Optimizing Inference Time for Car Detection with Faster R-CNN MobileNet V3 Large FPN and Quantization

Luc Caselles

August 11, 2024

1 Introduction

Real-time object detection, specifically car detection, demands efficient models capable of processing images rapidly while maintaining high accuracy. This report investigates the optimization of inference time for car detection using the Faster R-CNN MobileNet V3 Large FPN architecture. To further enhance performance, model quantization is explored as a technique to reduce computational complexity.

2 Model Selection: Faster R-CNN MobileNet V3 Large FPN

The Faster R-CNN architecture, coupled with the MobileNet V3 backbone, was selected for its potential to balance accuracy and speed. This choice is motivated by the following factors:

MobileNet V3 Efficiency: The MobileNet V3 architecture is designed for efficient computation, making it a suitable candidate for real-time applications. MobileNet V3 incorporates several design choices to minimize computational cost:

- Inverted residual structure: This structure places bottleneck layers between expansion and projection layers, reducing the number of channels in computationally expensive layers.
- Efficient channel expansion and squeezing: By carefully balancing the number of channels in expansion and squeezing layers, MobileNet V3 achieves a good trade-off between accuracy and efficiency.
- Lightweight depth-wise separable convolutions: These convolutions decompose standard convolutions into depth-wise and point-wise convolutions, significantly reducing the number of parameters and computations.

- Global average pooling and linear layer: The final layers of MobileNet V3 are designed to be computationally efficient.

FPN for Multi-scale Detection: The Feature Pyramid Network (FPN) effectively addresses objects of varying sizes, improving overall detection performance.
Strong Baseline: This architecture represents a solid foundation for object detection tasks and has demonstrated competitive results in various benchmarks.

There might more optimized models to have fast inference, but the goal of this exercise was probably the exploration of optimization techniques.

3 Quantization for Inference Speedup

Quantization is a technique that reduces the numerical precision of weights and activations in a neural network. By representing numbers with fewer bits, the model size and computational cost are significantly reduced. This leads to faster inference times and lower memory consumption.

PyTorch's `torch.quantization.quantize_dynamic` function implements dynamic quantization. This method determines optimal quantization parameters at runtime based on the distribution of input data. The quantization process involves the following steps:

Data Collection: A representative dataset is used to gather input data for calibration. Observation: The model is executed with the calibration data, and statistics about the distribution of activations are collected. Parameter Determination: Based on the collected statistics, quantization parameters (scale and zero-point) are calculated for each tensor. Quantization: The model's weights and activations are quantized using the determined parameters. During inference, the model operates with quantized values, leading to faster computations and reduced memory footprint.

In our case the 50 random images of the provided dataset are used as calibration dataset (so the quantized model could be different for each run).

4 Dataset and evaluation

A dataset made up of 3000 images camera images with annotated bbox of cars if provided. The bbox are represented by top left/top right corners. The inference is done on all the dataset for quantized and non quantized models. This is done sequentially as camera probably do the inference in real time and can not do batched inferences. Predictions are dumps to ensure consistency (Figure 1). To evaluate the performance of both models, several metrics were employed:

- **Mean Max IoU per Prediction** (mean_max_iou_per_pred): This metric calculates the maximum IoU for each predicted bounding box and then averages these values across all predictions. It provides insights into the quality of individual predictions.

- **Mean Max IoU per Ground Truth** (mean_max_iou_per_gt): This metric calculates the maximum IoU for each ground truth bounding box and then averages these values across all ground truths. It assesses how well the model is able to detect all objects in the image.
- **Precision:** This metric measures the proportion of correct positive predictions among all positive predictions. It indicates the model's ability to avoid false positives.
- **Recall:** This metric measures the proportion of correct positive predictions among all actual positive instances. It indicates the model's ability to detect all relevant objects.

The given metrics are averaged on the 3000 images.
The inference time is also displayed to compare both models in terms of computational efficiency.

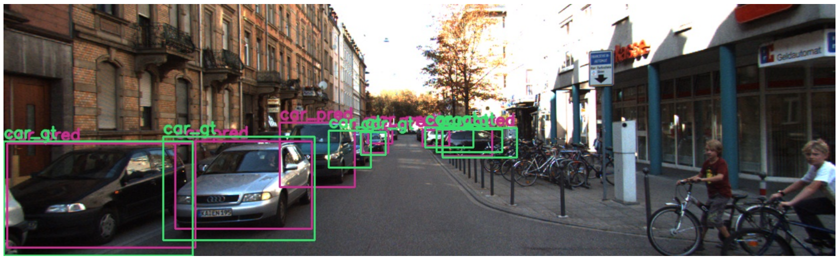

Figure 1: Exemple of prediction versus GT.

5 Comparison of Quantized and Non-Quantized Models

In this section, we compare the quantized and non-quantized models based on computational time and performance metrics. The hardware on which inference is run is an AMD Ryzen 5 5600 H.

5.1 Computational Time

The computational time for each model is as follows:

- **Quantized Model:** 1526.91 seconds
- **Non-Quantized Model:** 1845.23 seconds

The quantized model demonstrates a significant improvement in computational efficiency, with a reduction in processing time by approximately 318.32 seconds (about 17.2%).

5.2 Performance Metrics

Table 1 summarizes the performance metrics of both models. The metrics include mean maximum IoU per prediction, mean maximum IoU per ground truth, precision, and recall.

Table 1: Performance Metrics Comparison

Metric	**Quantized Model**	**Non-Quantized Model**
Mean Max IoU per Prediction	0.6397	0.6397
Mean Max IoU per Ground Truth	0.5863	0.5847
Precision	0.7341	0.7354
Recall	0.5956	0.5926

5.2.1 General Comments

- **Consistency in IoU:** Both models exhibit similar mean maximum IoU per prediction and mean maximum IoU per ground truth. This indicates that quantization has minimal impact on the object localization capabilities of the model.

- **Precision:** The quantized model shows a slight decrease in precision compared to the non-quantized model. Precision, which measures the accuracy of positive predictions, is slightly lower in the quantized model, suggesting a small trade-off in prediction accuracy.

- **Recall:** The quantized model demonstrates a slight improvement in recall, indicating marginally better performance in identifying true positives. This can be advantageous in scenarios where detecting all relevant instances is crucial.

In conclusion, while the quantized model shows minor variations in performance metrics, it achieves a notable reduction in computational time. The trade-offs in precision and recall are generally acceptable considering the efficiency benefits provided by quantization.