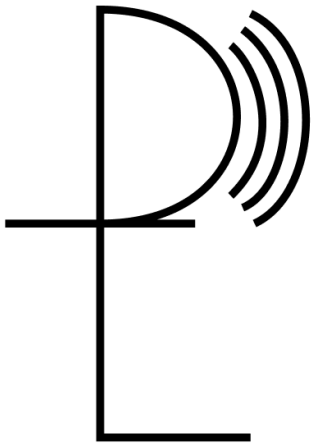
Boston University
Electrical & Computer Engineering

EC 463 Senior Design Project

Second Prototype Testing Plan

Portable Language Translator



By

Team 23

Portable Language Translator

Team Members

Yohan Kim yohank@bu.edu
Ryan Liao ryanliao@bu.edu
Cristian Palencia cris0211@bu.edu
Andrew Nguyen aynguyen@bu.edu

Required Materials:

Software

- Language Translation
    - Google Cloud Platform project with text-to-speech, speech-to-text, translate, and storage API's
    - Python3 Packages:
        - Numpy
        - Sounddevice
        - Google-cloud-speech
        - Google-cloud-texttospeech
        - Google-cloud-translate
        - Webrtcvad
        - Pydub
        - simpleaudio
        - Pynput
        - Flask
        - Flask-cors
- ASL Detection Model
    - Python3 Packages
        - OpenCV
        - Tensorflow
        - Mediapipe
        - Numpy
- User Interface
    - Python3 Packages
        - PyQt5
- Firmware
    - Python3 Packages
        - gpiozero

Hardware

- Raspberry Pi 5
- USB Devices
    - USB Camera
    - USB Speaker
    - USB Microphone
- Portable Power Bank (5000 mAh)

Set Up:

For the hardware setup we have a Raspberry Pi 5 that interfaces with all the relevant components. Specifically, it interfaces with three USB devices: camera, speaker, and microphone.

For the language translation, a react web app is used to control the base language and voice settings of the program, which it sends using a flask api. One device is used for the conversation. The users must wait for the translation and playback to fully finish before beginning the next phrase. The program captures audio from the USB microphone, splits it into chunks and uses voice activity detection software to determine if someone is speaking. Once it detects no speech, it sends the audio file to the google cloud for the transcription, translation, and finally text-to-speech. The audio file is sent back to the program which it plays from the system's speaker and to display the translated text on the UI.

For the ASL gesture detection and prediction, the external camera (USB 2.0 Camera) will record a live video stream and the script will initialize this video feed with OpenCV. First processing camera frames is done with Mediapipe's hand landmarks model to extract hand landmarks and then grouping these landmarks from the last 30 frames into a sequence. This sequence is fed into our LSTM (Long Short Term Memory) model that outputs the probability for each gesture. When a specific gesture's confidence is greater than the threshold value, that gesture will be then chosen as the detected gesture. An asynchronous thread handles this inference process, ensuring that consistent and confident predictions are added to a running sentence, which is eventually converted into speech for translation. Once the sentence is done playing, the script will then listen for the audio and save the transcription in a .txt file where it will be displayed in the UI. Then the script will go back to detecting ASL gestures and the cycle will restart.

Pre-Testing Setup Procedure:

Preliminary Hardware Setup:

1. Connect power bank to Raspberry Pi 5 using a USB-A to USB-C cord
2. Connect relevant USB devices including microphone, speaker, and camera
    a. To properly setup the microphone and speaker control on the Pi we need to run the following: sudo apt install alsa-utils
3. Connect three buttons to GPIO pins 4, 17, and 27. Additionally, connect the other terminals to ground.

Initial Setup:

1. Install dependencies using: “pip install -r requirements.txt”
2. Run the script “python3 everything.py”

Testing Procedure:

1. Open web application at portlangtrans.netlify.app and send base language and voice settings to the Raspberry Pi
2. Run python3 everything.py
3. Start in Language Translation mode. Have a conversation switching between english, spanish, and/or korean. Also, we can press the volume control buttons to modify volume out.
    a. English - Spanish script: (male)
        i. ¿Hola, como estas? (Hi, how are you?)
        ii. I am very good thank you! How was your day?
        iii. ¡Mi dia a sido bueno! ¿Que comiste hoy? (My day has been good! What did you eat today?)
        iv. I ate a hamburger and drank water
    b. English - Korean script: (male)
        i. 병원 어디 있어? (Where is the hospital?)
        ii. It is just down the street
        iii. Do you need help getting there?
        iv. 병원 얼마나 멀어요? (How far is your hospital?)
        v. It should take you 15 minutes to walk there
    c. Spanish - Korean script: (female)
        i. ¿Hola, como estas? (Hi, how are you?)
        ii. 나는 오늘 좋았어 고마워 너는 어땠어 (I am good, thank you! How was your day?)
        iii. ¡Mi dia a sido bueno! (My day has been good!)
        iv. ¿Que comiste hoy? (What did you eat today?)
        v. 나는 햄버거를 먹고 물을 마셨어 (I ate a hamburger and drank water)

4. Press the mode toggle button to switch to ASL mode.
5. Go through and gesture “hello”, “thank you”, and “I love you” to see ASL translation. After gesturing, speak into the microphone to test the ASL to speech conversation.

Measurable Criteria:

Language:

I. Settings are able to be sent from the deployed web app from a mobile device.
II. Device is able to translate between English, Spanish, and Korean smoothly and without manual input.

ASL:

I. Camera is able to detect ASL hand gestures, form a sentence, and translate them to words
II. Words translated from ASL with be spoken aloud from the speaker.
III. The program will automatically switch from using the camera to detect hand gestures to listening for speech.
IV. Spoken words will be transcribed and displayed on the on-screen UI as well as a live view of the camera feed.
V. The program will go back to detecting ASL hand gestures automatically after spoken words are displayed on UI.

User Interface:

I. Displays relevant text translation to the user
II. Demonstrates camera live feed such that the user can position themselves properly in the view of the camera.

Hardware / Firmware:

I. Three buttons are fully functional. One button is in charge of toggling between ASL and Language Translation modes. The other two are in charge of adjusting the volume of the speaker.
II. USB microphone properly picks up audio such that the Language Translation is able to perform without issue.
III. USB speaker outputs audio when necessary and can clearly to be heard by users.

Score Sheet:

Test	Correct?
English - Spanish conversation	
Transitions to next conversation w/o manual input	
English - Korean conversation	
Able to translate same language in succession	
Spanish - Korean conversation	
ASL - Individual gesture	
ASL - Form a sentence from gesture	
ASL - audio transcribes onto UI	
Button - Mode Toggling	
Buttons - Volume Control	