MOTORVEHICLE
UNIVERSITY OF
EMILIA-ROMAGNA

Università di Parma

3D Perception, Learning-Based Data Fusion Exam

2D Object Detection On NuImages

Francesco Marotta
Simone Maravigna

October 23, 2023

Project Report

Goal of the Project

The goal of this project is to develop an object detection system for autonomous vehicles operating in urban environments using the nuImages dataset. To achieve this, we employed the Faster R-CNN model with a ResNet-50-FPN backbone, initialized with pretrained weights from the COCO dataset, and subsequently fine-tuned to recognize the 23 classes specific to the nuImages dataset.

Dataset and Dataloader

The nuImages dataset contains 93,000 labeled images captured by a total of six cameras, divided equally into 3 front cameras and 3 rear cameras. Each image in the dataset is associated with a label that provides both semantic segmentation masks and 2D bounding boxes. This dataset classifies 23 distinct classes for foreground objects (e.g 'human.pedestrian.adult', 'vehicle.car', etc.).

The dataset is structured as a relational database, where each row in every table can be uniquely identified by its primary key token. Based on this organization, we have implemented the data loading process, specifically within the function `__get_item__()` of the dataset class.

Therefore, the data loader is able to navigate in such database to retrieve the required information and format them appropriately.

More in detail, the `__get_item__()` function returns the image along with its corresponding annotations. These annotations are composed of a tensor of shape `[N,4]`, which contains the bounding boxes, and a second tensor of shape `[N]`, which contains the labels. Where `N` is the number of objects in each image.

When the model acquires the data through the data loader, which allows it to access multiple images simultaneously, the images are normalized to a range of `[0,1]`. Subsequently, a data structure in the form of a list of dictionaries is constructed. Each dictionary represents one image and contains both the image itself and its corresponding annotations.

Model and Finetuning

We employed the Faster R-CNN model with a ResNet-50-FPN backbone. Firstly, we took several images from the nuImages dataset and input them into the pre-trained model. The results indicated that the model successfully detected and accurately classified objects based on the COCO dataset labels (the model was pretrained on the COCO dataset).

Consequently, we proceeded with fine-tuning the network to adapt it to our dataset. We modified the box predictor layer since our dataset had fewer classes compared to COCO (24 vs 91). As a result, we implemented a standard training cycle using SGD as the optimizer, Mean Average Precision as the evaluation metric, learning rate equal 0.005 and trained it for 10 epochs.

Results

The results indicated a consistently decreasing loss during training, with a mean average precision of nearly 48% on the validation dataset. It's worth noting that these results could likely have been improved with more extensive training. Subsequently, we applied the fine-tuned model to the nuImages-mini dataset, specifically on the images captured from the front camera (i.e., 8 images). The model achieved a mean average precision of approximately 53%, demonstrating its capability to detect and classify the objects of interest within the dataset with a reasonable level of precision.

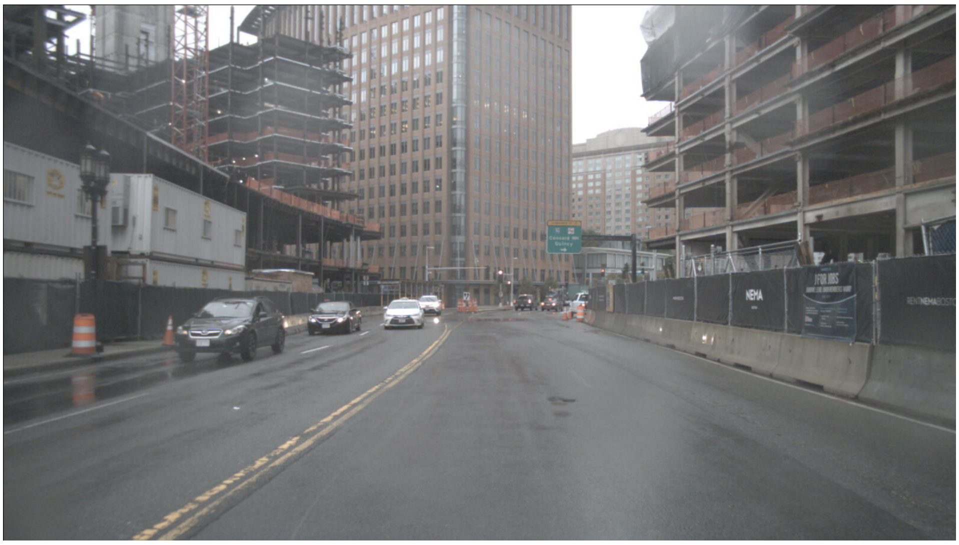

Figure 1: Example of an input raw image

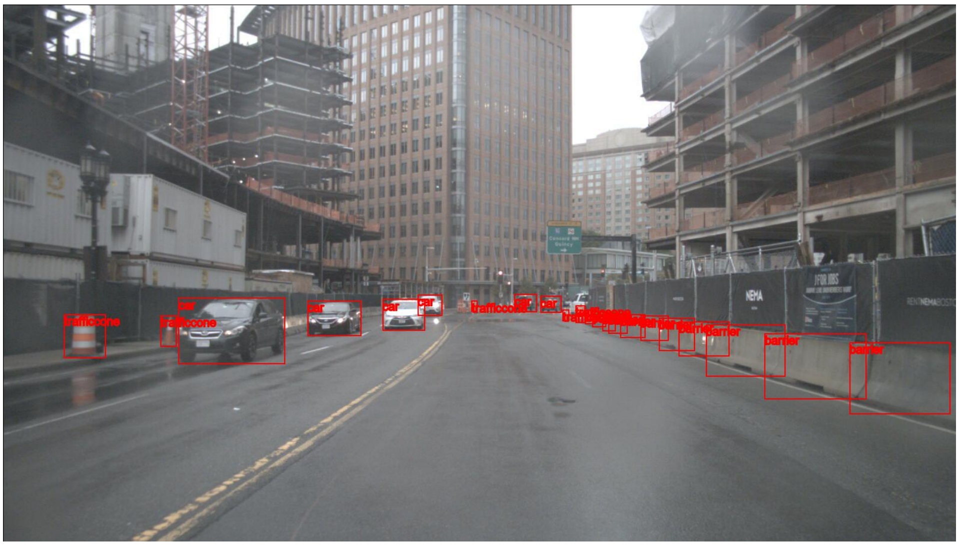

Figure 2: Example the output of the tuned model