GUVI | HCL

Skill Up. Level Up

நான் முதல்வன்

உலகை வெல்லும் இளைய தமிழகம்

GUVI - Naan Mudhalvan

Engineering Hackathon 2025

www.guvi.in

MULTILINGUAL SPEECH-TEXT ENGINE FOR A GLOBAL CALL CENTER

A.V.C COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION TECHNOLOGY

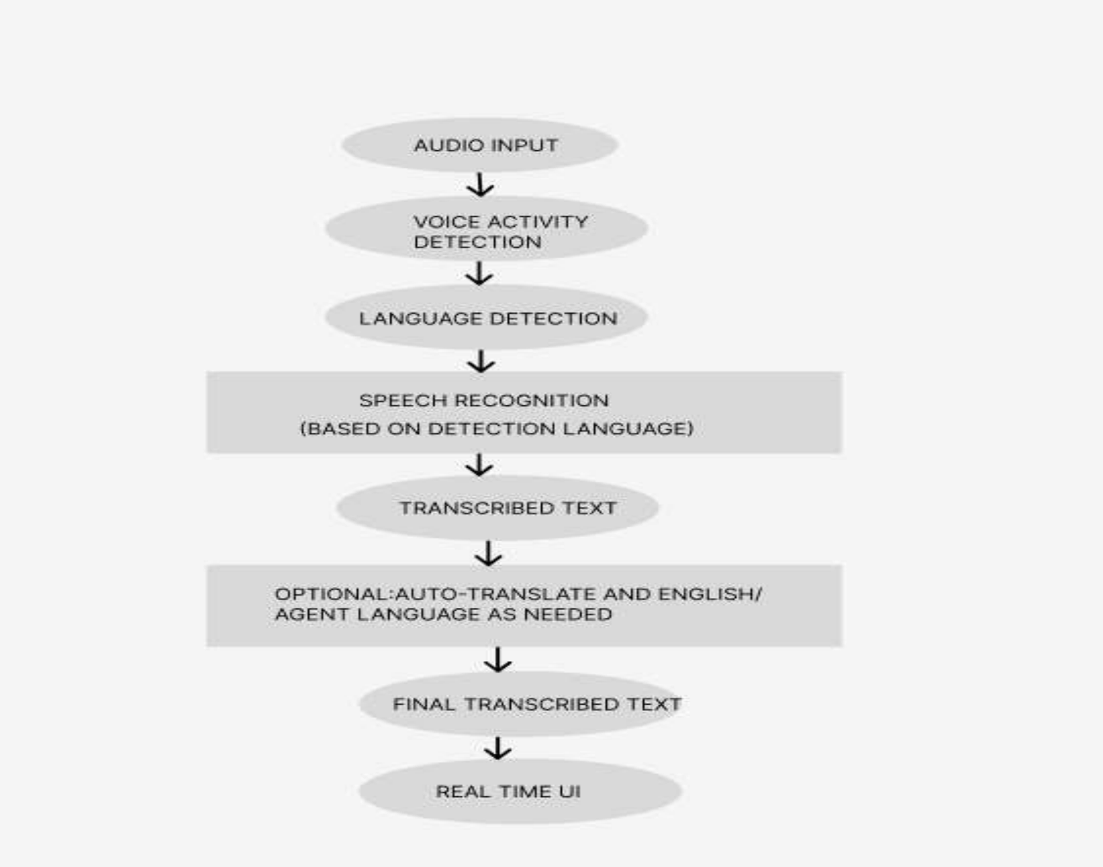
TEAM MEMBERS:

V.SWETHA
A.I.SUTHIKSHA
M.SUBHASREE
A.ABARNA
M.AMIRTHAVARSHINI

ABSTRACT

This project implements an accessible web-based interface for OpenAI's Whisper automatic speech recognition system using the Gradio framework. Whisper is a state-of-the-art multilingual ASR model trained on 680,000 hours of diverse audio data, capable of transcribing speech in 99 languages and translating non-English speech into English text. The implementation leverages Gradio's intuitive UI components to create a user-friendly interface that allows for real-time speech capture through microphone input or file upload. The system processes audio through the Whisper model pipeline, performing language detection, transcription, and optional translation. The web interface provides immediate feedback with detected language identification and text output, making advanced speech recognition technology accessible to users without technical expertise. This application demonstrates practical applications in accessibility, content creation, language learning, and cross-lingual communication, while also showcasing how complex machine learning models can be deployed for everyday use through simple yet effective web interfaces..

FLOWCHART


AUDIO INPUT
VOICE ACTIVITY DETECTION
LANGUAGE DETECTION
SPEECH RECOGNITION (BASED ON DETECTION LANGUAGE)
TRANSCRIBED TEXT
OPTIONAL:AUTO-TRANSLATE AND ENGLISH/ AGENT LANGUAGE AS NEEDED
FINAL TRANSCRIBED TEXT
REAL TIME UI

PROPOSED METHODOLOGY

Model Selection:
Use OpenAI Whisper, a Transformer-based model trained on diverse multilingual data, achieving high accuracy (~80%) even in noisy environments.

Audio Preprocessing:
Load audio, standardize length (e.g., 30s) using padding/trimming, and convert to log-Mel spectrogram features for model input.

Language Detection:
Automatically detect spoken language with model.detect_language() to guide transcription.

Decoding & Transcription:
Decode spectrogram to text using whisper.decode() with optimized settings for accuracy and speed.

Output Handling:
Produce recognized text output, which can be used directly or for further tasks like translation or NLP.

TOOLS AND LIBRARIES

Frontend: Gradio Web UI for user interaction
Backend: Python implementation using Whisper

SOFTWARE:

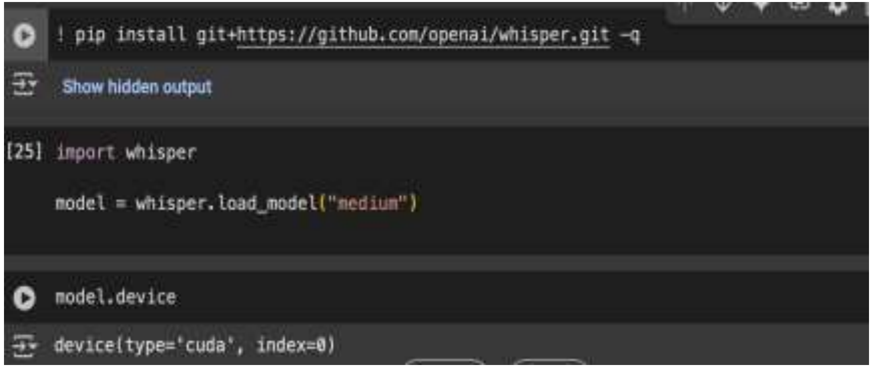
Google Colab

PROGRAMMING LANGUAGE:

Python

GRADIO:

Python library designed for ML model interfaces
Minimal code for functional UI
Built-in hosting capabilitiesResponsive design

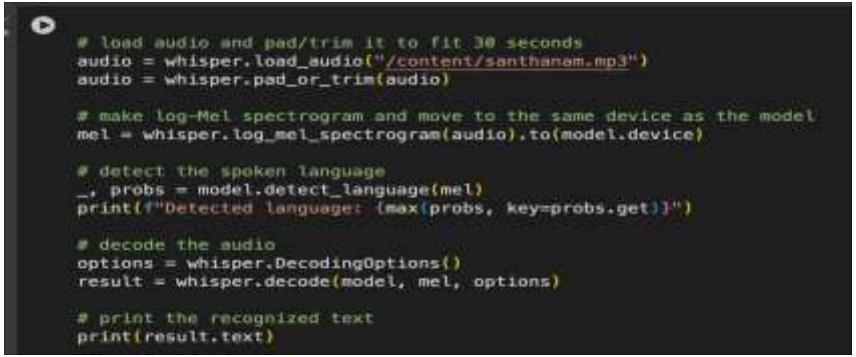


```
! pip install git+https://github.com/openai/whisper.git -q
```

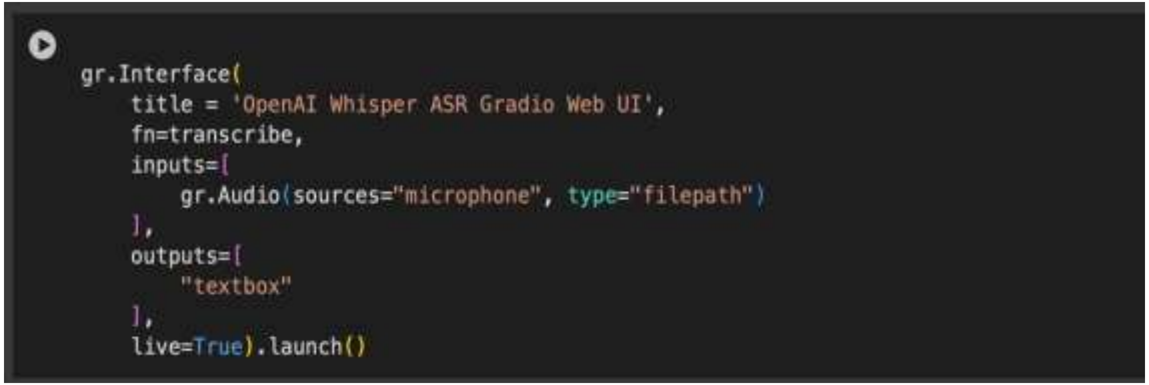
```
[25] import whisper

     model = whisper.load_model("medium")
```

```
model.device
```

```
device(type='cuda', index=0)
```

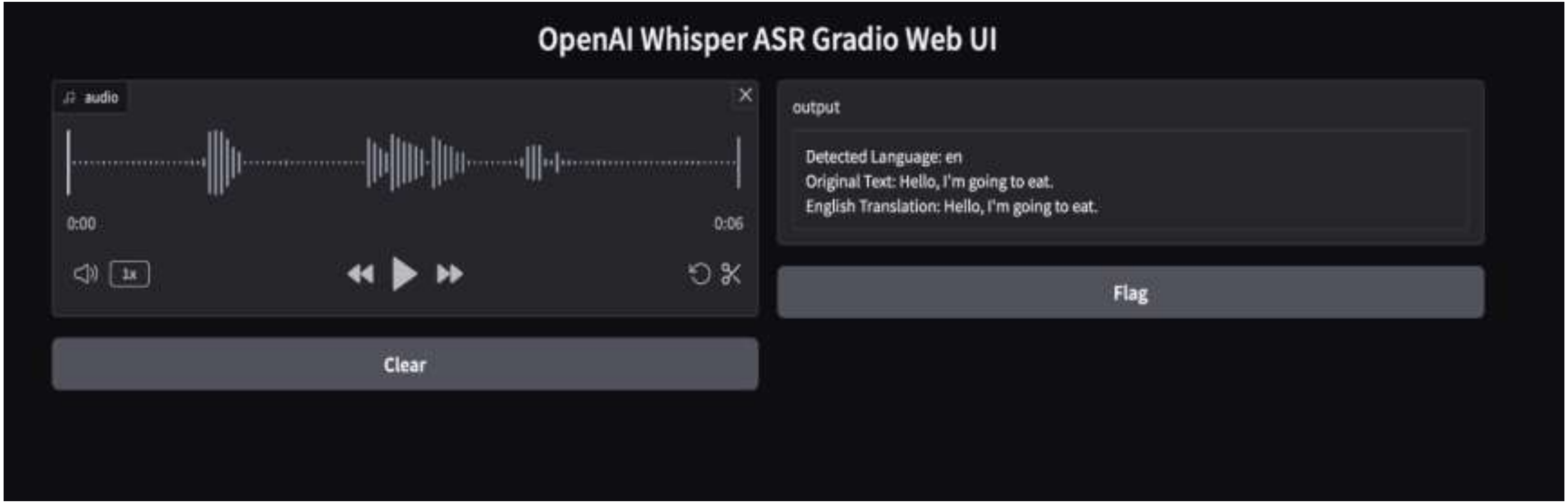


```
# load audio and pad/trim it to fit 30 seconds
audio = whisper.load_audio("/content/santhanam.mp3")
audio = whisper.pad_or_trim(audio)

# make log-Mel spectrogram and move to the same device as the model
mel = whisper.log_mel_spectrogram(audio).to(model.device)

# detect the spoken language
_, probs = model.detect_language(mel)
print(f"Detected language: {max(probs, key=probs.get)}")

# decode the audio
options = whisper.DecodingOptions()
result = whisper.decode(model, mel, options)

# print the recognized text
print(result.text)
```



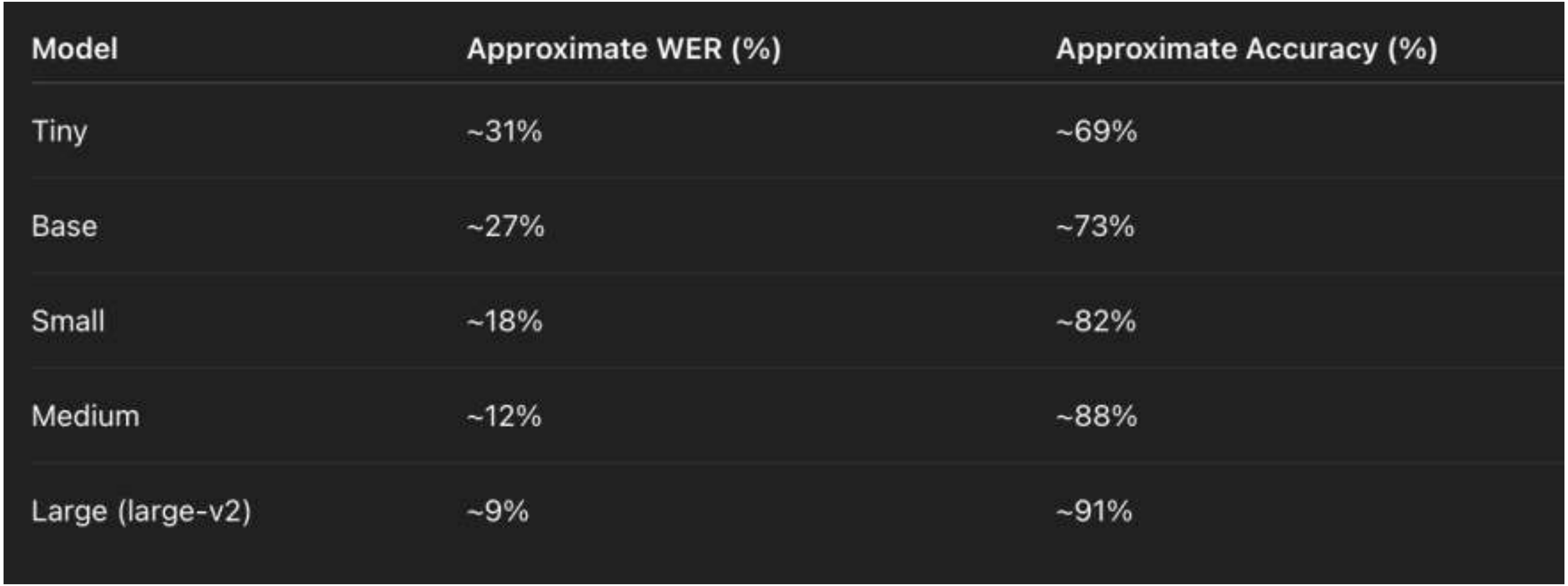
```
gr.Interface(
    title = 'OpenAI Whisper ASR Gradio Web UI',
    fn=transcribe,
    inputs=[
        gr.Audio(sources="microphone", type="filepath")
    ],
    outputs=[
        "textbox"
    ],
    live=True).launch()
```

OUTPUT:



Detected Language: en
Original Text: Hello, I'm going to eat.
English Translation: Hello, I'm going to eat.

ACCURACY:



Model	Approximate WER (%)	Approximate Accuracy (%)
Tiny	~31%	~69%
Base	~27%	~73%
Small	~18%	~82%
Medium	~12%	~88%
Large (large-v2)	~9%	~91%

FUTURE ENHANCEMENT:

Technical Enhancements:

- Fine-tuning for domain-specific vocabulary
- Integration with text
- summarization models
- Custom wake word detectionExtended translation language pairsInterface

Improvements:

- User accounts and saved transcripts
- Batch processing capabilities
- Mobile-optimized interface
- Offline functionality option