

```
import gradio as gr
import whisper
from deep_translator import GoogleTranslator
import os
import uuid
import asyncio
import edge_tts

# Load the Whisper model for speech-to-text
model = whisper.load_model("base")

# Language and voice settings with correct Edge TTS voice
names
language_settings = {
    "English": {"code": "en", "voice": "en-US-GuyNeural"},
    "Tamil": {"code": "ta", "voice": "ta-IN-ValluvarNeural"},
    "Hindi": {"code": "hi", "voice": "hi-IN-MadhurNeural"},
    "Malayalam": {"code": "ml", "voice": "ml-IN-
MidhunNeural"},
    "Kannada": {"code": "kn", "voice": "kn-IN-GaganNeural"},
    "French": {"code": "fr", "voice": "fr-FR-HenriNeural"},
    "Arabic": {"code": "ar", "voice": "ar-SA-HamzaNeural"},
}
```

```

# Asynchronous TTS audio generator using Edge TTS
async def generate_edge_tts(text, voice, output_path):
    communicate = edge_tts.Communicate(text=text,
    voice=voice)

    await communicate.save(output_path)

# Main function to transcribe, translate, and generate speech
def transcribe_translate_speak(audio_path,
target_language_name):
    try:
        lang_code =
language_settings[target_language_name]["code"]
        voice =
language_settings[target_language_name]["voice"]

        # Step 1: Transcribe the audio
        result = model.transcribe(audio_path)
        original_text = result["text"]

        # Step 2: Translate to target language
        translated_text = GoogleTranslator(source='auto',
target=lang_code).translate(original_text)

```

# Step 3: Convert translated text to speech using Edge  
TTS

```
output_audio_path =  
f'translated_{uuid.uuid4().hex}.mp3'  
  
    asyncio.run(generate_edge_tts(translated_text, voice,  
output_audio_path))
```

```
    return f' 🗣️ Original Text:\n{original_text}\n\n 🌐  
Translated to {target_language_name}:\n{translated_text}',  
output_audio_path
```

```
except Exception as e:
```

```
    return f' ❌ Error: {str(e)}", None
```

# Gradio UI Interface

```
interface = gr.Interface(  
    fn=transcribe_translate_speak,  
    inputs=[  
        gr.Audio(type="filepath", label=" 🎤 Upload Your Voice  
Message"),  
        gr.Dropdown(choices=list(language_settings.keys()),  
label=" 🌐 Select Target Language"),  
    ],  
    outputs=[
```

```
gr.Textbox(label="📄 Transcription & Translation"),
gr.Audio(label="🔊 Translated Voice Output (Natural
Male Voice)"),
],
title="🌐 Multilingual Voice Translator",
description="Upload an audio file in any language (Tamil,
Hindi, Malayalam, Kannada, Arabic, French, English) and get
a translated audio response in the selected language with a
natural male voice."
)

# Launch the app
interface.launch()
```