Project 3: Sign Language Recognition

Index: **Page No.**

Acknowledgement

It gives me immense pleasure to present the report of the 3rd Project of my internship at Invenio Business Solutions, conducted from May 20, 2025 to July 18,2025. I would like to express my gratitude to my reporting manager, Mrs. Saritha Koroth, and my mentors, Mr. Harsh Singh and Ms. Malika Kaur for their constant guidance. It was a privilege to intern under the mentorship of Mr. Harsh Singh for two enriching months.

Introduction

Sign-language recognition (SLR) systems have long been proposed as a bridge between Deaf and hearing communities, but research momentum has only recently accelerated thanks to advances in computer vision, lightweight deep-learning frameworks and commodity GPUs.

MediaPipe's real-time hand-tracking models, for instance, resolve 21 three-dimensional landmarks per hand at mobile-phone frame rates.

Simultaneously, open libraries such as `SpeechRecognition` and `pyttsx3` offer offline speech-to-text (STT) and TTS capabilities, making multimodal translation pipelines feasible on consumer hardware.

Objectives:

1. **Real-Time Recognition:** Achieve ≥24 fps landmark estimation and ≥10 fps classification on a standard laptop webcam feed.
2. **Bidirectional Communication:**
    - Convert recognised sign letters → synthesised speech through TTS.
    - Accept spoken English → display textual transcript → render sequential ASL alphabet images.
3. **User-Friendly Interface:** Provide clickable buttons for common actions (word commit, sentence commit, delete, clear, speak)
4. **Modular Codebase:** Abstract detection, feature extraction, classification, audio and UI layers for ease of substitution or upgrade.

Requirements:

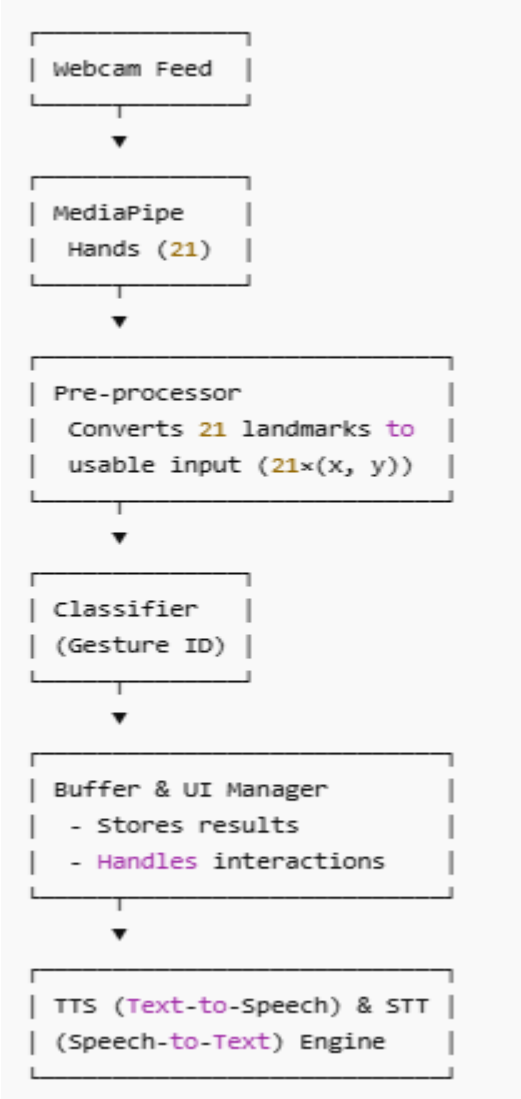
- matplotlib==3.7.1
- mediapipe==0.10.13
- numpy==1.24.3
- opencv-python==4.6.0.66
- pandas==2.0.1
- Pillow==9.5.0
- python-dotenv==1.0.0
- scikit-learn==1.2.2
- seaborn==0.12.2
- tensorflow==2.12.0

Use virtual environment for requirements.txt
Step 1: pip install virtualenv
Step 2: python -m venv venv
Step 3: source venv/Scripts/activate

To install the necessary requirements:
pip install -r requirements.txt

Methodology:

1. System-level Architecture



The pipeline separates perception (CV) from dialogue (NLP/audio) allowing parallel execution threads.

2. Data Acquisition & Pre-processing

- Frame Capture: `cv.VideoCapture` polls the default camera at 30 fps, scaled to 400 × 300 pixels to balance detail and throughput.
- Frame Mirroring: Horizontally flipped images match the user's mirror expectation and stabilise handedness features.
- Color Space Conversion: BGR → RGB because MediaPipe expects RGB.

3. Hand Landmark Extraction

MediaPipe returns 21 landmark triples `(x, y, z)` mapping to anatomical joints.

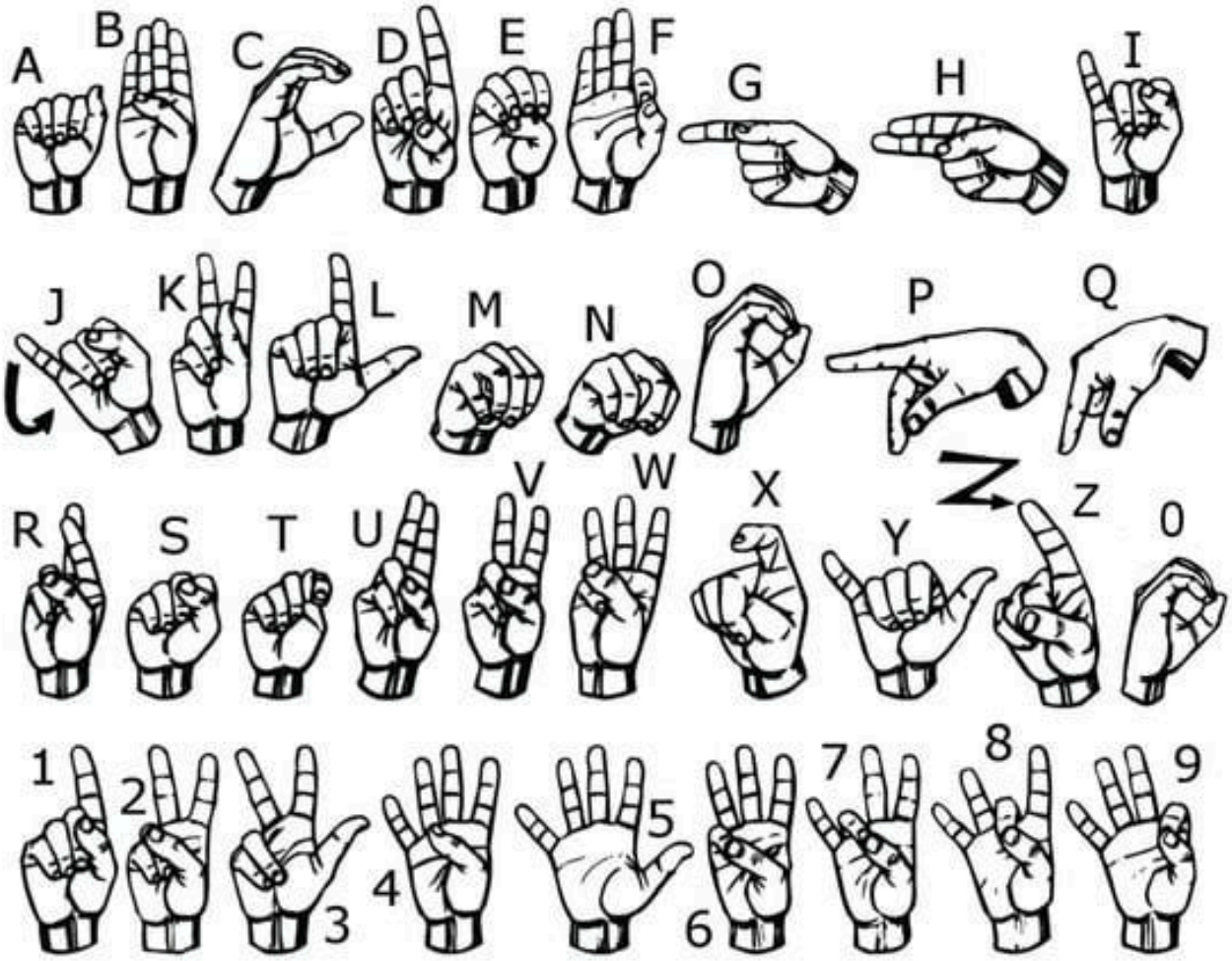
4. Feature Engineering

1. Bounding-box derivation using min/max of landmarks → ROI for optional cropping.
2. Relative co-ordinates normalisation: Translate landmarks so the wrist joint sits at origin; scale by maximum inter-landmark distance → viewpoint/scale invariance.
3. Flatten vector → 42-dimensional feature (21 points × 2) for classifier input.

5. Speech-to-Text Sub-system

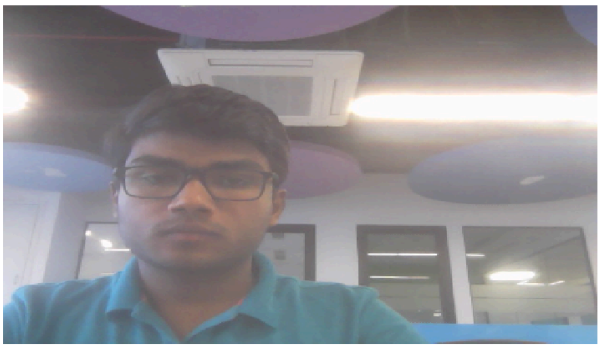
- Triggered by Speech button; launches a non-blocking thread calling `recognize_speech`.
- Default recogniser: Google Web API through `speech_recognition.Recognizer.recognize_google`.
- On completion, recognised text:
    1. Displays beside the Speech button.
    2. Generates an image queue (`prepare_speech_images`) mapping each alphabetic character to a stored PNG in `alphabet_images/`.
    3. Streams images at 1 letter/s with 2 s inter-word pause—mirroring how Deaf viewers might fingerspell the spoken utterance.

American Sign Language Dataset:



Project Results:

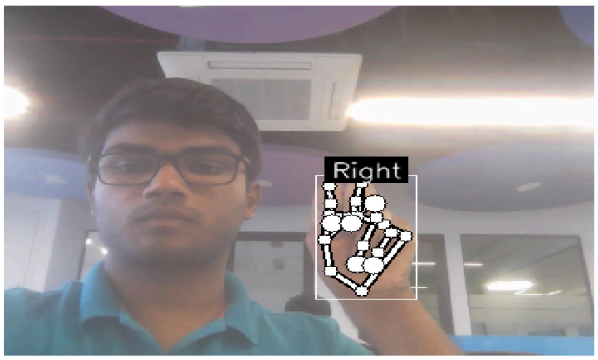
1. User Interface



Functionalities:

- Speak current word
- Add a word to a sentence
- Allows the user to delete the last character of a word if it was recognized incorrectly.
- Allows the user to delete the last word of a sentence
- Text to speech functionality: speak current word,
  Speak complete sentence
- Clear complete sentence
- Speech to text input: the user can give speech as input and the model will convert it into text and signs.

2. Give hand gesture as input



Current: MAN

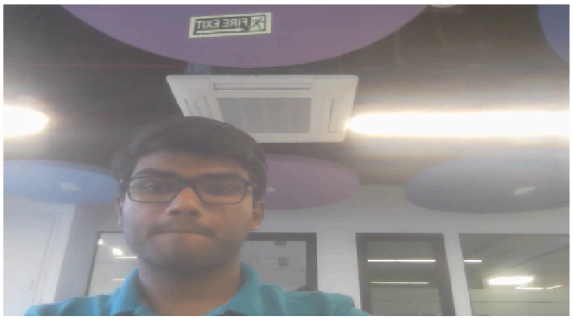
Last Word:

Sentence:

Speak Word | Add to Sentence | Delete Character | Delete Word | Speak Sentence | Clear

Speech

3. Add words in a sentence



Current:

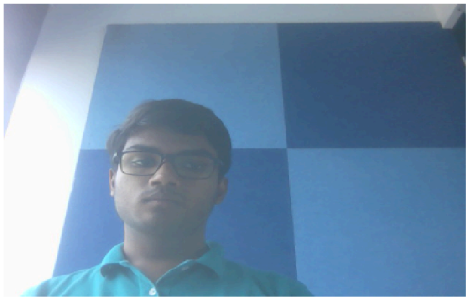
Last Word: HARSH

Sentence: I AM HARSH

Speak Word | Add to Sentence | Delete Character | Delete Word | Speak Sentence | Clear

Speech

4. Speech as input



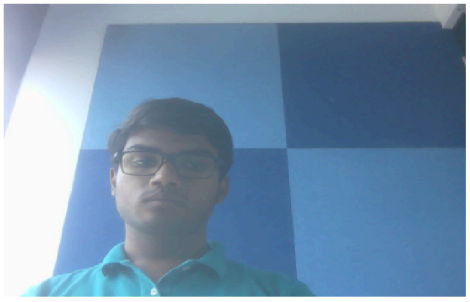
Current:

Last Word:

Sentence:

Speak Word | Add to Sentence | Delete Character | Delete Word | Speak Sentence | Clear

Speech Text: Horsh



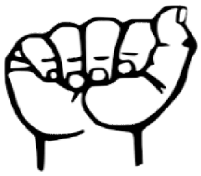
Current:

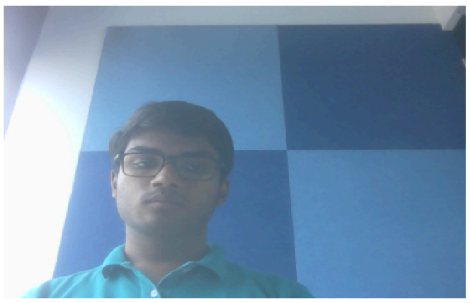
Last Word:

Sentence:

Speak Word | Add to Sentence | Delete Character | Delete Word | Speak Sentence | Clear

Speech Text: Horsh





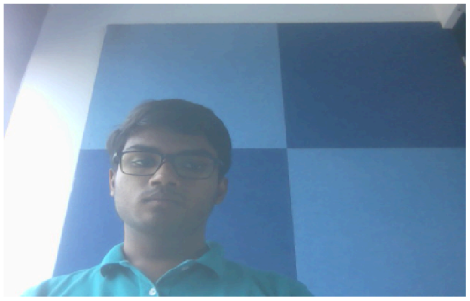
Current:

Last Word:

Sentence:

Speak Word | Add to Sentence | Delete Character | Delete Word | Speak Sentence | **Clear**

Speech **Text: Harsh**



Current:

Last Word:

Sentence:

Speak Word | Add to Sentence | Delete Character | Delete Word | Speak Sentence | **Clear**

Speech **Text: Harsh**