# ASL Translation Project — Draft Skeleton

## Prefilled Draft (based on your repo + the example report)

**Repo:** ellahazelbar/DeepWS → *ASL Translation System* with CNN+LSTM baseline, offline word-level translation, config/entry points under asl\_translator/src/training/train.py and demo at demo/app.py. Data expected in asl\_translator/data/raw. Use this section as the starting point for your draft write‑up; keep the rest of the template for depth.

### 1) Project Overview (filled)

* **One‑liner:** Translate short ASL video clips (single-word signs) into English using a CNN+LSTM video model; explore a transformer upgrade.
* **Motivation/Impact:** Lower the barrier for non-signers to understand isolated ASL words; towards accessible interfaces.
* **Contributions:** (i) Reproducible CNN+LSTM baseline; (ii) Clean data pipeline for grayscale clips; (iii) Ablations on clip length/resolution/augmentation; (iv) Error analysis & demo app.
* **Success criteria:** Top‑1 ≥ **70%** and macro‑F1 ≥ **0.70** on a signer‑held‑out validation set; latency ≤ **120 ms/clip** on consumer GPU.

### 2) Problem Statement & Scope (filled)

* **Task:** *Isolated sign classification* (word‑level). Stretch: short **sequence → text** with CTC.
* **Output:** English word from a fixed vocabulary (size TBD from dataset). Stretch: gloss sequence.
* **Constraints:** Grayscale input; no external pretraining beyond standard CV backbones unless permitted.

### 3) Related Work (pointer)

* CNN+RNN for sign recognition; video transformers (TimeSformer/ViViT); pose‑keypoint pipelines (MediaPipe/OpenPose) as an optional fusion track.

### 4) Data (filled)

* **Primary:** ASL Alphabet/Word datasets (Kaggle) + any course-provided set.
* **Splits:** Train/Val/Test with signer independence if possible.
* **Preprocessing:** Extract T=**32** frames per clip at ~**25 FPS**; resize **224×224**; grayscale normalization; optional horizontal flip (beware handedness), temporal jitter.

### 5) Pipeline & Augmentations (filled)

* torch.utils.data dataset returning [T,1,H,W]; padding/masks for variable T.
* Augs: random crop/scale, flip, brightness/contrast; toggleable via YAML.
* Optional cache of preprocessed tensors for fast training.

### 6) Models (filled)

* **Baseline:** 2D‑CNN (ResNet‑18) frame encoder → BiLSTM(2×512) → classifier (vocab size K).
* **Sequence head (stretch):** CTC over gloss tokens.
* **Upgrade path:** Video Transformer (TimeSformer‑B) with AdamW + cosine decay; optional late fusion with pose keypoints.

### 7) Training & Repro (filled)

* AMP on; grad‑clip 1.0; AdamW lr=3e‑4, cosine, warmup 5 epochs; batch 8; epochs 40–60.
* Checkpoint best **macro‑F1**; log with W&B; deterministic seeds; config‑driven runs.

### 8) Evaluation (filled)

* **Metrics:** Top‑1/Top‑5, macro‑F1. (Stretch: WER/CER for CTC.)
* **Reports:** Learning curves, confusion matrix, per‑sign length breakdown, signer breakdown.

### 9) Experiments & Ablations (filled)

1. Frames **16/32/48**; 2) Res **112/224**; 3) CNN+LSTM vs. Video Transformer; 4) Aug on/off; 5) Grayscale vs. replicated 3‑ch; 6) With/without keypoints.

### 10) Results Tables (prefilled headers)

| Model | Frames | Res | Params(M) | Top‑1 ↑ | Top‑5 ↑ | F1 ↑ | FPS ↑ |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CNN+LSTM (baseline) | 32 | 224 |  |  |  |  |  |
| TimeSformer‑B (strong) | 32 | 224 |  |  |  |  |  |

### 11) Error Analysis (filled)

* 8–10 failure cases: frame strips + predicted vs. gold; analyze confusions (e.g., *M* vs *N*, *F* vs *T*), motion blur, occlusion.

### 12) Inference & Demo (filled)

* CLI: python src/inference/predict.py --video path.mp4 → top‑k.
* Demo: python demo/app.py (local). Export ONNX/TorchScript for portability.

### 13) Risks & Ethics (filled)

* Signer bias; dataset imbalance; privacy of video data. Mitigate with signer‑held‑out splits, per‑class weighting, and documented consent.

### 14) Project Management (filled)

* **Roles:** Data | Modeling | Eval | Demo.
* **Timeline (4 weeks):** W1 data+baseline; W2 strong model; W3 ablations+analysis; W4 demo+report.

### 15) Deliverables (filled)

* Code + configs; Draft report PDF; 10–12‑slide deck; short demo video.

**How to use this doc:** Fill the TODO: fields. Delete any sections that aren’t required by your instructor. I kept defaults assuming *video-to-gloss/text* with grayscale clips, but the structure also fits isolated sign classification.

## 1) Project Overview

**One-liner:** TODO (e.g., "Translate American Sign Language videos to English glosses using a video transformer with CTC decoding.")

**Motivation / Impact:** TODO

**Contributions (bullet list):** - TODO Baseline reimplementation - TODO Strong model variant - TODO Ablations + error analysis - TODO Lightweight demo

**Success criteria (quantitative):** TODO (e.g., WER ≤ 40% on val; BLEU-4 ≥ 0.35; real-time ≥ 10 FPS on RTX 3060)

## 2) Problem Statement & Scope

* **Task type:** TODO (Isolated classification vs. continuous translation)
* **Target output:** TODO (Gloss sequence vs. English text)
* **Labeling granularity:** TODO (video-level / clip-level / frame-level)
* **Constraints:** Grayscale videos (default), no external data unless stated.
* **Assumptions:** TODO

## 3) Related Work (short)

Cite 3–6 core papers / baselines you compare against. - TODO [1] (Video 3D CNNs + RNNs + CTC) - TODO [2] (Video Transformers / TimeSformer / MViT) - TODO [3] (Keypoint-based approaches: OpenPose/MediaPipe + seq models)

Include 1–2 lines per reference: dataset, method gist, and metric.

## 4) Data

**Datasets:** TODO (WLASL / Phoenix-2014T / Custom)

**Split policy:** TODO (speaker-independent? temporal?)

**Stats (template):** | Split | #Videos | Total Hours | Avg Frames | Avg Len (glosses) | |—|—:|—:|—:|—:| | Train | | | | | | Val | | | | | | Test | | | | |

**Ethics & rights:** TODO (licenses, consent, privacy)

**Preprocessing:** - Read MP4 → frames at TODO (e.g., 25 FPS) - Resize to TODO (e.g., 224×224), grayscale normalization - Clip sampling TODO (fixed T frames / sliding windows) - Optional: hand/pose keypoints cache TODO

## 5) Data Pipeline & Augmentations

* Dataloader w/ torch.utils.data.Dataset + DataLoader
* Padding/masking for variable length
* Augmentations (toggleable): temporal jitter, random crop/scale, horizontal flip (note ASL handedness!)
* Caching strategy (LMDB / memmap / .pt tensors) TODO

## 6) Models

### 6.1 Baseline (reproducible)

* **Backbone:** 3D-ResNet18 (grayscale in-ch=1)
* **Head A (Classification):** GlobalAvgPool → MLP → Softmax
* **Head B (Sequence):** 3D-ResNet features → BiLSTM (2×512) → CTC over gloss vocab
* **Loss:** Cross-entropy (A) or CTC (B)

### 6.2 Strong Model (primary)

* **Video Transformer:** TimeSformer/MViT/ViViT (patchify over space, factorized time)
* **Decoder options:**
  + CTC over gloss tokens, or
  + Seq2Seq: Transformer decoder (teacher forcing) for gloss/English
* **Regularization:** dropout, stochastic depth
* **Optimization:** AdamW, cosine decay, warmup

### 6.3 Optional Fusion (if keypoints available)

* Keypoint encoder (MLP or GCN) + late fusion with video tokens

**Vocabulary:** TODO (gloss set size, special tokens: <pad>, <sos>, <eos>, blank)

## 7) Training & Reproducibility

* Mixed precision (AMP)
* Gradient clipping (e.g., 1.0)
* Batch size TODO, Accumulate grads if needed
* Checkpointing: best val WER/BLEU + last
* Determinism: seed all RNGs, cudnn flags
* Config-driven runs (.yaml) and experiment logging (W&B)

**Hyperparameters (template):**

run\_name: TODO  
seed: 42  
optimizer: adamw  
lr: 3.0e-4  
weight\_decay: 0.05  
epochs: 50  
sched: cosine  
warmup\_epochs: 5  
batch\_size: 8  
clip\_len: 32 # frames  
image\_size: 224  
model: timesformer\_base  
head: ctc  
vocab\_size: TODO  
ctc\_blank\_id: 0  
amp: true

## 8) Evaluation

**Metrics (pick by task):** - Classification: Top-1/Top-5 accuracy, macro-F1 - Translation/Gloss: **WER**, **CER**, BLEU-{1..4}, ROUGE-L - Latency (ms/clip) CPU vs GPU, throughput (clips/s)

**Reporting:** - Learning curves (train/val) - Confusion matrix (classification) - Per-length WER, per-signer breakdown

## 9) Experiments & Ablations (plan)

1. **Backbone**: 3D-ResNet18 → ResNet34 → Video-Transformer
2. **Clip length**: 16 vs 32 vs 48 frames
3. **Head**: CTC vs Seq2Seq decoder
4. **Input**: grayscale vs 3-channel grayscale (replicated) vs RGB (if allowed)
5. **Resolution**: 112 vs 224
6. **Keypoints**: off vs on (if feasible)

Cap at 5–8 runs for the draft; full report can expand.

## 10) Results (tables to fill)

**Main results (val):** | Model | Head | Frames | Res | Params(M) | WER ↓ | CER ↓ | BLEU-4 ↑ | FPS ↑ | |—|—|—:|—:|—:|—:|—:|—:|—:| | Baseline-3DRes18 | CTC | 32 | 224 | | | | | | | VideoTransformer-B | CTC | 32 | 224 | | | | | |

**Ablations:** | Change | Metric | Δ | |—|—|—| | Frames 16→32 | WER | TODO |

## 11) Error Analysis

* Show 5–10 failure cases with input frame strips + predicted vs gold gloss/English
* Class-wise errors (top confusions)
* Token-level alignments from CTC (where insertions/deletions occur)
* Correlate errors with clip length, signer, motion blur

## 12) Inference & Demo

* Export TorchScript/ONNX TODO
* Simple Streamlit/Gradio demo: upload MP4 → show predicted gloss/text + confidence
* Performance: measure end-to-end latency on laptop GPU/CPU

## 13) Risks & Mitigations

* **Data scarcity / domain shift:** augmentations, transfer learning, regularization
* **Overfitting:** early stopping, dropout, label smoothing
* **Ethical use:** bias across signers; accessibility goals; consent & privacy

## 14) Project Management

**Team roles:** TODO (e.g., Model, Data, Evaluation, Demo)

**Timeline (example, adapt dates):** - Week 1: Data audit + baseline reproduced - Week 2: Strong model, initial tuning - Week 3: Ablations + error analysis - Week 4: Demo + report + slides

**Milestones:** TODO

## 15) Deliverables

* **Code** (repo, reproducible configs)
* **Draft report** (this doc → PDF)
* **Slides** (10–12 slides)
* **Demo** (optional)