Introduction

- Being able to accurately detect vehicles from videos has many practical applications, including autonomous vehicles.
- Current state-of-the-art methods for doing this include YOLO, which is capable of real-time detections.
- Most object detectors are not optimized for video detections and do not take into account temporal information from the video.
- Techniques such as sequential non-maximum suppression aim to improve video detections by using neighboring frames to improve weak detections [1].

Dataset

- We trained our model with the UA-DETRAC dataset consisting of traffic videos and their annotations.
- The dataset consists of 60 videos of urban traffic with a total 140K frames, 8250 vehicles and 1.21 million labeled bounding boxes [1].
- The video data was preprocessed into 416x416 images before feeding into YOLO, along with their list of annotated ground-truth object labels.

Methodology

For this task, we first trained different variations of the YOLO object detection architectures [2] to perform the object detections, including YOLOv2 and Tiny-YOLO. Below is a summary of the YOLOv2 architecture. The architecture for Tiny-YOLO is similar, but only with 8 convolutional layers in the bulk of the network.

In addition, we use sequential NMS instead of NMS as a postprocessing technique

Sequence NMS iterates three steps:
1. Find the max sequence subject to the constraint that adjacent frames must be similar (IoU > 0.5).
2. Weak detections in the sequence are then rescoring.
3. Frames close to the max sequence are then suppressed.

Results

For training and testing, we split our data into a 90/10 train/test ratio. Our test data contains 6 videos from a variety of environments (day/night, rainy/clear) to test the performance of the algorithm under different conditions.

To compare performance between models, we use the average precision (AP), which is the area under the precision/recall curve.

Preliminary results indicate that Sequential NMS postprocessing does worse than normal NMS on a toy subset. Further testing, debugging, and tuning is needed to confirm these results.

References