PR2: Vehicle Detection

Satya Harthik Sonpole
SCU ID: 07700012343, mAP Score: 84.73%, Rank: 4

February 2025

Abstract

This report presents an object detection approach using Transfer Learning for detecting three types of vehicles: cars, medium trucks, and large trucks. The model was trained using Faster R-CNN with a ResNet-50 backbone, leveraging a pre-trained network and fine-tuning on the given dataset. Hyperparameter tuning was conducted to optimize learning rate, batch size, and training duration. The final model achieved a mean Average Precision (mAP) score of 84.73%, securing **Rank 4** on the leaderboard. The decision to stop training at epoch 2 was influenced by validation loss trends, bias-variance tradeoff, and computational efficiency.

1 Introduction

Object detection is a fundamental problem in computer vision, widely used in applications such as autonomous driving and surveillance. This task involves identifying object locations and classifying them into predefined categories.

For this assignment, the objective was to train an object detection model via Transfer Learning to detect three vehicle types. The dataset consisted of 14,000 training images, 2,000 validation images, and 2,000 test images. The evaluation metric used was Mean Average Precision (mAP), a standard measure in object detection challenges, which balances precision and recall across multiple Intersection over Union (IoU) thresholds.

2 Approach

The dataset was analyzed to understand class distribution and bounding box annotations. The Faster R-CNN model was chosen due to its ability to produce high detection accuracy through a two-stage detection mechanism:

1. **Region Proposal Network (RPN)**: Generates a set of potential bounding boxes for objects in the image.
2. **Classification and Regression Head**: Refines the proposals and classifies detected objects.

The pre-trained ResNet-50 backbone was used for feature extraction, and fine-tuned on the dataset. The primary training phases involved:

- Data preprocessing and augmentation to enhance generalization.
- Implementing Faster R-CNN with a ResNet-50 backbone.
- Hyperparameter tuning for learning rate, batch size, and optimizer selection.
- Training and validation monitoring to prevent overfitting.
- Evaluation using mAP on the test set.

3 Methodology

3.1 Model Selection and Architecture

Faster R-CNN was chosen over single-stage detectors like YOLO due to its superior accuracy, albeit at the cost of higher inference time. The ResNet-50 backbone was selected for its deep feature representation capability, and the model was initialized with weights pre-trained on the COCO dataset.

3.2 Hyperparameters and Training Setup

The following hyperparameters were used:

- **Learning Rate:** 0.005, chosen based on empirical results from training convergence.
- **Batch Size:** 8, constrained by GPU memory limitations.
- **Optimizer:** Adam with weight decay for stability and convergence.
- **Augmentation:** Random horizontal flips and color jitter to improve robustness.

Training was conducted using an NVIDIA RTX 1080 GPU, with losses monitored at each epoch to determine the optimal stopping point. Table 1 presents loss values across epochs.

Epoch	Training Loss	Validation Loss
0	1149.99	290.86
1	1036.55	283.84
2	1005.29	286.68

Table 1: Training and Validation Loss across Epochs

4 Results and Stopping Criterion

The model was evaluated using the test set, achieving an mAP score of **84.73%**. The decision to stop training at epoch 2 was based on:

- **Validation Loss Trends:** The increase in validation loss at epoch 2 suggested that additional training could lead to overfitting, as the model started to fit noise rather than generalizable patterns.
- **Bias-Variance Tradeoff:** Faster R-CNN is known for low bias but high variance. Stopping early prevented excessive variance, maintaining generalization ability.
- **Computational Efficiency:** Given GPU constraints and diminishing returns from further training, stopping at epoch 2 was a balanced choice.

5 Conclusion

The Faster R-CNN model with a ResNet-50 backbone was successfully trained to detect three types of vehicles with high accuracy. The final mAP score of 84.73% demonstrated the effectiveness of Transfer Learning and fine-tuning. The decision to stop training at epoch 2 was justified by validation loss trends and computational considerations. Future work could explore:

- Fine-tuning with additional augmentation techniques.
- Experimenting with alternative architectures such as YOLO for real-time detection.
- Incorporating techniques like knowledge distillation to reduce model size while maintaining accuracy.