Patch-Based Real Time Road Object Detection Using YOLOv3

Megan Rowe[1], Naveen Krishnamurthi[2], Puyang Ma[3]

[1]. Mechanical Engineering [2]. Computer Science [3]. Data Science | Email: {mrowe2, naveenk1, pym86} @stanford.edu

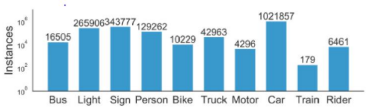
Stanford Deep Learning

Motivation

One of the key challenges in autonomous vehicles is building accurate real time object detection algorithms. Traditionally, there tends to be a tradeoff between speed and accuracy, and a standard approach is to decrease the resolution of input images in order to increase the speed of detection. The drawback of this approach is that it can make it difficult to detect small or more distant objects. In this project, we examine how running parallel object detection on multiple patches of an HD image can allow us to detect smaller objects in the overall image without decreasing its resolution. We found that precision for smaller object classes tended to improve, precision for larger object classes decreased slightly, and overall mean average precision was comparable..

Dataset & Features

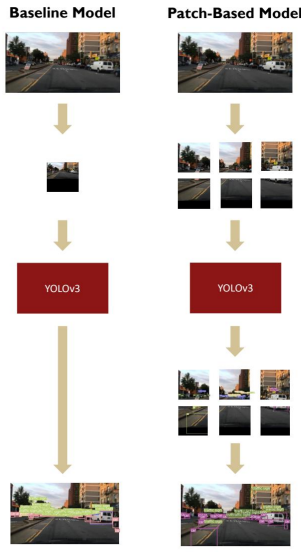
Our dataset consists of 100,000 images and labels from the BDD-100K dataset [1]. The images for the BDD-100K dataset were collected from over 1,100 hours of driving in a variety of different locations, driving scenarios, and weather conditions. They contain RGB channels and are of size 1280 x 720.



Statistics of different types of objects.

Models and Methodology

We used a PyTorch implementation of YOLOv3 that was pretrained on ImageNet. [2] We modified the repository to account for our object classes (traffic lights, traffic signs, people, and cars) and training data. We trained 2 different models on the dataset. The baseline model was trained on 70,000 images that were resized to 416x416. The second model was trained on 12,000 416x416 crops generated from 2,000 randomly selected full sized images of the same distribution as the baseline model.

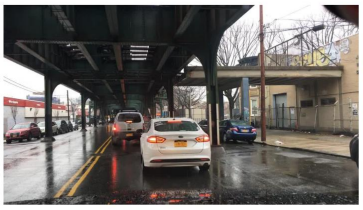


The two models used the same standard YOLOv3 network architecture. The architecture's loss function is provided below[3]

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)$$

Results

ORIGINAL INPUT IMAGE



BASELINE MODEL OUTPUT

PATCH-BASED MODEL OUTPUT

Discussion

Model	Input Size	Train Set Size	Validation Set Size	Epochs	Training AP (%)	Validation mAP (%)
Baseline	1280 x 720	70,000	500	10	35.2	25.9
Patch-Based	416x416	12,000	500	144	68.4	24.1

Overall, the mAP for our baseline model and patch-based model are very similar. When we examined the AP of each class, we found that our baseline model performed better on larger objects like cars and pedestrians while our patch-based model performed better on smaller objects like traffic lights and traffic signs. Our baseline model was unable to learn these smaller objects because our input image size of 1280x720px was shrunk down and padded to be the 416x416 size that YOLOv3 takes as input. When the image was shrunk down, many of the already small objects were sized down to very few pixels and were too small for our network to learn, leading the model to predict traffic lights and traffic signs everywhere in our validation images. Cropping eliminates this issue since the input image is not resized down leaving the smaller objects still large enough for the network to learn. One disadvantage of the patch-based model is that it cannot detect objects that are larger than the 416x416 patch size, however, the majority of the objects in our images are smaller than 416x416.

Future Work

- Productionize the real-time elements of the patch-based model.
- Train both of our models on all of the images provided in the BDD-100K dataset.
- Experiment with more advanced strategies for merging patch detections.

[1]. Fisher Yu, Wenqi Xian, Yingying Chen, et al. "BDD 100K: A Diverse Driving Video Database with Scalable Annotation Tooling"
[2]. Erik Lindernoren. "PyTorch-YOLOv3." https://github.com/eriklindernoren/PyTorch-YOLOv3.
[3]. Redmon, Joseph et al. "You Only Look Once: Unified, Real-Time Object Detection."