COMPS492F Machine Learning
Car Classification Project

Jiawei WANG
UID: 12395870

April 15, 2023

Contents

0.1 Overview

In this project, we implemented a car classification system using the ResNet-50 deep learning model. The primary goal was to classify images of cars into their respective classes based on their make and model. The process involved data preparation, model training, and testing.

0.2 Data Preparation

The dataset comprised images of various car models, organized into different folders based on their class. We performed the following steps to prepare the data:

- **Data Transformation**: To ensure consistency in input size, we resized the images to 256x256 pixels, performed center cropping to obtain 224x224 pixels, and converted them to PyTorch tensors. We also normalized the images.
- **Data Split**: The dataset was split into training and validation sets to facilitate model evaluation during training. The training set was used to train the model, while the validation set was used to monitor the model's performance.
- **Dataloader**: We utilized PyTorch's DataLoader to manage the training and validation datasets, which allowed efficient loading and batching of images during training.

0.3 Classification Implementation

We used the `ResNet-50` model, a popular deep learning architecture network that has shown excellent performance on various image classification tasks. It has 50 layers and uses residual connections (skip connections) to improve the training of deep networks., for car classification. We made the following modifications to the pre-trained model:

- We replaced the final fully connected layer with a new one that has 20 output units, corresponding to the number of car classes in our dataset.
- We used the cross-entropy loss function as our classification criterion and the stochastic gradient descent (SGD) optimizer with momentum for optimization.
- The model was trained for 100 epochs, and the model was saved in a device-agnostic way every 5 epochs.

Algorithm 1: Classification Implement

```
/* Modifying the ResNet-50 model */

criterion = nn.CrossEntropyLoss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)


/* Training process */

model = models.resnet50(pretrained=True)
num_features = model.fc.in_features
model.fc = nn.Linear(num_features, len(class_mapping))


/* train your own model */

python3 train.py --datapath <your datapath>
```

0.4 Model Testing

To evaluate the trained model's performance, we created a separate script, `test_model.py`. This script accepts the test data path, trained model path, and path for saving the classification results. The model evaluation process involves the following steps:

- **Loading the trained model**: The trained model is loaded from the specified path, and its weights are transferred to the appropriate device (CUDA or CPU).
- **Test data preparation**: The test dataset is created using the CarTestDataset class, which applies the same data transformations as the training dataset.
- **Model evaluation**: We set the model to evaluation mode and performed inference on the test dataset using DataLoader. The model's predictions were compared with the ground truth labels to calculate the accuracy.
- **Saving the results**: The classification results were saved to a text file in the format 'image_name: predicted_class_name'.

0.5 Conclution

In conclusion, this project demonstrated the successful implementation of a car classification system using the ResNet-50 model. The process involved data preparation, model training, and testing. The trained model can now be used to classify car images into their respective classes with reasonable accuracy (on the test dataset, the accuracy is around `97.42%`).