Vehicle Image Classification Project Report

1. Abstract

This report summarizes the Vehicle Classification Project aimed at classifying vehicle images into predefined categories
using a deep learning model. A ResNet-18 architecture was selected for its effectiveness in image classification tasks
due to its balance of accuracy and computational efficiency. After training and evaluating the model on the provided
dataset, we achieved a mean Average Precision (mAP) of X% on the test set. The project involved addressing class
imbalances, optimizing hyperparameters, and applying data augmentation to enhance model performance.

2. Data Analysis and Cleaning

Observations:

- The dataset comprises images of 12 vehicle categories, with noticeable class imbalances.
- Some images were duplicates or corrupted, which could impact the model's accuracy.

Cleaning Steps:

- Duplicate Removal: Utilized fastdup to identify and eliminate duplicate images.
- Corrupted Image Removal: Excluded corrupted images from the dataset.
- Class Balancing: Implemented oversampling for underrepresented classes to address imbalance issues.

3. Data Preprocessing

Steps and Rationale:

- Resizing: All images were resized to 224x224 pixels to ensure consistency in input dimensions for the model.

- Normalization: Applied normalization using the ImageNet dataset statistics to standardize pixel values.

- Augmentation: Employed data augmentation techniques (e.g., rotation, flipping, color jitter) to increase dataset variability and improve model generalization.

4. Model Architecture

Model: ResNet-18

- Architecture: ResNet-18, a deep residual network with residual blocks, was chosen for its effective feature extraction and reduced risk of overfitting.

- Rationale: The ResNet-18 model offers a good balance between accuracy and computational efficiency, making it suitable for the task.

5. Training and Experimentation

Training Settings:

- Loss Function: CrossEntropyLoss was used for multi-class classification.

- Optimizer: Adam optimizer with an initial learning rate of 0.001 was employed.

- Batch Size: 32, balancing between computational efficiency and memory constraints.

- Epochs: Trained for 25 epochs with early stopping based on validation loss.

Modifications:

- Data Augmentation: Applied to enhance model generalization.
- Learning Rate Scheduling: Reduced learning rate upon performance plateau.
- Class Weights: Applied to address class imbalance.

6. Results and Key Findings

Performance Metrics:

- Accuracy: Achieved 60% on the validation set.
- Precision, Recall, F1-Score: Metrics varied by category, with some categories (e.g., bus) showing high recall but low precision.
- mAP: The model attained a mean Average Precision of X% on the test set.

Learning Curves:

- Illustrated improvements in training and validation loss over epochs.
- A confusion matrix was generated to visualize model performance across categories.

7. Future Work

Potential Improvements:

- Advanced Models: Explore models like ResNet-50 or EfficientNet for potentially better performance.
- Class Imbalance: Implement additional methods such as SMOTE or advanced augmentation techniques.
- Hyperparameter Tuning: Further optimize hyperparameters and learning rate schedules.
- Transfer Learning: Consider using transfer learning with models pre-trained on larger datasets.