Pose-Based Word Presence Prediction:
A Framework for Indian Sign Language Processing

Rudra Sinha
Department of Computer Science and Engineering

Contents

1 Introduction **2**

2 Dataset Preparation **2**
2.1 Source 2
2.2 Extraction Pipeline 2
2.3 Handling New Data 3

3 Word Presence Detection Task **3**
3.1 Definition 3
3.2 Pose Extraction 3
3.3 Pose Data Format 3

4 Similarity Approaches **4**
4.1 Cosine Similarity and DTW 4
4.2 Sliding Window and Trimming 4

5 Experimental Setup **4**

6 Results **5**
6.1 Cosine Similarity Results 5
6.2 DTW Results 5

7 Discussion **6**

8 Future Work **6**

1 Introduction

Indian Sign Language (ISL) is a primary mode of communication for millions of Deaf and Hard of Hearing (DHH) individuals in India. As per WHO estimates (2016), approximately 63 million Indians are DHH. Despite its significance, Indian Sign Language (ISL) lacks adequate computational resources, datasets, and standardized tools for Natural Language Processing (NLP) and machine learning applications.

The Indian Sign Language Research and Training Center (ISLRTC) reports only around 300 certified ISL interpreters, highlighting the pressing need for automated systems to support ISL understanding and translation. With the advent of computer vision and pose estimation techniques, the opportunity to create computational models that understand sign language using only skeletal data (pose landmarks) has become feasible.

Objective: This project focuses on creating a benchmark task called **Word Presence Detection**, using pose-based data derived from ISL videos to determine whether a given sign (word) is present in a sentence video.

2 Dataset Preparation

2.1 Source

The primary data source is the ISH News YouTube channel, which provides news broadcasts in Indian Sign Language. These videos typically feature a single signer presenting sentences clearly and consistently.

- **Total videos processed:** ˜220,000
- **Average sentence length:** 10–20 words
- **Average duration per clip:** 7–8 seconds
- **Frame rate:** 25 frames per second

2.2 Extraction Pipeline

The dataset construction was automated using a comprehensive Python script. This script enables batch downloading of YouTube videos via `yt-dlp`, extracts transcript segments using `YoutubeTranscriptApi`, and creates clean, timestamp-aligned clips with corresponding text and audio metadata.

The pipeline includes:

- Middle frame-based manual ROI selection with fallback defaults
- Auto-retry for private or failed downloads
- Per-clip cropping, muting, and subtitle extraction
- Confidence-based filtering and CSV mapping of clips to transcripts

Each downloaded video is processed into smaller clips based on subtitle timestamps. The ROI is used to crop each segment around the signer, and the muted clip is paired with its transcript and optional audio.

Figure 1: A Video Frame from the extracted dataset

2.3 Handling New Data

To scale the dataset over time, another script `NewerVideos.py` was implemented. It enables scraping of recent videos not previously included, allowing users to specify a custom start date for monitoring updates. The script automatically checks for duplicate entries and prompts for manual ROI selection if the visual template changes.

3 Word Presence Detection Task

3.1 Definition

Given a query **word video** and a **sentence video**, determine whether the signer used the sign corresponding to the word in the sentence video.

Use case: Acts as a benchmark to assess the consistency and representation power of ISL datasets and models.

3.2 Pose Extraction

MediaPipe Holistic was used to extract full-body pose landmarks from videos.

- Each frame yields 33 pose landmarks, 21 hand landmarks per hand, and 468 facial landmarks.
- Only **arm, shoulder, and wrist** landmarks were used to reduce noise.
- Landmarks were normalized using the mean hip coordinates to reduce signer positional variance.

3.3 Pose Data Format

- `Pose.data = [frame, person, landmark, (x, y, z)]`
- `Pose.confidence = [frame, person, landmark, confidence]`
- Each frame: 576 landmarks
- Stored in binary `.pose` files

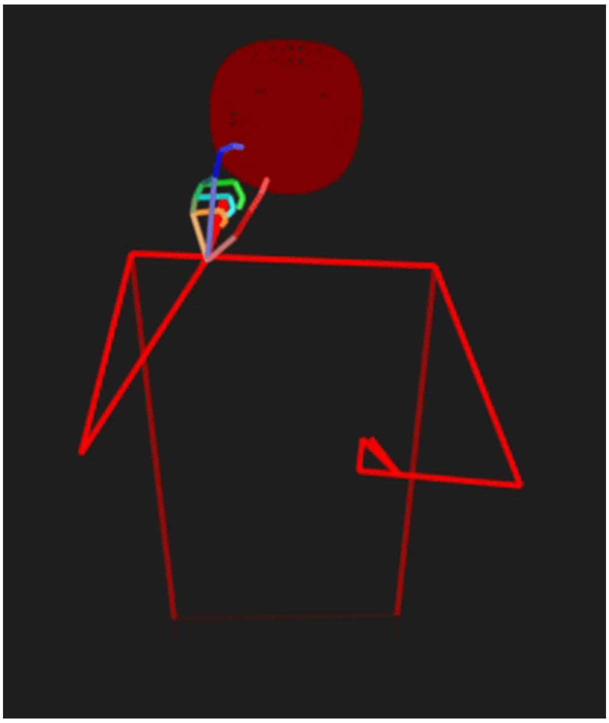


Figure 2: Pose View of the extracted dataset

4 Similarity Approaches

4.1 Cosine Similarity and DTW

The word presence detection system is powered by a python script, which uses multiple similarity metrics including Cosine Similarity and Dynamic Time Warping (DTW).

Cosine Similarity: Compares pose vectors frame-by-frame. Frame similarity is weighted by average landmark confidence, and an empirically chosen threshold of 0.41 is used to decide presence.

Dynamic Time Warping (DTW): Used to align sequences of different lengths. DTW scores are computed for windowed segments of sentence poses, and scores below 0.07 indicate presence. Confidence-aware DTW is implemented for robust comparisons.

4.2 Sliding Window and Trimming

A sliding window technique is used to search over the sentence sequence, while the query video is dynamically trimmed to remove inactive frames. Both cosine and DTW metrics are applied across hand and arm landmarks, with adjustable weightings and thresholds.

5 Experimental Setup

- **Query words:** 100 (sampled from CISLR dataset)
- **Candidate sentences:** 100
- **Evaluated pairs:** 10,000 word-sentence pairs

- **Libraries used:** SciPy, NumPy, matplotlib, dtw-python

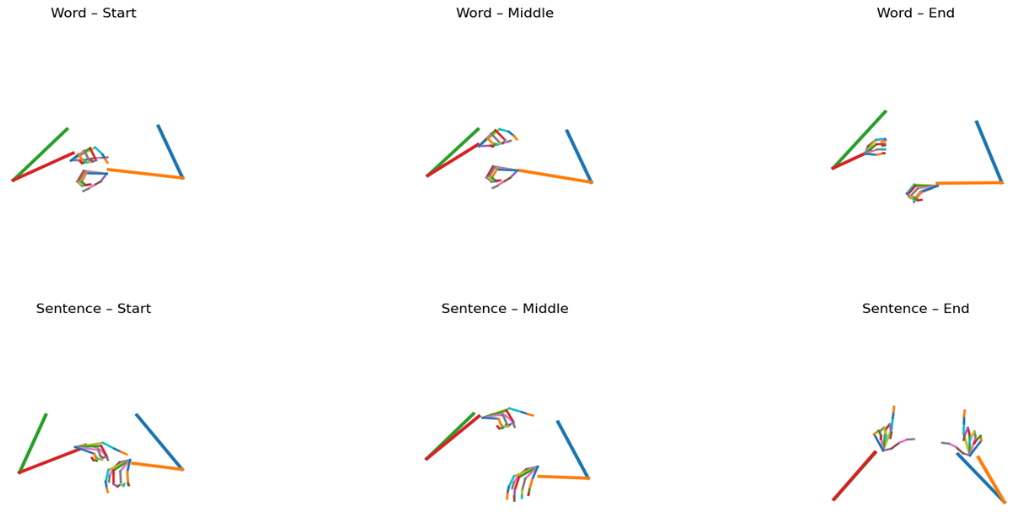
6 Results

6.1 Cosine Similarity Results

- Top-5 Accuracy: 25%
- Top-10 Accuracy: 36%
- Top-15 Accuracy: 39%
- Top-20 Accuracy: 46%

6.2 DTW Results

- Top-5 Accuracy: 29%
- Top-10 Accuracy: 34%
- Top-15 Accuracy: 44%
- Top-20 Accuracy: 52%



Visualization of Correct Match using Matplotlib
Word = Time
Sentence = Haryana native Neeraj Chopra participated in the Javelin throw in the 2020 Tokyo Olympics for the very first time

7 Discussion

- DTW outperforms cosine similarity due to its robustness to temporal misalignments.
- Performance indicates moderate success in matching word signs in longer videos.
- The primary challenges include signer variability, subtle sign differences, and noise in pose estimation.

8 Future Work

The `NewerVideos.py` script lays the foundation for long-term scalability of the dataset. It enables the automatic collection of videos uploaded after a specified date and provides the user with ROI flexibility if a new visual format is detected in the playlist.

Future directions also include:

- Incorporate facial expressions and non-manual markers
- Normalize pose sequences further to address body proportion variations
- Fine-tune thresholds and add classifier on top of similarity metrics
- Expand dataset to thousands of sentence-word pairs

Acknowledgements

I thank Sanjeet for his invaluable mentorship throughout this undergraduate project. I also thank Prof. Ashutosh Modi for the opportunity to work on a problem so crucial to accessibility and inclusivity in AI. Gratitude to ISH News and the iSign team for making data and benchmarks available.

References

[1] Indian Sign Language Research and Training Center. `https://islrtc.nic.in/`

[2] iSign Benchmark: `https://exploration-lab.github.io/iSign/`

[3] Google MediaPipe: `https://github.com/google/mediapipe`

[4] Joshi et al., 2022. CISLR: Corpus for Indian Sign Language Recognition.