An Industry Oriented Mini Project Report on

REAL TIME MULTIMODAL LANGUAGE TRANSLATOR

Submitted in partial fulfillment of the

Requirements for the award of degree of

Bachelor of Technolog In

Computer Science and Engineering

By

BHUKYA BHARATH

(21EG505807)

DASARI NAMRATHA

(21EG505815)

PANGUNURI SAI AVANIESSH

(21EG505855)

Under the Guidance of

Mrs. J. Himabindu Priyanka

Assistant Professor

Department of CSE

Department of Computer Science and Engineering

ANURAG UNIVERSITY

Venkatapur(v)Ghatkesar(M)Medchal(D)T.S-500088

(2023-2024)

DECLARATION

We hereby declare that the project work entitled "**REAL TIME MULTIMODAL LANGUAGE TRANSLATOR**" submitted to the **Anurag University** in partial fulfillment of the requirements for the award the degree of **Bachelor of Technology (B. Tech)** in Computer Science and Engineering is a record of an original work done by us under the guidance of **Mrs. J.Himabindu Priyanka, Assistant Professor** and this project work has not been submitted to any other university of for the award of any other degree or diploma.

Place:

Date:

Bhukya Bharath
21EG505807

Dasari Namratha
21EG505815

Pangunuri Sai Avaniessh
21EG505855

ANURAG UNIVERSITY | SCHOOL OF ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"REAL TIME MULTIMODAL LANGUAGE TRANSLATOR"** being submitted by **Bhukya Bharath** bearing the Hall Ticket number **21EG505807, Dasari Namratha** bearing the Hall Ticket number **21EG505815** and **Pangunuri Sai Avaniessh** bearing the Hall Ticket number **21EG505855** in partial fulfillment of the requirements for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering** to **Anurag University** is a record of bonafide work carried out by them under my guidance and supervision from June 2023 to November 2023

The results presented in this project have been verified and found to be satisfactory. The results embodied in this project report have not been submitted to any other University for the award of any other degree or diploma.

Signature of Supervisor **Dean,CSE**

Mrs. J.Himabindu Priyanka

Asst. Professor,CSE

External Examiner 1 **External Examiner 2**

ACKNOWLEDGEMENT

We would like to express our sincere thanks and deep sense of gratitude to project supervisor **Mrs. J. Himabindu Priyanka** for her constant encouragement and inspiring guidance without which this project could not have been completed. His critical reviews and constructive comments improved our grasp of the subject and steered to the fruitful completion of the work. His patience, guidance and encouragement made this project possible.

We would like express my special thanks to **Dr. V. Vijaya Kumar**, Dean School of Engineering, Anurag University, for their encouragement and timely support in our B.Tech program.

We would like acknowledge our sincere gratitude for the support extended by **Dr.G.Vishnu Murthy**, Dean, Dept. of CSE, Anurag University. We also express my deep sense of gratitude to **Dr. V V S S S Balaram**, Academic coordinator, **Dr.Pallam Ravi**, Project in-Charge, **Mrs. J. Himabindu Priyanka**, Project Coordinator and Project review committee members, whose research expertise and commitment to the highest standards continuously motivated me during the crucial stage our project work.

Bhukya Bharath

(21EG505807)

Dasari Namratha

(21EG505815)

Pangunuri Sai Avaniessh

(21EG505855)

ABSTRACT

Real-Time Multimodal Language Translator is a powerful software application that offers live translation of spoken language inputs, complemented by voice output in the desired language. This dynamic tool not only transcribes spoken words but also employs a language translation service for real-time interpretation into the user's chosen language. The application actively displays both the transcribed and translated text, ensuring a fluid and context-aware conversation.

Incorporating image text extraction, the application enriches its functionality by being able to process and translate text extracted from images in real time. Each completed phrase is smoothly converted into spoken words using region-specific voices, guaranteeing a seamless and natural dialogue. To maintain clarity and prevent overlapping audio, the application vocalizes only fully processed phrases, typically brief sentences with natural pauses.

At its core, this technology leverages Google's formidable Google Translate API and the Python recognition library, allowing not only text translation but also the storage of translations as MP3 audio files. These audio files can be effortlessly played back using the 'playsound' module. Furthermore, the application seamlessly integrates a range of technologies through REST APIs, encompassing language translation, speech-to-text, and text-to-speech. With the added capability for image text translation, this web-based application becomes an invaluable asset for multicultural, multilingual technology teams. It offers a cost-effective alternative to human translators for automated meeting translation, streamlining communication and promoting collaboration across diverse linguistic backgrounds. In a connected and globalized world, this tool is essential for breaking down language barriers and facilitating effective communication.

CONTENTS

List of Tables

List of Figures

1.INTRODUCTION

Real-time voice and image translation is a transformative technology that has revolutionized the way people communicate across language barriers. It seamlessly combines two essential elements of human interaction: spoken language and visual content, to break down the barriers of linguistic diversity and enhance global communication. This innovation has found applications in various sectors, from international business and travel to healthcare and emergency response.

Voice Translation

Real-time voice translation involves the conversion of spoken language from one language to another instantly. It utilizes advanced speech recognition and natural language processing algorithms to capture and understand the source language, and then generates an accurate translation in the target language. This technology empowers individuals and organizations to have conversations, negotiate deals, or simply connect with people who speak different languages, all without the need for a human interpreter.

Image Translation

In addition to voice translation, real-time image translation complements the process by enabling the translation of visual content. This functionality allows users to point their device's camera at text, signs, menus, or any other written material in a foreign language, and receive instant translations in their preferred language. It's particularly valuable for travelers exploring unfamiliar destinations and businesses seeking to interact with diverse customers or partners.

1.1 Motivation

The motivation for a real-time voice and image translator is rooted in the desire to promote communication, understanding, and inclusivity across linguistic and cultural boundaries. By addressing these motivations, this technology has the potential to bring people closer together, facilitate cooperation, and improve the quality of life for individuals around the world.

1.2 Problem Definition

The problem of real-time multimodal language translation, which focuses on translating spoken language and textual content extracted from images in real time, is a highly intricate challenge. This multifaceted issue necessitates the simultaneous processing of two distinct input sources: spoken language (speech) and visual information derived from images. Meeting the real-time requirement is of utmost importance, ensuring minimal latency for instantaneous translations in live conversations, video content, or any scenario that combines spoken and visual inputs.

Accurate speech recognition is a fundamental component, as it dictates the system's ability to correctly interpret spoken words, even in the face of varied accents, dialects, and background noise. Extracting textual content from images is equally essential, requiring robust optical character recognition (OCR) and efficient image analysis techniques. These processes must be reliable and swift.

Contextual understanding plays a critical role in providing accurate translations. The system must grasp the context of the conversation and the visual scene, accounting for idiomatic expressions, cultural references, and domain-specific terminology. Ambiguity resolution techniques are crucial to select the most suitable translations based on the context.

1.3 Objective of the Project

The objective of a project focused on developing a real-time voice and image translator is to create a comprehensive system that seamlessly translates spoken language and text within images, enhancing communication and breaking down language barriers. The project's objectives encompass various aspects like Speech Recognition, Image Processing, Privacy and Data Security, Accuracy and Naturalness, Real-Time Perform

2.LITERATURE SURVEY

[1]Traditional speech translation systems use a cascade manner that concatenates speech recognition (ASR), machine translation (MT), and text-to-speech (TTS) synthesis to translate speech from one language to another language in a step-by-step manner. Unfortunately, since those components are trained separately, MT often struggles to handle ASR errors, resulting in unnatural translation results. Recently, one work attempted to construct direct speech translation in a single model. The model used a multi-task scheme that learns to predict not only the target speech spectrograms directly but also the source and target phoneme transcription as auxiliary tasks. However, that work was only evaluated Spanish-English language pairs with similar syntax and word order. With syntactically distant language pairs, speech translation requires distant word order, and thus direct speech frame-to-frame alignments become difficult. Another direction was to construct a single deep-learning framework while keeping the step-by-step translation process. However, such studies focused only on speech-to-text translation. Furthermore, all of these works were based on a recurrent neural net-work (RNN) model. In this work, we propose a step-by-step scheme to a complete end-to-end speech-to-speech translation and propose a Transformer-based speech translation using Transcoder. We compare our proposed and multi-task model using syntactically similar and distant language pairs.

[2]We investigate the problem of simultaneous machine translation of long-form speech content. We target a continuous speech-to-text scenario, generating translated captions for a live audio feed, such as a lecture or play-by-play commentary. As this scenario allows for revisions to our incremental translations, we adopt a re-translation approach to simultaneous translation, where the source is repeatedly translated from scratch as it grows. This approach naturally exhibits very low latency and high final quality, but at the cost of incremental instability as the output is continuously refined. We experiment with a pipeline of industry-grade speech recognition and translation tools, augmented with simple inference heuristics to improve stability. We use TED Talks as a source of multilingual test data, developing our techniques on English-to-German spoken language translation. Our minimalist approach to simultaneous

translation allows us to scale our final evaluation to several other target languages, dramatically improving incremental stability for all of them

[3]Nokia offers a speech translation solution for mobile network operators: Operators can provide real-time machine translation of speech, as a useful extension to their currently available voice service. The concept offers a beneficial difference against Over-the-Top (OTT) service providers. The successfully implemented solution presented in this paper works with any device, that is capable of voice calls, even with simple mobile handsets (feature phones). In the service, the Operator's MSS (Mobile Switching Center Server) and/or IMS (IP Multimedia Subsystem) core is involved. The core network is connected to the Microsoft Azure Data Center that provides the speech translation through a Nokia Translation Application Server, running in a Nokia Data Center.

[4]Currently both language and images are essential to the texts we use while language and images can combine for literacy education. In this research work, an application is developed to translate the text integrated to images for visual literacy. Further, several approaches for image-to-text multilingual translator are reviewed in detail. By overcoming the gaps, which are identified by thorough review of the literature, an improved methodology is proposed. As a result, the development of application goes through four major phases including: capturing, extraction, recognition and translation. Moreover, Optical Character Recognition algorithm is particularly used for character extraction and recognition with high accuracy under different environmental circumstances. It translates text just by capturing an image with user smart phone camera and translation instantly appears on user's mobile screen in language selected by the user. The proposed solution may particularly be helpful in literacy education, for learning different languages and possibly can work as visitors' assistant.

Comparsion of Existing Methods

S.no	Authors	Methodology	Advantages	Disadvantage
1	Takatomo Kano, Sakriani Sakti, Satoshi Nakamura	the learning scheme changes from single-task to multi-task learning	Naturalness,End-to-End Processing,Multilingual Support,Fine-Tuning	Data Requirements,Errors, Privacy and Security,Resource-Intensive Decoding
2	Naveen Arivazhagan, Colin Cherry, I Te, Wolfgang Macherey, Pallavi Baljekar, George Foster	NMT heuristics in our English-to-German TED Talk simultaneous spoken language translation scenario	Improved Accuracy,Enhanced Clarity,Confidence Building	Potential for Delays,Potential for Error Propagation,Audience Perception
3	Mate Akos TUNDIK, Attila HILT, Gergo BOTA, Lorand NAGY, Kalle LUUKKANEN	to provide adequate input for the translation with the best possible audio quality.	Accessibility,Real-Time Communication,Quality Improvements,Cross-Platform Compatibility	Privacy Concerns,Legal and Regulatory Challenges
4	Muhammad Ajmal,Farooq Ahmad,Martinez-Enriquez A.M.,Mudasser Naseer,Aslam Muhammad,Mohsin Ashraf	Device camera will start automatically and user can focus and take a snapshot of text easily	Multilingual Accessibility,Customization ,Scalability,Language Preservation	Quality of Conversion,Loss of Visual Information,Linguistic and Cultural Sensitivity,Quality Control

3.ANALYSIS

3.1 Existing System:

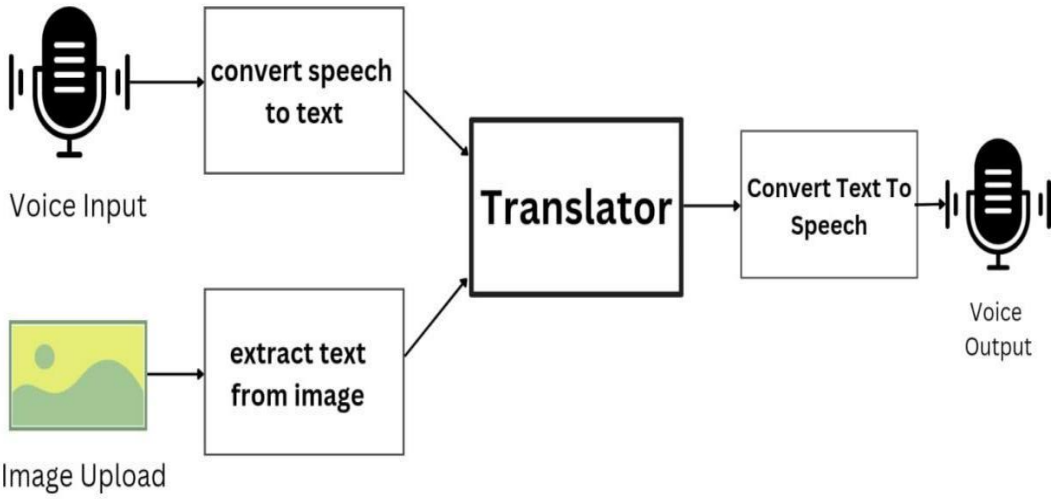
According to understanding the need of speech translation in now day's. we are going to developing speech translation system. The details about system architecture of our system. The input speech is recognized using an automatic speech recognizer and then parsed by a statistical natural language understanding module.

Here are some disadvantages to consider:

- **Accuracy of ASR:** ASR systems may not always provide 100% accurate transcriptions of spoken language. Variations in accents, background noise, and complex linguistic elements can lead to transcription errors, affecting the quality of translations.
- **Challenges in Multiple Languages:** Handling multiple languages and dialects can be complex. ASR and NLU models may perform differently for various languages, requiring extensive training data and resources.
- **Limited Context Understanding:** While NLU modules can parse and understand text, they may have limitations in capturing the full context of a conversation. Contextual understanding can be especially challenging in multilingual and cross-cultural interactions.

3.2 Proposed System

The proposed real time speech-to-speech language translator consists of 4 modules i.e. Login, Input, Translation, Output.



Input Module: This module consists of taking an input in form of voice and sending it into the system.

Translation Module: This module does the hard work of translating the given voice input into the desired output by using the speech-to-speech translation technology by means of Google's Translate API.

Output Module: This module provides the result of work performed by the system in the form of either text or voice.

Advantages of Proposed System:

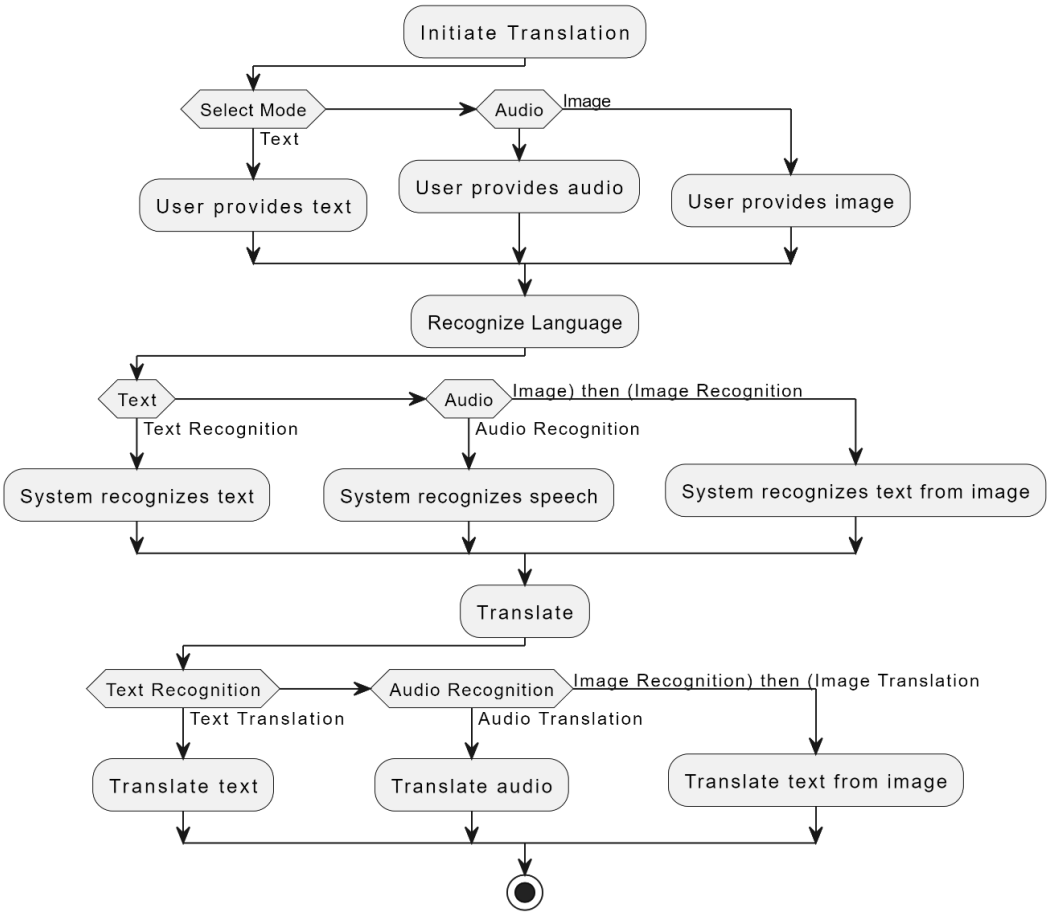
Breaking Language Barriers: The most apparent advantage is the ability to break down language barriers, allowing individuals who speak different languages to communicate effectively. This promotes inclusivity, cross-cultural understanding, and global collaboration.

Cultural Exchange: These systems promote cultural exchange and appreciation by allowing individuals to communicate with people from different cultures, fostering cross-cultural understanding.

Education and Collaboration: Real-time translation supports international collaborations in education, research, and professional settings, enhancing learning and research opportunities.

Emergency Response: In emergency situations, such as natural disasters or medical emergencies, real-time translation can help first responders communicate with affected populations quickly and accurately.

Travel and Tourism: Travelers benefit from real-time translation when navigating foreign destinations, ordering food, or understanding local customs, enhancing the tourism experience.



3.3 Software Requirement Specification

3.3.1 Purpose

The main purpose of an image and voice translator is to facilitate effective communication and break down language barriers between individuals who speak different languages. This technology serves as a bridge that enables people to understand and communicate with each other, even if they do not share a common language

3.3.2 Scope

The scope of image and voice translation technology in real life is extensive and continues to grow as advancements in artificial intelligence and natural language processing continue.

3.3.3 Overall Description

Software Requirements

- **Technology (or) Language**:Python
- **Operating System** :Windows, Linux,MAC
- **IDE**: Visual Studio Code(VS Code) or Python IDLE

Hardware Requirements

- **Processor:**Intel or Ryzen
- **Speed**:1.8Hz
- **Processor Monitor**:14” MONITOR
- **Hard Disk**:32GB or Above
- **Internet:**Proper internet Connection
- **RAM:**2GB or Above

4.IMPLEMENTATION

Real-time multimodal language translator project involves several key steps. Here is an outline of those steps:

TECHNOLOGY:

PYTHON: Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook , Instagram , Dropbox , Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

- □ GUI Applications (like Kivy, Tkinter, PyQt etc.)
- □ Web frameworks like Django (used by YouTube, Instagram, Dropbox)
- □ Image processing (like Opencv, Pillow)
- □ Web scraping (like Scrapy, BeautifulSoup, Selenium)
- □ Test frameworks
- □ Multimedia

GUI Design using Tkinter:Creating a graphical user interface (GUI) using the Tkinter library to provide the user with input fields, buttons, and display areas for text and images.

Language Translation:Using the googletrans library to fetch a list of supported languages.Provide input and output text boxes for the user to enter text and display the translation.

Use the textblob library to perform language translation from the source language to the target language. The selected languages are determined by the user through comboboxes.

Voice Input:Implement voice input functionality, allowing the user to speak in the source language.Utilize the speech_recognition library to capture audio from the user's microphone.Recognize the spoken text using Google's speech recognition service.

Voice Output:Convert the translated text into speech using the gTTS (Google Text-to-Speech) library.Play the generated speech using the playsound library.Remove the temporary audio file after it has been played.

Image Processing:Allow the user to load an image using a file dialog.Utilize the pytesseract library to extract text from the loaded image.Display the extracted text in the original text box.Display the loaded image in a separate image label widget.

Error Handling:Implement error handling to manage exceptions that may occur during voice recognition, text-to-speech conversion, and image processing. Display appropriate error messages to the user.

Clearing Text Fields:Create a "Clear" button to clear the content of both the original and translated text boxes.

User Interaction:Allow the user to select the source and target languages from the comboboxes.Provide buttons for translation, voice input, voice output, and image loading.

Main Application Loop:Create an instance of the Run class to initialize the GUI.Enter the Tkinter event loop (self.root.mainloop()) to handle user interactions and keep the application running.

Configuration and Dependencies:Ensure that the required external tools and libraries are properly configured and available on the system, such as Tesseract for OCR and a working microphone for voice input.

Testing and Debugging:Test the application thoroughly to identify and fix any bugs or issues.Ensure that all functionalities work as expected.

Customization and Enhancements:Customize the GUI, color schemes, and layout to suit your preferences.Add more features or integrate additional translation services if needed.

Deployment:Deploy the application to the target environment, and make it accessible to users.

4.1.List of program files:

main.py:

- This is the main Python script that contains the application's code. It serves as the entry point for the project.
- main.py is responsible for creating the graphical user interface (GUI) using the Tkinter library and defining the functionality of the translator application.
- It includes the implementation of language translation, voice input, voice output, image processing, and other related features.
- The code in main.py initializes the application and enters the Tkinter event loop to handle user interactions.

captured_voice.mp3:

- This is an audio file in the MP3 format that is generated by the project during the "Voice Output" functionality.
- The file contains the translated text converted into speech using the gTTS (Google Text-to-Speech) library.
- It is played to provide the user with the translated text as spoken output.

- The file is generated temporarily for each translation request and is played using the playsound library. After playing, it is removed to avoid cluttering the system.

image.jpg:

- This file is mentioned in the code for the "Load Image" functionality.
- It represents an image that the user can load into the application for text extraction and translation.
- The code uses the PIL (Python Imaging Library) and pytesseract to process the image and extract text from it. The extracted text is then displayed in the original text box.
- Additionally, the image is displayed in a separate image label widget to provide a visual representation of the loaded image.

4.2.List of Libraries

tkinter: This library is used for creating the graphical user interface (GUI) of the application. It provides various widgets and functions for building the application's user interface.

googletrans: This library is used for language translation. It accesses Google Translate to perform translations between different languages.

textblob: TextBlob is used for natural language processing tasks. In this program, it is used for translating text from one language to another.

gtts (gTTS - Google Text-to-Speech): This library is used for converting text to speech. It allows you to generate audio files from text.

playsound: The playsound library is used for playing audio files. In this program, it plays the generated audio file from gTTS.

PIL (Python Imaging Library): The Python Imaging Library is used for working with images, including loading, displaying, and resizing images. In this program, it's used to load images and extract text from them.

pytesseract: This library is used for text recognition in images. It utilizes an OCR (Optical Character Recognition) engine to extract text from images.

speech_recognition (sr): The speech_recognition library is used for capturing voice input from a microphone. It is utilized to recognize spoken words and convert them to text.

os: The os library provides operating system-dependent functionality. In this program, it is used for file handling and directory operations.

4.3.Dataset

It relies on several external libraries and tools for its functionalities, such as:

Google Translate:

The project uses Google Translate, accessed through the googletrans library, for language translation. Google Translate itself uses a vast and constantly updated dataset of multilingual text to perform translations.

Speech Recognition:

The project utilizes the speech_recognition library for voice input. This library connects to Google's speech recognition service, which uses its dataset for speech recognition.

Google Text-to-Speech (gTTS):

The gTTS library is used for text-to-speech conversion. It accesses Google's text-to-speech service to generate speech from text.

Tesseract (OCR):

For image-based text extraction, the project uses the pytesseract library, which is a Python wrapper for the Tesseract OCR engine. Tesseract doesn't use a dataset included with the project but relies on pre-trained models and language data.

5.Experimental Results

5.1.Experiment Setup:

Experiment Setup for the Real-Time Multimodal Language Translator Project

The Real-Time Multimodal Language Translator project using Visual Studio Code (VSCode). This project is designed to translate text, perform voice input and output, and extract text from images.

1. Clone the Project:Begin by cloning or downloading the project's source code from our repository on GitHub.

2. Open the Project in VSCode:Launch Visual Studio Code and use the "Open Folder" option to open the project directory in the VSCode workspace.

3. Create a Virtual Environment (Optional but recommended):Open a terminal within VSCode.Navigate to the project directory.Create a virtual environment for the project to isolate its dependencies.

On Windows:python -m venv venv

Activate the virtual environment:

On Windows:venv\Scripts\activate

4. Install Dependencies:Install the required Python packages by running the following command within the activated virtual environment:

pip install googletrans==4.0.0-rc1

pip install textblob

pip install gTTS

pip install playsound==1.2.2

pip install pillow

pip install pytesseract

pip install SpeechRecognition

5. Install Tesseract (OCR Engine):If we haven't already, we need to install the Tesseract OCR engine. Download and install Tesseract from the official repository:https://github.com/tesseract-ocr/tesseract/releases.Ensure that to add Tesseract to your system's PATH during the installation

6. Run the Application:In VSCode, open the main.py file.

Ensure that the system is correctly set up for any external tools or libraries mentioned in the code (e.g., Tesseract).

Start the application by running the main.py script within VSCode.

7. Interact with the Application:The application's GUI should appear, allowing us to interact with its features.

Use the provided functionalities like language translation, voice input, voice output, and image processing.

5.2.Parameters With Formulas

In Real-Time Multimodal Language Translator there are various parameters and formulas related to its functionality.

Voice Input Recognition Rate (VIR):This parameter measures the accuracy of the system in recognizing spoken language inputs.

Formula: VIR = (Number of correctly recognized voice inputs / Total number of voice inputs) * 100%

Translation Accuracy (TA):Translation accuracy represents the correctness of language translation from the source language to the target language.

Formula: TA = (Number of correctly translated phrases / Total number of translated phrases) * 100%

Text Extraction Success Rate (TESR):This parameter measures the system's ability to accurately extract text from images.

Formula: TESR = (Number of successfully extracted text from images / Total number of image-based text extractions) * 100%

Response Time (RT):Response time evaluates how quickly the system can process and translate voice inputs or image-based text.

Formula: RT = (Time taken to process and translate a voice input or image text) / (Number of voice inputs or image texts processed)

Memory Usage (MU):Memory usage indicates the amount of RAM or memory required by the application to function smoothly.

Formula: MU = (Total memory consumption of the application)

User Satisfaction (US):User satisfaction is a qualitative parameter representing how satisfied users are with the application's performance.

Formula: US = (Number of positive user feedback / Total user feedback) * 100%

Cost of Operation (CO):Cost of operation quantifies the expenses associated with utilizing external services and maintaining the application.

Formula: CO = (Total cost of using external services/APIs) + (Development and maintenance costs)

Error Rate (ER):Error rate measures the frequency of errors or issues encountered during the application's operation.

Formula: ER = (Number of errors encountered during voice recognition, translation, or image text extraction) / (Total number of voice inputs and image text extractions)

Throughput (TP):Throughput represents the system's processing capacity in terms of voice inputs and image text extractions per unit of time.

Formula: TP = (Total number of voice inputs and image text extractions processed) / (Total time of operation)

5.3 Sample Code

```
""" Real Time Multimodal LanguageTranslator."""
import tkinter as tk
from tkinter import filedialog
from tkinter import ttk, messagebox
import os
import googletrans
import textblob
from gtts import gTTS
from playsound import playsound
from PIL import Image, ImageTk
import pytesseract
import speech_recognition as sr
import easyocr

class Run:
    """This class represents the Language Translator application."""
    def __init__(self):
        # GUI Screen
        self.root = tk.Tk()
        self.root.title("Language - Translator")
        self.root.geometry("1040x500")  # Increased the height to accommodate image
        self.root.resizable(False, False)
        self.root.config(bg='#b2c2cf')
        self.top_frame = tk.Frame(self.root, bg="white", width=1020, height=15)
        self.top_frame.place(x=0, y=0)
        # Calling Widgets Function
        self.placing_widgets()
        self.root.mainloop()
```

```
    self.reco = None
    self.text = None


def placing_widgets(self):
    """GUI"""
    # Language List
    self.languages = googletrans.LANGUAGES
    self.language_list = list(self.languages.values())
    # Text Boxes
    self.original_text = tk.Text(
        self.root, height=10, width=40, bg="#ffffff", font=("Times New Roman", 15))
    self.original_text.grid(row=1, column=1, pady=20, padx=10, rowspan=3)
    self.translated_text = tk.Text(
        self.root, height=10, width=40, bg="#ffffff", font=("Times New Roman", 15))
    self.translated_text.grid(row=1, column=3, pady=20, padx=10, rowspan=3)
    # Buttons
    self.translated_button = tk.Button(self.root, text="Translate!", font=(
        "Helvetica", 20), command=self.translate_it, bg="#4db4d6", fg="black")
    self.translated_button.grid(row=2, column=2, padx=15)
    self.input_button = tk.Button(self.root, text="Voice_input", font=(
        "Helvetica", 20), command=self.input, bg='#4db4d6')
    self.input_button.grid(row=12, column=1, padx=10)
    self.translated_button = tk.Button(self.root, text="Voice_output", font=(
        "Helvetica", 20), command=self.speak, bg='#4db4d6')
    self.translated_button.grid(row=12, column=3, padx=10)
    self.image_button = tk.Button(self.root, text="Load Image", font=(
        "Helvetica", 20), command=self.load_image, bg='#4db4d6')
    self.image_button.grid(row=12, column=2, padx=10)
    # combo Boxes
```

```
    self.original_combo = ttk.Combobox(
        self.root, width=50, value=self.language_list)
    self.original_combo.current(21)
    self.original_combo.grid(row=6, column=1)

    self.translated_combo = ttk.Combobox(
        self.root, width=50, value=self.language_list)
    self.translated_combo.current(38)
    self.translated_combo.grid(row=6, column=3)
    # clear Button
    self.clear_btn = tk.Button(
        self.root, text="Clear", command=self.clear, bg='#4db4d6', width=6, height=2)
    self.clear_btn.grid(row=3, column=2)
    # Image display
    self.image_label = tk.Label(self.root, bg='white')
    self.image_label.grid(row=1, column=4, rowspan=3)
# Translation
def translate_it(self):
    """Translate text from the original language to the selected language."""
    self.translated_text.delete(1.0, tk.END)
    try:
        for key, value in self.languages.items():
            if value == self.original_combo.get():
                from_language_key = key
        for key, value in self.languages.items():
            if value == self.translated_combo.get():
                to_language_key = key
        words = textblob.TextBlob(self.original_text.get(1.0, tk.END))
        words = words.translate(
```

```
                from_lang=from_language_key, to=to_language_key)
            self.translated_text.insert(1.0, words)
        except FileNotFoundError as exception:
            messagebox.showerror("Translator", exception)
    # Taking Input from Mic
    def input(self):
        """Capture voice input and display it in the original_text box."""
        for key, value in self.languages.items():
            if value == self.original_combo.get():
                from_language_key = key
        #import speech_recognition as sr
        self.reco = sr.Recognizer()
        with sr.Microphone() as source:
            self.reco.adjust_for_ambient_noise(source, duration=0.3)
            print("Speak now")
            audio = self.reco.listen(source)
            try:
                self.text = self.reco.recognize_google(
                    audio, language=from_language_key)
                self.original_text.insert(1.0, self.text)
            except sr.UnknownValueError:
                messagebox.showerror('Not Recognized', 'No speech detected. Please try
again.')
            except sr.RequestError as e:
                messagebox.showerror('Request Error', f'Error making a request to the
recognition service: {e}')
    # Output Through Speaker
    def speak(self):
        """Convert translated text to speech and play it."""
        for key, value in self.languages.items():
```

```
        if value == self.translated_combo.get():
            to_language_key = key
    words = self.translated_text.get(1.0, tk.END)
    # Converting text into a mp3 File
    obj = gTTS(text=words, slow=False, lang=to_language_key)
    obj.save('captured_voice.mp3')

    # Playing the converted mp3 File
    playsound("E:\\projects\\captured_voice.mp3")
    # Removing the mp3 File After Playing
    os.remove("E:\\projects\\captured_voice.mp3")
# Load Image and extract text
def load_image(self):
    """Load an image, extract text, and display it in the original_text box."""

    pytesseract.pytesseract.tesseract_cmd=r'C:\Users\LENOVO\AppData\Local\Prog
    rams\Tesseract-OCR\tesseract.exe'
    file_path = filedialog.askopenfilename()
    if file_path:
        try:
            image = Image.open(file_path)
            text = pytesseract.image_to_string(image)
            self.original_text.delete(1.0, tk.END)
            self.original_text.insert(1.0, text)
            # Display the loaded image
            image = Image.open(file_path)
            image = image.resize((200, 200), Image.LANCZOS)
            image = ImageTk.PhotoImage(image)
            self.image_label.config(image=image)
            self.image_label.image = image
```

```
            except FileNotFoundError as fnfe:
                messagebox.showerror("Error", f"File not found: {fnfe}")
            except PermissionError as pe:
                messagebox.showerror("Error", f"Permission denied: {pe}")
            #except Exception as exception:
                #messagebox.showerror("Error", f"Error loading image: {exception}")
    # Clearing All The Inputed And Outputed Text
    def clear(self):
        """Clear the content of both text boxes."""
        self.original_text.delete(1.0, tk.END)
        self.translated_text.delete(1.0, tk.END)
# Creating Object
my_run_instance = Run()
```

6.Test cases

Test case Id	INPUT	Expected output	Actual output	Rate
1	Hii	హలో	హలో	success
2	నమస్కతం	hello	hello	success
3	आप कैसे है	మీరు ఎలా ఉన్నారు	మీరు ఎలా ఉన్నారు	success
4	வணக்கம்	hello	hello	success
5	(Image)In order to succeed we must first believe that we can	सफल होने के लिए हमें पहले विश्वास करना चाहिए कि हम कर सकते हैं	सफल होने के लिए हमें पहले विश्वास करना चाहिए कि हम कर सकते हैं	success
6	జీవితం అనేది ఒక ప్రశ్న లాంటిది ఎవరూ సమాధానం చెప్పలేరు! చావు అనేది సమాధానం లాంటిది కానీ ప్రశ్నించే ధైర్యం ఎవరికీ ఉండదు!	Life is like a question No one can answer! Death is like an answer But no one has the courage to question!	Life is like a question No one can answer! Death is like an answer But no one has the courage to question!	success

7.Screenshots

7.1 Voice Translation

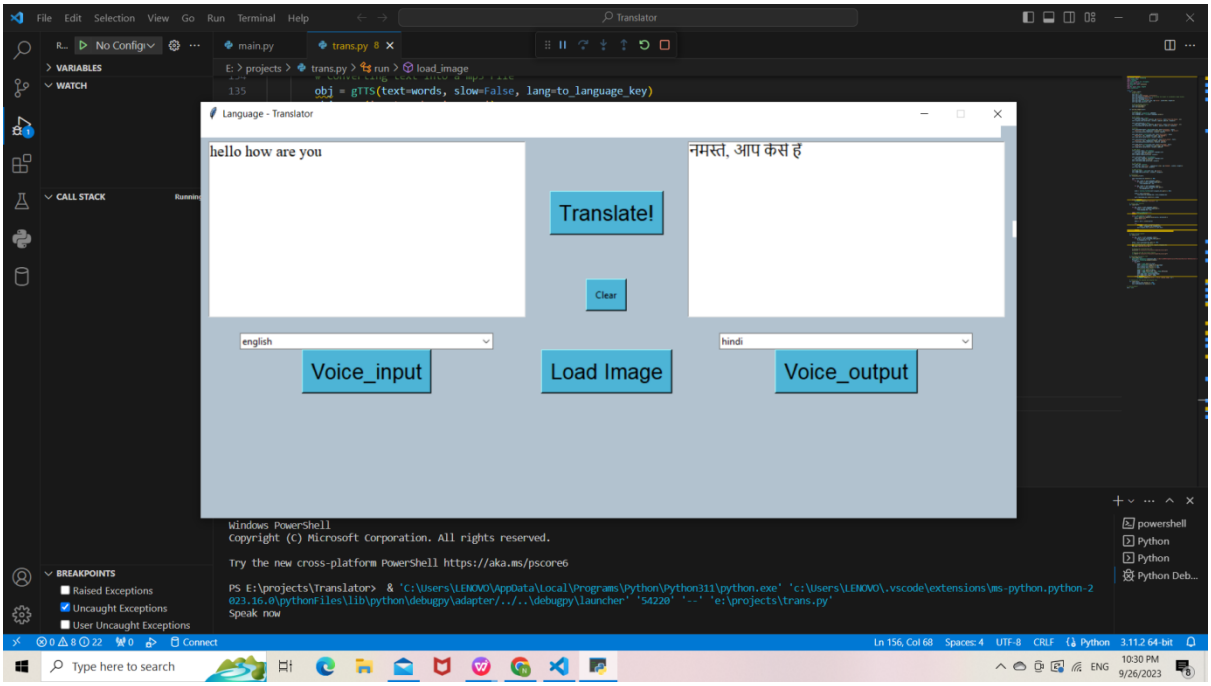


Fig 7.1

In Figure 7.1, we see a voice translation system that takes voice input and outputs translated voice in the user's desired language.

7.2 Image text extract translation

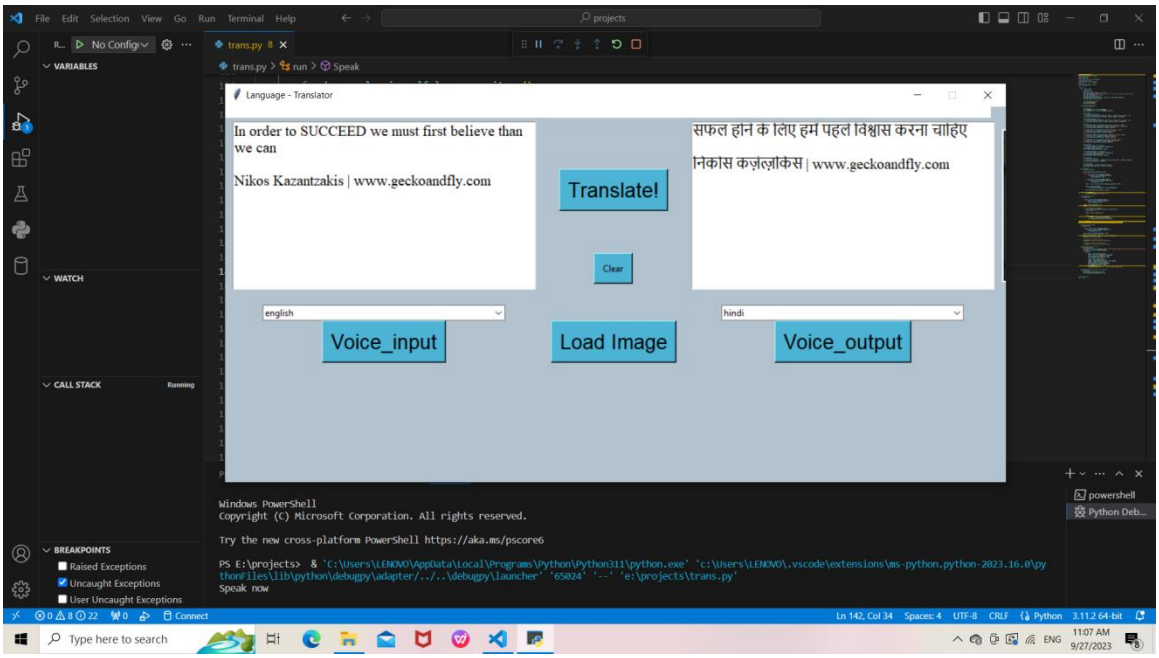


Fig 7.2

In Figure 7.2, we see a system that extracts text from an image and outputs translated voice in the user's desired language.

8.CONCLUSION:

In conclusion, real-time voice and language translation technology represents a transformative breakthrough in bridging linguistic divides and fostering global communication. Its integration of cutting-edge machine learning and natural language processing has revolutionized the way individuals and businesses communicate across languages. This technology has empowered international travel, global commerce, and cross-cultural connections, making the world more accessible and interconnected. While challenges like context preservation and low-resource language support persist, ongoing advancements promise increasingly accurate and context-aware translations. As real-time translation technology continues to evolve, it holds the potential to break down language barriers further, fostering a more inclusive and interconnected world where effective communication knows no linguistic bounds.

9.Future Enhancement

The future of real-time voice and language translation technology is poised to be even more transformative and seamless. Anticipated advancements in this field include enhanced contextual understanding, enabling translation systems to capture and interpret the subtleties of tone, intent, and cultural nuances in conversations. Further improvements in low-resource language support will make the technology accessible to a wider range of communities and languages. Integration with emerging technologies like augmented reality (AR) and virtual reality (VR) holds the promise of immersive language experiences, where users can interact in their native language while seeing translations in real time. Additionally, real-time translation technology may become more personalized, adapting to individual user preferences and speech patterns. As research and development continue to push the boundaries of AI and machine learning, we can look forward to a future where language is no longer a barrier, fostering global communication, understanding, and cooperation on an unprecedented scale.

Bibliography

1. Takatomo Kano, Sakriani Sakti, Satoshi Nakamura”**Transformer-Based Direct Speech-To-Speech Translation with Transcoder**”

2. Naveen Arivazhagan, Colin Cherry, I Te, Wolfgang Macherey, Pallavi Baljekar, George Foster **“Re-Translation Strategies for Long Form, Simultaneous, Spoken Language Translation**”

3. Mate Akos TUNDIK, Attila HILT, Gergo BOTA, Lorand NAGY, Kalle LUUKKANEN **“Access-independent Cloud-based Real-Time Translation Service for Voice Calls in Mobile Networks**”

4. Muhammad Ajmal,Farooq Ahmad,Martinez-Enriquez A.M.,Mudasser Naseer,Aslam Muhammad,Mohsin Ashraf**” Image to Multilingual Text Conversion for Literacy Education”**

5. L.R. Bahl,R. Bakis,J. Bellegarda,P.F. Brown,D. Burshtein,S.K. Das,P.V. de Souza,P.S. Gopalakrishnan,F. Jelinek,D. Kanevsky,R.L. Mercer,A.J. Nadas,D. Nahamoo,M.A. Picheny **"Large vocabulary natural language continuous speech recognition"**

6.M. Ingleby,Danhui Li**"Intermediate level speech processing as an aim of low level signal analysis"**

7. Zhihua Lu,Philipp Heidenreich,Abdelhak M. Zoubir**"Objective quality assessment of speech enhancement algorithms using bootstrap-based multiple hypotheses tests"**

8. Yukai Ju; Wei Rao; Xiaopeng Yan; Yihui Fu; Shubo Lv; Luyao Cheng; Yannan Wang; Lei Xie; Shidong Shang**"TEA-PSE: Tencent-Ethereal-Audio-Lab Personalized Speech Enhancement System for ICASSP 2022 DNS Challenge"**