ROB599 Perception -Team 12

Approach 1 -- Yolov2

In [1], Redmon and Farhadi presented a state-of-the-art real-time object detection algorithm YOLOv2. They use Darknet-19 as the base of YOLO-v2. Darknet-19 contains 19 convolutional layers and 5 max-pooling layers. In this model, they use mostly 3x3 filters and double the number of channels after every pooling step. And, they use global average pooling to make predictions as well as 1×1 filters to compress the feature representation between 3 × 3 convolution. The overall network detail shows in Table 1. We basically use this method as reference, and use this ImageNet pretrained model. The advantage of YOLO is its fast speed and high accuracy, so the network is accurate and at the same time small and easy to train.

Before we train on top of this pretrained model, we apply several methods to given training image. It means that we are not directly use given training image and its labels. Our detailed training procedure include:

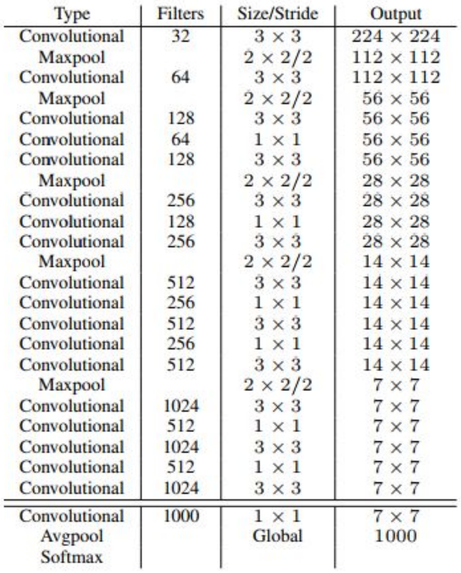
1. We apply *bilateral filter* (found in wikipedia) to compute depth image from 3D point cloud. The formula shows below. Because we can get 3D information directly from 3D point cloud, we then remove the area that not satisfy the filtering criteria from depth image.

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

2. Since we only have 3D bounding box, we convert this 3D bounding box to 2D bounding box by taking minimum/maximum x-y value from eight points of 3D bounding box.
3. Now, we original RGB image, 2D bounding box and depth image. Then, we replace last channel of RGB image to be depth image. Specifically, replace B as depth image. Lastly, we use this so called RGD image and 2D bounding box for training. All result of image shows in Table.2. And counting 2D bounding box of car.

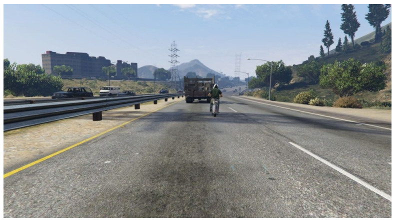
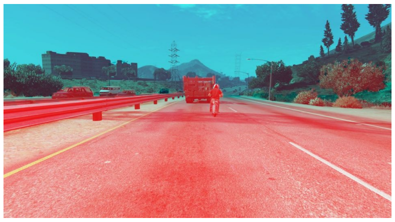
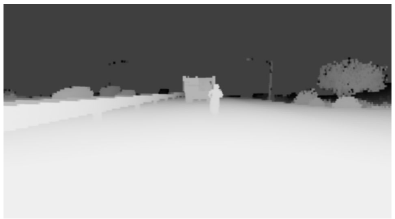
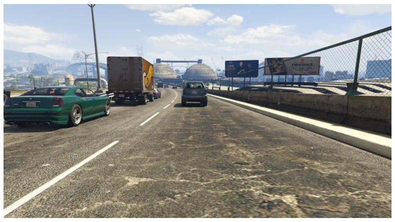
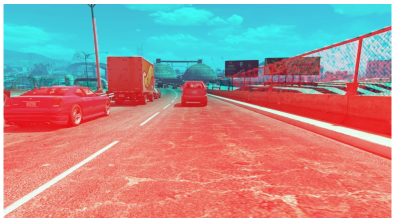
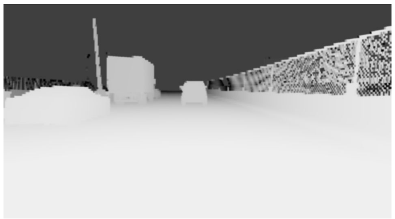
The approach 1 is the one that we submitted and get final result, we also have other approach 2. This approach is also attach at the end of this report.

Table 1: DarkNnet-19



Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Table 2: modified image

Original image	After remove one channel	Depth map from lidar
		
		

Approach 2 -- Deeplab v3+Filtering

For the second approach, one uses semantic segmentation to extract the car masks in the image and then use projected depth map to filter the masks.

During the first stage, Deeplab v3 is used to extract the mask of cars. ResNet-101 pre-trained on Imagenet is used, the final atrous convolution layers are finetuned on Playing for Data, which is a dataset containing around 25k images in GTA and corresponding labels for each class. Two rounds of finetuning were performed. The first one achieves 35.4% mIOU with all 35 classes presented. And the second one achieves 88.9% mIOU for just cars. One used the second training for car masks.

During the second stage, depth map are used to separate cars. Since semantic segmentation cannot distinguish instances of cars, for multiple cars close to each other, a big mask is often predicted. To alleviate this kind of problem, the following process is performed. First, the gradient for depth map is calculated. Secondly, a threshold is used to extract the region for dramatic depth change which often corresponds to the boundary of the objects. Thirdly, those region are subtracted from the car masks. An additional dilation process is performed to enlarge those regions so that instances of cars can be better separated. Finally, convex hull of the connected masks are calculated and the number of pixels for each mask is calculated for filtering output small masks which are the error from segmentation.

Compared to only filtering the size, use depth information gives additionally 0.12 gain in accuracy. And the main error is from the network's ability to detect small car instances.

References

[1] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." *arXiv preprint arXiv:1612.08242* (2016).

[2] Richter et al. 'Playing for Data: Ground Truth from Computer Games' website: https://download.visinf.tu-darmstadt.de/data/from_games/

[3] DeepLab-ResNet-TensorFlow, GitHub: https://github.com/DrSleep/tensorflow-deeplab-resnet