# “Talkie Text: Text to Speech”

Submitted as Third Year Mini Project 2B

### by:

| **NAME** | **ROLL NO.** |
| --- | --- |
| Ruchita Dolas | 15 |
| Aditi Mishra | 29 |
| Mehul Nikumbh | 33 |
| Atharva Bhaindarkar | 52 |

***Mentor : Dr. T Rajani Mangala***



Department of Electronics Engineering

V.E.S. Institute of Technology 2022-23

**CERTIFICATE**

This is to certify that the project entitled “**Talkie Text: Text to Speech**” is a bonafide work of **“Ruchita Dolas (15)**, **Aditi Mishra (29)** and **Mehul Nikumbh(33), Atharva Bhaindarkar (52)**.” submitted to the V.E.S. Institute of Technology as a Third Year Mini Project 2B during Academic year 2022-23.

Dr. Rajani Mangala

### Supervisor/Guide

Mrs. Kavita Tewari Name & Sign

**Head of Department, Electronics Engineering Principal**

## PROJECT REPORT APPROVAL

This project report is entitled “**Talkie Text: Text to Speech**” by **Atharva Bhaindarkar (52)**, **Ruchita Dolas (15)**, **Aditi Mishra (29)** and **Mehul Nikumbh (33)**, is approved as Third Year mini project 2B during the academic year **2022- 2023**.

## Examiners

### 1.

**2.**

## DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature Signature

### Ruchita Dolas (15) Aditi Mishra (29)

Signature Signature

### Mehul Nikumbh (33) Atharva Bhaindarkar (52)

Date:

**INDEX**

| **Sr.no** | **Content** | **PageNo.** |
| --- | --- | --- |
| 1 | Introduction | 1 |
| 2 | Literature Review | 3 |
| 3 | Block Diagram and Working | 6 |
| 4 | Software | 14 |
| 5 | Results and Discussion | 16 |
| 6 | Conclusion | 23 |
| 7 | References | 25 |

# CHAPTER 1 INTRODUCTION

## Problem Statement

Visually impaired and illiterate people have difficulty in operating smartphones, due to which they are unable to decipher any text or message they have received on their smartphones and they have to rely on others to read it aloud to them. Hence there is a need for a system that reads aloud the received message/text.

## Proposed Solution

We propose using OCR to convert text appearing in any image. This image will be accepted from there user and the end output will be a speech conversion of the text in image. The user will have to upload the image on the website and the result will be generated within 10-15 seconds.

# CHAPTER 2 LITERATURE REVIEW

1. **N.K. Srivastava, et al., in their work “Netra: Smart Hand Gloves Comprises Obstacle Detection, Object Identification & OCR Text to Speech Converter for Blinds''** presented a Smart Glove that can overcome the problems faced by visually impaired people in their day to day life. This glove can extract text from any image which contains text and can convert text into speech. So that blinds can easily hear the text which they can not see. One more feature of this glove allows the blinds to identify the objects around them. The experimental results show good accuracy.
2. **Christophe Ponsard et al., in their work “An OCR-Enabled Digital Comic Books Viewer”** presented thegeneralization of user-friendly and mobile interfaces like smart phones, eBook readers and tablets has accelerated the transition of comic books to the digital format. This paper explores how these new technologies can improve the digital access to comic books. Their main contribution is the inclusion of optical character recognition within text bubble associated to comics characters. The recognised text can then be fed into a text-to-speech engine for an improved experience.
3. **Himank Dave, et al., in their work “OCR Text Detector and Audio Convertor”** presented a geometrical rectification framework which is crucial to revive the frontal-flat view of a document from one camera captured image. Tesseract is implementing an extended Short-Term Memory (LSTM) based recognition engine which may be a Recurrent Neural Network (RNN). They converted the OCR text output into an Audio output using gTTS, a screen reader application developed to convert text into speech for the OCR system.
4. **Pooja Shree H R, et al., in their work “OCR Oriented Reading System for Blind People”** presented an OCR scanning system that uses a camera application present in your smartphone consolidated with Optical Character Recognition (OCR). An AI-based reading system was developed to scan the detected content present in the image and convert it into digital text which is recognized by the system and displays the translated content and provides voice/speech output.

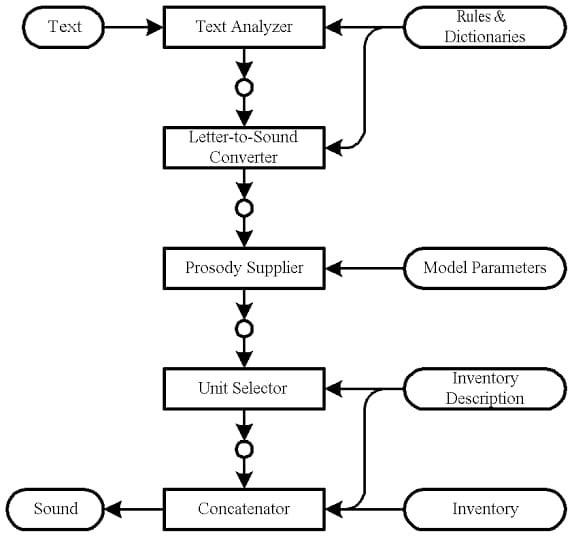
# CHAPTER 3

**BLOCK DIAGRAM & WORKING**

## Block Diagram for Text-to-Speech

In the [Fig 3.1.1] data flows from top to bottom. The boxes represent modules, and the rounded rectangles are permanent storage. The circles are the interfaces between the modules.

* + - Message is fed to the Text Analyzer, where it is analyzed and enriched with syntactic and (possibly) semantic information.
    - Letter-To-Sound Converter transcribes this enriched information into a string of phones guided by Rules and Dictionaries, where rules are used by default and dictionaries for exceptions
    - Prosody Supplier computes the prosody (intonation, duration, and intensity) information adjusted by Model Parameters. [1]
    - Unit Selector selects the appropriate units (diphones) based on the string of phones and the Inventory Description.
    - Concatenator needs the selected units and the prosody information to decompress the units from the inventory, apply pitch-shift and stretching, and concatenate the units together.



**Fig 3.1.1 Block diagram of TTS [1]**

## Block Diagram for OCR

In the [Fig 3.2.1] an API request is sent to an input image and then it is pre-processed where binarization, skew correction and noise removal is performed.

* Binarization: Binarisation means converting a coloured image into an image which consists of only black and white pixels (Black pixel value=0 and White pixel value=255).
* Skew Correction: While scanning a document, it might be slightly skewed (image aligned at a certain angle with horizontal) sometimes. While extracting the information from the scanned image, detecting & correcting the skew is crucial.
* Noise Removal: It smoothens the image by removing small dots/patches which have higher intensity than the rest of the image. Noise removal can be performed for both Coloured and Binary images.

Leptonica is an open source library containing software that is broadly useful for image processing and image analysis applications. Leptonica and trained data set is sent to the tesseract OCR engine. OCR Post Processing involves correcting errors that were generated during the OCR process. In the case that the OCRed text is a continuous thought written in a language that a reader understands, it is likely they could correct almost all errors. If the text was encrypted or has no context or meaning. Then correcting these errors is nearly impossible without the original images. After the data is post-processed, the test is generated as an API response.

****

**Fig 3.2 OCR Process Flow [2]**

## Working for Talkie Text: Text to Speech

**How does OCR work?** [8]

Optical character recognition (OCR) uses a scanner to process the physical form of a document. Once all pages are copied, OCR software converts the document into a two-color or black-and-white version. The scanned-in image or bitmap is analyzed for light and dark areas, and the dark areas are identified as characters that need to be recognized, while light areas are identified as background. The dark areas are then processed to find alphabetic letters or numeric digits. This stage typically involves targeting one character, word or block of text at a time. Characters are then identified using one of two algorithms — pattern recognition or feature recognition.

Pattern recognition is used when the OCR program is fed examples of text in various fonts and formats to compare and recognize characters in the scanned document or image file.

Feature detection occurs when the OCR applies rules regarding the features of a specific letter or number to recognize characters in the scanned document. Features include the number of angled lines, crossed lines or curves in a character. For example, the capital letter “A” is stored as two diagonal lines that meet with a horizontal line across the middle. When a character is identified, it is converted into an ASCII code (American Standard Code for Information Interchange) that computer systems use to handle further manipulations.An OCR program also analyzes the structure of a document image. It divides the page into elements such as blocks of texts, tables or images. The lines are divided into words and then into characters. Once the characters have been singled out, the program compares them with a set of pattern images. After processing all likely matches, the program presents you with the recognized text.

The final step is to assess how well our model has performed. Even if it gives high confidence scores, we need to measure performance with objective metrics. Since we cannot improve what we do not measure, these metrics serve as a vital benchmark for the iterative improvement of our OCR model.

**Character Error Rate (CER)**

We will look at the metric used to evaluate OCR output, namely Character Error Rate (CER).

(i) Equation

CER calculation is based on the concept of Levenshtein distance, where we count the minimum number of character-level operations required to transform the ground truth text (aka reference text) into the OCR output.

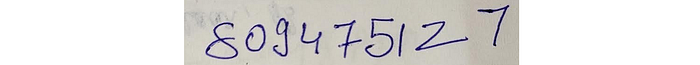
It is represented with this formula:

where,

* S = Number of Substitutions
* D = Number of Deletions
* I = Number of Insertions
* N = Number of characters in reference text (aka ground truth)

## (ii) Illustration with Example

Let’s look at an example:



Ground Truth Reference Text: 809475127

* OCR Transcribed Output Text: 80g475Z7

Several errors require edits to transform OCR output into the ground truth:

1. g instead of 9 (at reference text character 3)
2. Missing 1 (at reference text character 7)
3. Z instead of 2 (at reference text character 8)

With that, here are the values to input into the equation:

* Number of Substitutions (S) = 2
* Number of Deletions (D) = 1
* Number of Insertions (I) = 0
* Number of characters in reference text (N) = 9

Based on the above, we get (2 + 1 + 0) / 9 = 0.3333. When converted to a percentage value, the CER becomes 33.33%. This implies that every 3rd character in the sequence was incorrectly transcribed.

We repeat this calculation for all the pairs of transcribed output and corresponding ground truth, and take the mean of these values to obtain an overall CER percentage.

## (iii) CER Normalization

One thing to note is that CER values can exceed 100%, especially with many insertions. For example, the CER for ground truth ‘ABC’ and a longer OCR output ‘ABC12345’ is 166.67%.

## (iv) What is a good CER value?

There is no single benchmark for defining a good CER value, as it is highly dependent on the use case. Different scenarios and complexity (e.g., printed vs. handwritten text, type of content, etc.) can result in varying OCR performances. Nonetheless, there are several sources that we can take reference from.

* Good OCR accuracy: CER 0‐2% (i.e. 98–99% accurate)
* Average OCR accuracy: CER 2-10%
* Poor OCR accuracy: CER >10% (i.e. below 90% accurate).

# CHAPTER 4 HARDWARE & SOFTWARE

**OVERVIEW**

### Table 4.1 Tabular overview of the hardware specification

| **Sr.No** | **Hardware Component** | **Description** |
| --- | --- | --- |
| 1 | Speaker | To listen to the voice generated output |

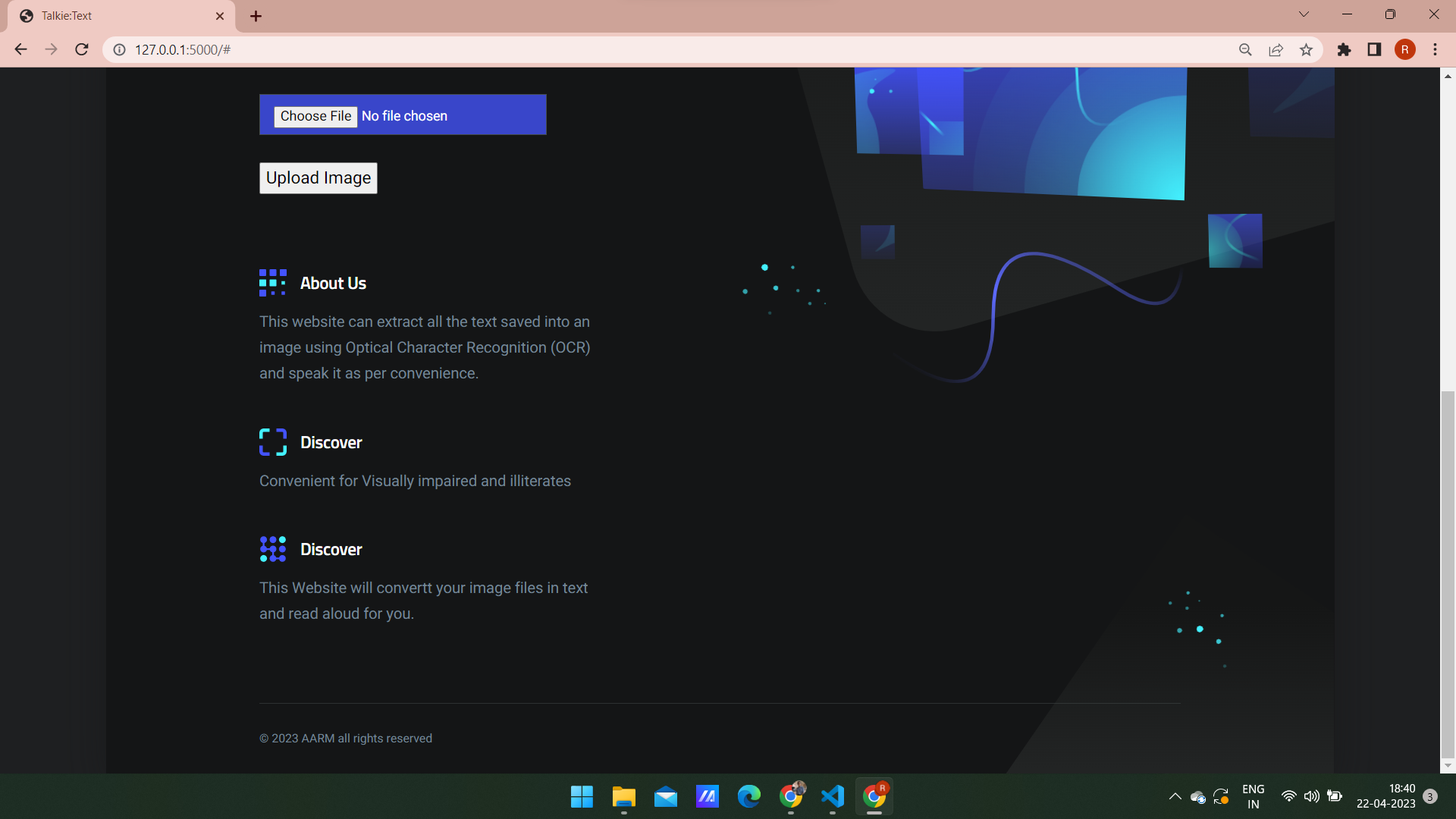
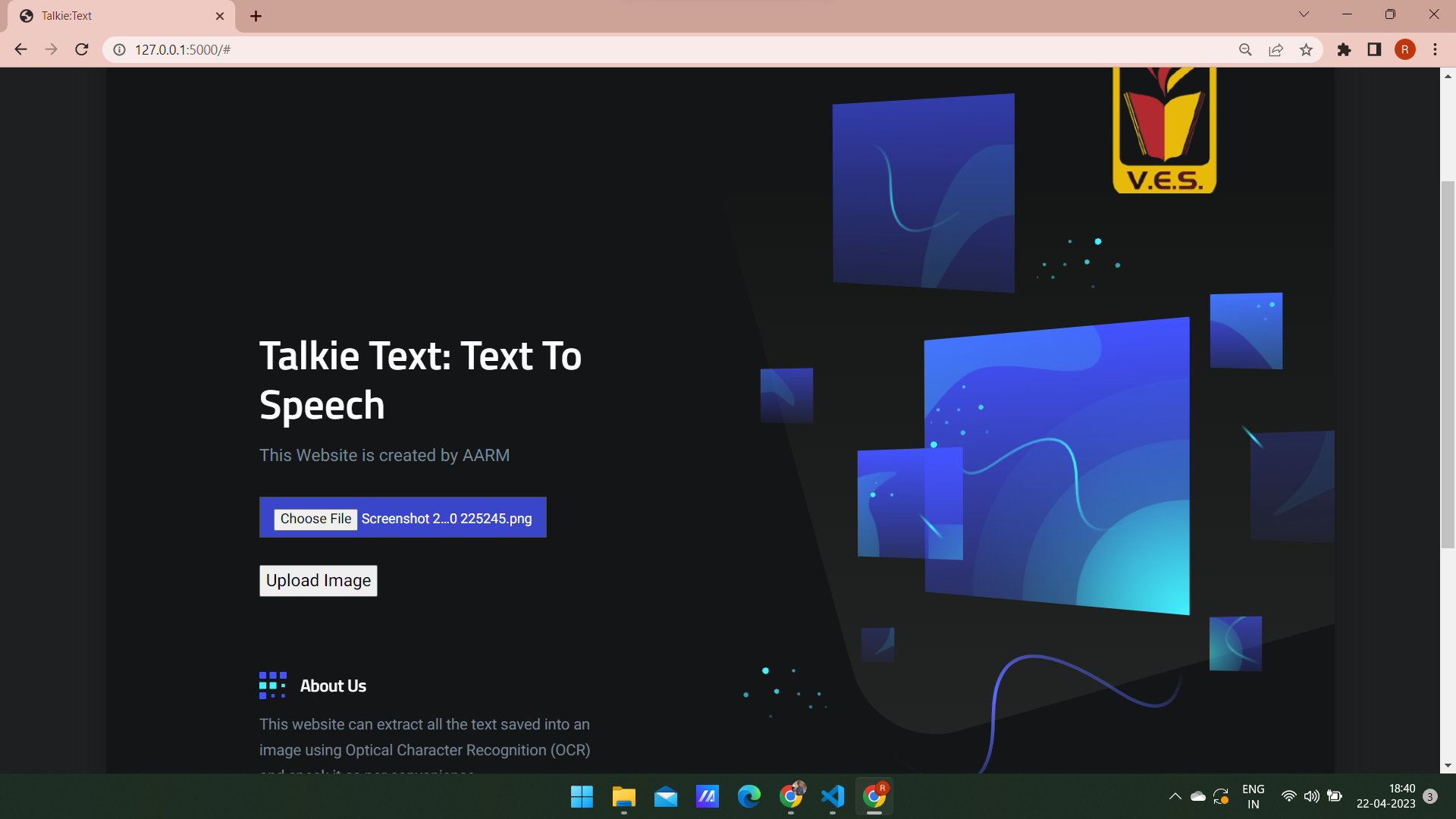
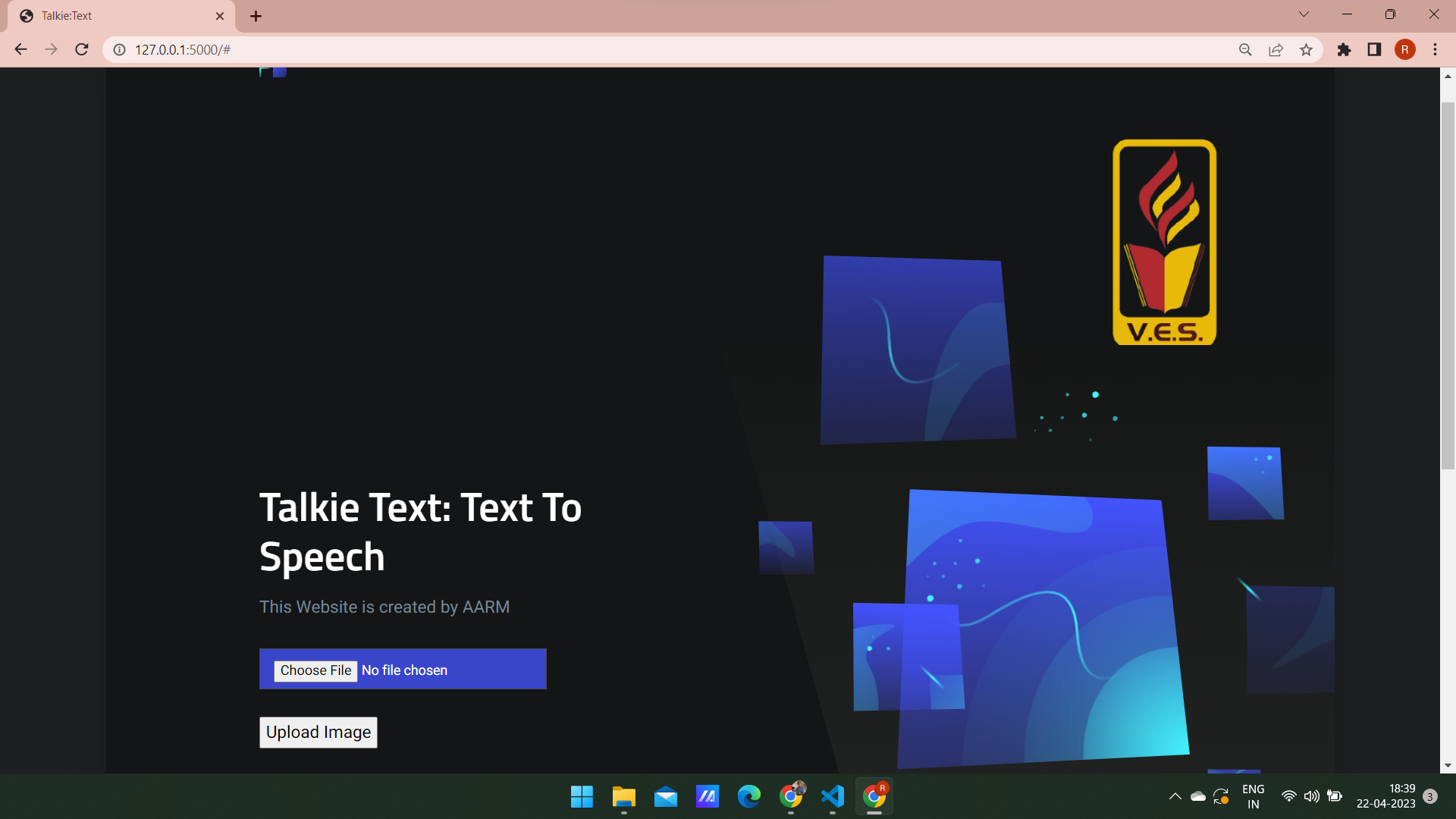
**Table 4.2 Tabular overview of the software specifications**

| **Sr.No** | **Software Required** | **Description** |
| --- | --- | --- |
| 1 | Python | Coding purpose |
| 2 | TTS | Library for advanced Text- To-Speech generation. |
| 3 | Tesseract | Tesseract tests the text lines to determine whether they are fixed pitch. |
| 4 | Google Colab | Execute arbitrary Python code |

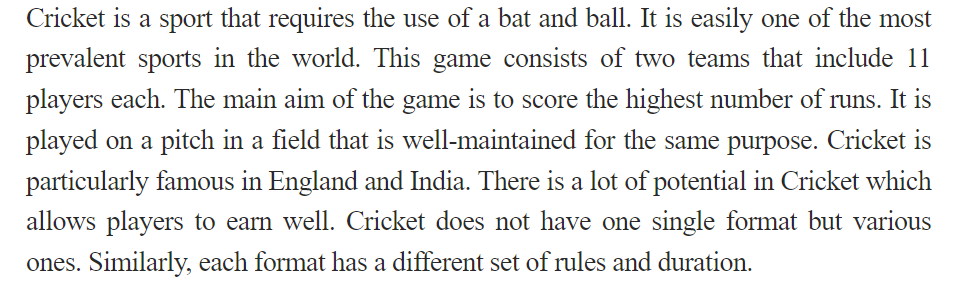
# CHAPTER 5 RESULTS & DISCUSSION

## Results

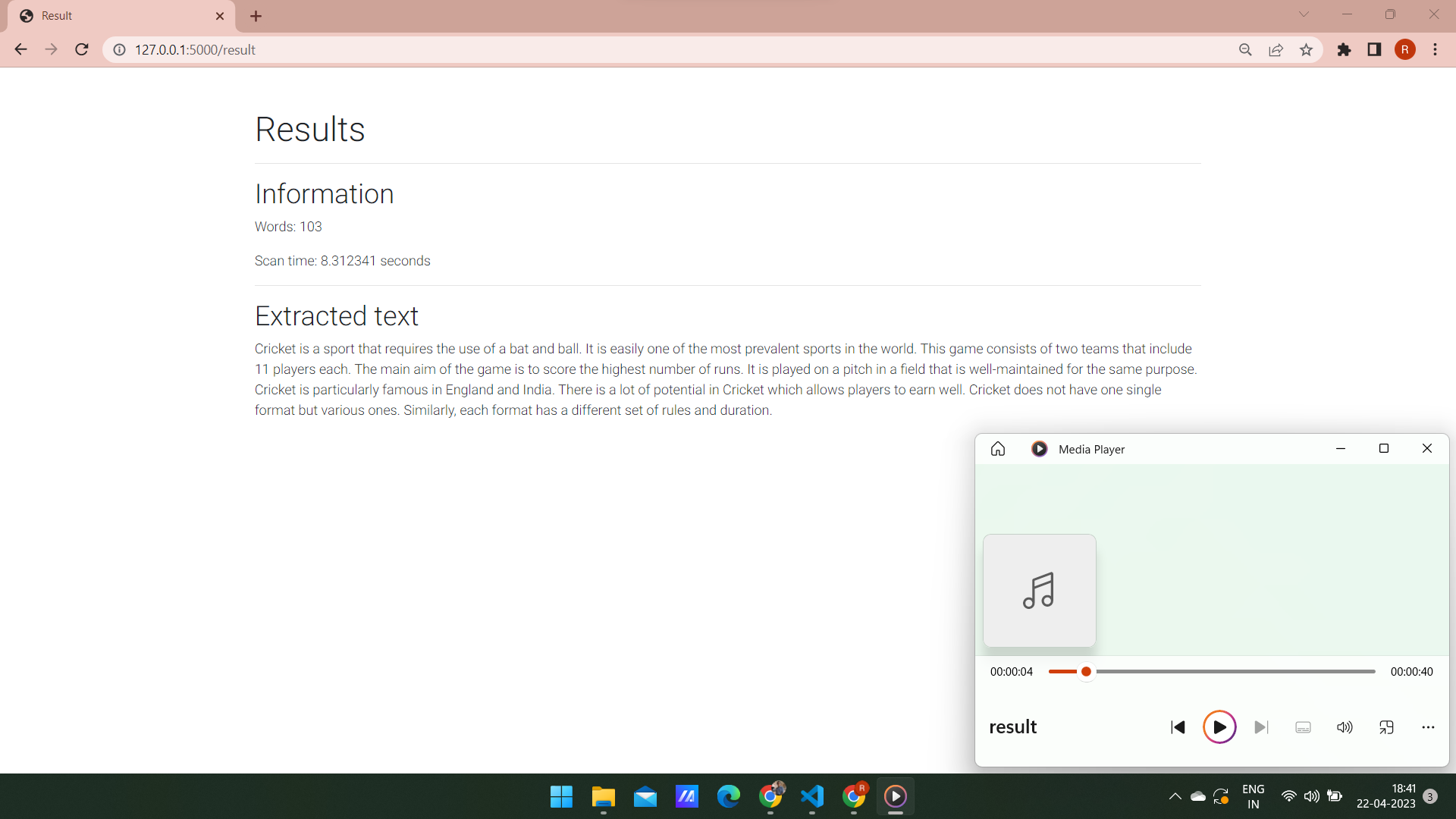
* + - Talkie Text can control the voice speed as well as part of the prosody by adjusting the phoneme duration, which cannot be supported by other end- to-end TTS systems.
    - Talkie Text can add breaks between adjacent words by lengthening the duration of the space characters in the sentence, which can improve the prosody of voice. We show where we add breaks in two positions of the sentence to improve the prosody.



**Fig 5.1 Images captured of the website**

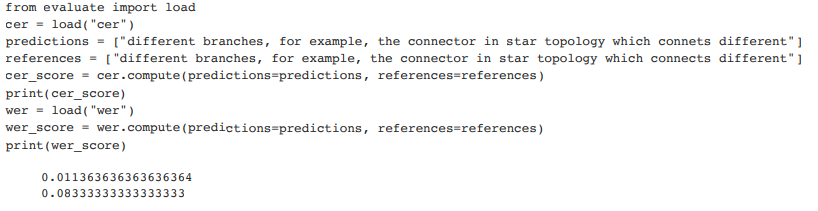
****

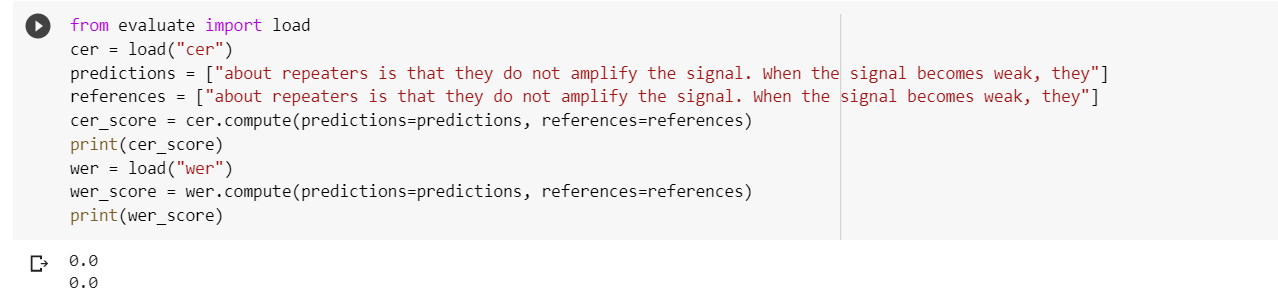
**Fig 5.2 Image uploaded for image to speech conversion**

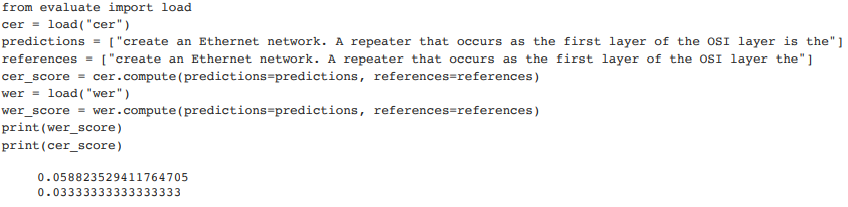
****

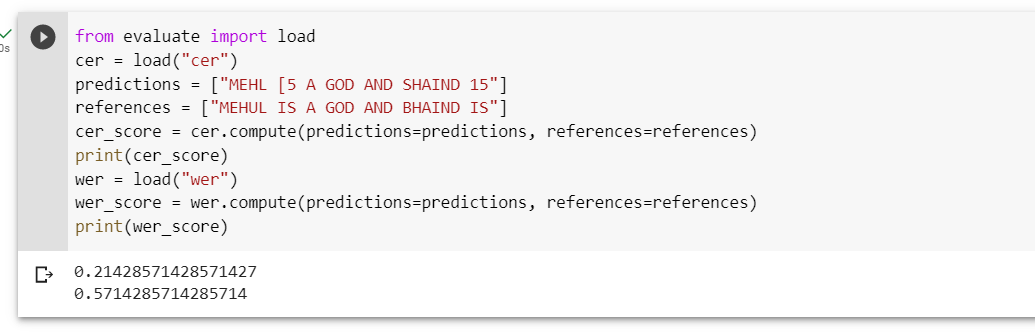
**Fig 5.3 Text from the image converted into audio**

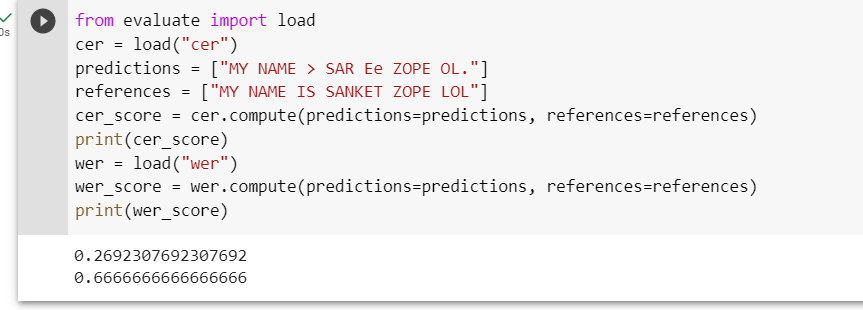
We tried our OCR model on 25 different images containing texts and we got a CER (character error rate) of 0.7 percent. Also on that same dataset of 25 images we got a WER (word error rate) of 1.17 percent which makes our OCR fall under the category of having ‘Good accuracy’ which makes it a perfect fit for our purpose.





**Fig 5.4 CER and WER calculated for printed text**

****

****

**Fig 5.5 CER and WER of handwritten text**

## Discussion

The character error rate and word error rate is 0 for 23 out of the 25 printed text images uploaded and hence it has a good accuracy which suits our purpose .

But in the case of handwritten text the image uploaded does not have a good accuracy.This is because the handwritten files are difficult to understand for a machine as it depends majorly on the handwriting of the user .We got a CER and WER of 24.16% and 62% respectively .Hence improvement needs to be done in that area.

This can be performed by training another model on handwritten dataset which will help in increasing its accuracy, but since most of the requirement in our project is on printed text we dont require extra model for handwritten text.

# CHAPTER 6 CONCLUSION

## Conclusion

In this work, we have proposed Talkie Text, a fast, robust and controllable neural TTS system. We created a responsive web page that lets a person upload an image and using OCR techniques the web pages will convert the text in the image to audio. This can be beneficial for people who understand the English language but can’t read. Audiobooks in image form are a great application of this system.

## Future Scope

* + - For future scope, the quality of synthesized speech can be improved.
    - Also, not only images but pdf files can also be converted to text and this can also be converted to audio
    - A screen recording API can be used and the text captured in the video can be read aloud.

# CHAPTER 7

# REFERENCES

## 7.1 References

1. Markus Schnell, Michael Küstner “Text-to-Speech for Low-Resource Systems”, Dresden University of Technology Dresden, Germany, in IEEE, 2002.
2. Himank Dave, Aryaman Gobse, Aryika Goel, Swati Bairagi, “OCR Text Detector and Audio Convertor”, in IJRASET, 2020.
3. N.K. Srivastava, Satyam Singh, “Netra: Smart Hand Gloves Comprises Obstacle Detection, Object Identification & OCR Text to Speech Converter for Blinds”, in IEEE, 2018.
4. Christophe Ponsard, Ravi Ramdoyal, Daniel Dziamski, “An OCR-Enabled Digital Comic Books Viewer”, in ICCHP, 2012.
5. Pooja Shree H R, Dr. Revathi, “OCR Oriented Reading System for Blind People”, in AjCt, in 2022.
6. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5820628/
7. <https://towardsdatascience.com/pre-processing-in-ocr-fc231c6035a7>

[8] <https://www.ibm.com/cloud/blog/optical-character-recognition>