Car Brand Classification via Image Recognition

1. Introduction

This project tackles the automatic identification of car brands from images using deep learning. It aims to develop a robust model capable of classifying vehicle manufacturers using only visual input.

Motivation: This supports automated vehicle cataloging, smart search, and price modeling. The model uses supervised learning with ResNet50 and transfer learning, powered by progressive training and preprocessing.

Repository: https://github.com/RiadDePauwUCLL/car-brand-prediction

2. Data & Pipeline

Dataset: 250,000+ labeled car images across 120 brands. Each image depicts one vehicle, captured from various angles and conditions.

Preprocessing:
- Image resizing to 256x256
- Brand name normalization
- Stratified splits into train/validation/test
- Augmentation: flips, rotations, color jitter, perspective warp, and random erasing

Data was loaded using PyTorch's `ImageFolder`, with custom `transforms.Compose` pipelines.

3. Model Architecture & Training

Architecture: ResNet50 backbone (pretrained on ImageNet), with a custom classification head.

```
model.fc = nn.Sequential(
    nn.Linear(model.fc.in_features, 2048),
    nn.BatchNorm1d(2048), nn.ReLU(), nn.Dropout(0.3),
    nn.Linear(2048, 1024),
    nn.BatchNorm1d(1024), nn.ReLU(), nn.Dropout(0.3),
    nn.Linear(1024, num_classes)
)
```

Training strategy:
- Progressive unfreezing of layers
- Class-weighted loss to mitigate imbalance
- OneCycleLR learning rate scheduler
- Gradient accumulation, early stopping, checkpointing
- PyTorch DirectML enabled training on AMD GPUs

Car Brand Classification via Image Recognition

4. Evaluation & Results

Results:
- 75% top-1 accuracy across 120 classes
- 90% top-5 accuracy
- Avg Precision: 0.77 | Avg Recall: 0.75

Insights:
- Strong performance for iconic designs (e.g. BMW, Mercedes)
- Confusion mostly between similar-looking brands
- Accuracy >50 samples/class improves significantly
- Confusion matrix indicates visual & ownership relations

5. Testing & Visualization

The model's predictions were validated using confusion matrices, per-class classification reports, and top-k accuracy metrics. Evaluation includes:
- Confusion matrix plots for error analysis
- Sample predictions visualized using `imshow`
- Use of `model.eval()` for consistent inference mode

6. Contributions & Next Steps

Individual Contributions:
- Collected, cleaned, and augmented a large dataset
- Designed a custom training loop with multiple optimizations
- Built tools for evaluation, error tracking, and visualization

Next Steps:
- Explore model ensembles & attention mechanisms
- Extend classification to model, year, body style
- Merge structured data to predict car prices via multi-task learning