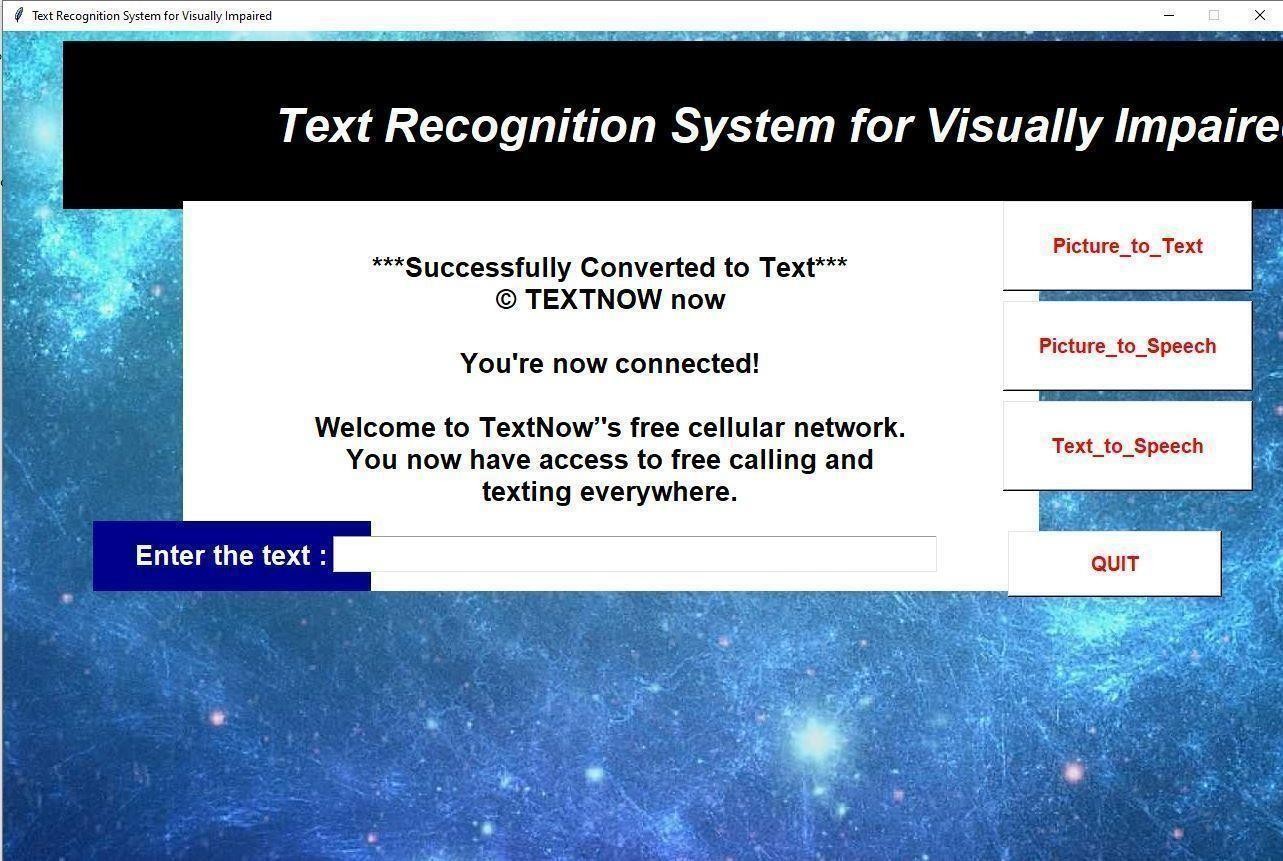
outputfile = arg elif opt == "-t":

duration = arg start(inputfile, outputfile, duration)

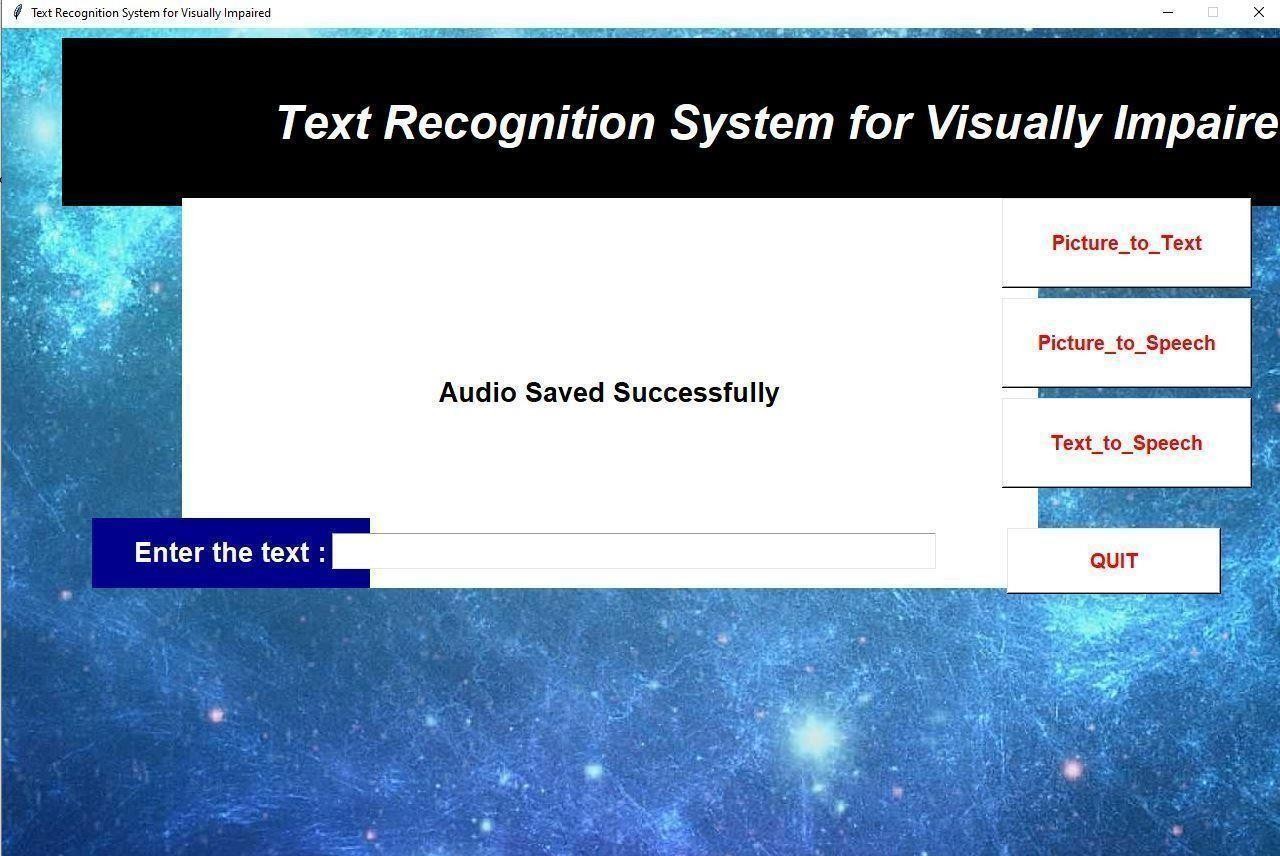
## APPENDIX 2



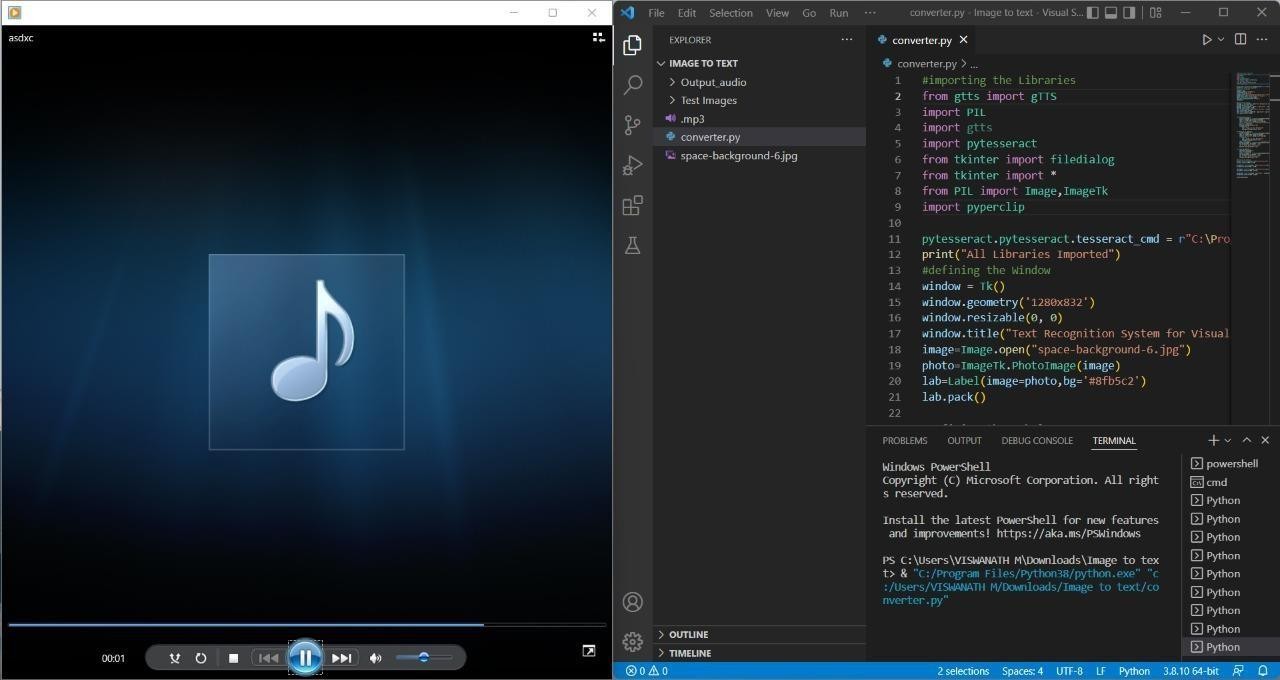
#### Fig 7.1 Input screen.



**Fig 7.2 text converted**



#### Fig 7.3 conversion of audio.



**Fig 7.4 Audio Output.**

## REFERENCES

1. Baskaran S:Lakshmi D, Image to Audio Conversion using Portable Camera January 2018 Journal of Electrical & Electronic Systems
2. Sai Aishwarya Edupuganti; Vijaya Durga Koganti; Cheekati Sri Lakshmi; Ravuri Naveen Kumar, “Text and Speech Recognition for Visually Impaired People using Google Vision,” 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), 2021
3. Asha G. Hagargund, Sharsha Vanria Thota, Mitadru Bera, Eram Fatima Shaik “Image to speech conversion for visually impaired,” International Research Journal of Engineering and Technology (IRJET), Volume 03, Issue, 2017
4. Samruddhi Deshpande; Revati Shriram, “Real time text detection and recognition on hand held objects to assist blind people,” 2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 2016
5. Prabhakar Manage; Veeresh Ambe; Prayag Gokhale; Vaishnavi Patil, “An Intelligent Text Reader based on Python,” 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020

[6]. D.Velmurugan, M.S.Sonam, S.Umamaheswari, S.Parthasarathy, K.R.Arun[2016]. A Smart Reader for Visually Impaired People Using Raspberry PI. International Journal of Engineering Science and Computing IJESC Volume 6 Issue No. 3.

[7]. 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.

[8]. Silvio Ferreira, C´eline Thillou, Bernard Gosselin. From Picture to Speech: An Innovative Application for Embedded Environment. Faculté Polytechnique de Mons, Laboratoire de Théorie des Circuits et Traitement du Signal Bˆatiment Multitel - Initialis, 1, avenue Copernic, 7000, Mons, Belgium.

[9]. Nagaraja L, Nagarjun R S, Nishanth M Anand, Nithin D, Veena S Murthy [2015]. Vision based Text Recognition using Raspberry Pi. International Journal of Computer Applications (0975 – 8887) National Conference on Power Systems & Industrial Automation.

[10]. Poonam S. Shetake, S. A. Patil, P. M. Jadhav [2014] Review of text to speech conversion methods.

[11]. Michael McEnancy Finger Reader Is audio reading gadget for Index Finger IJECCE Vol. 5, Issue 4 July2014.

[12] N Giudice, G Legge, Blind navigation and the role of technology, in The Engineering Handbook of Smart Technology for Aging, Disability and Independence, AA Helal, M Mokhtari, B Abdulrazak, Eds. Hoboken, NJ, USA: Wiley, 2008