Boston University
Electrical & Computer Engineering

EC 464 Senior Design Project

Final 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
- 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)
- Touch Display
- 3 Buttons (Audio control and Mode toggling)
- Encasing (3D print)

Set Up:

For the hardware setup we have a Raspberry Pi 5 that interfaces with all the relevant components. Specifically, it interfaces with the following: USB camera, USB speaker USB microphone and a touch display all of which are mounted throughout the 3D printed encasing. The Raspberry Pi will be connected to the user's phone hotspot using 1 of 3 UI tabs where a user can select their internet and insert their password through a virtual keyboard displayed on the touch display.

For the language translation, a dedicated UI tab allows users to configure settings for the language translation, such as, base language and voice gender before the translation starts. 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 translation UI tab.

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 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
2. Connect touch screen display via DSI.
3. Connect three buttons to GPIO pins 4, 17, and 27. Additionally, connect the other terminals to ground.
4. Mount the display / Raspberry Pi to the encasing using screws.
5. Glue down the speaker devices onto the appropriate location on the encasing.
6. Fasten the camera in the provided slot at the top of the encasing.
7. Connect the Raspberry Pi to the power bank using a usb cable.
8. Place the back plate onto the device and screw into place.
9. Turn on the power bank using the opening on the encasing.

Initial Setup:

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

Testing Procedure:

1. Connect to mobile hotspot.
2. Start in Language Translation mode. Have a conversation switching between english, spanish, and/or korean (change using settings tab). Also, we can press the volume control buttons to modify volume out.
    a. English - Spanish script: (male)
        i. ¿Hola, como estas? (Hello, 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. 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)
3. Press the mode toggle button to switch to ASL mode.
4. Go through ASL script

    a. ASL: hello help bathroom
    b. Speech: Yes the bathroom is down the hall, will you get there ok?
    c. ASL: Yes, thanks
    d. Speech: Have a nice day
5. Press the mode toggle button to switch back to Language Translation mode
6. Press power button to turn off device

Measurable Criteria:

Language:

I. Device is able to translate between the set base language of English, Spanish or Korean and the other detected language.

ASL:

I. Camera is able to detect ASL hand gestures, form a sentence, and translate them to words
II. Words translated from ASL will 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.
III. Settings are able to be sent from the dedicated UI tab on the display.
IV. Users can use the Wifi tab to connect the device to their phone hotspot.

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
III. USB speaker outputs audio when necessary and can clearly be heard by users.

Score Sheet:

Test	Correct?
English - Spanish conversation	
Able to translate same language in succession	
Spanish - Korean conversation	
ASL - Form a sentence from gesture	
ASL - Audio transcribes onto UI	
Button - Mode Toggling	
Buttons - Volume Control	
Device Wearability	
Independent of Wall Brick	