ASL Recognition Pipeline with Neural Sequence Modeling (Work in Progress)

DEMO Video link:
https://drive.google.com/file/d/1-FCDnimxuKwMv2kciRjpx8JcUPfeQU0H/view?usp=sharing

1. Overview

This project aims to build an end-to-end system that converts American Sign Language (ASL) video input into fluent English text.

Existing ASL recognition systems typically output isolated sign labels (glosses) but do not address the grammatical differences between ASL and English. For example, an ASL sequence such as:

YESTERDAY STORE I GO

needs to be transformed into natural English:

I went to the store yesterday.

This project focuses on building a modular pipeline that combines:

- Computer vision (gesture extraction)
- Temporal sequence modeling
- (Planned) language modeling

with an emphasis on **offline inference, reproducibility, and robustness under imperfect real-world data**.

2. System Architecture

Pipeline:

Video Input
→ MediaPipe (hand keypoint extraction)
→ Temporal feature engineering (normalization + velocity)
→ BiGRU-based sequence model
→ ASL gloss prediction
→ (Planned) LLM post-processing
→ Fluent English output

Key design decision:

- Use **landmark-based representation (instead of raw video)** to reduce computational cost and enable CPU-only inference.

3. Completed Work

Data Pipeline

- Processed the MS-ASL dataset (1000 classes, YouTube-based video sources)
- Implemented a full pipeline:
    - URL extraction → batch download (yt-dlp)
    - Clip segmentation based on timestamps
    - Landmark extraction using MediaPipe (dual-hand tracking)
- Designed robust handling for:
    - Missing videos
    - Corrupted clips
    - Partial extraction failures

Feature Engineering

- Extracted **126-dim hand keypoints per frame (both hands)**
- Applied:
    - Per-frame normalization (relative to wrist)
    - Scale normalization
    - Temporal velocity features
- Final input representation:
  (T, 252) = [position + velocity]

Model

- Implemented **BiGRU-based sequence classifier** in PyTorch
- Added:
    - Attention pooling (instead of last-frame representation)
    - Label remapping for filtered subsets
    - Weighted sampling to address class imbalance

Training System

- Built full training pipeline:
    - Dataset filtering (by label frequency)
    - Train/val/test split validation
    - Model checkpointing
    - Evaluation pipeline
- Implemented improvements:
    - AdamW optimizer
    - learning rate scheduling
    - gradient clipping
    - label smoothing

4. Current Results

- Model: BiGRU + Attention
- Input: MediaPipe dual-hand keypoints + velocity
- Dataset: filtered MS-ASL subset

Data Reality (important engineering insight)

- ~40% of original MS-ASL videos were unavailable (deleted/private)
- Final usable dataset significantly reduced
- Required rebuilding dataset from scratch

Final Training Setup

- Classes: ~529
- Train samples: ~9100
- Val samples: ~2200
- Test samples: ~1800

Performance

- Best validation accuracy: ~0.17–0.18
- Test accuracy: ~0.17

Smaller subset experiments

- With stricter filtering (higher samples per class):
    - Accuracy improves significantly

    - Demonstrates strong dependence on data quality

5. Key Technical Insights

1. Dataset Quality Dominates Performance

- Major bottleneck is not model capacity
- Missing videos + class imbalance severely degrade performance
- Filtering high-quality subsets improves accuracy more than architecture changes

2. Overfitting Behavior

- Observed:
    - Loss ↓ continuously
    - Accuracy plateaus or drops
- Indicates:
    - Model memorization
    - insufficient per-class diversity

3. Temporal Modeling Matters but Has Limits

- GRU captures short-term dependencies well
- Struggles with:
    - long sequences
    - subtle transitions
    - signer variability

4. Landmark Representation Tradeoff

Pros:

- Fast
- Lightweight
- Works on CPU

Cons:

- Loses:
    - facial expressions
    - body posture
    - fine visual context

6. Challenges

1. Dataset Issues (Major Real-World Constraint)

- ~40% video failure rate (YouTube links invalid)
- Many classes have:
    - very few samples
    - or missing val/test splits
- Required:
    - dynamic filtering
    - rebuilding label space

2. Class Imbalance

- Long-tail distribution:
    - few classes with many samples
    - many classes with very few samples

3. Cross-Signer Variability

- Different:
  - speed
  - style
  - hand shape
- Leads to poor generalization

4. Temporal Alignment Problem

- Fixed-length sampling (T=64) may:
  - include irrelevant frames
  - miss critical motion details

5. Representation Limitation

- Only using hands ignores:
  - facial expressions (important in ASL)
  - body orientation

7. Ongoing Work

Model Improvements

- Experimenting with:
  - Attention-based sequence models
  - Transformer-style architectures
  - Temporal convolution (TCN)

Data Strategy

- Building **high-quality subset filtering pipeline**
- Focusing on:
  - classes with sufficient samples
  - balanced splits

Feature Expansion

- Considering:
  - MediaPipe Holistic (pose + face + hands)
  - richer spatio-temporal features

Language Modeling

- Designing integration with local LLM:
  - gloss → English translation
- Exploring:
  - lightweight offline inference (llama.cpp)

8. Future Direction

Short-Term

- Improve classification accuracy via:
  - better sampling strategies
  - stronger regularization
  - improved feature extraction

Mid-Term

- Replace GRU with:

    - Transformer / attention models
    - or skeleton-based graph models (e.g., ST-GCN)

Long-Term

- Full pipeline:
    - continuous sign recognition
    - sentence-level modeling
    - grammar-aware translation

9. Demo

Current Demo

- Input: single ASL video (user-recorded)
- Output:
    - predicted sign label
    - top-5 predictions with confidence

System Characteristics

- Fully offline
- CPU-compatible
- End-to-end pipeline (video → prediction)

Planned Demo

- Video → gloss sequence → fluent English sentence

10. Tech Stack

- Python
- PyTorch
- MediaPipe (Tasks API)
- OpenCV
- NumPy / Pandas

(Planned)

- Local LLM inference (llama.cpp or similar)