Static ASL Hand Gesture Classification via Deep Convolutional Neural Networks

by Leisha Khapre & Gabriel Gonzalez

December 08, 2025

***Note**: The Dynamic Words (Bonus) part of the project was not completed.*

Submitted to fulfill the requirement for Final Project Report
DAEN 429, Fall 2025
Department of Industrial & Systems Engineering
Texas A&M University

1. Introduction

American Sign Language (ASL) is a vital means of communication for millions of Deaf and hard-of-hearing individuals. With advances in human–computer interaction, there's a growing interest in systems that can reliably recognize hand gestures in real time. Static ASL alphabet recognition, with its structured set of poses, aligns well with computer vision approaches.

Deep learning, particularly Convolutional Neural Networks (CNNs), has proven effective for image classification, and transfer learning allows pretrained models to adapt to new domains with limited data. This project explores ResNet-18 with several fine-tuning strategies, evaluating performance on both official ASL datasets and real-world gestures collected by one of the report's authors. The goal is to identify the training setup that best generalizes and achieves high accuracy in static ASL alphabet classification.

2. Data & Experimental Setup

2.1 Datasets

This study draws on a publicly available ASL alphabet dataset containing static hand gesture images across 29 classes: the 26 English letters (A–Z) plus three additional gestures (SPACE, DELETE, NOTHING). Each class includes thousands of samples with diverse hand shapes, backgrounds, and lighting conditions, providing a strong foundation for model training. To test generalizability, two evaluation sets were used:

- **Official test set (28 images)**: one representative sampler per class from the ASL dataset
- **Real-world test set (20 images)**: hand gestures photographed by the author

The second set mirrors real deployment scenarios and introduces domain-shift challenges, offering a more rigorous measure of model performance beyond controlled-dataset environments.

2.2 Preprocessing

All images were standardized using the same preprocessing pipeline:

- Resized to 224 × 224 pixels, the required input size for ResNet-18
- Converted to normalized tensors using ImageNet mean and standard deviation

Data augmentations, including random horizontal flips, color jitter, and random rotations, were applied only during training to increase variability and reduce overfitting. Validation and test images were processed without augmentation to preserve real-world conditions.

2.3 Train/Validation Split

The original ASL dataset was divided into an 80/20 stratified train–validation split to preserve class balance. A fixed random seed ensured reproducibility, and the validation set was reserved solely for model selection, never incorporated into training.

3. Models & Fine-Tune Strategy

3.1 Base Architecture - ResNet-18

ResNet-18 is a widely used deep residual CNN that employs skip connections to address the vanishing gradient problem. It remains computationally lightweight while still learning highly discriminative visual features. For this work, the pretrained ImageNet version of ResNet-18 served as the foundation for transfer learning. The original classification head was replaced with a fully connected layer of size 29, matching the number of ASL gesture classes.

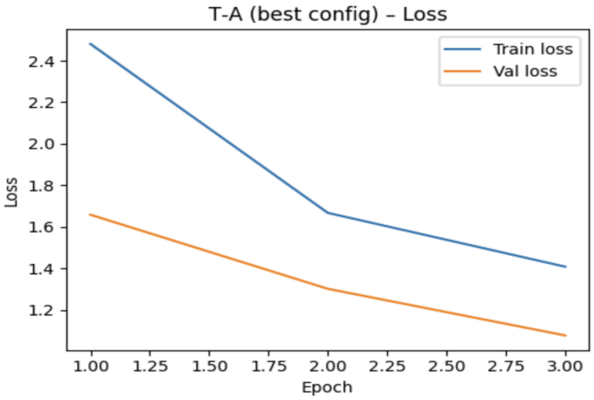
3.2 Ablation Policies

To investigate the effectiveness of transfer learning for ASL recognition, four different model configurations were developed. The first method (T‑A) kept the entire pretrained ResNet‑18 backbone frozen while training only the newly added classifier head, allowing us to evaluate how well ImageNet‑learned features transfer to ASL gestures with minimal adaptation. The second strategy (T‑B) expanded this by unfreezing the final residual block, enabling the network to refine higher‑level spatial features more specialized for hand shape recognition. A more advanced approach, T‑C, employed a progressive fine‑tuning policy in which training began with only the head unfrozen and gradually extended to deeper layers, the last one or two blocks, allowing the optimizer to adjust deeper filters without destabilizing pretrained knowledge. Finally, a baseline model (S-A), created and trained entirely from scratch, was included to assess the benefit of transfer learning compared to learning all weights directly from the ASL dataset. Together, these four strategies enabled a controlled ablation study of how the extent of fine‑tuning affects performance, generalization, and training efficiency.

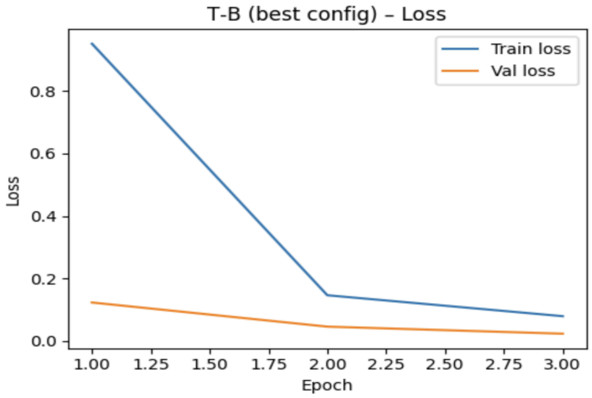
4. Results

4.1 Validation Ablation Study

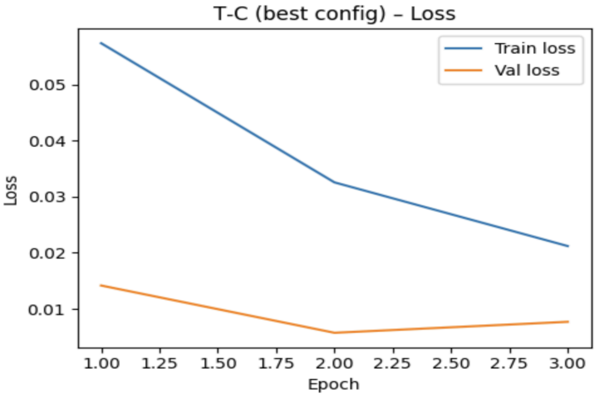
The best model for T-A had a learning rate of 1e-3 with three epochs.



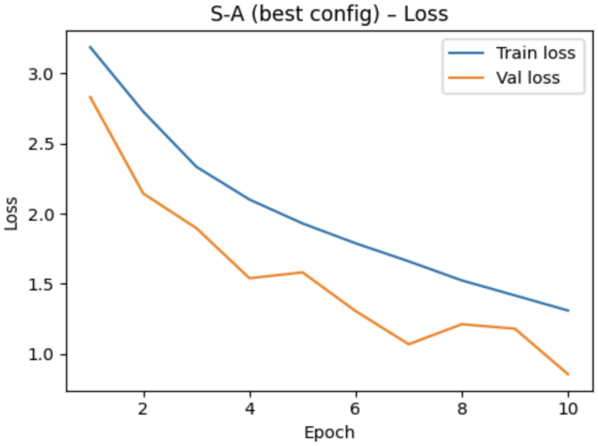
The best T-B model ran for three epochs with a learning rate of 1e-4.



The best T-C model had a learning rate of 1e-4 and ran for three epochs.



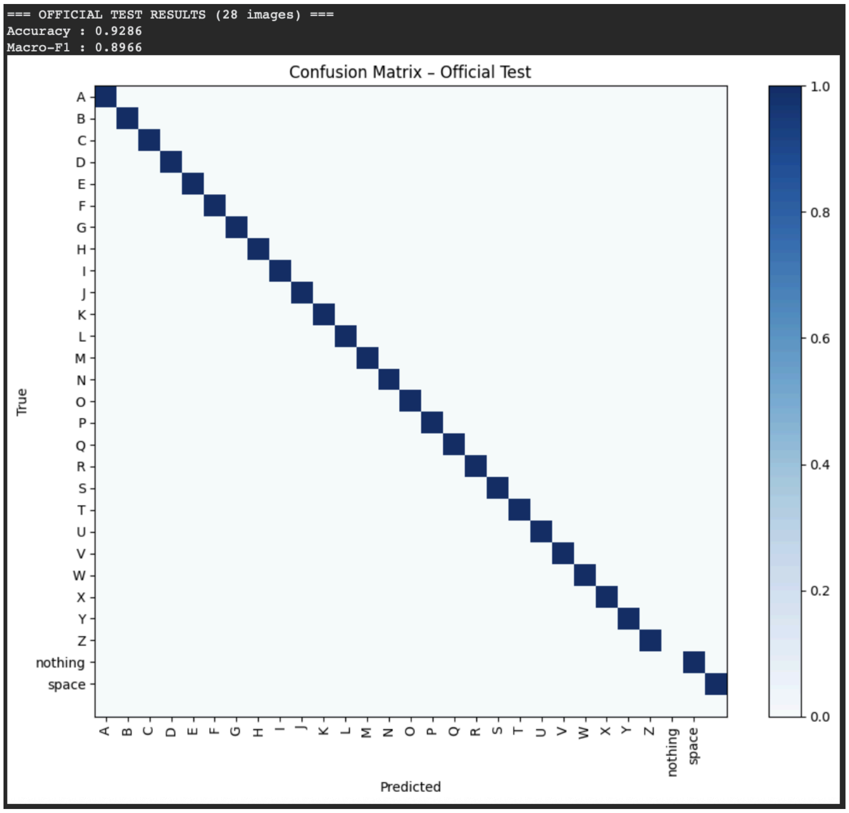
The best S-A model ran for 10 epochs with a learning rate of 1e-3.



As seen in the table below, the T-C model performed the best. It had the best macro-F1 score among the four models on the validation dataset. The T-C model was, therefore, selected for testing on the official Kaggle and custom testing sets.

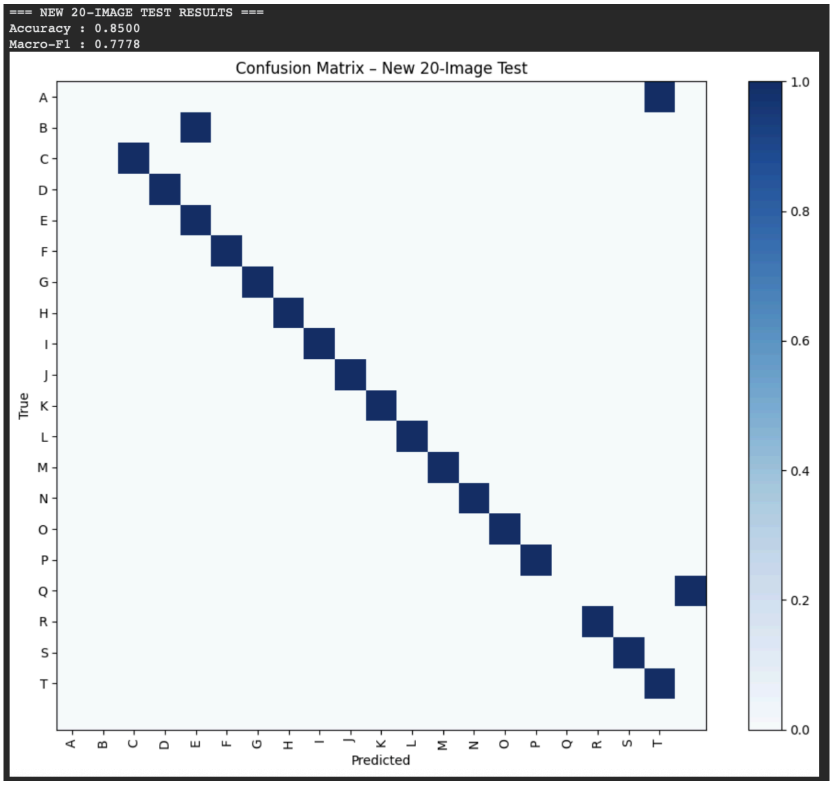
Model	**Validation Macro F1**
TA	0.713592
TB	0.995864
TC	0.999540
SA (Custom CNN)	0.657838

4.2 Testing on Official 28-Image Set



```
=== OFFICIAL TEST RESULTS (28 images) ===
Accuracy : 0.9286
Macro-F1 : 0.8966
```

4.3 Testing on Real New (Custom) 20-Image Set



```
=== NEW 20-IMAGE TEST RESULTS ===
Accuracy : 0.8500
Macro-F1 : 0.7778
```

4.4 Discussion

Of all the models, T-C performed the best; T-B was next, and T-A was third. The S-A model, created and trained from scratch, performed the worst.

Upon testing the T-C model on the official test set, an accuracy of 0.9286 (92.86%) was obtained. The macro F1 score on this was 0.8966. For the customs dataset, the accuracy was 0.8500 (85%) and the F1 score was 0.7778. The model performed worse on the custom dataset by a small margin.

5. Conclusions

S-A performed the worst because it had considerably fewer layers than the pretrained ResNet. It also had no pretrained weights. Limited training data and a low number of epochs due to computational constraints led to a low F1 score. T-A did not perform well because it was too restricted- only the head was trainable. It froze the entire network except for the last layer. T-B was a significant improvement over T-A because it was more flexible. The model could reshape higher-level features for ASL signs while also referencing the robust ImageNest features. T-C was the best model since it was built on top of the T-B model and used the already adapted ResNet block and head classifiers, along with layer 4. T-C demonstrated that progressive fine-tuning leads to better generalization on both the test set and on the more challenging real-world image set than the other models.