

Deep Learning for Autonomous Vehicles

Subproject 1 – Module 2

Traffic Light recognition

We've all been there: a light turns green and the car in front of you doesn't budge. No one likes to get stuck behind a vehicle that doesn't notice when a light change. Also, a system that can countdown on red light the time remaining until a change to green can save a significant quantity of fuel in city driving (e.g., restart engine five seconds before green) and advise driver to start braking early if it will not make it through a green light. Aim of this project is to develop a model to recognize traffic-light state in the car driving direction.

This dataset contains 13427 camera images at a resolution of 1280x720 pixels and contains about 24000 annotated traffic lights. The annotations include bounding boxes of traffic lights as well as the current state (active light) of each traffic light. The camera images are provided as raw 12bit HDR images taken with a red-clear-clear-blue filter and as reconstructed 8-bit RGB color images. The RGB images are provided for debugging and can also be used for training. However, the RGB conversion process has some drawbacks. Some of the converted images may contain artifacts and the color distribution may seem unusual.

Network: The R-CNN detector first generates region proposals using an algorithm such as Edge Boxes. The proposal regions are cropped out of the image and resized. Then, the CNN classifies the regions which were cropped and resized. Finally, the region proposal bounding boxes are refined by a support vector machine (SVM) that is trained using CNN features. The Fast R-CNN detector also uses an algorithm like Edge Boxes to generate region proposals. Unlike the R-CNN detector, which crops and resizes region proposals, the Fast R-CNN detector processes the entire image. Whereas an R-CNN detector must classify each region, Fast R-CNN pools CNN features corresponding to each region proposal. Fast R-CNN is more efficient than RCNN, because in the Fast R-CNN detector, the computations for overlapping regions are shared.

A post processing technique of Non-maximal suppression is used to select the bounding box with the highest probability of the object being there. The image is scaled back up and this box is displayed.

Hyperparameters summary:-

Number of samples for training- 1230

Number of samples for testing- 156

ROIs- 5

epoch length- 200

Epochs- 30

Anchors- 9

Results:



