Sign Language Recognition

Static ASL Alphabets and Dynamic Words

Randil Wijayananda | DAEN 429 | December 2025

1. Introduction

This project implements a sign language recognition system in two phases. Static ASL alphabet classification using transfer learning with ResNet-18. Phase two involves dynamic word-level video recognition, using temporal modeling with LSTM.

Phase 1 Dataset: ASL Alphabet dataset from Kaggle containing 87,000 images across 29 classes (A-Z plus SPACE, DELETE, NOTHING).

Phase 2 Dataset: WLASL100 dataset containing 100 sign language words as short video clips.

2. Phase 1: Static ASL Classification

2.1 Methodology

Data Split: Stratified 80/20 split with seed=429. Training: 69,600 images, Validation: 17,400 images.

Model: ResNet-18 pretrained on ImageNet with modified fc layer (1000 to 29 classes). BatchNorm layers set to eval() mode when frozen.

Freezing Strategies:

• T-A: Head-only (fc layer). Trainable: 14,877 params (0.1%)

• T-B: layer4 + fc. Trainable: 8.4M params (75.1%)

• T-C: Progressive unfreezing from T-B checkpoint (layer3 + layer4 + fc). Trainable: 10.5M params (93.9%)

• S-A: Training from scratch. Trainable: 11.2M params (100%)

2.2 Phase 1 Results

Table 1: Phase 1 Ablation Study Results (Validation Set)

Experiment	Layers Trained	Val Accuracy	Val Macro-F1	Epochs
T-A (Head Only)	fc only	94.35%	0.9434	15
T-B (Last Block)	layer4 + fc	99.97%	0.9997	15
T-C (Progressive)	layer3+4 + fc	**100.00%**	**1.0000**	10
S-A (Scratch)	All layers	100.00%	1.0000	30

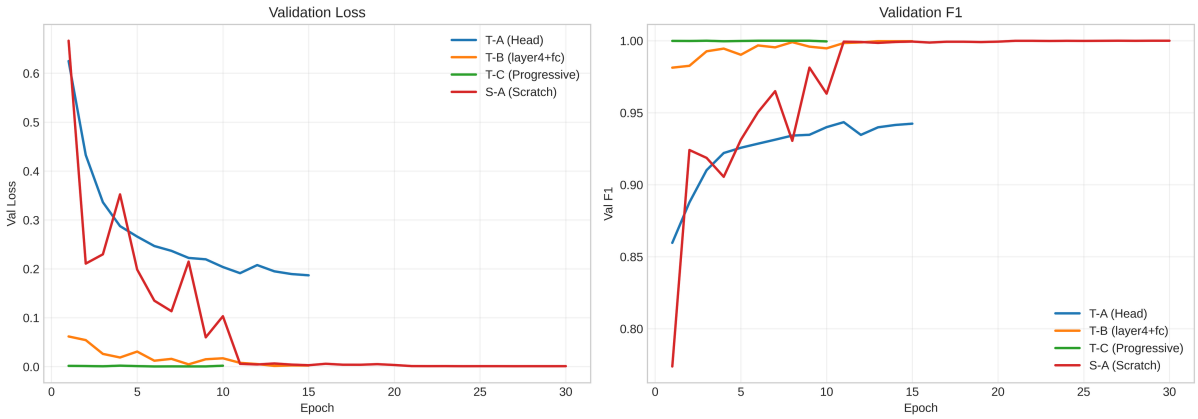

Figure 1: Validation loss and F1 comparison across all Phase 1 experiments. T-C and S-A both achieve F1=1.0, but T-C converges significantly faster.

Model Selection: T-C was selected as the best model. Both T-C and S-A achieved perfect validation performance (F1=1.0000). T-C converged in only 3 epochs (from T-B checkpoint) compared to 27 epochs for S-A. This shows the efficiency advantage of transfer learning.

Table 2: Phase 1 Test Set Performance (Model: T-C)

Test Set	Images	Accuracy	Macro-F1
Original Test Set	26	100.00%	1.0000
Custom Test Set (10 classes)	20	50.00%	0.2980

The performance gap between the original test set (100%) and the custom set (50%) show a clear domain shift. The model generalizes well to data similar to the training distribution but struggles with real world variations in lighting and background.

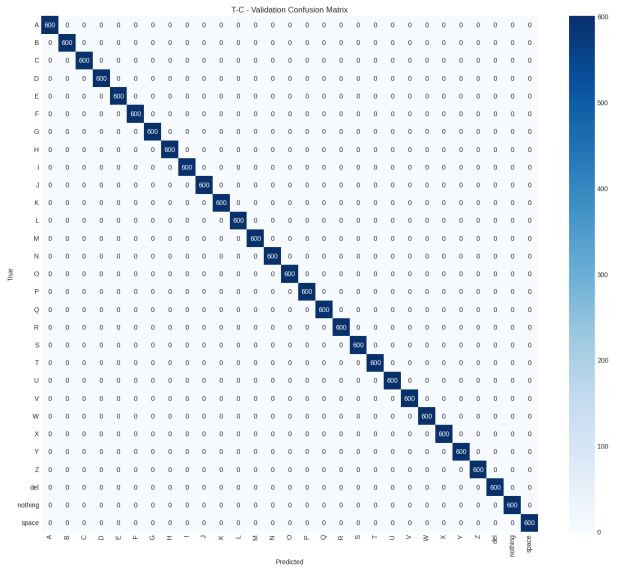

Figure 2: T-C validation confusion matrix (17,400 images, 29 classes).

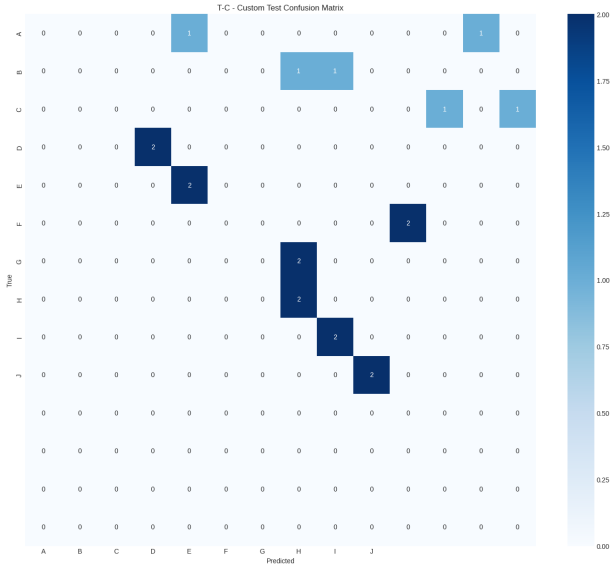

Figure 3: T-C custom test confusion matrix (20 images, 10 classes).

3. Phase 2: Dynamic Word Recognition (Bonus)

3.1 Methodology

Feature Extractor: The best Phase 1 model (T-C) was used as a per frame feature extractor. The classification head was removed. This outputs 512 dimensional features per frame via global average pooling.

Temporal Model: Bidirectional LSTM with 2 layers (256 hidden units) to capture temporal dynamics. The final hidden states were concatenated and passed through a classification head (512 to 256 to 100).

Video Processing: 16 frames extracted per video (sampling every 2nd frame). Batch size: 8 videos.

Training Stages:

• 2A: Freeze CNN, train LSTM + classifier only (20 epochs)

• 2B: Unfreeze layer4, train layer4 + LSTM + classifier (15 epochs, starting from 2A checkpoint)

3.2 Phase 2 Results

Table 3: Phase 2 Ablation Study Results (Validation Set)

Experiment	Configuration	Val Accuracy	Val Macro-F1
2A (Freeze CNN)	LSTM only	2.48%	0.0121
2B (Unfreeze layer4)	layer4 + LSTM	**7.02%**	**0.0472**

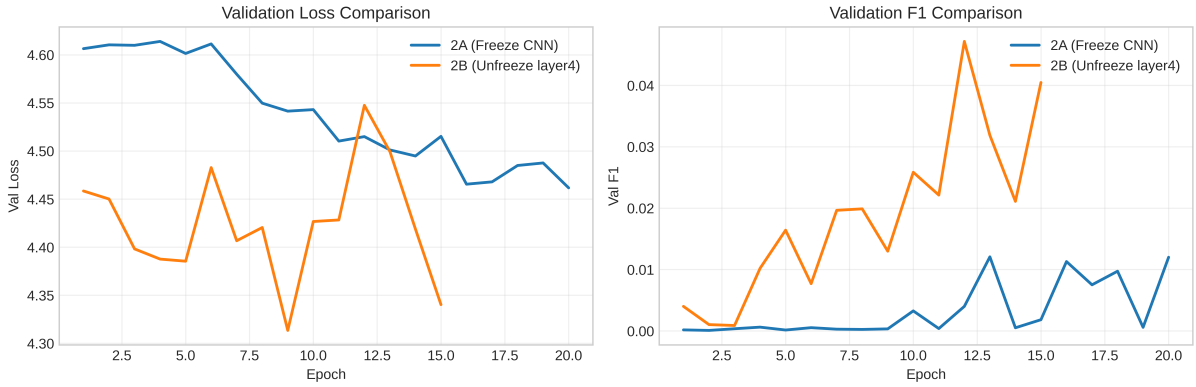

Figure 4: Phase 2 validation loss and F1 comparison (2A vs 2B).

Test Set Results (Model 2B): Accuracy: 4.00%, Macro-F1: 0.0205

Analysis: Unfreezing layer4 (2B) improved F1 by 3.51% over frozen CNN (2A). The low overall performance is expected given the challenging 100 class video classification task with limited training data (~10 videos per class). The best performing classes included 'play' (F1=0.40), 'thin' (F1=0.33), and 'go' (F1=0.33).

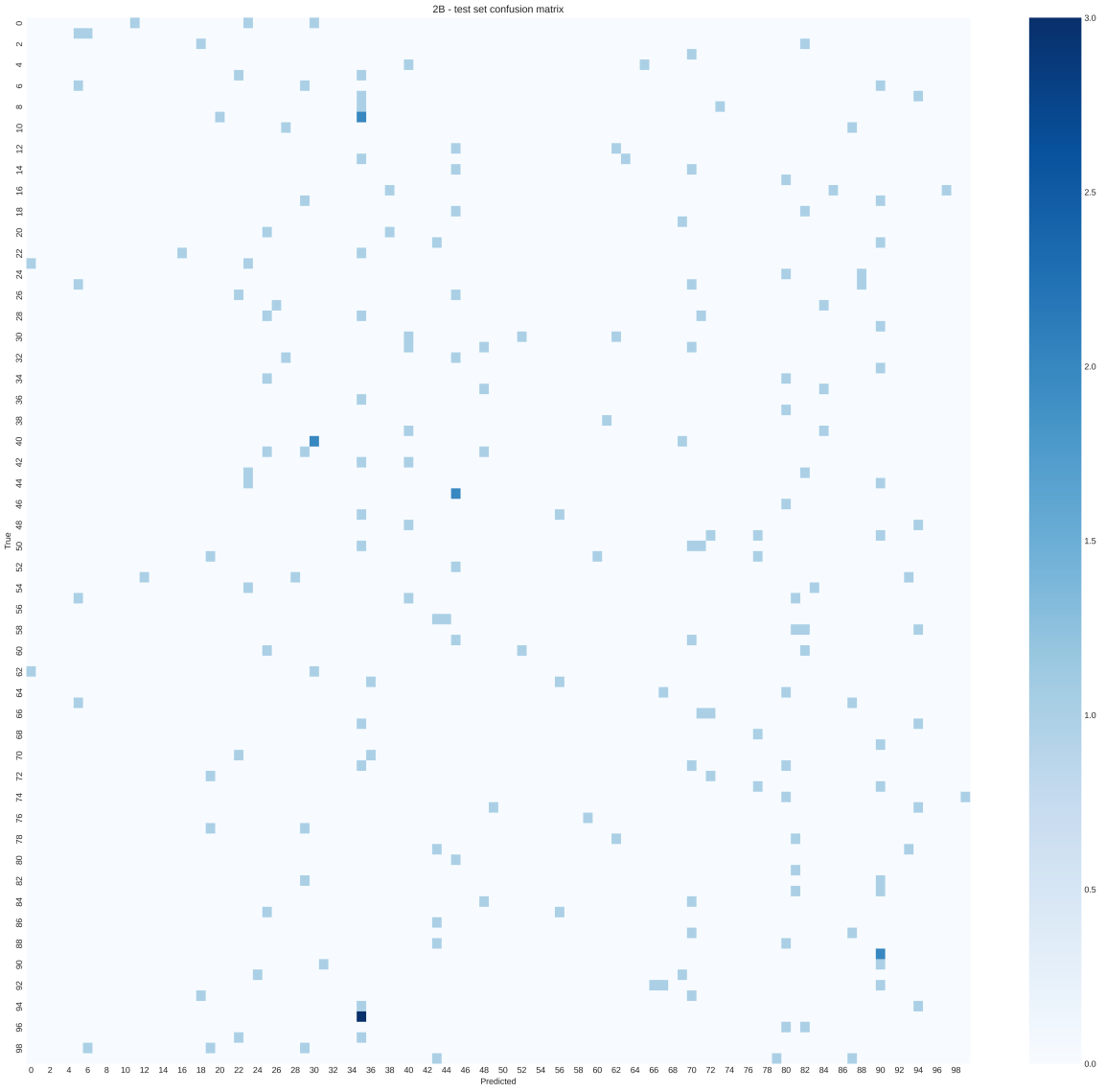

Figure 5: Model 2B test confusion matrix (100 classes).

4. Analysis & Discussion

4.1 Transfer Learning vs Training from Scratch

In Phase 1, both T-C (transfer learning with progressive unfreezing) and S-A (training from scratch) achieved identical final performance of 100% accuracy. However, T-C demonstrated significant advantages:

- **Convergence Speed:** T-C reached 100% in 3 epochs vs. 27 epochs for S-A (9x faster)
- **Training Stability:** Transfer learning showed smoother loss curves with less variance
- **Computational Efficiency:** Fewer trainable parameters and epochs reduce GPU time significantly

4.2 Domain Gap and Real World Performance

The performance drop from 100% (original test) to 50% (custom test) in Phase 1 shows a significant domain gap. The training data was captured under controlled conditions, while custom images had various lighting, backgrounds, and hand positions. This highlights the importance of diverse training data for real world deployment.

4.3 Challenges in Video Classification

Phase 2's lower performance (4-7%) compared to Phase 1 (100%) stems from several factors: (1) increased complexity of 100 classes vs. 29, (2) temporal variation in sign execution, (3) limited training data (approx.10 videos per class), and (4) the challenge of distinguishing similar signs that differ only in motion patterns.

5. Conclusion

This project successfully demonstrated sign language recognition across two modalities:

Phase 1: Progressive unfreezing (T-C) achieved 100% accuracy with 9x faster convergence than training from scratch. This validates transfer learning's effectiveness for static ASL classification.

Phase 2: Temporal modeling with BiLSTM achieved 7.02% validation accuracy on 100-class video recognition. Despite the low accuracy, unfreezing layer4 improved performance by 3.51% over frozen features.

Future Work: Improve Phase 2 performance through data augmentation, attention mechanisms, or other temporal models (Transformers). Address domain gap through diverse training data or domain adaptation techniques.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. CVPR.

[2] ASL Alphabet Dataset: https://www.kaggle.com/datasets/grassknoted/asl-alphabet

[3] WLASL100 Dataset: https://www.kaggle.com/datasets/thtrnphc/wlasl100-new