Final Report: Hand Gesture Recognition Using CNNs

Mohammed Almassri Jeremy Forest Hossam Khalifa Hagop Minassian

Laith Qasem

Abstract

This report investigates hand-gesture recognition using Segmentation-Guided Background Augmentation (SGBA) and standard computer-vision techniques. We evaluated eleven specific models: nine trained from scratch and two via transfer learning. Experiments spanned three datasets—SignAlphaSet, ASL Digits, and HaGRID—utilizing ShuffleNetV2, ResNet-34, and VGG-16 backbones. Using accuracy as the primary metric and training time as a tiebreaker, ResNet-34 emerged as the best overall performer, securing top placements on two datasets. Surprisingly, ResNet-34 consistently matched or even exceeded the training efficiency of the lighter ShuffleNetV2 model architecture. Additionally, the VGG-16 transfer-learning model significantly outperformed its from-scratch counterpart on ASL Digits and SignAlphaSet, improving accuracy by 10% and reducing average training time by approximately 42%. Key methodologies included AdamW optimization, data augmentation, and checkpoint-based model selection. Ultimately, while individual model performance varies by environment, lighter architectures remain better suited for limited training setups. ResNet-34 provides an optimal balance between high accuracy and low computational cost overall.

1. Introduction

With global projections indicating rising hearing loss [16], more accessible assistive technologies are urgently needed. Hand-sign and gesture recognition can both support accessibility and evolve into a general human–computer interaction modality [9]. We also aim to enable interactive learning and other real-world applications beyond pure accessibility [10].

This project develops several AI-driven gesture-recognition models using Segmentation-Guided Background Augmentation (SGBA) and standard computer-vision techniques in order to achieve high accuracy while maintaining computational efficiency.

There are many challenges associated with computer vision techniques and machine learning, specifically when attempting to transition from training-controlled environments, often oversimplified ones, to real-life environments with many unpredictable backgrounds and visuals. Possible solutions for these challenges utilized in other research include segmentation and data augmentation.

Segmentation is the process of extracting and isolating the regions of interest within an image, such as the hand for gesture recognition systems. This approach can help increase a model's robustness and training accuracy by reducing background noise and forcing the model to focus on relevant features [12]. However, segmentation has limitations: difficulty in accurately separating regions of interest in diverse backgrounds, reduced suitability for real-time applications, and an increase in computational overhead [12].

Data augmentation involves modifying training images to improve model generalization and reduce risks of overfitting [1]. Random rotations, flipping, image blurring, and lighting changes are common techniques. These transformations introduce more diversity to the training data, helping models adapt better to real-world conditions. This is often employed when the dataset is too simple to reflect real-world occurrences. However, data augmentation can hardly replicate true real-life environments on its own [17]. It is important to note that when applied only during training, augmentation does not negatively impact real-time inference performance.

The solution proposed in this research addresses the challenge of oversimplified datasets, which fail to highlight real-world diversity, by combining segmentation and data augmentation. First, hand regions are isolated from the original dataset to generate masks mirroring segmentation. Second, these masks are composited onto a separate background dataset prepared in advance. Finally, augmentations such as blur and color transformations are applied to further simulate real-world conditions.

Importantly, this preprocessing and augmentation was performed only during model training. This allows the

models to benefit from the increased data variability and improved robustness while still avoiding segmentation's computational overhead during real-time inference. As a result, this proposed approach enhances model performance without sacrificing efficiency in deployment.

We developed 11 models across three datasets (SignAlphaSet, ASL Digits, and HaGRID) comprising nine trained from scratch and two utilizing transfer learning via VGG-16. Each dataset follows a dedicated preprocessing and augmentation pipeline before training on ShuffleNetV2 1.0x, ResNet-34, and VGG-16 architectures. To ensure a controlled comparison, models within each dataset share identical hyperparameters, excluding two parameters tuned specifically for architectural optimization. All models utilize the AdamW optimizer, with the final selection based on the lowest validation loss. Evaluation includes confusion matrices, training logs, Grad-CAM visualizations, and classification reports ($Precision$, $Recall$, and F_1-score), alongside training time to analyze the computational cost of each implementation.

Across datasets, our experiments report strong accuracy while highlighting trade-offs between model capacity and computational cost; full quantitative results are presented in Sec. 3.

1.1. Related Works

Hand-gesture recognition has moved from traditional hand-crafted features to deep CNN-based methods; recent work emphasizes robustness in unconstrained environments. Segmentation and saliency methods improve focus on the hand region and reduce background noise [12], but reliable segmentation in the wild adds runtime cost. Data-centric approaches use geometric, photometric, and compositing augmentations to increase training diversity [10, 14, 17], yet synthetic variability can still fall short of natural scene complexity.

Architecture and transfer-learning work compare lightweight backbones (e.g., ShuffleNetV2 1.0x) against deeper models (ResNet-34, VGG-16) to trade off accuracy and compute [4, 8, 15]. Transfer learning from large-scale datasets often improves convergence and accuracy when domains align. Our SGBA combines segmentation-guided compositing at training time with standard augmentations to increase realism while avoiding inference-time segmentation, positioning the method between costly segmentation pipelines and augmentation-only strategies.

2. Methodology

2.1. Datasets

Three different datasets with varying degrees of complexity were employed in this report. See Table 1 for a statistical summary of each dataset.

Dataset	Total Images	Resolution	Classes	Img/Class	Source
SignAlphaSet [2]	26,000	296×296	26	1,000	Mendeley
ASL Digits [13]	5,000	400×400	10	500	Kaggle
HaGRID 512p [5]	10,000	512×512	2	5,000	Kaggle

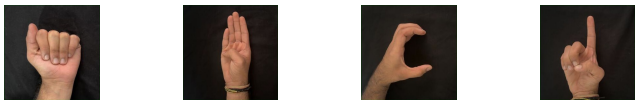
Table 1. Summary Statistics of the Selected Datasets

The following covers specific details about each dataset:

- SignAlphaSet: Images contain a single hand on a black background with controlled illumination. This dataset represents a medium-complexity domain: more classes and finer-grained hand poses require stronger discrimination and benefit from targeted augmentations.
- ASL Digits: Images were preprocessed with MediaPipe to isolate the hand and provide a uniform black background. ASL Digits is a low-complexity, constrained domain (uniform backgrounds, low intra-class variability), useful for testing baseline performance.
- HaGRID 512p: This dataset exhibits high visual complexity (diverse subjects, backgrounds, and lighting). HaGRID stresses model robustness and generalization due to wide intra-class variation and domain shifts.



(a) ASL Digits



(b) SignAlphaSet



(c) HaGRID

Figure 1. Visual comparison of dataset samples across different levels of environmental and class complexity.

All the datasets were filtered and preprocessed before being integrated into the training pipeline. The images

were split into training, validation, and test sets using a fixed ratio of $70\% : 10\% : 20\%$. This same split was applied consistently across all datasets to allow the models to learn from the training data, utilize the validation set for real-time monitoring, and undergo final evaluation on a consistent test set. A fixed hold-out split was chosen over K-fold cross-validation primarily due to the high computational wall-clock time required to retrain deep architectures like VGG-16 multiple times. Furthermore, given the large cumulative scale of the datasets (approximately $41,000$ images), a 10% validation slice is statistically representative enough to ensure stable performance metrics without the need for multiple folds [3]. To prepare the data for the CNN models, all images were resized to a target resolution of 224×224 pixels using the `torchvision.transforms.v2` library. We utilized `RandomResizedCrop` with `antialias=True` to ensure high-quality downsampling and provide scale invariance. We implemented two distinct preprocessing and augmentation strategies based on the environmental complexity and native characteristics of the datasets. For the SignAlphaSet and ASL Digits datasets, which feature highly controlled environments and uniform black backgrounds, we applied an intensive augmentation pipeline to improve model robustness. This included a custom background replacement technique to swap the uniform backgrounds with various environmental scenes, alongside random rotations up to $15°$, random horizontal flips, and color jitter (adjusting brightness and contrast by 0.2). In contrast, a more streamlined augmentation strategy was applied to the HaGRID dataset; because it already contains significant natural variation in backgrounds and lighting, we limited augmentations to the random resized cropping and color jitter. For the validation and test sets, images were simply resized and center-cropped without any augmentations to ensure evaluation was performed on real, unseen images. Finally, all images were normalized according to the ImageNet mean ($[0.485, 0.456, 0.406]$) and standard deviation ($[0.229, 0.224, 0.225]$) [6] to align with our goal of comparing the different models fairly.

2.2. CNN Models

To classify hand gestures in this study, we selected ShuffleNetV2 1.0x, ResNet-34, and VGG-16 to represent a broad spectrum of computational complexity. While other popular architectures such as MobileNetV2 or DenseNet were considered, our chosen trio was prioritized because they offer distinct, well-established benchmarks for lightweight, moderate, and high-capacity performance. This specific range allows for a clear analysis of the trade-offs between a model's structural capacity and its resulting accuracy and computational cost.

All three architectures utilize a standard input size of $224 \times 224 \times 3$ and incorporate ReLU activation functions within their convolutional layers to introduce non-linearity [4, 8, 15]. The final fully connected layer of each model employs a Softmax function to provide a probability distribution across the gesture classes. Architectural specifications and theoretical requirements for each model are summarized in Table 2.

Model	Params (M)	GFLOPs	Tier	Epoch Speed
ShuffleNetV2 1.0x	2.3	0.15	Lightweight	Fast
ResNet-34	21.8	3.6	Moderate	Moderate
VGG-16	138	15.5	High	Slow

Table 2. Architectural Complexity and Theoretical Requirements

The following provides a detailed description of each architecture and its relevance to our hand-gesture recognition task:

- ShuffleNetV2 1.0x: Designed for efficiency, this model utilizes channel shuffle operations to facilitate information flow between feature groups. While this approach significantly reduces computational demand and increases training speed compared to VGG or ResNet, its lightweight nature may capture less granular detail than deeper networks. This allows us to test the lower bound of accuracy and computational cost for our task.

- ResNet-34: This architecture introduces residual connections that allow information to skip layers, mitigating the vanishing gradient problem. By creating shortcuts between convolutional layers, ResNet-34 facilitates a manageable training process and the learning of significant image features, providing a balance between accuracy and efficiency. This serves as a strong baseline for moderate complexity, allowing us to evaluate the benefits of deeper architectures without excessive computational overhead.

- VGG-16: Characterized by a deep stack of convolutional layers, VGG-16 is capable of learning highly intricate features. However, this high capacity results in prolonged training times and more extensive computational demands than the other two models. While it may achieve the highest accuracy, it also represents the upper bound of computational cost in our experiments, allowing us to analyze the trade-offs between model complexity and performance.

2.3. Optimization Algorithm and Hyperparameters

All models were trained using the AdamW optimizer, which decouples weight decay (1×10^{-4}) from the adaptive gradient update to ensure consistent regularization across all layers. The training objective is Cross-Entropy Loss:

$\mathcal{L} = -\sum_{i=1}^{C} y_i \log(\hat{y}_i)$, where C represents the gesture classes. Optimization was validated by monitoring epoch-wise training and validation logs (accuracy and loss) to identify the optimal convergence point and prevent overfitting. Hyperparameters were tuned based on model capacity and dataset complexity (Table 3): a batch size of 32 was used throughout, with a learning rate of 1×10^{-3} for ShuffleNetV2 1.0x and ResNet-34 (HaGRID/ASL Digits) and 1×10^{-4} for VGG-16 and all models on SignAlphaSet to ensure stability. Performance is reported using Confusion Matrices and Classification Reports (Precision, Recall, F1-score) to evaluate per-class reliability. To ensure the models prioritize hand geometry over background artifacts, we utilize Grad-CAM for visual interpretability. This confirms that the CNN's attention is localized on the hand's morphological features rather than environmental noise. Weight decay is 1×10^{-4} for all entries.

Dataset	Batch	Epochs	LR (ShuffleNet / ResNet)	LR (VGG-16)
ShuffleNetV2 1.0x	2.3	0.15	Lightweight	Fast
ResNet-34	21.8	3.6	Moderate	Moderate
VGG-16	138	15.5	High	Slow

Table 3. Final Hyperparameter Configuration per Dataset

To select the best-performing checkpoint, validation loss was monitored at every epoch and the model weights achieving the minimum were saved:

$$\theta_{best} = \arg\min_{\theta_e} (\mathcal{L}_{val}) \tag{1}$$

where θ_e denotes the model weights at epoch e. This checkpoint, rather than the final epoch state, is used for all reported test evaluations to reflect best generalization performance.

To evaluate model performance, we utilized a comprehensive suite of metrics focused on both classification quality and computational efficiency. Accuracy provided the overall fraction of correct predictions, while Precision and Recall were used to balance the minimization of false alarms against the goal of capturing every intended user gesture. The F1-Score served as a harmonic mean of these two, acting as our primary metric for navigating uneven class costs. Additionally, Training Time was recorded as wall-clock time per model to compare the computational overhead across different architectures. A detailed rationale for these hyperparameter choices, along with comparisons against alternative configurations, is provided in the Ablation Study (see 3.4).

3. Results

3.1. Experiment Setup

3.1.1 Initial Experimental Phase

During the exploratory phase of the project, development and testing were distributed across multiple local workstations and Google Colab environments. This decentralized approach facilitated parallel experimentation with various CNN architectures and early-stage preprocessing techniques. However, the diversity of hardware platforms introduced inconsistencies in training latency and performance metrics. Consequently, this phase was primarily utilized to independently evaluate hyperparameter strategies and preprocessing configurations for each model to identify optimal baselines.

The baseline hyperparameters included a batch size of 16 and a learning rate of 1×10^{-3} for all models and datasets. These values were selected based on common default settings in the literature and preliminary testing [11, 18]. Regarding the training duration, the number of epochs was initially set to higher values for the HaGRID dataset due to its complexity, while 20 epochs were used as a starting point for the other two datasets, as they were expected to converge faster. We originally employed the Adam optimizer as this was the most commonly used optimization algorithm in the literature for similar tasks [18].

3.1.2 Transition to Unified Training and Optimization Strategy

To determine the optimal configuration for each architecture, training and validation loss were monitored alongside Grad-CAM visualizations and confusion matrices. Initial findings suggested that a uniform hyperparameter approach was insufficient across different architectures and datasets. While a learning rate of 1×10^{-3} was effective for the HaGRID and ASL Digits datasets, the VGG-16 models and the SignAlphaSet required a reduced learning rate of 1×10^{-4} to ensure stable convergence. This adjustment was necessary to prevent the optimizer from overshooting local minima, likely due to the higher parameter density of VGG-16 and the specific feature complexity of the SignAlphaSet. A more granular analysis of how these specific learning rate and batch size adjustments impacted model performance is provided in the Ablation Study (Section 3.4). Furthermore, the batch size was increased to 32 for all models. The rationale for this change was to leverage the high VRAM capacity of the available hardware to achieve more stable gradient estimates, which generally leads to better convergence. Finally, the optimization strategy shifted from Adam to AdamW with a weight decay of 1×10^{-4}. This transition was motivated by the desire to utilize decoupled weight decay, which has been shown to improve generalization per-

formance compared to standard L_2 regularization in Adam, particularly for complex classification tasks where overfitting is a primary concern [7]. The impact of this optimizer change on training dynamics and final performance is further analyzed in the Ablation Study 3.4.

3.2. Main Results and Analysis

3.2.1 Performance Metrics

The metrics used to evaluate and rank model performance overall were accuracy, precision, recall and the training time. Greater emphasis was placed on overall accuracy and training time since we did not assign different weights to false positives versus false negatives. This project does not involve implementations where one type of misclassification carries greater importance than the other, resulting in accuracy being the primary indicator for success. The training time metric was included as an indicator of computational cost, providing insight into the relative difficulty of training different model-dataset pairs and preparing them for real-world deployment. To ensure fair comparisons, all models were trained and evaluated on the same hardware environment to minimize discrepancies that can arise from differences in available resources rather than model-dataset pair characteristics.

Table 4 summarizes the comprehensive performance metrics for all 11 model-dataset combinations, including additional F1 score and training time per epoch.

Model Architecture	Acc.	Prec.	Rec.	F1	Time	Time/Ep.
HaGRID (35 Epochs)						
ShuffleNetV2 1.0x	0.98	0.98	0.98	0.98	6m 13s	10.7s
ResNet-34	0.98	0.98	0.98	0.98	5m 58s	10.2s
VGG-16	0.99	0.99	0.99	0.99	15m 39s	26.8s
ASL Digits (20 Epochs)						
ShuffleNetV2 1.0x	0.99	0.99	0.99	0.99	2m 51s	8.6s
ResNet-34	0.90	0.91	0.90	0.90	2m 51s	8.6s
VGG-16	0.88	0.90	0.88	0.87	4m 50s	14.5s
VGG-16-TL	0.98	0.98	0.98	0.98	2m 57s	8.9s
SignAlphaSet (15 Epochs)						
ShuffleNetV2 1.0x	0.96	0.96	0.96	0.96	8m 30s	34.0s
ResNet-34	1.00	1.00	1.00	1.00	8m 22s	33.5s
VGG-16	1.00	1.00	1.00	1.00	16m 31s	1m 6.1s
VGG-16-TL	1.00	1.00	1.00	1.00	8m 51s	35.4s

Table 4. Summary of performance metrics for all model-dataset combinations.

3.2.2 Additional Results

Additional result types such as training curves and Grad-CAM visualizations were collected as secondary data that offer information on deeper insight into model behavior beyond their final performance. These supplementary data types enable analysis of aspects such as convergence speed, training stability, and the ability of models to recognize the relevant regions in an image — all of which can help for future optimizations of each model. All data which falls under this category for each model-dataset combination are included in the Appendix (see A, B, C, and D) as they are too numerous to include in the main text. These include full epoch training logs, confusion matrices, classification reports, and Grad-CAM outputs.

3.3. Discussion of Results

3.3.1 Analysis

The experimental results in Table 4 highlight clear differences in model performance across datasets, architectures, and training durations. These differences shed light on important trade-offs between computational efficiency and accuracy, with several outcomes diverging from our expectations.

All models achieved generally strong and consistent performance across each of the datasets. The accuracies were all within similar ranges, aside from the exception of the ASL Digits dataset.

The ASL Digits dataset showed the largest performance variation. ShuffleNetV2 1.0x and VGG-16-TL achieved near-perfect results, while VGG-16 and ResNet-34 remained around 90%. This may be due to incomplete convergence within the 20 training epochs and instability observed in the training curves A, suggesting that additional training with more epochs could improve performance, with efficiency being the trade-off.

In contrast, HaGRID results were consistent across models, with each of the training curves highlighting proper generalization A. This made training time the primary differentiator. ResNet-34 achieved competitive accuracy with the lowest training cost, while VGG-16 remained the most expensive.

On SignAlphaSet, multiple models achieved perfect accuracy, indicating that the dataset is relatively easy to classify, as presumed earlier. ResNet-34 offered the best overall performance, maintaining perfect accuracy with the lowest training times.

Another surprising result was the comparison between the transfer-learning VGG-16 models and the from-scratch versions. The transfer-learning models performed better across the board, maintaining similar or better accuracy levels while nearly halving the training times. An explanation for this observation is that the datasets were simple enough that transfer-learning models adapted most of the needed features within a shorter time period. This can be inferred from the Grad-CAM visualizations C, where the VGG-16-TL models show different focus patterns compared to the regular VGG-16, suggesting faster and more stable adaptation in simpler domains.

Overall, performance depends on dataset characteristics and training conditions rather than model complexity alone. Each model type presents its own benefits, but generally requires tailored hyperparameters to perform at peak efficiency. Lightweight, moderate, and transfer-learning models offer the best trade-offs consistently across the presented datasets in constrained environments compared to architecturally heavier models.

3.3.2 Ranking methodology

Models were ranked using a priority-based approach that considers both accuracy and training time, with accuracy as the primary criterion.

Models were first compared by accuracy. When differences were significant, higher accuracy determined the ranking. When accuracy was similar ($\pm$ 3%), training time was used as a tiebreaker, favoring models that are more efficient. This ensures that accuracy remains the priority evaluation and training time only influences results when performance is comparable. Under this ranking methodology, efficiency outweighs small performance improvements.

This explains the observed rankings:

- ShuffleNetV2 1.0x ranked first on ASL Digits due to having the highest accuracy and lowest training time, with VGG-16-TL being a close second.
- ResNet-34 ranked first on HaGRID due to achieving similarly high accuracy with lower cost compared to other models.
- ResNet-34 also ranked first on SignAlphaSet among top-performing models due to better efficiency, though it could be interchangeable with VGG-16-TL.

Overall, this ranking system balances predictive performance and computational cost while ensuring that accuracy remains the dominant factor in model evaluation.

ASL Digits	HaGRID	SignAlphaSet
🥇 ShuffleNetV2	🥇 ResNet-34	🥇 ResNet-34
🥈 VGG-16-TL	🥈 ShuffleNetV2	🥈 VGG-16-TL
🥉 ResNet-34	🥉 VGG-16	🥉 ShuffleNetV2
4th VGG-16		4th VGG-16

Figure 2. Ranking of models for each dataset

3.4. Ablation Study

The following points summarize the key trends we observed during our preliminary ablation experiments. These findings directly informed our final choices for hyperparameters and preprocessing:

3.4.1 Optimizer and Weight Decay

For our training setup, we switched from the standard Adam optimizer to AdamW with a weight decay of 1×10^{-4}. While the final performance results were roughly the same for both optimizers, we noticed a significant difference in how they behaved during training. When testing ResNet-34 on the HaGRID dataset, the standard Adam optimizer remained quite jittery—showing fluctuations in the validation curves—throughout the epochs. In contrast, AdamW was initially unstable but settled as training progressed. We believe the decoupled weight decay in AdamW helped the model stabilize and prevented it from overfitting too early. Because AdamW was more stable in later stages of training, we decided to use it for all our models to keep our methodology consistent (see F.1).

3.5. Learning Rate

We experimented between using 1×10^{-3} and 1×10^{-4} for all models. The results showed that while 1×10^{-3} worked well for the lightweight ShuffleNetV2, it caused noisy and unstable validation loss for VGG-16 and occasionally ResNet-34. In some cases, like with VGG-16 on SignAlphaSet, the higher learning rate caused the model to overshoot local minima, leading to a learning plateau (see F.2). Dropping the rate to 1×10^{-4} for these deeper networks allowed the model to learn and converge much more smoothly. This was especially important for VGG-16, which has a much larger parameter space and is more sensitive to optimization instability.

3.6. Batch Size

Experiments were conducted comparing batch sizes of 16 and 32, while maintaining the optimized learning rates previously established for each model. Observations indicated that a batch size of 16 resulted in less consistent training curves for specific configurations, such as VGG-16 on the SignAlphaSet, and increased the total training time by approximately two minutes (see F.3). These results suggest that the smaller batch size was not as well-aligned with the chosen learning rates, leading to more fluctuations during training.

The slight increase in execution time for the smaller batch size is likely due to hardware efficiency. Processing smaller batches often introduces more overhead, as the system performs more frequent updates per epoch compared to the more efficient processing seen with a batch size of 32.

Ultimately, a batch size of 32 was selected for the final configurations. This decision was based on the fact that the larger batch size consistently produced lower average validation loss and better overall stability across all models and datasets, ensuring more reliable convergence.

References

[1] Ahmed Eid and Friedhelm Schwenker. Visual static hand gesture recognition using convolutional neural network. *Algorithms*, 16(8):361, 2023. 1

[2] Bindu Garg et al. Signalphaset. Mendeley Data, V3, 2025. 2

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 3

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 3

[5] Innominate817. Hagrid classification 512p. Kaggle, 2023. 2

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 25, pages 1097–1105, 2012. 3

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[8] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. ShuffleNet V2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018. 2, 3

[9] G. R. S. Murthy and R. S. Jadon. A review of vision based hand gestures recognition. *International Journal of Information Technology and Knowledge Management*, 2(2):405–410, 2009. 1

[10] Sothea Ngin, Toch Ensophea, et al. Feature extraction and convolutional neural networks for static hand gesture recognition and context sentence generation. *Cambodia Journal for Business and Professional Practice*, 1:34–52, 2024. 1, 2

[11] Johan Obando-Ceron, Marc G. Bellemare, and Pablo Samuel Castro. Small batch deep reinforcement learning, 2023. 4

[12] Xia Qi et al. Real-time hand gesture monitoring model based on mediapipe's registerable system. *Sensors*, 24(19):6262, 2024. 1, 2

[13] M. Rayeed. American sign language digit dataset. Kaggle, 2022. 2

[14] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 2019. 2

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 2, 3

[16] World Health Organization. Deafness and hearing loss. WHO Fact Sheets, 2025. 1

[17] M. Xiao et al. Image data augmentation techniques based on deep learning: A survey. *Mathematical Biosciences and Engineering*, 21(6):6190–6224, 2024. 1, 2

[18] Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenxi Sun, Zhuoran Chu, Bin Dong, and Zhi-Quan He. Adam-mini: Use fewer learning rates to gain more, 2024. 4

Appendix: Model Results and Visualizations

A. Training Curves

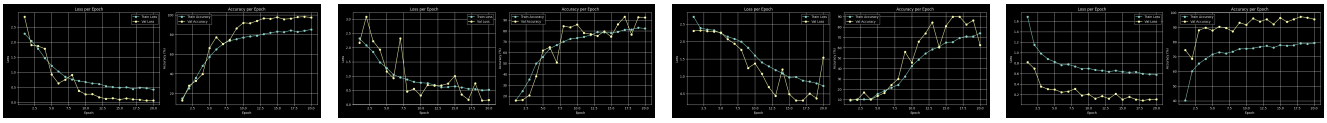


Figure 3. Training curves on ASL Digits: ShuffleNetV2 1.0x, ResNet-34, VGG-16, VGG-16-TL (left to right).

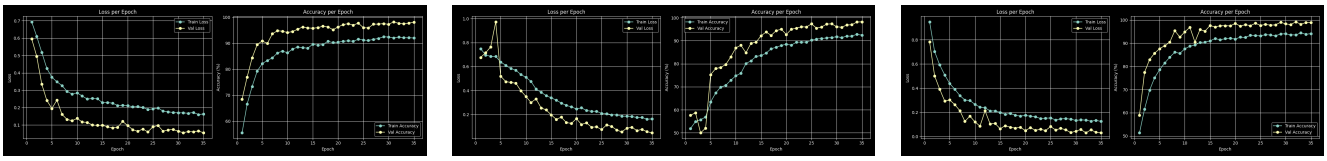


Figure 4. Training curves on HaGRID@: ShuffleNetV2 1.0x, ResNet-34, VGG-16 (left to right).

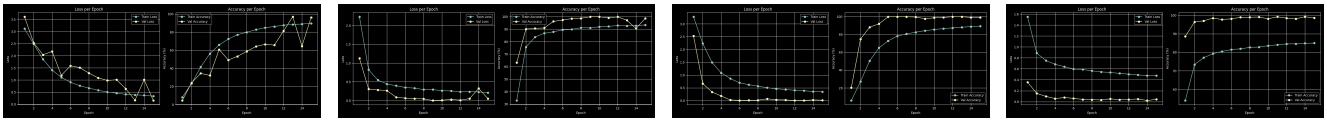


Figure 5. Training curves on SignAlphaSet: ShuffleNetV2 1.0x, ResNet-34, VGG-16, VGG-16-TL (left to right).

B. Confusion Matrices

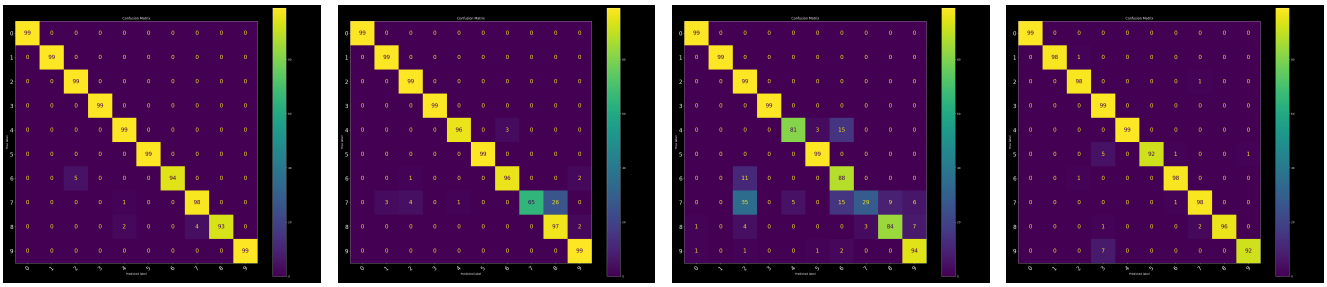


Figure 6. Confusion matrices on ASL Digits: ShuffleNetV2 1.0x, ResNet-34, VGG-16, VGG-16-TL (left to right).

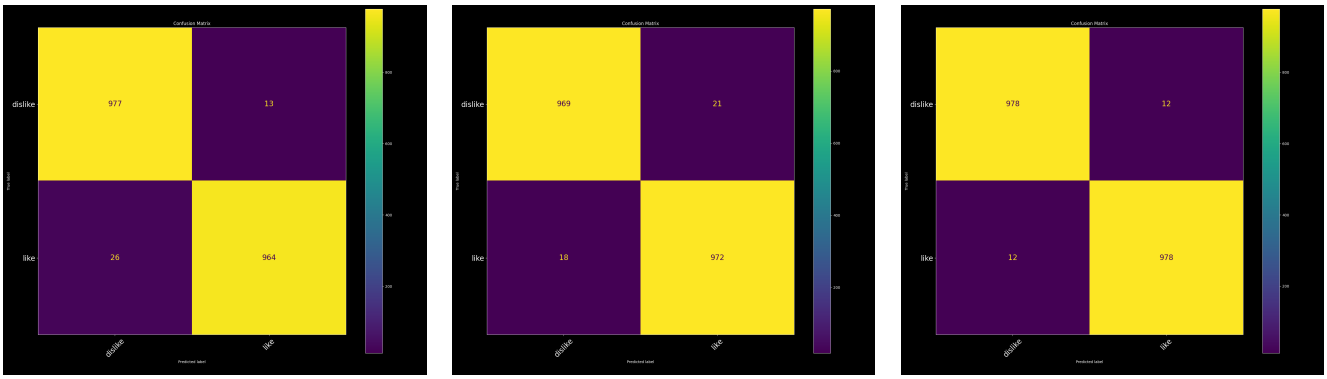


Figure 7. Confusion matrices on HaGRID@: ShuffleNetV2 1.0x, ResNet-34, VGG-16 (left to right).

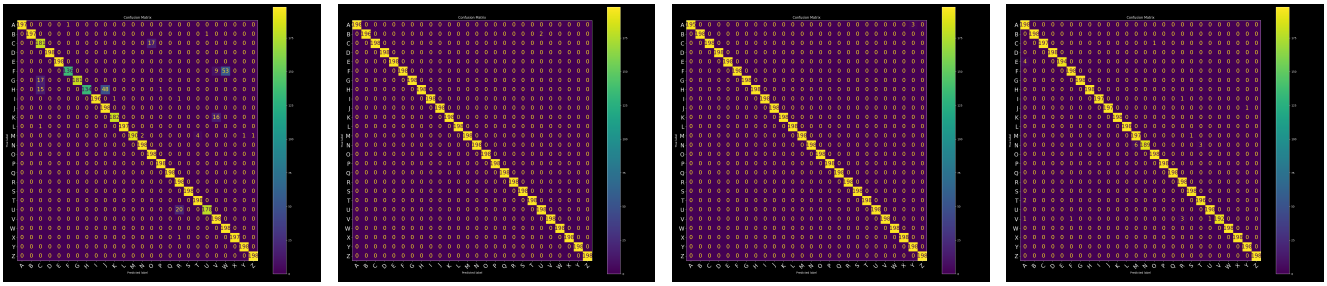


Figure 8. Confusion matrices on SignAlphaSet: ShuffleNetV2 1.0x, ResNet-34, VGG-16, VGG-16-TL (left to right).

C. Grad-CAM Visualizations

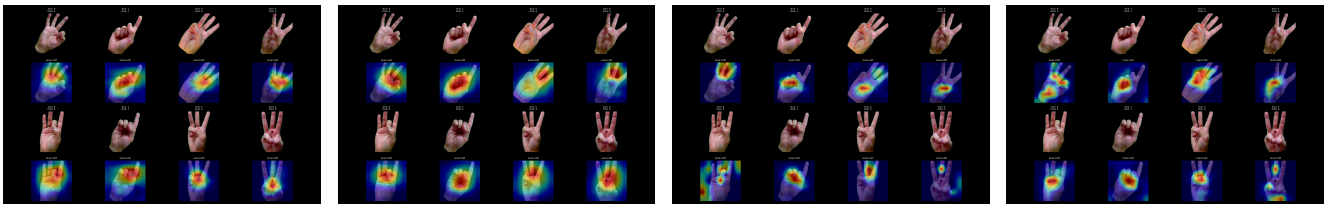


Figure 9. Grad-CAM visualizations on ASL Digits: ShuffleNetV2 1.0x, ResNet-34, VGG-16, VGG-16-TL (left to right).

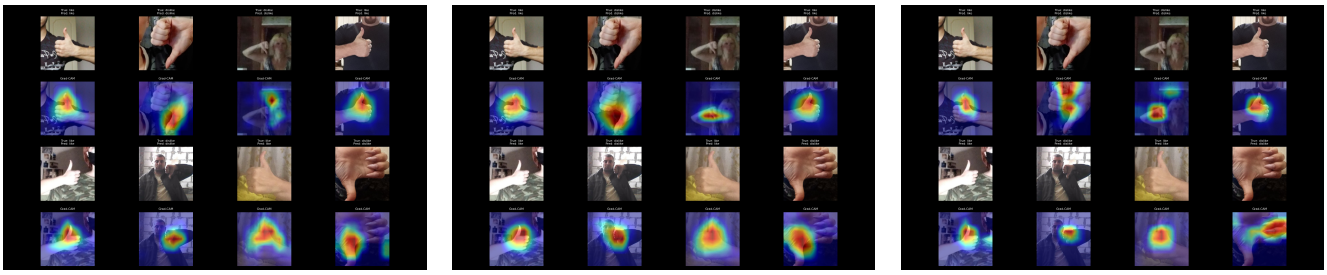


Figure 10. Grad-CAM visualizations on HaGRID@: ShuffleNetV2 1.0x, ResNet-34, VGG-16 (left to right).

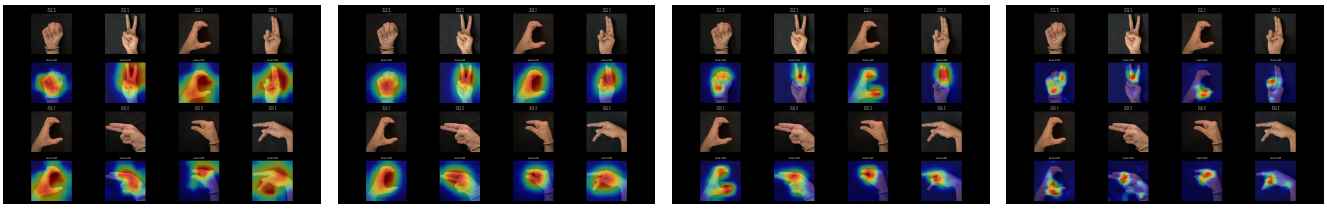


Figure 11. Grad-CAM visualizations on SignAlphaSet: ShuffleNetV2 1.0x, ResNet-34, VGG-16, VGG-16-TL (left to right).

D. Classification Reports

D.1. ASL Digits

ShuffleNetV2 (trained from scratch)

```
Test Accuracy  : 98.79%
Test Precision : 98.83%
Test Recall    : 98.79%
Test F1-Score  : 98.78%

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        99
           1       1.00      1.00      1.00        99
           2       0.95      1.00      0.98        99
           3       1.00      1.00      1.00        99
           4       0.97      1.00      0.99        99
           5       1.00      1.00      1.00        99
           6       1.00      0.95      0.97        99
           7       0.96      0.99      0.98        99
           8       1.00      0.94      0.97        99
           9       1.00      1.00      1.00        99

    accuracy                           0.99       990
   macro avg       0.99      0.99      0.99       990
weighted avg       0.99      0.99      0.99       990
```

ResNet-34 (trained from scratch)

```
Test Accuracy  : 95.76%
Test Precision : 96.32%
Test Recall    : 95.76%
Test F1-Score  : 95.56%

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        99
           1       0.97      1.00      0.99        99
           2       0.95      1.00      0.98        99
           3       1.00      1.00      1.00        99
           4       0.99      0.97      0.98        99
           5       1.00      1.00      1.00        99
           6       0.97      0.97      0.97        99
           7       1.00      0.66      0.79        99
           8       0.79      0.98      0.87        99
           9       0.96      1.00      0.98        99

    accuracy                           0.96       990
   macro avg       0.96      0.96      0.96       990
weighted avg       0.96      0.96      0.96       990
```

VGG-16 (trained from scratch)

```
Test Accuracy  : 87.98%
Test Precision : 89.65%
Test Recall    : 87.98%
Test F1-Score  : 86.75%

Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99        99
           1       1.00      1.00      1.00        99
           2       0.66      1.00      0.80        99
           3       1.00      1.00      1.00        99
           4       0.94      0.82      0.88        99
           5       0.96      1.00      0.98        99
           6       0.73      0.89      0.80        99
           7       0.91      0.29      0.44        99
           8       0.90      0.85      0.88        99
           9       0.88      0.95      0.91        99

    accuracy                           0.88       990
   macro avg       0.90      0.88      0.87       990
weighted avg       0.90      0.88      0.87       990
```

VGG-16-TL (transfer learning)

```
Test Accuracy  : 97.88%
Test Precision : 98.03%
Test Recall    : 97.88%
Test F1-Score  : 97.89%

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        99
           1       1.00      0.99      0.99        99
           2       0.98      0.99      0.98        99
           3       0.88      1.00      0.94        99
           4       1.00      1.00      1.00        99
           5       1.00      0.93      0.96        99
           6       0.98      0.99      0.98        99
           7       0.97      0.99      0.98        99
           8       1.00      0.97      0.98        99
           9       0.99      0.93      0.96        99

    accuracy                           0.98       990
   macro avg       0.98      0.98      0.98       990
weighted avg       0.98      0.98      0.98       990
```

D.2. HaGRID

ShuffleNetV2 (trained from scratch)

```
Test Accuracy  : 98.03%
Test Precision : 98.04%
Test Recall    : 98.03%
Test F1-Score  : 98.03%

Classification Report:
              precision    recall  f1-score   support

     dislike       0.97      0.99      0.98       990
        like       0.99      0.97      0.98       990

    accuracy                           0.98      1980
   macro avg       0.98      0.98      0.98      1980
weighted avg       0.98      0.98      0.98      1980
```

ResNet-34 (trained from scratch)

```
Test Accuracy  : 98.03%
Test Precision : 98.03%
Test Recall    : 98.03%
Test F1-Score  : 98.03%

Classification Report:
              precision    recall  f1-score   support

     dislike       0.98      0.98      0.98       990
        like       0.98      0.98      0.98       990

    accuracy                           0.98      1980
   macro avg       0.98      0.98      0.98      1980
weighted avg       0.98      0.98      0.98      1980
```

VGG-16 (trained from scratch)

```
Test Accuracy  : 98.79%
Test Precision : 98.79%
Test Recall    : 98.79%
Test F1-Score  : 98.79%

Classification Report:
              precision    recall  f1-score   support

     dislike       0.99      0.99      0.99       990
        like       0.99      0.99      0.99       990

    accuracy                           0.99      1980
   macro avg       0.99      0.99      0.99      1980
weighted avg       0.99      0.99      0.99      1980
```

D.3. SignAlphaSet

ShuffleNetV2 (trained from scratch)

```
Test Accuracy  : 95.92%
Test Precision : 96.48%
Test Recall    : 95.92%
Test F1-Score  : 95.83%

Classification Report:
              precision    recall  f1-score   support

           A       1.00      0.99      1.00       198
           B       1.00      0.99      1.00       198
           C       0.85      0.91      0.88       198
           D       1.00      1.00      1.00       198
           E       1.00      1.00      1.00       198
           F       0.99      0.69      0.81       198
           G       1.00      0.91      0.96       198
           H       1.00      0.68      0.81       198
           I       1.00      0.99      0.99       198
           J       0.80      1.00      0.89       198
           K       0.99      0.92      0.96       198
           L       1.00      0.99      1.00       198
           M       1.00      0.96      0.98       198
           N       0.99      1.00      0.99       198
           O       0.92      1.00      0.96       198
           P       0.99      1.00      1.00       198
           Q       1.00      1.00      1.00       198
           R       0.90      1.00      0.95       198
           S       1.00      1.00      1.00       198
           T       0.98      1.00      0.99       198
           U       0.99      0.90      0.94       198
           V       0.89      1.00      0.94       198
           W       0.79      1.00      0.88       198
           X       1.00      0.99      1.00       198
           Y       0.99      1.00      1.00       198
           Z       0.99      1.00      1.00       198

    accuracy                           0.96      5148
   macro avg       0.96      0.96      0.96      5148
weighted avg       0.96      0.96      0.96      5148
```

ResNet-34 (trained from scratch)

```
Test Accuracy  : 99.96%
Test Precision : 99.96%
Test Recall    : 99.96%
Test F1-Score  : 99.96%

Classification Report:
              precision    recall  f1-score   support

           A       1.00      1.00      1.00       198
           B       1.00      0.99      0.99       198
           C       1.00      1.00      1.00       198
           D       1.00      1.00      1.00       198
```

E	1.00	1.00	1.00	198
F	1.00	1.00	1.00	198
G	1.00	1.00	1.00	198
H	1.00	1.00	1.00	198
I	1.00	1.00	1.00	198
J	1.00	1.00	1.00	198
K	1.00	1.00	1.00	198
L	1.00	1.00	1.00	198
M	1.00	1.00	1.00	198
N	1.00	1.00	1.00	198
O	1.00	1.00	1.00	198
P	1.00	1.00	1.00	198
Q	1.00	1.00	1.00	198
R	1.00	1.00	1.00	198
S	1.00	1.00	1.00	198
T	1.00	1.00	1.00	198
U	0.99	1.00	0.99	198
V	1.00	1.00	1.00	198
W	1.00	1.00	1.00	198
X	1.00	1.00	1.00	198
Y	1.00	1.00	1.00	198
Z	1.00	1.00	1.00	198
accuracy			1.00	5148
macro avg	1.00	1.00	1.00	5148
weighted avg	1.00	1.00	1.00	5148

VGG-16 (trained from scratch)

```
Test Accuracy  : 99.94%
Test Precision : 99.94%
Test Recall    : 99.94%
Test F1-Score  : 99.94%

Classification Report:
```

	precision	recall	f1-score	support
A	1.00	0.98	0.99	198
B	1.00	1.00	1.00	198
C	1.00	1.00	1.00	198
D	1.00	1.00	1.00	198
E	1.00	1.00	1.00	198
F	1.00	1.00	1.00	198
G	1.00	1.00	1.00	198
H	1.00	1.00	1.00	198
I	1.00	1.00	1.00	198
J	1.00	1.00	1.00	198
K	1.00	1.00	1.00	198
L	1.00	1.00	1.00	198
M	1.00	1.00	1.00	198
N	1.00	1.00	1.00	198
O	1.00	1.00	1.00	198
P	1.00	1.00	1.00	198

```
           Q       1.00      1.00      1.00       198
           R       1.00      1.00      1.00       198
           S       1.00      1.00      1.00       198
           T       1.00      1.00      1.00       198
           U       1.00      1.00      1.00       198
           V       1.00      1.00      1.00       198
           W       1.00      1.00      1.00       198
           X       1.00      1.00      1.00       198
           Y       0.99      1.00      0.99       198
           Z       1.00      1.00      1.00       198

    accuracy                           1.00      5148
   macro avg       1.00      1.00      1.00      5148
weighted avg       1.00      1.00      1.00      5148
```

VGG-16-TL (transfer learning) — Epoch Log

```
Test Accuracy  : 99.51%
Test Precision : 99.52%
Test Recall    : 99.51%
Test F1-Score  : 99.51%

Classification Report:
              precision    recall  f1-score   support

           A       0.97      1.00      0.98       198
           B       1.00      1.00      1.00       198
           C       1.00      0.99      1.00       198
           D       1.00      1.00      1.00       198
           E       1.00      0.98      0.99       198
           F       0.99      1.00      1.00       198
           G       1.00      1.00      1.00       198
           H       1.00      1.00      1.00       198
           I       1.00      0.99      1.00       198
           J       1.00      0.99      1.00       198
           K       1.00      1.00      1.00       198
           L       1.00      1.00      1.00       198
           M       0.97      0.99      0.98       198
           N       1.00      0.95      0.98       198
           O       0.99      1.00      1.00       198
           P       1.00      1.00      1.00       198
           Q       1.00      1.00      1.00       198
           R       0.98      1.00      0.99       198
           S       0.99      1.00      1.00       198
           T       0.98      0.99      0.99       198
           U       0.99      1.00      1.00       198
           V       1.00      0.97      0.98       198
           W       1.00      1.00      1.00       198
           X       1.00      1.00      1.00       198
           Y       0.99      1.00      1.00       198
           Z       1.00      1.00      1.00       198

    accuracy                           1.00      5148
```

macro avg	1.00	1.00	1.00	5148
weighted avg	1.00	1.00	1.00	5148

E. Epoch Logs

E.1. ASL Digits

ShuffleNetV2 (trained from scratch) — Epoch Log

```
Epoch [1/20] Loss: 2.2863 | Acc: 14.97% | Val Loss: 2.8473 | Val Acc: 12.75%
--> Model Saved
Epoch [2/20] Loss: 2.0426 | Acc: 25.23% | Val Loss: 1.9028 | Val Acc: 27.84%
--> Model Saved
Epoch [3/20] Loss: 1.7852 | Acc: 35.80% | Val Loss: 1.8704 | Val Acc: 32.75%
--> Model Saved
Epoch [4/20] Loss: 1.4690 | Acc: 48.06% | Val Loss: 1.7830 | Val Acc: 39.80%
--> Model Saved
Epoch [5/20] Loss: 1.2174 | Acc: 57.31% | Val Loss: 0.9244 | Val Acc: 66.27%
--> Model Saved
Epoch [6/20] Loss: 1.0216 | Acc: 64.89% | Val Loss: 0.6311 | Val Acc: 77.25%
--> Model Saved
Epoch [7/20] Loss: 0.8529 | Acc: 70.31% | Val Loss: 0.7517 | Val Acc: 70.39%
Epoch [8/20] Loss: 0.7573 | Acc: 73.80% | Val Loss: 0.9113 | Val Acc: 74.71%
Epoch [9/20] Loss: 0.7132 | Acc: 75.29% | Val Loss: 0.3825 | Val Acc: 86.27%
--> Model Saved
Epoch [10/20] Loss: 0.6717 | Acc: 77.11% | Val Loss: 0.2671 | Val Acc: 91.96%
--> Model Saved
Epoch [11/20] Loss: 0.6284 | Acc: 78.26% | Val Loss: 0.2711 | Val Acc: 91.57%
Epoch [12/20] Loss: 0.6086 | Acc: 78.97% | Val Loss: 0.1706 | Val Acc: 93.92%
--> Model Saved
Epoch [13/20] Loss: 0.5386 | Acc: 80.51% | Val Loss: 0.1162 | Val Acc: 96.47%
--> Model Saved
Epoch [14/20] Loss: 0.5069 | Acc: 81.80% | Val Loss: 0.1383 | Val Acc: 96.08%
Epoch [15/20] Loss: 0.4832 | Acc: 83.00% | Val Loss: 0.0879 | Val Acc: 97.65%
--> Model Saved
Epoch [16/20] Loss: 0.5006 | Acc: 82.51% | Val Loss: 0.1276 | Val Acc: 95.69%
Epoch [17/20] Loss: 0.4436 | Acc: 84.26% | Val Loss: 0.0988 | Val Acc: 96.27%
Epoch [18/20] Loss: 0.4837 | Acc: 82.77% | Val Loss: 0.0833 | Val Acc: 97.84%
--> Model Saved
Epoch [19/20] Loss: 0.4678 | Acc: 83.94% | Val Loss: 0.0589 | Val Acc: 98.24%
--> Model Saved
Epoch [20/20] Loss: 0.4224 | Acc: 85.14% | Val Loss: 0.0582 | Val Acc: 97.45%
--> Model Saved

Total training time: 2m 51s
```

ResNet-34 (trained from scratch) — Epoch Log

```
Epoch [1/20] Loss: 2.3235 | Acc: 16.03% | Val Loss: 2.1718 | Val Acc: 15.88%
--> Model Saved
Epoch [2/20] Loss: 2.0812 | Acc: 24.66% | Val Loss: 3.0866 | Val Acc: 16.47%
Epoch [3/20] Loss: 1.8499 | Acc: 35.23% | Val Loss: 2.2306 | Val Acc: 20.59%
Epoch [4/20] Loss: 1.4847 | Acc: 49.97% | Val Loss: 1.9275 | Val Acc: 38.24%
--> Model Saved
Epoch [5/20] Loss: 1.2864 | Acc: 56.11% | Val Loss: 1.1628 | Val Acc: 61.96%
--> Model Saved
Epoch [6/20] Loss: 1.0499 | Acc: 63.71% | Val Loss: 0.9330 | Val Acc: 65.10%
```

```
--> Model Saved
Epoch [7/20] Loss: 0.9539 | Acc: 66.97% | Val Loss: 2.3233 | Val Acc: 50.98%
Epoch [8/20] Loss: 0.8803 | Acc: 70.09% | Val Loss: 0.4583 | Val Acc: 84.51%
--> Model Saved
Epoch [9/20] Loss: 0.7988 | Acc: 72.77% | Val Loss: 0.5499 | Val Acc: 83.33%
Epoch [10/20] Loss: 0.7660 | Acc: 73.54% | Val Loss: 0.3207 | Val Acc: 85.88%
--> Model Saved
Epoch [11/20] Loss: 0.7523 | Acc: 74.54% | Val Loss: 0.6875 | Val Acc: 78.24%
Epoch [12/20] Loss: 0.6895 | Acc: 76.26% | Val Loss: 0.6631 | Val Acc: 77.06%
Epoch [13/20] Loss: 0.6111 | Acc: 78.91% | Val Loss: 0.6728 | Val Acc: 75.49%
Epoch [14/20] Loss: 0.6167 | Acc: 78.86% | Val Loss: 0.7365 | Val Acc: 79.61%
Epoch [15/20] Loss: 0.6405 | Acc: 78.17% | Val Loss: 1.0050 | Val Acc: 74.71%
Epoch [16/20] Loss: 0.6051 | Acc: 79.14% | Val Loss: 0.3524 | Val Acc: 86.08%
Epoch [17/20] Loss: 0.5529 | Acc: 81.11% | Val Loss: 0.1727 | Val Acc: 93.14%
--> Model Saved
Epoch [18/20] Loss: 0.5439 | Acc: 81.40% | Val Loss: 0.7466 | Val Acc: 76.67%
Epoch [19/20] Loss: 0.5068 | Acc: 82.91% | Val Loss: 0.1425 | Val Acc: 92.75%
--> Model Saved
Epoch [20/20] Loss: 0.5206 | Acc: 82.31% | Val Loss: 0.1639 | Val Acc: 92.55%

Total training time: 2m 51s
```

VGG-16 (trained from scratch) — Epoch Log

```
Epoch [1/20] Loss: 2.7146 | Acc: 9.26% | Val Loss: 2.3084 | Val Acc: 10.00%
--> Model Saved
Epoch [2/20] Loss: 2.3908 | Acc: 10.51% | Val Loss: 2.3197 | Val Acc: 10.00%
Epoch [3/20] Loss: 2.3474 | Acc: 10.40% | Val Loss: 2.3015 | Val Acc: 17.06%
--> Model Saved
Epoch [4/20] Loss: 2.3267 | Acc: 10.63% | Val Loss: 2.2890 | Val Acc: 10.20%
--> Model Saved
Epoch [5/20] Loss: 2.2418 | Acc: 15.60% | Val Loss: 2.2609 | Val Acc: 13.73%
--> Model Saved
Epoch [6/20] Loss: 2.1403 | Acc: 18.57% | Val Loss: 2.0681 | Val Acc: 16.67%
--> Model Saved
Epoch [7/20] Loss: 2.0787 | Acc: 21.69% | Val Loss: 1.9376 | Val Acc: 24.31%
--> Model Saved
Epoch [8/20] Loss: 1.9956 | Acc: 24.34% | Val Loss: 1.7635 | Val Acc: 30.00%
--> Model Saved
Epoch [9/20] Loss: 1.8204 | Acc: 32.46% | Val Loss: 1.2414 | Val Acc: 56.08%
--> Model Saved
Epoch [10/20] Loss: 1.5994 | Acc: 42.54% | Val Loss: 1.3705 | Val Acc: 45.88%
Epoch [11/20] Loss: 1.4012 | Acc: 48.66% | Val Loss: 1.0756 | Val Acc: 66.08%
--> Model Saved
Epoch [12/20] Loss: 1.2910 | Acc: 54.54% | Val Loss: 0.6930 | Val Acc: 73.92%
--> Model Saved
Epoch [13/20] Loss: 1.1823 | Acc: 58.49% | Val Loss: 0.4379 | Val Acc: 84.51%
--> Model Saved
Epoch [14/20] Loss: 1.0903 | Acc: 60.94% | Val Loss: 1.2026 | Val Acc: 60.59%
Epoch [15/20] Loss: 0.9698 | Acc: 65.23% | Val Loss: 0.4938 | Val Acc: 80.98%
Epoch [16/20] Loss: 0.9833 | Acc: 65.57% | Val Loss: 0.3078 | Val Acc: 90.00%
--> Model Saved
Epoch [17/20] Loss: 0.8780 | Acc: 69.54% | Val Loss: 0.3063 | Val Acc: 90.00%
```

```
--> Model Saved
Epoch [18/20] Loss: 0.8510 | Acc: 71.09% | Val Loss: 0.5055 | Val Acc: 82.55%
Epoch [19/20] Loss: 0.8018 | Acc: 71.14% | Val Loss: 0.3700 | Val Acc: 86.47%
Epoch [20/20] Loss: 0.7274 | Acc: 74.14% | Val Loss: 1.5396 | Val Acc: 62.75%

Total training time: 4m 50s
```

VGG-16-TL (transfer learning) — Epoch Log

```
Epoch [1/20] Loss: 1.6869 | Acc: 40.20% | Val Loss: 0.8207 | Val Acc: 74.51%
--> Model Saved
Epoch [2/20] Loss: 1.1493 | Acc: 60.14% | Val Loss: 0.6978 | Val Acc: 68.63%
--> Model Saved
Epoch [3/20] Loss: 0.9855 | Acc: 65.34% | Val Loss: 0.3566 | Val Acc: 88.04%
--> Model Saved
Epoch [4/20] Loss: 0.8804 | Acc: 68.57% | Val Loss: 0.3062 | Val Acc: 89.80%
--> Model Saved
Epoch [5/20] Loss: 0.8263 | Acc: 71.66% | Val Loss: 0.2990 | Val Acc: 87.84%
--> Model Saved
Epoch [6/20] Loss: 0.7662 | Acc: 73.06% | Val Loss: 0.2475 | Val Acc: 90.39%
--> Model Saved
Epoch [7/20] Loss: 0.7773 | Acc: 72.29% | Val Loss: 0.2607 | Val Acc: 89.61%
Epoch [8/20] Loss: 0.7398 | Acc: 73.49% | Val Loss: 0.3120 | Val Acc: 87.25%
Epoch [9/20] Loss: 0.6926 | Acc: 75.37% | Val Loss: 0.1870 | Val Acc: 93.14%
--> Model Saved
Epoch [10/20] Loss: 0.7006 | Acc: 75.46% | Val Loss: 0.2090 | Val Acc: 91.76%
Epoch [11/20] Loss: 0.6733 | Acc: 75.83% | Val Loss: 0.1262 | Val Acc: 96.08%
--> Model Saved
Epoch [12/20] Loss: 0.6608 | Acc: 76.74% | Val Loss: 0.1723 | Val Acc: 93.92%
Epoch [13/20] Loss: 0.6398 | Acc: 77.43% | Val Loss: 0.1250 | Val Acc: 95.49%
--> Model Saved
Epoch [14/20] Loss: 0.6599 | Acc: 76.20% | Val Loss: 0.2159 | Val Acc: 92.16%
Epoch [15/20] Loss: 0.6374 | Acc: 77.91% | Val Loss: 0.1093 | Val Acc: 96.47%
--> Model Saved
Epoch [16/20] Loss: 0.6268 | Acc: 77.43% | Val Loss: 0.1583 | Val Acc: 93.73%
Epoch [17/20] Loss: 0.6358 | Acc: 77.86% | Val Loss: 0.1094 | Val Acc: 95.49%
Epoch [18/20] Loss: 0.6004 | Acc: 79.06% | Val Loss: 0.0892 | Val Acc: 97.25%
--> Model Saved
Epoch [19/20] Loss: 0.5892 | Acc: 78.77% | Val Loss: 0.1101 | Val Acc: 96.67%
Epoch [20/20] Loss: 0.5812 | Acc: 79.29% | Val Loss: 0.1108 | Val Acc: 95.49%

Total training time: 2m 57s
```

E.2. HaGRID

ShuffleNetV2 (trained from scratch) — Epoch Log

```
Epoch [1/35] Loss: 0.6942 | Acc: 55.51% | Val Loss: 0.5968 | Val Acc: 68.43%
--> Model Saved
Epoch [2/35] Loss: 0.6107 | Acc: 66.54% | Val Loss: 0.4957 | Val Acc: 76.96%
--> Model Saved
Epoch [3/35] Loss: 0.5181 | Acc: 73.39% | Val Loss: 0.3352 | Val Acc: 84.41%
--> Model Saved
Epoch [4/35] Loss: 0.4263 | Acc: 79.29% | Val Loss: 0.2414 | Val Acc: 89.51%
```

```
--> Model Saved
Epoch [5/35] Loss: 0.3750 | Acc: 82.17% | Val Loss: 0.1950 | Val Acc: 90.88%
--> Model Saved
Epoch [6/35] Loss: 0.3487 | Acc: 83.37% | Val Loss: 0.2433 | Val Acc: 89.90%
Epoch [7/35] Loss: 0.3270 | Acc: 84.37% | Val Loss: 0.1608 | Val Acc: 93.63%
--> Model Saved
Epoch [8/35] Loss: 0.2937 | Acc: 86.26% | Val Loss: 0.1321 | Val Acc: 94.80%
--> Model Saved
Epoch [9/35] Loss: 0.2777 | Acc: 86.99% | Val Loss: 0.1245 | Val Acc: 94.61%
--> Model Saved
Epoch [10/35] Loss: 0.2851 | Acc: 86.31% | Val Loss: 0.1379 | Val Acc: 94.12%
Epoch [11/35] Loss: 0.2666 | Acc: 87.76% | Val Loss: 0.1169 | Val Acc: 94.51%
--> Model Saved
Epoch [12/35] Loss: 0.2492 | Acc: 88.51% | Val Loss: 0.1150 | Val Acc: 95.39%
--> Model Saved
Epoch [13/35] Loss: 0.2527 | Acc: 88.29% | Val Loss: 0.0997 | Val Acc: 96.27%
--> Model Saved
Epoch [14/35] Loss: 0.2509 | Acc: 88.10% | Val Loss: 0.0979 | Val Acc: 95.88%
--> Model Saved
Epoch [15/35] Loss: 0.2286 | Acc: 89.63% | Val Loss: 0.0988 | Val Acc: 95.78%
Epoch [16/35] Loss: 0.2293 | Acc: 89.21% | Val Loss: 0.0895 | Val Acc: 95.98%
--> Model Saved
Epoch [17/35] Loss: 0.2255 | Acc: 89.49% | Val Loss: 0.0838 | Val Acc: 96.57%
--> Model Saved
Epoch [18/35] Loss: 0.2126 | Acc: 90.73% | Val Loss: 0.0861 | Val Acc: 96.27%
Epoch [19/35] Loss: 0.2126 | Acc: 90.24% | Val Loss: 0.1206 | Val Acc: 95.20%
Epoch [20/35] Loss: 0.2114 | Acc: 90.44% | Val Loss: 0.0963 | Val Acc: 96.27%
Epoch [21/35] Loss: 0.2050 | Acc: 90.73% | Val Loss: 0.0733 | Val Acc: 97.16%
--> Model Saved
Epoch [22/35] Loss: 0.2057 | Acc: 90.96% | Val Loss: 0.0632 | Val Acc: 97.55%
--> Model Saved
Epoch [23/35] Loss: 0.2010 | Acc: 90.73% | Val Loss: 0.0769 | Val Acc: 96.96%
Epoch [24/35] Loss: 0.1896 | Acc: 91.56% | Val Loss: 0.0600 | Val Acc: 97.75%
--> Model Saved
Epoch [25/35] Loss: 0.1921 | Acc: 91.24% | Val Loss: 0.0897 | Val Acc: 95.98%
Epoch [26/35] Loss: 0.1980 | Acc: 91.07% | Val Loss: 0.0969 | Val Acc: 95.88%
Epoch [27/35] Loss: 0.1798 | Acc: 91.47% | Val Loss: 0.0645 | Val Acc: 97.35%
Epoch [28/35] Loss: 0.1759 | Acc: 91.86% | Val Loss: 0.0716 | Val Acc: 97.35%
Epoch [29/35] Loss: 0.1718 | Acc: 92.49% | Val Loss: 0.0741 | Val Acc: 97.55%
Epoch [30/35] Loss: 0.1700 | Acc: 92.39% | Val Loss: 0.0641 | Val Acc: 97.25%
Epoch [31/35] Loss: 0.1697 | Acc: 92.07% | Val Loss: 0.0551 | Val Acc: 98.24%
--> Model Saved
Epoch [32/35] Loss: 0.1673 | Acc: 92.36% | Val Loss: 0.0626 | Val Acc: 97.65%
Epoch [33/35] Loss: 0.1723 | Acc: 92.11% | Val Loss: 0.0602 | Val Acc: 97.45%
Epoch [34/35] Loss: 0.1602 | Acc: 92.11% | Val Loss: 0.0660 | Val Acc: 97.75%
Epoch [35/35] Loss: 0.1633 | Acc: 92.03% | Val Loss: 0.0545 | Val Acc: 98.14%
--> Model Saved

Total training time: 6m 13s
```

ResNet-34 (trained from scratch) — Epoch Log

```
Epoch [1/35] Loss: 0.7499 | Acc: 51.90% | Val Loss: 0.6767 | Val Acc: 57.65%
```

--> Model Saved
Epoch [2/35] Loss: 0.6970 | Acc: 54.91% | Val Loss: 0.7151 | Val Acc: 58.73%
Epoch [3/35] Loss: 0.6854 | Acc: 55.73% | Val Loss: 0.7649 | Val Acc: 50.00%
Epoch [4/35] Loss: 0.6844 | Acc: 56.90% | Val Loss: 0.9739 | Val Acc: 52.06%
Epoch [5/35] Loss: 0.6382 | Acc: 63.44% | Val Loss: 0.5190 | Val Acc: 75.29%
--> Model Saved
Epoch [6/35] Loss: 0.6105 | Acc: 67.33% | Val Loss: 0.4747 | Val Acc: 77.94%
--> Model Saved
Epoch [7/35] Loss: 0.5838 | Acc: 69.91% | Val Loss: 0.4678 | Val Acc: 78.43%
--> Model Saved
Epoch [8/35] Loss: 0.5699 | Acc: 70.66% | Val Loss: 0.4612 | Val Acc: 79.71%
--> Model Saved
Epoch [9/35] Loss: 0.5331 | Acc: 72.89% | Val Loss: 0.3975 | Val Acc: 83.04%
--> Model Saved
Epoch [10/35] Loss: 0.5106 | Acc: 74.79% | Val Loss: 0.3505 | Val Acc: 86.96%
--> Model Saved
Epoch [11/35] Loss: 0.4759 | Acc: 75.93% | Val Loss: 0.2998 | Val Acc: 88.04%
--> Model Saved
Epoch [12/35] Loss: 0.4121 | Acc: 80.10% | Val Loss: 0.3306 | Val Acc: 84.80%
Epoch [13/35] Loss: 0.3864 | Acc: 81.30% | Val Loss: 0.2551 | Val Acc: 88.92%
--> Model Saved
Epoch [14/35] Loss: 0.3570 | Acc: 83.20% | Val Loss: 0.2381 | Val Acc: 89.41%
--> Model Saved
Epoch [15/35] Loss: 0.3395 | Acc: 83.59% | Val Loss: 0.1974 | Val Acc: 92.16%
--> Model Saved
Epoch [16/35] Loss: 0.3198 | Acc: 84.64% | Val Loss: 0.1596 | Val Acc: 93.82%
--> Model Saved
Epoch [17/35] Loss: 0.2955 | Acc: 86.56% | Val Loss: 0.1778 | Val Acc: 92.35%
Epoch [18/35] Loss: 0.2762 | Acc: 87.17% | Val Loss: 0.1332 | Val Acc: 94.41%
--> Model Saved
Epoch [19/35] Loss: 0.2636 | Acc: 87.96% | Val Loss: 0.1256 | Val Acc: 95.00%
--> Model Saved
Epoch [20/35] Loss: 0.2465 | Acc: 88.53% | Val Loss: 0.1654 | Val Acc: 92.75%
Epoch [21/35] Loss: 0.2559 | Acc: 88.13% | Val Loss: 0.1167 | Val Acc: 95.00%
--> Model Saved
Epoch [22/35] Loss: 0.2331 | Acc: 89.37% | Val Loss: 0.1305 | Val Acc: 95.49%
Epoch [23/35] Loss: 0.2269 | Acc: 89.44% | Val Loss: 0.0937 | Val Acc: 96.08%
--> Model Saved
Epoch [24/35] Loss: 0.2276 | Acc: 89.44% | Val Loss: 0.0968 | Val Acc: 95.98%
Epoch [25/35] Loss: 0.2083 | Acc: 90.27% | Val Loss: 0.0741 | Val Acc: 97.35%
--> Model Saved
Epoch [26/35] Loss: 0.2073 | Acc: 90.77% | Val Loss: 0.1132 | Val Acc: 95.49%
Epoch [27/35] Loss: 0.1976 | Acc: 90.97% | Val Loss: 0.0999 | Val Acc: 95.98%
Epoch [28/35] Loss: 0.1971 | Acc: 91.27% | Val Loss: 0.0706 | Val Acc: 97.25%
--> Model Saved
Epoch [29/35] Loss: 0.1857 | Acc: 91.39% | Val Loss: 0.0545 | Val Acc: 97.35%
--> Model Saved
Epoch [30/35] Loss: 0.1859 | Acc: 91.81% | Val Loss: 0.0859 | Val Acc: 96.08%
Epoch [31/35] Loss: 0.1830 | Acc: 91.39% | Val Loss: 0.0939 | Val Acc: 95.88%
Epoch [32/35] Loss: 0.1750 | Acc: 92.03% | Val Loss: 0.0654 | Val Acc: 96.86%
Epoch [33/35] Loss: 0.1750 | Acc: 92.01% | Val Loss: 0.0757 | Val Acc: 96.96%
Epoch [34/35] Loss: 0.1614 | Acc: 92.84% | Val Loss: 0.0565 | Val Acc: 98.14%
Epoch [35/35] Loss: 0.1630 | Acc: 92.43% | Val Loss: 0.0467 | Val Acc: 98.24%

```
--> Model Saved

Total training time: 5m 58s
```

VGG-16 (trained from scratch) — Epoch Log

```
Epoch [1/35] Loss: 0.9484 | Acc: 51.34% | Val Loss: 0.7821 | Val Acc: 58.92%
--> Model Saved
Epoch [2/35] Loss: 0.7014 | Acc: 61.57% | Val Loss: 0.5002 | Val Acc: 77.45%
--> Model Saved
Epoch [3/35] Loss: 0.5923 | Acc: 69.71% | Val Loss: 0.3911 | Val Acc: 82.94%
--> Model Saved
Epoch [4/35] Loss: 0.5089 | Acc: 75.11% | Val Loss: 0.2923 | Val Acc: 85.69%
--> Model Saved
Epoch [5/35] Loss: 0.4370 | Acc: 78.59% | Val Loss: 0.3015 | Val Acc: 87.65%
Epoch [6/35] Loss: 0.3901 | Acc: 81.46% | Val Loss: 0.2627 | Val Acc: 88.92%
--> Model Saved
Epoch [7/35] Loss: 0.3382 | Acc: 83.94% | Val Loss: 0.2133 | Val Acc: 90.39%
--> Model Saved
Epoch [8/35] Loss: 0.3007 | Acc: 86.10% | Val Loss: 0.1244 | Val Acc: 95.39%
--> Model Saved
Epoch [9/35] Loss: 0.2969 | Acc: 85.63% | Val Loss: 0.1682 | Val Acc: 92.65%
Epoch [10/35] Loss: 0.2648 | Acc: 87.59% | Val Loss: 0.1198 | Val Acc: 94.90%
--> Model Saved
Epoch [11/35] Loss: 0.2426 | Acc: 88.70% | Val Loss: 0.0855 | Val Acc: 96.86%
--> Model Saved
Epoch [12/35] Loss: 0.2369 | Acc: 89.24% | Val Loss: 0.2117 | Val Acc: 90.49%
Epoch [13/35] Loss: 0.2111 | Acc: 90.30% | Val Loss: 0.1037 | Val Acc: 95.88%
Epoch [14/35] Loss: 0.2104 | Acc: 90.51% | Val Loss: 0.1065 | Val Acc: 95.20%
Epoch [15/35] Loss: 0.1986 | Acc: 90.99% | Val Loss: 0.0634 | Val Acc: 97.65%
--> Model Saved
Epoch [16/35] Loss: 0.1823 | Acc: 92.04% | Val Loss: 0.0890 | Val Acc: 96.96%
Epoch [17/35] Loss: 0.1900 | Acc: 91.44% | Val Loss: 0.0755 | Val Acc: 97.55%
Epoch [18/35] Loss: 0.1756 | Acc: 91.97% | Val Loss: 0.0687 | Val Acc: 97.55%
Epoch [19/35] Loss: 0.1681 | Acc: 92.10% | Val Loss: 0.0754 | Val Acc: 97.45%
Epoch [20/35] Loss: 0.1741 | Acc: 91.79% | Val Loss: 0.0482 | Val Acc: 98.53%
--> Model Saved
Epoch [21/35] Loss: 0.1668 | Acc: 92.61% | Val Loss: 0.0722 | Val Acc: 97.35%
Epoch [22/35] Loss: 0.1633 | Acc: 92.49% | Val Loss: 0.0503 | Val Acc: 98.04%
Epoch [23/35] Loss: 0.1474 | Acc: 93.39% | Val Loss: 0.0591 | Val Acc: 97.55%
Epoch [24/35] Loss: 0.1508 | Acc: 93.30% | Val Loss: 0.0483 | Val Acc: 98.63%
Epoch [25/35] Loss: 0.1527 | Acc: 93.17% | Val Loss: 0.0815 | Val Acc: 97.65%
Epoch [26/35] Loss: 0.1363 | Acc: 93.71% | Val Loss: 0.0530 | Val Acc: 98.14%
Epoch [27/35] Loss: 0.1453 | Acc: 93.51% | Val Loss: 0.0666 | Val Acc: 97.75%
Epoch [28/35] Loss: 0.1480 | Acc: 93.20% | Val Loss: 0.0544 | Val Acc: 97.94%
Epoch [29/35] Loss: 0.1420 | Acc: 94.01% | Val Loss: 0.0316 | Val Acc: 99.02%
--> Model Saved
Epoch [30/35] Loss: 0.1326 | Acc: 94.04% | Val Loss: 0.0435 | Val Acc: 98.33%
Epoch [31/35] Loss: 0.1366 | Acc: 93.64% | Val Loss: 0.0573 | Val Acc: 98.04%
Epoch [32/35] Loss: 0.1349 | Acc: 93.54% | Val Loss: 0.0298 | Val Acc: 99.22%
--> Model Saved
Epoch [33/35] Loss: 0.1258 | Acc: 94.46% | Val Loss: 0.0540 | Val Acc: 98.14%
Epoch [34/35] Loss: 0.1328 | Acc: 93.84% | Val Loss: 0.0340 | Val Acc: 98.92%
```

```
Epoch [35/35] Loss: 0.1243 | Acc: 94.13% | Val Loss: 0.0294 | Val Acc: 98.92%
--> Model Saved

Total training time: 15m 39s
```

E.3. SignAlphaSet

ShuffleNetV2 (trained from scratch) — Epoch Log

```
Epoch [1/15] Loss: 3.1179 | Acc: 8.14% | Val Loss: 3.6103 | Val Acc: 4.52%
--> Model Saved
Epoch [2/15] Loss: 2.4876 | Acc: 23.14% | Val Loss: 2.5300 | Val Acc: 24.06%
--> Model Saved
Epoch [3/15] Loss: 1.8594 | Acc: 41.51% | Val Loss: 2.0406 | Val Acc: 34.43%
--> Model Saved
Epoch [4/15] Loss: 1.4096 | Acc: 56.22% | Val Loss: 2.1774 | Val Acc: 32.24%
Epoch [5/15] Loss: 1.1101 | Acc: 65.84% | Val Loss: 1.1933 | Val Acc: 60.75%
--> Model Saved
Epoch [6/15] Loss: 0.9061 | Acc: 72.37% | Val Loss: 1.5892 | Val Acc: 49.21%
Epoch [7/15] Loss: 0.7658 | Acc: 77.07% | Val Loss: 1.5117 | Val Acc: 53.24%
Epoch [8/15] Loss: 0.6650 | Acc: 79.85% | Val Loss: 1.2846 | Val Acc: 58.67%
Epoch [9/15] Loss: 0.5776 | Acc: 82.67% | Val Loss: 1.0893 | Val Acc: 64.14%
--> Model Saved
Epoch [10/15] Loss: 0.5039 | Acc: 84.87% | Val Loss: 0.9829 | Val Acc: 66.55%
--> Model Saved
Epoch [11/15] Loss: 0.4618 | Acc: 86.09% | Val Loss: 1.0096 | Val Acc: 65.61%
Epoch [12/15] Loss: 0.4076 | Acc: 87.63% | Val Loss: 0.6401 | Val Acc: 81.22%
--> Model Saved
Epoch [13/15] Loss: 0.3872 | Acc: 88.36% | Val Loss: 0.1658 | Val Acc: 96.95%
--> Model Saved
Epoch [14/15] Loss: 0.3728 | Acc: 88.87% | Val Loss: 1.0131 | Val Acc: 64.56%
Epoch [15/15] Loss: 0.3339 | Acc: 89.69% | Val Loss: 0.1543 | Val Acc: 96.23%
--> Model Saved

Total training time: 8m 30s
```

ResNet-34 (trained from scratch) — Epoch Log

```
Epoch [1/15] Loss: 2.2343 | Acc: 32.86% | Val Loss: 1.1275 | Val Acc: 63.24%
--> Model Saved
Epoch [2/15] Loss: 0.8229 | Acc: 75.69% | Val Loss: 0.3086 | Val Acc: 90.16%
--> Model Saved
Epoch [3/15] Loss: 0.5482 | Acc: 83.72% | Val Loss: 0.2854 | Val Acc: 90.57%
--> Model Saved
Epoch [4/15] Loss: 0.4420 | Acc: 86.73% | Val Loss: 0.2673 | Val Acc: 91.03%
--> Model Saved
Epoch [5/15] Loss: 0.3971 | Acc: 87.92% | Val Loss: 0.0901 | Val Acc: 96.23%
--> Model Saved
Epoch [6/15] Loss: 0.3504 | Acc: 89.47% | Val Loss: 0.0652 | Val Acc: 97.36%
--> Model Saved
Epoch [7/15] Loss: 0.3326 | Acc: 89.88% | Val Loss: 0.0555 | Val Acc: 98.45%
--> Model Saved
Epoch [8/15] Loss: 0.2935 | Acc: 91.12% | Val Loss: 0.0460 | Val Acc: 98.76%
--> Model Saved
```

```
Epoch [9/15] Loss: 0.2943 | Acc: 91.07% | Val Loss: 0.0027 | Val Acc: 99.96%
--> Model Saved
Epoch [10/15] Loss: 0.2648 | Acc: 91.89% | Val Loss: 0.0116 | Val Acc: 100.00%
Epoch [11/15] Loss: 0.2574 | Acc: 92.23% | Val Loss: 0.0314 | Val Acc: 99.06%
Epoch [12/15] Loss: 0.2343 | Acc: 92.95% | Val Loss: 0.0129 | Val Acc: 99.89%
Epoch [13/15] Loss: 0.2372 | Acc: 92.66% | Val Loss: 0.0528 | Val Acc: 97.17%
Epoch [14/15] Loss: 0.2319 | Acc: 92.90% | Val Loss: 0.3244 | Val Acc: 90.57%
Epoch [15/15] Loss: 0.2118 | Acc: 93.42% | Val Loss: 0.0523 | Val Acc: 98.64%

Total training time: 8m 23s
```

VGG-16 (trained from scratch) — Epoch Log

```
Epoch [1/15] Loss: 3.2834 | Acc: 5.89% | Val Loss: 2.5356 | Val Acc: 20.25%
--> Model Saved
Epoch [2/15] Loss: 2.2380 | Acc: 27.20% | Val Loss: 0.6659 | Val Acc: 74.85%
--> Model Saved
Epoch [3/15] Loss: 1.4994 | Acc: 50.61% | Val Loss: 0.3390 | Val Acc: 88.20%
--> Model Saved
Epoch [4/15] Loss: 1.0873 | Acc: 64.94% | Val Loss: 0.1811 | Val Acc: 91.86%
--> Model Saved
Epoch [5/15] Loss: 0.8612 | Acc: 72.75% | Val Loss: 0.0257 | Val Acc: 99.85%
--> Model Saved
Epoch [6/15] Loss: 0.6990 | Acc: 78.24% | Val Loss: 0.0086 | Val Acc: 99.92%
--> Model Saved
Epoch [7/15] Loss: 0.6218 | Acc: 80.57% | Val Loss: 0.0154 | Val Acc: 99.85%
Epoch [8/15] Loss: 0.5653 | Acc: 82.57% | Val Loss: 0.0147 | Val Acc: 99.59%
Epoch [9/15] Loss: 0.5028 | Acc: 84.25% | Val Loss: 0.0697 | Val Acc: 97.62%
Epoch [10/15] Loss: 0.4637 | Acc: 85.51% | Val Loss: 0.0305 | Val Acc: 98.72%
Epoch [11/15] Loss: 0.4323 | Acc: 86.57% | Val Loss: 0.0270 | Val Acc: 99.13%
Epoch [12/15] Loss: 0.4036 | Acc: 87.37% | Val Loss: 0.0026 | Val Acc: 99.92%
--> Model Saved
Epoch [13/15] Loss: 0.3866 | Acc: 88.07% | Val Loss: 0.0064 | Val Acc: 99.89%
Epoch [14/15] Loss: 0.3550 | Acc: 88.86% | Val Loss: 0.0252 | Val Acc: 99.17%
Epoch [15/15] Loss: 0.3454 | Acc: 89.37% | Val Loss: 0.0160 | Val Acc: 99.17%

Total training time: 16m 31s
```

VGG-16-TL (transfer learning) — Epoch Log

```
Epoch [1/15] Loss: 1.5521 | Acc: 53.99% | Val Loss: 0.3571 | Val Acc: 88.57%
--> Model Saved
Epoch [2/15] Loss: 0.8877 | Acc: 73.37% | Val Loss: 0.1522 | Val Acc: 96.34%
--> Model Saved
Epoch [3/15] Loss: 0.7505 | Acc: 77.10% | Val Loss: 0.0970 | Val Acc: 96.98%
--> Model Saved
Epoch [4/15] Loss: 0.6833 | Acc: 79.24% | Val Loss: 0.0575 | Val Acc: 98.45%
--> Model Saved
Epoch [5/15] Loss: 0.6391 | Acc: 80.52% | Val Loss: 0.0757 | Val Acc: 97.66%
Epoch [6/15] Loss: 0.5969 | Acc: 81.43% | Val Loss: 0.0616 | Val Acc: 98.08%
Epoch [7/15] Loss: 0.5881 | Acc: 81.90% | Val Loss: 0.0411 | Val Acc: 98.87%
--> Model Saved
```

```
Epoch [8/15] Loss: 0.5642 | Acc: 82.63% | Val Loss: 0.0372 | Val Acc: 98.94%
--> Model Saved
Epoch [9/15] Loss: 0.5446 | Acc: 82.85% | Val Loss: 0.0316 | Val Acc: 99.02%
--> Model Saved
Epoch [10/15] Loss: 0.5333 | Acc: 83.55% | Val Loss: 0.0517 | Val Acc: 98.08%
Epoch [11/15] Loss: 0.5168 | Acc: 84.05% | Val Loss: 0.0370 | Val Acc: 99.17%
Epoch [12/15] Loss: 0.5028 | Acc: 84.52% | Val Loss: 0.0416 | Val Acc: 98.45%
Epoch [13/15] Loss: 0.4900 | Acc: 84.59% | Val Loss: 0.0498 | Val Acc: 98.08%
Epoch [14/15] Loss: 0.4760 | Acc: 84.86% | Val Loss: 0.0217 | Val Acc: 99.21%
--> Model Saved
Epoch [15/15] Loss: 0.4778 | Acc: 84.98% | Val Loss: 0.0432 | Val Acc: 98.53%

Total training time: 8m 51s
```

F. Ablation Study

F.1. Optimizer Comparison

ResNet-34 using Adam vs AdamW on HaGRID

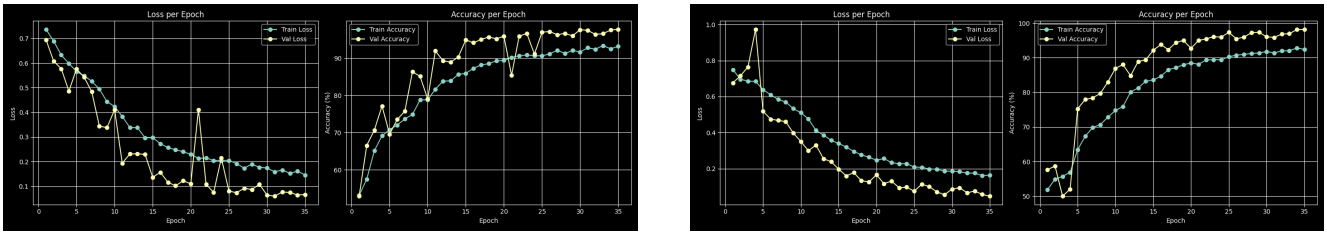


Figure 12. Training curves on HaGRID: ResNet-34 using Adam vs AdamW (left to right).

F.2. Learning Rate Comparison

VGG-16 using 1×10^{-3} vs 1×10^{-4} on SignAlphaSet
Epoch logs:
VGG-16 (higher LR — unstable run)

```
Epoch [1/15] Loss: 3.6229 | Acc: 4.23% | Val Loss: 3.2600 | Val Acc: 3.85%
--> Model Saved
Epoch [2/15] Loss: 3.2605 | Acc: 3.67% | Val Loss: 3.2583 | Val Acc: 3.85%
--> Model Saved
Epoch [3/15] Loss: 3.2590 | Acc: 3.66% | Val Loss: 3.2582 | Val Acc: 3.85%
--> Model Saved
Epoch [4/15] Loss: 3.2589 | Acc: 3.73% | Val Loss: 3.2582 | Val Acc: 3.85%
--> Model Saved
Epoch [5/15] Loss: 3.2587 | Acc: 3.68% | Val Loss: 3.2581 | Val Acc: 3.85%
--> Model Saved
```

VGG-16 (lower LR — stable run)

```
Epoch [1/15] Loss: 3.2834 | Acc: 5.89% | Val Loss: 2.5356 | Val Acc: 20.25%
--> Model Saved
Epoch [2/15] Loss: 2.2380 | Acc: 27.20% | Val Loss: 0.6659 | Val Acc: 74.85%
--> Model Saved
Epoch [3/15] Loss: 1.4994 | Acc: 50.61% | Val Loss: 0.3390 | Val Acc: 88.20%
--> Model Saved
Epoch [4/15] Loss: 1.0873 | Acc: 64.94% | Val Loss: 0.1811 | Val Acc: 91.86%
--> Model Saved
Epoch [5/15] Loss: 0.8612 | Acc: 72.75% | Val Loss: 0.0257 | Val Acc: 99.85%
--> Model Saved
Epoch [6/15] Loss: 0.6990 | Acc: 78.24% | Val Loss: 0.0086 | Val Acc: 99.92%
--> Model Saved
Epoch [7/15] Loss: 0.6218 | Acc: 80.57% | Val Loss: 0.0154 | Val Acc: 99.85%
Epoch [8/15] Loss: 0.5653 | Acc: 82.57% | Val Loss: 0.0147 | Val Acc: 99.59%
Epoch [9/15] Loss: 0.5028 | Acc: 84.25% | Val Loss: 0.0697 | Val Acc: 97.62%
Epoch [10/15] Loss: 0.4637 | Acc: 85.51% | Val Loss: 0.0305 | Val Acc: 98.72%
Epoch [11/15] Loss: 0.4323 | Acc: 86.57% | Val Loss: 0.0270 | Val Acc: 99.13%
Epoch [12/15] Loss: 0.4036 | Acc: 87.37% | Val Loss: 0.0026 | Val Acc: 99.92%
--> Model Saved
```

```
Epoch [13/15] Loss: 0.3866 | Acc: 88.07% | Val Loss: 0.0064 | Val Acc: 99.89%
Epoch [14/15] Loss: 0.3550 | Acc: 88.86% | Val Loss: 0.0252 | Val Acc: 99.17%
Epoch [15/15] Loss: 0.3454 | Acc: 89.37% | Val Loss: 0.0160 | Val Acc: 99.17%

Total training time: 16m 31s
```

F.3. Batch Size Comparison

ResNet-34 using batch size 16 vs 32 on HaGRID

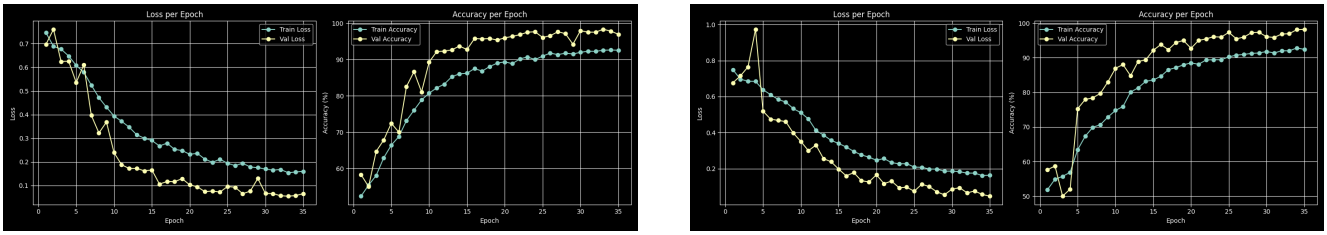


Figure 13. Training curves on HaGRID: ResNet-34 using batch size 16 vs 32 (left to right).

Epoch logs: ResNet-34 (batch size 16)

```
Epoch [1/35] Loss: 0.7461 | Acc: 52.37% | Val Loss: 0.6967 | Val Acc: 58.33%
--> Model Saved
Epoch [2/35] Loss: 0.6888 | Acc: 55.41% | Val Loss: 0.7606 | Val Acc: 55.00%
Epoch [3/35] Loss: 0.6769 | Acc: 58.07% | Val Loss: 0.6248 | Val Acc: 64.71%
--> Model Saved
Epoch [4/35] Loss: 0.6478 | Acc: 62.94% | Val Loss: 0.6257 | Val Acc: 67.84%
Epoch [5/35] Loss: 0.6080 | Acc: 66.43% | Val Loss: 0.5360 | Val Acc: 72.45%
--> Model Saved
Epoch [6/35] Loss: 0.5790 | Acc: 68.79% | Val Loss: 0.6100 | Val Acc: 70.00%
Epoch [7/35] Loss: 0.5245 | Acc: 73.17% | Val Loss: 0.3968 | Val Acc: 82.55%
--> Model Saved
Epoch [8/35] Loss: 0.4717 | Acc: 76.04% | Val Loss: 0.3224 | Val Acc: 86.67%
--> Model Saved
Epoch [9/35] Loss: 0.4312 | Acc: 78.87% | Val Loss: 0.3685 | Val Acc: 81.08%
Epoch [10/35] Loss: 0.3940 | Acc: 80.74% | Val Loss: 0.2391 | Val Acc: 89.31%
--> Model Saved
Epoch [11/35] Loss: 0.3718 | Acc: 82.20% | Val Loss: 0.1881 | Val Acc: 92.16%
--> Model Saved
Epoch [12/35] Loss: 0.3469 | Acc: 83.23% | Val Loss: 0.1723 | Val Acc: 92.35%
--> Model Saved
Epoch [13/35] Loss: 0.3147 | Acc: 85.27% | Val Loss: 0.1731 | Val Acc: 92.65%
Epoch [14/35] Loss: 0.3003 | Acc: 86.06% | Val Loss: 0.1621 | Val Acc: 93.73%
--> Model Saved
Epoch [15/35] Loss: 0.2909 | Acc: 86.24% | Val Loss: 0.1662 | Val Acc: 92.75%
Epoch [16/35] Loss: 0.2675 | Acc: 87.53% | Val Loss: 0.1064 | Val Acc: 95.78%
--> Model Saved
Epoch [17/35] Loss: 0.2793 | Acc: 86.81% | Val Loss: 0.1174 | Val Acc: 95.69%
Epoch [18/35] Loss: 0.2530 | Acc: 88.09% | Val Loss: 0.1165 | Val Acc: 95.78%
Epoch [19/35] Loss: 0.2482 | Acc: 89.01% | Val Loss: 0.1295 | Val Acc: 95.39%
Epoch [20/35] Loss: 0.2321 | Acc: 89.36% | Val Loss: 0.1029 | Val Acc: 95.98%
--> Model Saved
Epoch [21/35] Loss: 0.2364 | Acc: 88.87% | Val Loss: 0.0951 | Val Acc: 96.37%
--> Model Saved
```

```
Epoch [22/35] Loss: 0.2118 | Acc: 90.20% | Val Loss: 0.0743 | Val Acc: 96.86%
--> Model Saved
Epoch [23/35] Loss: 0.1985 | Acc: 90.70% | Val Loss: 0.0771 | Val Acc: 97.55%
Epoch [24/35] Loss: 0.2115 | Acc: 90.00% | Val Loss: 0.0727 | Val Acc: 97.65%
--> Model Saved
Epoch [25/35] Loss: 0.1940 | Acc: 90.94% | Val Loss: 0.0962 | Val Acc: 96.08%
Epoch [26/35] Loss: 0.1847 | Acc: 91.79% | Val Loss: 0.0930 | Val Acc: 96.57%
Epoch [27/35] Loss: 0.1945 | Acc: 91.36% | Val Loss: 0.0656 | Val Acc: 97.65%
--> Model Saved
Epoch [28/35] Loss: 0.1786 | Acc: 91.76% | Val Loss: 0.0767 | Val Acc: 97.16%
Epoch [29/35] Loss: 0.1759 | Acc: 91.54% | Val Loss: 0.1302 | Val Acc: 94.22%
Epoch [30/35] Loss: 0.1702 | Acc: 91.99% | Val Loss: 0.0667 | Val Acc: 97.94%
Epoch [31/35] Loss: 0.1653 | Acc: 92.30% | Val Loss: 0.0662 | Val Acc: 97.55%
Epoch [32/35] Loss: 0.1680 | Acc: 92.26% | Val Loss: 0.0572 | Val Acc: 97.55%
--> Model Saved
Epoch [33/35] Loss: 0.1545 | Acc: 92.57% | Val Loss: 0.0552 | Val Acc: 98.33%
--> Model Saved
Epoch [34/35] Loss: 0.1576 | Acc: 92.63% | Val Loss: 0.0578 | Val Acc: 97.84%
Epoch [35/35] Loss: 0.1603 | Acc: 92.57% | Val Loss: 0.0659 | Val Acc: 96.96%

Total training time: 6m 29s
```

ResNet-34 (batch size 32)

```
Epoch [1/35] Loss: 0.7499 | Acc: 51.90% | Val Loss: 0.6767 | Val Acc: 57.65%
--> Model Saved
Epoch [2/35] Loss: 0.6970 | Acc: 54.91% | Val Loss: 0.7151 | Val Acc: 58.73%
Epoch [3/35] Loss: 0.6854 | Acc: 55.73% | Val Loss: 0.7649 | Val Acc: 50.00%
Epoch [4/35] Loss: 0.6844 | Acc: 56.90% | Val Loss: 0.9739 | Val Acc: 52.06%
Epoch [5/35] Loss: 0.6382 | Acc: 63.44% | Val Loss: 0.5190 | Val Acc: 75.29%
--> Model Saved
Epoch [6/35] Loss: 0.6105 | Acc: 67.33% | Val Loss: 0.4747 | Val Acc: 77.94%
--> Model Saved
Epoch [7/35] Loss: 0.5838 | Acc: 69.91% | Val Loss: 0.4678 | Val Acc: 78.43%
--> Model Saved
Epoch [8/35] Loss: 0.5699 | Acc: 70.66% | Val Loss: 0.4612 | Val Acc: 79.71%
--> Model Saved
Epoch [9/35] Loss: 0.5331 | Acc: 72.89% | Val Loss: 0.3975 | Val Acc: 83.04%
--> Model Saved
Epoch [10/35] Loss: 0.5106 | Acc: 74.79% | Val Loss: 0.3505 | Val Acc: 86.96%
--> Model Saved
Epoch [11/35] Loss: 0.4759 | Acc: 75.93% | Val Loss: 0.2998 | Val Acc: 88.04%
--> Model Saved
Epoch [12/35] Loss: 0.4121 | Acc: 80.10% | Val Loss: 0.3306 | Val Acc: 84.80%
Epoch [13/35] Loss: 0.3864 | Acc: 81.30% | Val Loss: 0.2551 | Val Acc: 88.92%
--> Model Saved
Epoch [14/35] Loss: 0.3570 | Acc: 83.20% | Val Loss: 0.2381 | Val Acc: 89.41%
--> Model Saved
Epoch [15/35] Loss: 0.3395 | Acc: 83.59% | Val Loss: 0.1974 | Val Acc: 92.16%
--> Model Saved
Epoch [16/35] Loss: 0.3198 | Acc: 84.64% | Val Loss: 0.1596 | Val Acc: 93.82%
--> Model Saved
Epoch [17/35] Loss: 0.2955 | Acc: 86.56% | Val Loss: 0.1778 | Val Acc: 92.35%
```

```
Epoch [18/35] Loss: 0.2762 | Acc: 87.17% | Val Loss: 0.1332 | Val Acc: 94.41%
--> Model Saved
Epoch [19/35] Loss: 0.2636 | Acc: 87.96% | Val Loss: 0.1256 | Val Acc: 95.00%
--> Model Saved
Epoch [20/35] Loss: 0.2465 | Acc: 88.53% | Val Loss: 0.1654 | Val Acc: 92.75%
Epoch [21/35] Loss: 0.2559 | Acc: 88.13% | Val Loss: 0.1167 | Val Acc: 95.00%
--> Model Saved
Epoch [22/35] Loss: 0.2331 | Acc: 89.37% | Val Loss: 0.1305 | Val Acc: 95.49%
Epoch [23/35] Loss: 0.2269 | Acc: 89.44% | Val Loss: 0.0937 | Val Acc: 96.08%
--> Model Saved
Epoch [24/35] Loss: 0.2276 | Acc: 89.44% | Val Loss: 0.0968 | Val Acc: 95.98%
Epoch [25/35] Loss: 0.2083 | Acc: 90.27% | Val Loss: 0.0741 | Val Acc: 97.35%
--> Model Saved
Epoch [26/35] Loss: 0.2073 | Acc: 90.77% | Val Loss: 0.1132 | Val Acc: 95.49%
Epoch [27/35] Loss: 0.1976 | Acc: 90.97% | Val Loss: 0.0999 | Val Acc: 95.98%
Epoch [28/35] Loss: 0.1971 | Acc: 91.27% | Val Loss: 0.0706 | Val Acc: 97.25%
--> Model Saved
Epoch [29/35] Loss: 0.1857 | Acc: 91.39% | Val Loss: 0.0545 | Val Acc: 97.35%
--> Model Saved
Epoch [30/35] Loss: 0.1859 | Acc: 91.81% | Val Loss: 0.0859 | Val Acc: 96.08%
Epoch [31/35] Loss: 0.1830 | Acc: 91.39% | Val Loss: 0.0939 | Val Acc: 95.88%
Epoch [32/35] Loss: 0.1750 | Acc: 92.03% | Val Loss: 0.0654 | Val Acc: 96.86%
Epoch [33/35] Loss: 0.1750 | Acc: 92.01% | Val Loss: 0.0757 | Val Acc: 96.96%
Epoch [34/35] Loss: 0.1614 | Acc: 92.84% | Val Loss: 0.0565 | Val Acc: 98.14%
Epoch [35/35] Loss: 0.1630 | Acc: 92.43% | Val Loss: 0.0467 | Val Acc: 98.24%
--> Model Saved

Total training time: 5m 58s
```