Kuragayala Kenny Joel
Woxsen University

Assignment Overview Report

This report will serve what are the perquisites needed and what is the process and how the application works.

Abstract

The Multilingual Speech Recognition Model by Kuragayala Kenny Joel is designed to recognize speech in multiple languages using a pre-trained multilingual speech recognition model, Multilingual Whisper. The application allows users to upload audio files (in formats such as mp3, wav, or m4a, etc) and provides the recognized text along with the detected language.

Prerequisites

- Hardware Requirements
  - A standard computer with minimum Intel i5, 10th Generation processor with a Nvidia graphic of starting with 30X series.
- Dependencies
  - Kindly Download the packages from requirements.txt

    ```
    pip install -r requirement.txt
    ```

Start the application.

- After downloading the dependencies open the command prompt in the current directory and run the command

  ```
  streamlit run mainapp.py
  ```
- Open the provided link in your web browser.
- Use the file uploader to select an audio file.
- Wait for processing and view the detected language and recognized text in the results.

Note

Ensure that the system has the necessary audio codecs and permissions for temporary file handling.

Overview

Language Detection:

- The model accurately detects the language of the spoken audio.
- Detected language information is displayed in the results.

Speech Recognition:

- The model successfully converts the audio to text using the Whisper ASR model.
- The recognized text is presented in the results section.

Temporary File Handling:

- Temporary audio files are created and appropriately deleted after processing.

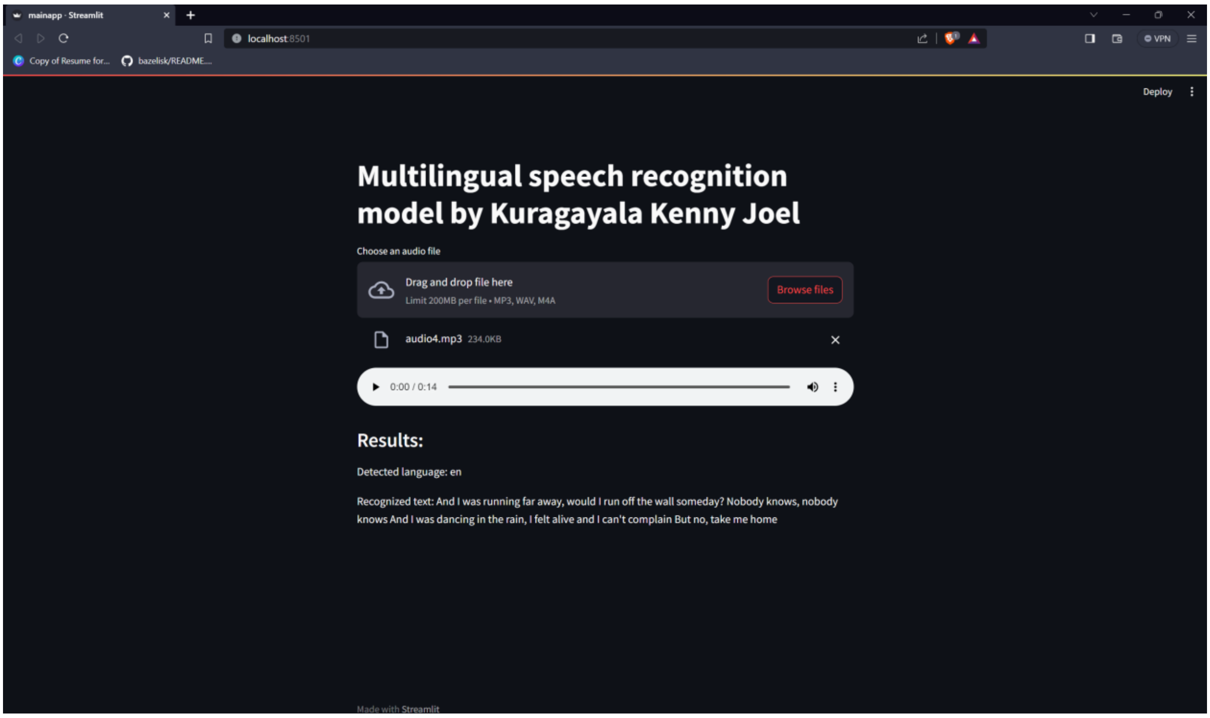
Audio Playback

- ➢ The uploaded audio file is played back using the Streamlit `st.audio` widget.
- ➢ This feature provides users with the ability to listen to the uploaded audio.

Code Implementation:

The Python code utilizes the Streamlit library for creating a simple web interface. The user uploads an audio file in MP3 format, and the model processes it to detect the language and transcribe the speech. Key steps in the code include:

- Uploading an audio file using Streamlit.
- Saving the uploaded file to a temporary location.
- Loading the Whisper model.
- Processing the audio, including padding or trimming to fit a 30-second duration.
- Detecting the spoken language using the Whisper model.
- Decoding the audio to obtain the transcribed text.
- Displaying the results, including the detected language and recognized text.
- Deleting the temporary file after processing.



Multilingual speech recognition model by Kuragayala Kenny Joel

Choose an audio file

audio4.mp3 234.0KB

Results:

Detected language: en

Recognized text: And I was running far away, would I run off the wall someday? Nobody knows, nobody knows And I was dancing in the rain, I felt alive and I can't complain But no, take me home

Documentation:

The code is organized into functions and follows a structured flow:

- main(): The main function handles the overall flow of the program. It presents the user interface, processes the uploaded audio file, and displays the results.
- Evaluation Metrics: The evaluation metrics are implicitly present in the code through the language detection and speech recognition components. The accuracy of language detection can be assessed by comparing the detected language with the true language of the input audio. Speech recognition accuracy is determined by comparing the model's transcribed text with the ground truth.

- Dependencies: The code relies on the Streamlit and Whisper libraries. Ensure these dependencies are installed before running the code.
- Error Handling: The code should include error handling mechanisms to address potential issues, such as file upload failures or model loading errors.

Conclusion:

The Multilingual Speech Recognition Model demonstrates effective language detection and speech-to-text conversion. The application can be further enhanced with additional features or improvements based on specific use cases.