ISL WORD RECOGNITION VIA I3D & POSE FUSION

UGP Report

R.Charan | 230819

Department of Computer Science and Engineering
IIT Kanpur

April 2025

Contents

1 Introduction

1.1 Problem Statement

Given a video of a signer performing gestures and actions, the task is to predict the corresponding gloss label(word). As the CISLR dataset[1] that we use contains a low number of average videos per word(1.5 videos per word), we formulate the gloss recognition task as a one-shot learning task. A single video is provided for all unique labels in the corpus as prototypes. The rest are used as test data for evaluating the model.

1.2 Challenges

One-shot recognition of sign-language words is challenging due to limited per-gloss video samples, especially in low-resource languages like Indian Sign Language (ISL). Unlike spoken language, where audio signals typically follow a consistent phonetic structure, sign language recognition must model complex spatio-temporal patterns that vary greatly across different signers. Variability arises not only in signing styles and speeds but also in the segmentation of gestures, with some signers repeating or reordering gestures even for the same word.

Moreover, while spoken languages benefit from abundant, well-annotated datasets, sign language resources are often sparse and fragmented, exacerbating the difficulty of developing robust recognition systems. For instance, CISLR (2022)[2] highlights that videos for the same gloss may involve different arm movements, hand shapes, and even gesture sequences depending on the signer. This demands strong feature aggregation methods capable of capturing both spatial articulation (hand shapes, facial expressions, body posture) and temporal dynamics (motion flow, rhythm, transitions) in a signer-agnostic manner.

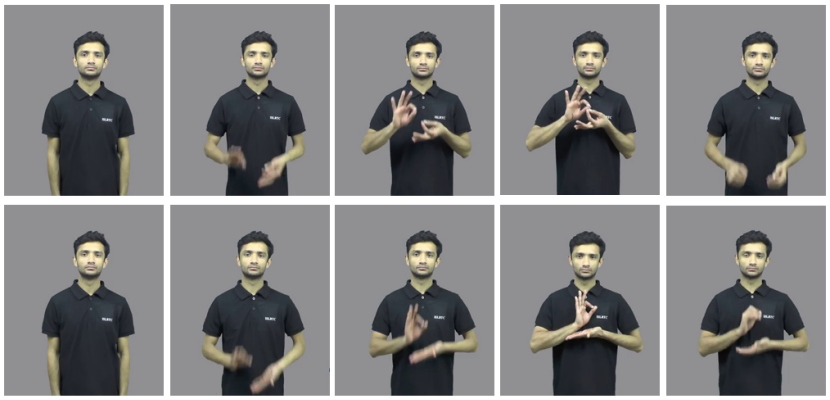


Figure 1: An example of the same signer showing different signs for the same word "Buddhist." Though the movement of the arm is similar, the hand gestures differ. *(CISLR)*[2]

1.3 Methodology

The I3D features[4] that we use are from the publicly available CISLR Dataset[1]. It uses the following methodology to extract these features,

- Given the limited availability of annotated data for Indian Sign Language (ISL), we employ a cross-lingual knowledge transfer approach to enhance recognition performance.
- Specifically, we leverage the Word-Level American Sign Language (WLASL) dataset[5], which comprises over 2,000 commonly used ASL signs, to pre-train an Inflated 3D ConvNet (I3D) model[4].
- This model is adept at capturing spatio-temporal features from video sequences, making it suitable for sign language recognition tasks.

After training the I3D model[4] on the WLASL dataset[5], we utilize the learned weights to extract rich spatio-temporal features from our ISL video samples. This transfer learning strategy enables us to harness the extensive knowledge embedded in the WLASL-trained model, thereby compensating for the scarcity of ISL-specific data.

In addition to the I3D-based features[4], we extract pose information from the same ISL videos using the MediaPipe Holistic model[6]. This model provides 543 landmarks per frame, encompassing 33 pose landmarks, 468 face

landmarks, and 21 hand landmarks for each hand. Each landmark is represented by (x, y, z) coordinates, capturing comprehensive spatial information pertinent to sign language gestures.

To effectively utilize these heterogeneous feature sets, we process the I3D[4] and pose features[3] through separate pipelines tailored to their respective modalities. The I3D features[4], capturing dynamic motion patterns, and the pose features[3], encapsulating static spatial configurations, are both crucial for accurate sign recognition.

2 Dataset and Features

2.1 CISLR Dataset[1]

The publicly available CISLR dataset[1] contains the following which are of great use for us

- I3D_features.pkl, which contains the I3D features[4] extracted for all the CISLR videos.
- dataset.csv, which contains the information about 7050 videos from the dataset
- test.csv, which contains the information about the 2285 test samples
- prototype.csv, which contains the information about the 4765 train samples

Each of the csv files have the following information- uid, gloss, duration, category. Each of the I3D features[4] are of the following dimensions, (1, 1024, S, 1, 1), where S is the number of temporal segments.

```
<class 'pandas.core.frame.DataFrame'>
           id                                        I3D_features
0    ZcVzjZeVwj0  [[[[[0.03112409]], [[0.00099271]], [[0.000154]...
1    ZIKWlQUff9c  [[[[[0.22031154]], [[0.22767238]], [[0.1475680...
2  FnTehW6ik-Y_2  [[[[[0.20237736]], [[0.38866925]], [[0.4987630...
3    0lrX7s_3ScY  [[[[[0.04070692]], [[0.00172393]], [[0.0040934...
4    bgwSjzhyPs8  [[[[[0.]], [[0.]], [[0.]], [[0.]], [[0.0005370...
Shape of first few I3D feature: (1, 1024, 11, 1, 1) (1, 1024, 8, 1, 1) (1, 1024, 9, 1, 1) (1, 1024, 9, 1, 1)
```

Figure 2: I3D_features.pkl from *cislr* Dataset[1]

2.2 Pose Data[3]

We utilize MediaPipe[6], a real-time framework for human pose estimation, to extract 2D coordinates of key landmarks such as hands, face, and upper body joints from each video frame. This results in a sequence of pose keypoints that represent the spatial configuration of the signer over time. For each frame in a particular video, we have 543 points split across the whole person as face, left-hand, right-hand and body. Each point has 3 co-ordinates,

- (x, y): Normalized co-ordinate location of that point in the frame
- z: Distance of the point from the camera, with respect to the middle of the hip as reference.

3 Methods

3.1 Baseline Models

We evaluate the baseline models of just I3D[4] and Pose[3] to understand how efficient the streams are for our task of word/gloss prediction.

3.1.1 I3D Only[4]

Aggregate raw I3D segments by simple mean → signed-sqrt → L2 normalization. This achieved 17.02% Top-1.

3.1.2 Pose Only[3]

Cosine matching on pose-velocity features: Only 0.31% Top-1—pose alone is too sparse and averaging over time might not carry any relevant information.

Method	Top-1	Top-5	Top-10
I3D Only	17.02	20.88	22.93
Pose Only	0.31	0.39	0.83

Table 1: Baseline Methods Accuracy

3.2 Motion Guided Pooling

3.2.1 Velocity-Weighted I3D

Compute per-segment weights from mean landmark speed. Weight I3D slices accordingly before SSR (mean, max, std) pooling → 15.45% Top-1.

3.2.2 Pose-Guided Segment Weighting

Same as above but SSR pooling first before weighting, 16.54% Top-1.

Method	Top-1	Top-5	Top-10
Velocity-Weighted I3D	15.45	19.12	21.27
Pose-Guided I3D	16.54	20.53	22.23

Table 2: Motion Guided Pooling

3.3 Learned Pooling Variants

3.3.1 Generalized-Mean (GeM) Pooling

GeM (power-mean) pooling over I3D segments for $p \in 1, 2, 3, 4$ (p=3 best): 19.04% Top-1.

3.3.2 Attention-Pooling for Velocity

Frame-wise speeds attendively pooled into 510-d velocity: 18.86% Top-1.

Method	Top-1	Top-5	Top-10
GeM-I3D (p=3)	19.04	24.33	27.44
Attention-Velocity	18.86	24.29	27.22

Table 3: Learned Pooling Variants

3.4 Normalization Techniques

3.4.1 PCA-Whitening

Projects features to d dimensions then whitens ($\Sigma \rightarrow$I) via PCA. The best dimension was d=768 (18.99% Top-1)

3.4.2 Mahalanobis Scaling

Scale each feature by $1/\sqrt{variance} \rightarrow$ per-feature decorrelation, 17.37% Top-1

3.4.3 ZCA Whitening

Whitening without axis swapping based on ZCA: 19.47% Top-1

Method	Top-1	Top-5	Top-10
PCA-Whitening (768d)	18.99	24.64	27.66
Mahalanobis Scaling	17.37	21.53	23.37
ZCA Whitening	19.47	24.55	27.40

Table 4: Normalization Techniques

3.5 Fusion Strategies

3.5.1 Score-Level Fusion

Weighted sum of similarity scores (α GeM, β PCA, γ Mahalanobis) with grid search. Best at (α,β,γ)=(0.6, 0.4, 0.0): 19.21% Top-1, 25.16% Top-5, 28.18% Top-10.

3.5.2 Feature-Level Fusion

Concatenate or weighted-sum normalized embeddings before matching: PCA-whitened GeM (1024-d) + Mahalanobis also explored.

Method	Top-1	Top-5	Top-10
Fused Score (α=0.6,β=0.4,γ=0.0)	19.21	25.16	28.18
PCA-Whitened GeM-I3D (1024d)	19.69	24.64	27.22

Table 5: Fusion Strategies

3.6 Ensemble Techniques

To improve robustness and leverage complementary information across multiple I3D feature[4] representations, we explored ensemble strategies combining SSR and GeM pooled features using both PCA whitening and ZCA whitening. These ensembles are intended to improve recognition performance by merging the strengths of different pooling methods and normalization strategies.

- **PCA-Whitened Ensemble:** In the PCA-based ensemble, both SSR and GeM pooled I3D features[4] were whitened using Principal Component Analysis (PCA) with whitening and dimensionality reduction (best at 1024 dimensions). After whitening, we evaluated,
  - **Max Fusion:** Element-wise max between cosine similarity scores of SSR and GeM.
  - **Weighted Fusion:** A combination of scores using weight α, selected through grid search.

Method	Top-1	Top-5	Top-10
Max-Ensemble	20.09	24.95	27.48
Whitened-Ensemble (α=0.4)	19.69	24.64	27.22

Table 6: PCA-Whitened Ensemble

- **ZCA-Whitened Ensemble:** To test whether spatial alignment and structure preservation would help further, we replaced PCA with ZCA whitening. ZCA not only decorrelates features and scales variances like PCA but also preserves the original feature directions, making it more interpretable and potentially more compatible with cosine similarity. We repeated the same pooling and fusion strategies:

- SSR and GeM pooled features
- ZCA whitening applied
- Cosine similarity-based max and weighted fusion (with best α=0.50)

Method	Top-1	Top-5	Top-10
Max-Ensemble	19.52	24.77	27.70
Whitened-Ensemble (α=0.5)	20.13	25.38	28.58

Table 7: ZCA-Whitened Ensemble

3.7 Results

We have boosted accuracy for the same dataset used in CISLR_v1-a[1] by just applying data augmentation techniques on the extracted I3D feature set[4].

Dataset	**# Test Samples**	**Top-1**	**Top-5**	**Top-10**
CISLR v1.0-a	2285	16.81	20.04	22.58

Figure 3: Original Accuracy

Method	Top-1	Top-5	Top-10
ZCA Whitened-Ensemble (α=0.5)	20.13	25.38	28.58

Table 8: Improved Accuracy

3.8 Directions for the Future

More data would be the way forward. Scraping and organizing more data would help us train bigger models and to fine-tune the I3D model[4] to the Indian context to assist better information capture.

4 Data Acquisition Automation

To complement the existing I3D[4] and pose-based[3] recognition pipeline, we developed a custom Python script (DEF-Scraping_Script.py) to automate video download, frame extraction, OCR-based segmentation, and region-of-interest cropping from YouTube playlists. This section describes its design, dependencies, and workflow.

4.1 Dependencies

- **yt-dlp**: Download videos and extract playlist metadata.
- **moviepy & ffmpeg**: Clip videos, extract frames at specified frame rates.
- **OpenCV (cv2)**: Interactive ROI selection and image processing.
- **Tesseract OCR (pytesseract)**: Extract text from frames for segmentation.
- **numpy, shutil, concurrent.futures**: Array handling, file management, parallel processing.

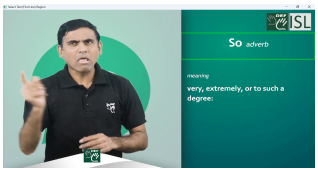
4.2 Workflow

1. Playlist Collection: The user is prompted to give all the playlists for scraping data. The DEF YouTube channel has Word-Of-The-Day organised for every month of the year. Any list of these that are needed can be entered into the program in seperate lines.

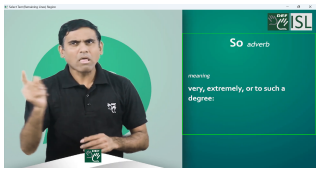
```
PS C:\Users\rchar\Desktop\New folder> python DEF-Scraping_Script.py
Enter YouTube playlist URLs (enter an empty line to finish):
https://www.youtube.com/watch?v=pSBVKX3tRtM&list=PL-oV5cJPCqmiB0LSp9Bgsl8HFel7buoDp
https://www.youtube.com/watch?v=H46VzWmACHs&list=PL-oV5cJPCqmhCYhqrPIcMA2FGUt-H9_fG
https://www.youtube.com/watch?v=L6dm6zJyCRE&list=PL-oV5cJPCqmjIqw3R-lRjk6UAJyfpOJ-Q
https://www.youtube.com/watch?v=xmNnflibm34&list=PL-oV5cJPCqmgDuovWNlAToMxUqOAFyPtB

Fetching playlist information from https://www.youtube.com/watch?v=pSBVKX3tRtM&list=PL-oV5cJPCqmiB0LSp9Bgsl8HFel7buoDp ...
Found 31 videos in playlist: January 2023 PV AND WOTD
1. Make out - 2 (Phrasal Verb) January 1st
2. Embittered (Adjective) Word of the Day for January 2nd
3. Cross something out (Phrasal Verb) January 3rd
```

Figure 4: Playlist collection

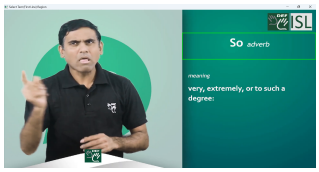
2. Playlist Metadata Extraction:
    - Parse the URL(s) of the YouTube playlist provided using ytdlp in extractflat mode.
    - Retrieve the title of the playlist and the list of video IDs for processing.
3. Initial ROI Selection (First Pass)
    - Download the first video to a temporary directory.
    - Capture a midpoint frame and prompt the user to draw three bounding boxes:
        - Complete text region except any DEF logo that might interfere.
        - Complete Text region.
        - Person region for final video cropping.



Selecting text region 1



Selecting text region 2



Selecting person region

4. Automated Video Processing: For each video in the playlist (parallelized via ProcessPoolExecutor):
    - Download the full video (.mp4).
    - Extract frames at 1 FPS into a temp folder.
    - For each expected description line from YouTube metadata:
        - Apply OCR within the appropriate text ROI.
        - Use a stability threshold (0.5) to mark segment start/end times.
        - Crop and save each detected segment as a separate clipped video using ffmpeg filters. Rewind frames by two seconds at segment boundaries for accuracy.

  - Merge duplicate segments by matching description texts and concatenating clips.

```
Detecting text intervals...
Started segment for 'So (adverb)' at 8s
Ended segment for 'So (adverb)' at 17s
Started segment for 'Meaning - very, extremely, or to such a degree:' at 19s
Ended segment for 'Meaning - very, extremely, or to such a degree:' at 49s
Started segment for 'Example - The house is so beautiful.' at 49s
Ended segment for 'Example - The house is so beautiful.' at 53s
```

Figure 5: Detection of text in the selected region

5. Second Pass (Retry for Failures)
   - Identify videos with no detected text intervals.
   - Prompt the user for a second set of ROIs on a failure sample.
   - Reprocess failed videos concurrently using the new ROI pair.

4.3 Output

The script creates a directory called "New" in the given base directory, set to "D:\WOTD" by default. The "D:\WOTD\New" directory contains the following saved by the script:

- Directory: Meaning with,
  - All the videos of signers performing the sign language for words, saved with a unique-ID.
  - segments.csv, a csv file containing the details about all the videos in the directory.
- Directory: Sentence with,
  - All the videos of signers performing the sign language for sentences, saved with a unique-ID.
  - segments.csv, a csv file containing the details about all videos in the directory.

	A	B	C
1	Unique_ID	Video_Title	Description
2	9ba8be23-0ab1-415c-a0bd-393200ff3cdb	So adverb - Word of the Day	So (adverb)
3	9ba8be23-0ab1-415c-a0bd-393200ff3cdb	So adverb - Word of the Day	Meaning - very, extremely, or to such a degree:
4	9ba8be23-0ab1-415c-a0bd-393200ff3cdb	So adverb - Word of the Day	Example - The house is so beautiful.

Figure 6: Format of the .csv files

- video_segments_common.csv, containing all the information about the complete set (meanings + sentences) of videos that were saved.
- Directory: temp, which contains the information of videos that failed the script.

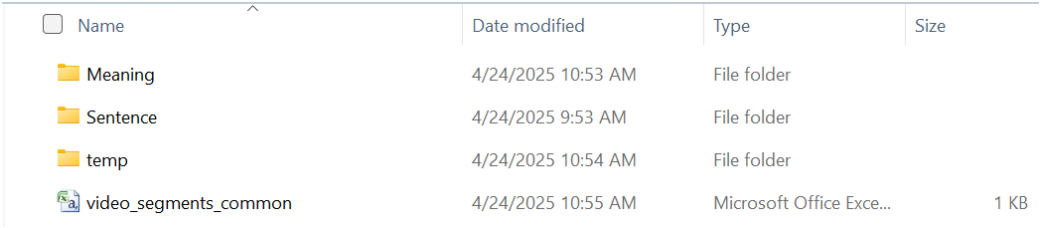


Figure 7: Directory contents

4.4 Rationale

The DEF videos have a fixed ROI containing the person and a fixed ROI containing the text. There are multiple such formats of fixed person ROI and text ROI used accross different playlists of DEF. The playlists for each year use the same format of videos, and in very few cases use 2 different formats. This is the rationale behind using 2 passes in the script.

The videos also contain the content of the video - word + sentences (meaning and example), in the description. So we check for the text in the description to occur on the selected text ROI and clip the corresponding person ROI for each line of the description.

The selection for the text and person ROI can be repeated multiple time by pressing the character 'r' before confirming it by pressing the 'enter' key.

4.5 Advantages and Integration

By automating data acquisition and segmentation, this script:

- Reduces manual annotation time when expanding the training set.
- Ensures consistent cropping and synchronization between textual descriptions and video segments.
- Can be integrated seamlessly with the existing feature-extraction pipeline by outputting uniformly cropped segment clips ready for I3D[4]/pose[3] feature extraction.

5 Acknowledgements

I would like to express my sincere gratitude to **Prof. Ashutosh Modi** for offering me the opportunity to undertake this Undergraduate Research Project. I am deeply thankful to my PhD mentor, **Sanjeet Singh**, for his valuable guidance, constant support, and insightful feedback throughout the course of this work.

5.1 Code Availability

All code and supplementary scripts for this project are publicly hosted at: `https://github.com/rcharan05/UGP`.

References

[1] Exploration-Lab, *cislr_v1-a dataset*, https://huggingface.co/datasets/Exploration-Lab/CISLR.

[2] Abhinav Joshi, Ashwani Bhat, Pradeep S, Priya Gole, Shashwat Gupta, Shreyansh Agarwal, Ashutosh Modi (2022) *CISLR: Corpus for Indian Sign Language Recognition*, Joshi et al., EMNLP 2022. https://aclanthology.org/2022.emnlp-main.707.pdf

[3] Amit Moryossef, Mathias Müller, and Rebecka Fahrni. *pose-format: Library for viewing, augmenting, and handling .pose files*, 2021. Available at: https://github.com/sign-language-processing/pose

[4] Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? A new model and the kinetics dataset. CVPR 2017. https://arxiv.org/abs/1705.07750

[5] Li, D., Rodriguez, C., Yu, X., & Li, S. (2020). Word-Level Deep Sign Language Recognition from Video: A New Large-Scale Dataset and Methods Comparison. WACV 2020. https://arxiv.org/abs/1910.11006

[6] Lugaresi, C., et al. (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv preprint arXiv:1906.08172. https://arxiv.org/abs/1906.08172