Speech-to-Speech LLM Bot

1. **Introduction:**
   This project aims to build an interactive bot that comprehends spoken input, processes it using a Large Language Model (LLM) in the background, and then forms responses. Such applications will allow users and bots real-time, effective, and efficient communication through natural language processing and speech technologies.

2. **Objectives:**
   - **Speech Recognition:** Correctly transcribe spoken input from a microphone or web camera.
   - **Natural Language Processing:** Use an LLM (Large Language Model) to generate context -appropriate responses from the captured text.
   - **Text-to-Speech Conversion:** Synthesize the obtained textual response back to speech.
   - **User Interaction:** Implement an interface for easy interaction of the bot with it's user.
   - **Performance:** The system should be capable enough to process input and deliver it's response in 3 seconds of time.

3. **System Architecture:**

   i. **Introduction**
   The system architecture contains three blocks, they are:
   - **Input Processing:** Records the speech from the user, through either a microphone or a webcam.
   - **Natural Language Processing (NLP):** Processes the text using an Large Language Model (LLM).
   - **Output Generation:** Converts back the processed text into speech.

   ii. **Realization Components**
   - **Speech Recognition:** Developed using Whisper by OpenAI. It converts the speech input to text.
   - **Natural Language Processing:** Llama 2 takes the converted text and gives suitable responses.
   - **Text-to-Speech:** Pyttsx3 library is used to generate speech for the text response.
   - **User Interface:** A simple UI for taking the input methods and shows the recognized / generated text.

- **User Input:** Microphone/Webcam/Text
- **Speech Recognition (SR):** Whisper model converts speech to text
- **Natural Language Processing (NLP):** Llama 2 model generates responses
- **Text-to-Speech (TTS):** Pyttsx3 converts text to speech
- **Output:** The system outputs the speech through a speaker.

4. Implementation Details:

I. Libraries Setup and Imports:

The project uses a python libraries like speech_recognition, opencv-python-headless, torch, transformers and pyttsx3.

Loading the Model

- Whisper Model: The whisper model is taken from the Hugging Face model hub pre-trained for the speech recognition task.
- Llama 2 Model: Pre-trained for the response generation task, too, taken from the Hugging Face.

II. Input Preprocessor:

- **Microphone Input:** Bot take input from the microphone by capturing and transcribing audio with the help of library speech_recognition.
- **Webcam Input:** Bot capture the video from the webcam by using OpevCV to trigger the input audio for speech recognition.
- **Input Text:** Enables direct input from the user through the command line.

III. NLP and Response Generation:

The transcribed text is passed to Llama 2 model, and generates a relevant response.

IV. Text-to-Speech (TTS):

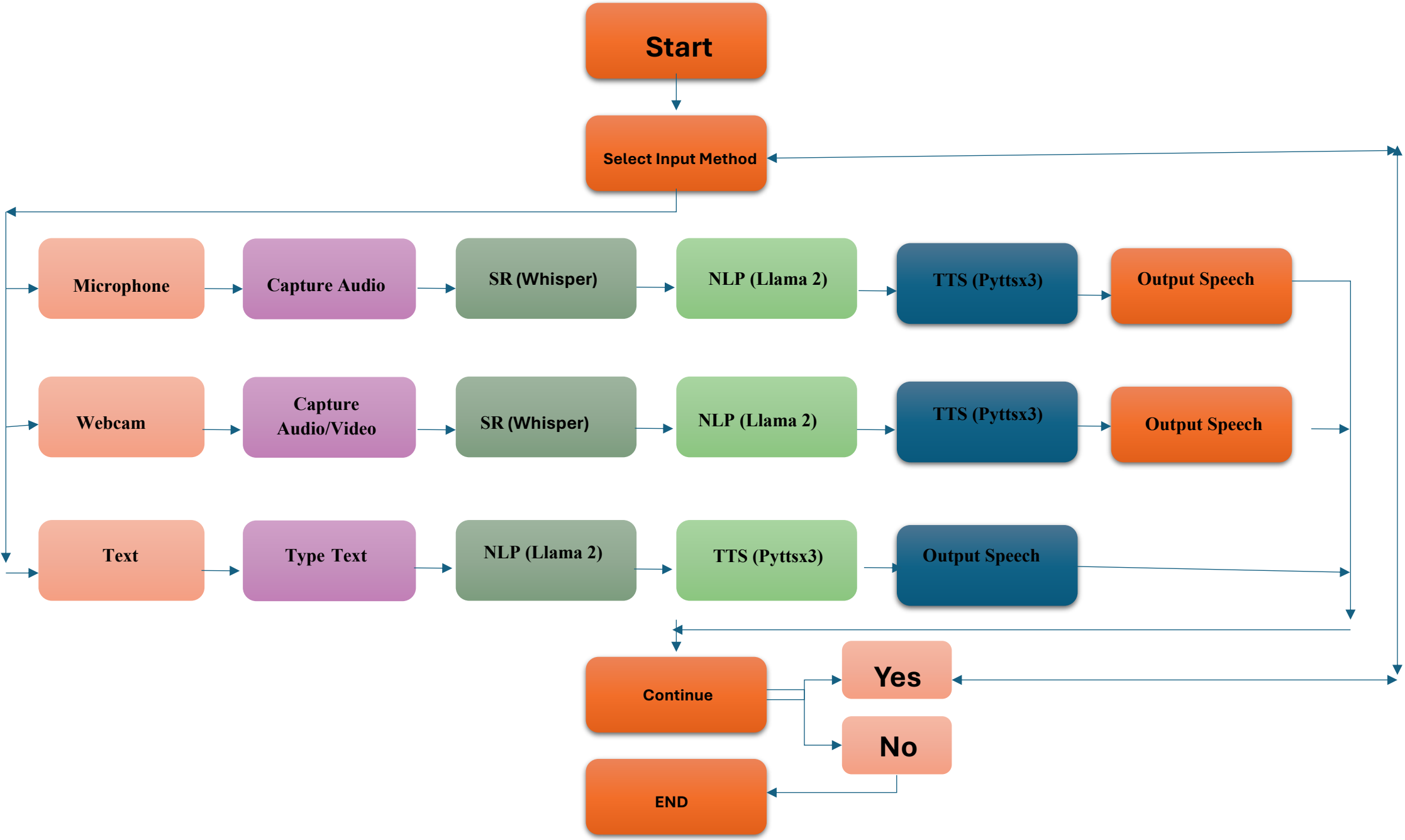
The generated text response is converted to speech using the Pyttsx3 engine.

5. End-User Flow of Interaction:

- **Input Method Selection:** The user selects either microphone, webcam, or text as the input method.
- **Speech Recognition:** The selected input is captured and transcribed.
- **NLP Processing:** Transcribed text is processed by the LLM, and a response is generated.
- **Speech Output:** The response is converted to speech and played back to the user.

6. Challenges Faced:

- **Model Limitations:** The large model couldn't be fully utilized in Google Colab due to resource constraints.
- **Colab Restrictions:** Lack of microphone and webcam in Colab led to the addition of text input as an alternative.
- **Latency Isses:** The system struggled to meet the 3-second response time due to the computatiosal demands of the models.
- **Incomplete Features:** Continuous interaction loops were not implemented due to time limits.



User Flow of Interaction

7. Conclusion:

The Speech-to-Speech LLM Bot project successfully integrates with the LLM technology to create an interactive, and real time communication tool. The system effectively processes speech input, generates contextually accurate responses, and give output them in natural sounding voice. The user-friendly interface and optimized processing ensure responses.

Click Here for the Code

Sample image of the output