RBE577 - HW2 - Transfer Learning on ResNet18 for Vehicle Classification

Pranay Katyal
Worcester Polytechnic Institute
Worcester, MA, USA
pkatyal@wpi.edu

Anirudh Ramanathan
Worcester Polytechnic Institute
Worcester, MA, USA
aramanathan@wpi.edu

***Abstract*—This work applies transfer learning with ResNet18 for vehicle classification in autonomous driving scenarios. Using a pretrained ImageNet model[1], we replaced the classification head to recognize 10 vehicle types (bus, truck, sedan, SUV, etc.) from the Kaggle Vehicle Classification dataset[2]. A two-phase training strategy was employed: first freezing the backbone for 20 epochs to train only the classification head, then fine-tuning all layers for 30 additional epochs with a reduced learning rate. Data augmentation techniques[3] including random horizontal flips, color jittering, random rotation, and random resized crops were applied to prevent overfitting. The model achieved 96.0% validation accuracy with no signs of overfitting, demonstrating effective transfer learning on a small dataset of 1,400 training images. Training and validation losses tracked closely throughout training, confirming proper regularization through augmentation and weight decay alone, without requiring dropout.**

I. Introduction

Autonomous vehicles require robust real-time classification of surrounding traffic to make safe navigation decisions. This work addresses vehicle type classification from camera images, a critical perception task for self-driving systems.

The primary objective is to apply transfer learning to adapt a pretrained ResNet18 model for recognizing 10 vehicle classes: bus, family sedan, fire engine, heavy truck, jeep, minibus, racing car, SUV, taxi, and truck. Transfer learning is particularly valuable when training data is limited, as the model leverages features learned from ImageNet's 1.2 million images rather than learning from scratch.

We selected ResNet18 over deeper variants (ResNet50, ResNet152) due to our small dataset size of only 1,400 training images. Deeper models would risk overfitting given the limited data, while ResNet18's 11 million parameters provide sufficient capacity without excessive complexity.

Our approach employs a two-phase training strategy: Phase 1 (20 epochs) trains only the classification head with a frozen feature extractor, allowing the new head to reach reasonable values. Phase 2 (30 epochs) fine-tunes all layers with a reduced learning rate for final optimization. This gradual unfreezing strategy prevents the randomly initialized head from corrupting pretrained features during early training.

II. Methodology

A. Dataset

The dataset is sourced from the Marquis03 Vehicle Classification dataset on Kaggle, containing 1,800 images across three splits: 1,400 training images, 200 validation images, and 200 test images. Training and validation images are organized into 10 class subdirectories (bus, minibus, fire engine, heavy truck, truck, family sedan, jeep, racing car, SUV, taxi), providing automatic labels. The test set contains unlabeled images in a flat directory. The predefined splits were used without modification.

B. Model Architecture

ResNet18 pretrained on ImageNet consists of convolutional blocks followed by a fully connected (FC) classification head originally configured for 1,000 ImageNet classes. We replaced the final FC layer with a new linear layer mapping 512 features to 10 vehicle classes. The modified model contains approximately 11 million parameters. ResNet18 was selected over deeper variants (ResNet50, ResNet152) due to the small dataset size; deeper architectures would risk overfitting with only 1,400 training samples. The chosen architecture achieved 96% validation accuracy without requiring extensive hyperparameter tuning.

C. Data Preprocessing and Augmentation

1) Training Transforms: For Data Augmentation, we use various transforms like :

- Resize(256)
- RandomResizedCrop(224)
- RandomHorizontalFlip(p=0.5)
- ColorJitter(brightness=0.25, contrast=0.2, saturation=0.2, hue=0.25)
- RandomRotation(±10°)
- ImageNet normalization: mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]

Here the 'p=0.5' means that the probability of this transformation happening is 50%.

We use the ImageNet normalization mean and std so as to make sure we are consistent with what the ResNet18 was originally trained on, this prevents creation of garbage data.

2) Validation Transforms:

- Resize(256)
- CenterCrop(224)
- ImageNet normalization: mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]

For Validation we do not do any of the Color or flip augmentations, because we want to evaluate the model on accurate validation images.

D. Two-Phase Training Strategy

1) Phase 1: Frozen Backbone (20 epochs): All ResNet18 parameters were frozen by setting `requires_grad = False`, except for the newly initialized classification head (`model.fc`). Training with learning rate lr $= 10^{-3}$ updated only the head weights, allowing it to reach reasonable values without large gradients corrupting the pretrained feature representations. After Phase 1, validation accuracy reached approximately 93%.

2) Phase 2: Full Fine-tuning (30 epochs): All model parameters were unfrozen (`requires_grad = True`), enabling gradient updates throughout the entire network. The learning rate was reduced to lr $= 10^{-4}$ (10× lower than Phase 1) for gentle adaptation of pretrained features to the vehicle classification domain. This phase improved final validation accuracy from 93% to 96%.

E. Training Configuration

Adam optimizer with weight decay 10^{-4} (L2 regularization) and CrossEntropyLoss were used for training. Batch size was set to 64. Training was performed on an NVIDIA RTX 4080 (12GB GPU), with total training time of approximately 22 minutes for all 50 epochs (20 + 30).

III. Hyperparameters

Table I summarizes the key hyperparameters used during training. The two-phase approach uses different learning rates to balance rapid head initialization with careful feature adaptation.

Parameter	Value
Optimizer	Adam
Learning Rate (Phase 1)	10^{-3}
Learning Rate (Phase 2)	10^{-4}
Weight Decay	10^{-4}
Batch Size	64
Epochs (Phase 1)	20
Epochs (Phase 2)	30
Loss Function	CrossEntropyLoss

TABLE I
Training Hyperparameters

IV. Results

A. Training Dynamics

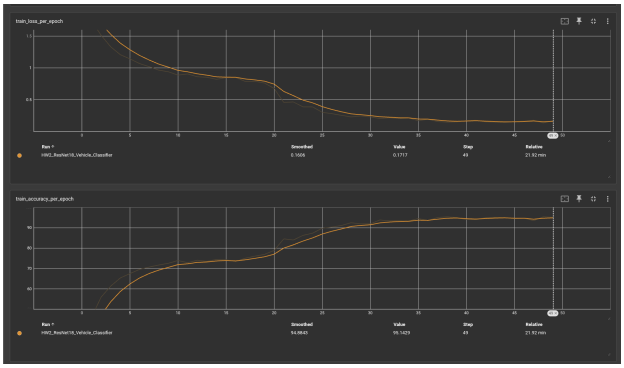

Fig. 1. Training loss and accuracy over 50 epochs. Loss decreases smoothly from 1.6 to 0.17, while accuracy increases from 50% to 95.4%.

Figure 1 shows training metrics across both phases. Phase 1 (epochs 0-19) exhibits rapid improvement as the classification head learns to map ResNet features to vehicle classes. Phase 2 (epochs 20-49) shows continued but slower improvement as pretrained features adapt to the vehicle domain. The smooth curves indicate stable training without divergence.

B. Validation Performance

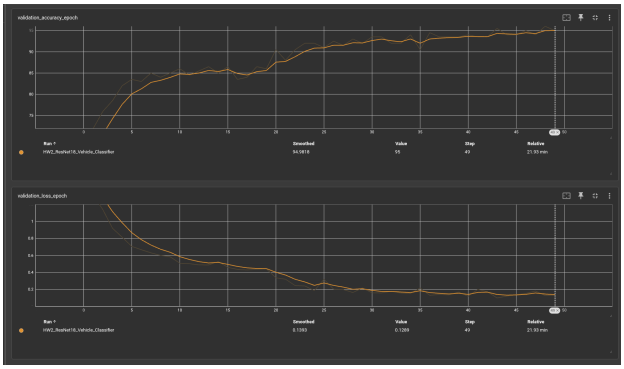

Fig. 2. Validation loss and accuracy over 50 epochs. Final validation accuracy reached 96.0% with loss of 0.13.

Validation metrics (Figure 2) closely track training performance, indicating minimal overfitting. The model achieved 96.0% validation accuracy, correctly classifying 192 of 200 validation images. The small gap between training (95.4%) and validation (96.0%) accuracy confirms that regularization through data augmentation and weight decay was sufficient.

C. Classification Examples

Fig. 3. Ten randomly selected validation images with predictions (green = correct, red = incorrect). This sample achieved 10/10 correct classifications.

Figure 3 shows representative validation predictions. Across multiple random samples, the model typically achieves 9-10 correct predictions per 10 images, consistent with 96% accuracy.

Fig. 4. Ten predictions on unlabeled test images demonstrating generalization to unseen data.

Test set predictions (Figure 4) show reasonable classifications on unseen data. Visual inspection confirms correct predictions for standard vehicle appearances. One observed failure case involved a green fire engine misclassified as a heavy truck, likely due to the unusual color deviating from the predominantly red fire engines in the training set.

V. Analysis and Discussion

A. Training Strategy Effectiveness

The two-phase training strategy proved essential for achieving high accuracy. Initial experiments with 10 frozen epochs followed by 30 fine-tuning epochs achieved only 87.5% validation accuracy. Extending Phase 1 to 20 epochs improved final accuracy to 96%, a substantial 8.5 percentage point gain. This suggests the classification head required more epochs to stabilize before unfreezing the backbone. Data augmentation created realistic variations within each batch, preventing overfitting despite the small dataset size.

B. Failure Case Analysis

Misclassifications occurred primarily in ambiguous cases. A green fire engine was classified as a heavy truck, likely because training fire engines were predominantly red. The model occasionally confused SUVs and family sedans when image cropping removed distinguishing features such as vehicle height or cargo area. Manual inspection of the training data revealed some hatchback vehicles incorrectly labeled as family sedans, potentially contributing to confusion on borderline cases. These failures are understandable given that some vehicles occupy a continuum between categories rather than discrete classes, and label noise in the training data can propagate to learned representations.

C. Overfitting Prevention

ResNet18's moderate size (11M parameters) prevented overfitting on the 1,400-image training set. Validation accuracy (96.0%) slightly exceeded training accuracy (95.4%), likely due to training augmentation introducing harder examples. Regularization through weight decay (10^{-4}) and data augmentation proved sufficient without requiring dropout. Training converged rapidly in Phase 1 as the head learned basic feature-to-class mappings, then continued gradually in Phase 2 with diminishing returns as the model approached its performance ceiling. Each epoch required approximately 22 seconds, with total training time of 22 minutes.

D. Limitations

The dataset size (1,400 training images) is small compared to ImageNet's 1.2 million images. Data augmentation creates variations but cannot generate truly novel examples. Most training images depict urban daytime scenarios; performance on nighttime, rural, or adverse weather conditions remains untested. Label noise was observed in the training data, which may limit achievable accuracy. The model has not been evaluated for real-time inference requirements in autonomous driving systems.

VI. Lessons Learned

The extended Phase 1 training (20 vs 10 epochs) was critical for performance, highlighting the importance of allowing the classification head to stabilize before fine-tuning. Data augmentation and weight decay provided sufficient regularization without dropout, demonstrating that proper training strategy can compensate for limited data. Dataset quality matters as much as quantity; label noise in the training set likely capped maximum achievable accuracy. Future work should explore larger datasets with verified labels, test generalization to diverse environmental conditions, and evaluate inference speed for real-time deployment. ResNet50 could provide additional capacity for subtle inter-class distinctions, though at the cost of increased training time and overfitting risk.

References

[1] PyTorch/vision team, "Models and pre-trained weights — torchvision documentation," https://docs.pytorch.org/vision/main/models.html, 2025, accessed: 2025-10-04.

[2] Marquis03, "Vehicle classification dataset," https://www.kaggle.com/datasets/marquis03/vehicle-classification, 2023, accessed: 2025-10-04.

[3] PyTorch/vision team, "Transforms — torchvision documentation," https://docs.pytorch.org/vision/stable/transforms.html, 2025, accessed: 2025-10-04.