Car Model Recognition using Deep Learning

Nikolay Sidorov

1 Introduction

The objective of this project was to develop a neural network pipeline that can identify the model of a car from an input image. The motivation behind this task stems from real-world applications such as automated surveillance, insurance assessment, and intelligent transportation systems.

To achieve this, a small dataset titled *"Car Images Dataset"* by Kshitij Kumar, available on Kaggle, was used. The project involved two core components: object detection and image classification. Additionally, utility tools were developed to handle image input from various sources.

2 Results

2.1 Milestone 1: Object Detection

The first step involved building an object detector using the pre-trained SSD MobileNet V2 FPNLite 320x320 model. This detector was fine-tuned to recognize cars using the MS COCO label map. The model successfully identified car objects and generated bounding boxes (green boxes) around them in input images.

2.2 Milestone 2: Car Model Classification

Once the car regions were localized by the detector, a classifier was trained using the MobileNetV2 architecture as the base model. The classifier was trained on cropped car images from the dataset to learn distinguishing features of different car models. Data was resized to 224×224 pixels to match the input expectations of MobileNetV2.

2.3 Highlight: Integrated Pipeline

An end-to-end pipeline was created to connect all components. It performs the following steps:

1. Load an image (from local directory or web) using custom loaders.
2. Detect car(s) using the object detection model.

3. Crop the detected regions (green boxes).

4. Classify each car region using the image classifier.

5. Display predictions and visualization.

Pipeline snippet (Python):

```
def pipeline(path):
  img = webImageLoaderToDetector(path)
  cropped = detectObj(img)
  for car in cropped:
    img_array = convertToArray(car)
    pred = classifierModel.predict(img_array)
    predicted_class_index = np.argmax(pred[0])
    confidence = pred[0][predicted_class_index]
    predicted_class_name = class_names[predicted_class_index]
    print(f"Predicted: {predicted_class_name} ({confidence:.2f})")
    plt.imshow(car)
    plt.axis('off')
    plt.show()
```

3 Conclusions

The project demonstrates a successful implementation of a multi-stage deep learning pipeline for car model recognition. Object detection and classification models were integrated effectively, and a clean interface was established to allow inference on both local and web-hosted images.

Future enhancements could include:

- Expanding the dataset with more car models for improved generalization.
- Fine-tuning the object detector for better car bounding box accuracy.
- Quantizing or optimizing models for faster inference at scale.