Comparing Multiple Techniques for Vehicle Classification

Jong Hoon Park   Devansh Dhrafani   Hridayesh Josh

`jonghoon, ddhrafan, hvj @andrew.cmu.edu`

Abstract

Vehicle classification is a critical task in machine learning and computer vision, with many real-world applications. In this project, we explore multiple vehicle classification techniques using pre-processing and deep learning models on driving scenes. We compare the performance of different approaches and demonstrate their potential for real-time vehicle classification. We hope our results contribute to advancing research in machine learning and computer vision and offer a starting point for improving accuracy in vehicle classification.

1. Introduction

[1] Vehicle classification has been an active area of research in the fields of machine learning and computer vision. We explored four different approaches for vehicle classification using 10,204 snapshots of scenes via various pre-processing methods for training data (RGB images of game scenes) and convolutional neural network architectures. The dataset included 7,573 training images and 2,631 test images, each containing an RGB image, a 3D bounding box, a camera matrix, and a label. The dataset had three unique labels corresponding to sets of similar vehicles. (e.g. sedans and SUVs are classified as the label 1).

2. Experiments

The first method involved fine-tuning a pre-trained ResNet18 model [1] [2] by passing training images through its 17 convolutional layers, concatenating bounding box coordinates to feed the result into a fully connected layer.

The second method transferred a pre-trained ResNet18 and replaced its last layer with a dense layer having 3 output neurons. The model was fine-tuned with training images.

In the third method, we combined a simple neural network, a feature extractor, and a pre-trained ResNet18. Unlike the first method, we converted the 3D bounding box co-

---

[1] Github

[2] ResNet18 is a convolutional neural network for image classification, introduced in "Deep Residual Learning for Image Recognition" by Kaiming He et al. 2016.

ordinates for each image to 2D coordinates and used them to crop vehicles from the training images. The feature extractor then used a pre-trained ResNet18 to extract features from the cropped images. These features were then fed into the simple neural network, consisting of 2 dense layers, to capture additional features. We replaced the last layer of another pre-trained ResNet18 with this simple neural network and fine-tuned it using the full-scale training images.

In the fourth method, we modified a pre-trained ResNet18 by replacing its ReLU activation function with a leaky ReLU with negative slope of 0.01, and its last fully connected layer with three dense layers. We added these extra layers to capture more complex features. We chose leaky ReLU based on empirical results reported in [2], which showed that it can lower test loss for convolutional neural networks such as CIFAR-10 and CIFAR-100 showing that it can lower test loss for convolutional neural networks using CIFAR benchmark. We set the negative slope as 0.001 to experiment.

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \frac{x_i}{a_i} & \text{if } x_i < 0 \end{cases} \quad (1)$$

3. Results

Our four experimental approaches were evaluated using 2,631 test images extracted from the game's 3D universe. Method 1 achieved 50.5% accuracy, while Method 2, Method 3, and Method 4 achieved 69.1%, 59.1%, and 60.8% accuracy, respectively. However, these results are dependent on the choice of hyperparameters and may vary accordingly.

4. Conclusion

Our second method achieved the highest test accuracy among all experimental methods suggesting that adding extra layers, activation functions, or data augmentation techniques does not always improve image classification performance. Experimenting with different combinations of these techniques may be necessary, and our findings provide additional insights for the field of autonomous vehicle perception.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778, 2016.

[2] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolution network. 2015.

Table 1. Different Vehicle Classification Approaches

Method	Description
Method 1	- Model - ResNet18 (17 convolutional layers & 1 dense layer; ReLU activation) - Data - Full-scale scene images - Additional Features - Coordinates of 3D bounding box around each vehicle - Preprocessing - Resizing, Normalization, Horizontal-Flipping - Stochastic Gradient Descent, Cross Entropy Loss
Method 2	- Model - ResNet18 (17 convolutional layers & 1 dense layer; ReLU activation); Last layer modified to accept 3 classes - Data - Full-scale scene images - Preprocessing - Resizing, Normalization - Adam Optimizer, Cross Entropy Loss
Method 3	- Model - ResNet18 (17 convolutional layers & 1 dense layer; ReLU activation); Feature Extractor (another ResNet18); Simple Neural Networks (2 dense layers) - Data - Full-scale scene images - Additional Features - Coordinates of 2D bounding box around each vehicle - Preprocessing - Resizing, Normalization, Horizontal Flipping, Rotation - Adam Optimizer, Cross Entropy Loss
Method 4	- Model - Modified ResNet18 (17 convolutional layers & 3 dense layers; Leaky ReLU activation) - Data - Full-scale scene images - Preprocessing - Resizing, Normalization - Adam Optimizer, Cross Entropy Loss