Gurtej Bhasin 1001103940 December 14, 2020

AER1515 – Assignment 3
3D Object Detection and Instance Segmentation

1.0 Depth Estimation

We are provided with a set of rectified stereo image pairs from the KITTI dataset. The disparity map for each stereo image pair along with the calibration information for the left and right camera. Using the given disparity map and calibration information, the depth map is estimated for each left stereo image. Pixels without a disparity have a value of 0 and as an extension these 0 disparity pixels correspond to 0 depth in our depth estimate map. Additionally, I assigned a depth of 0 if the depth at the pixel is greater than 80 m or less than 10 cm.

The formula for calculating depth from a disparity for a given pair of stereo images is as follows:

$$depth = \frac{fb}{disparity} \; where\, , f - focal\, length \wedge b - base\, length$$

1.1 Disparity and corresponding depth maps for training dataset

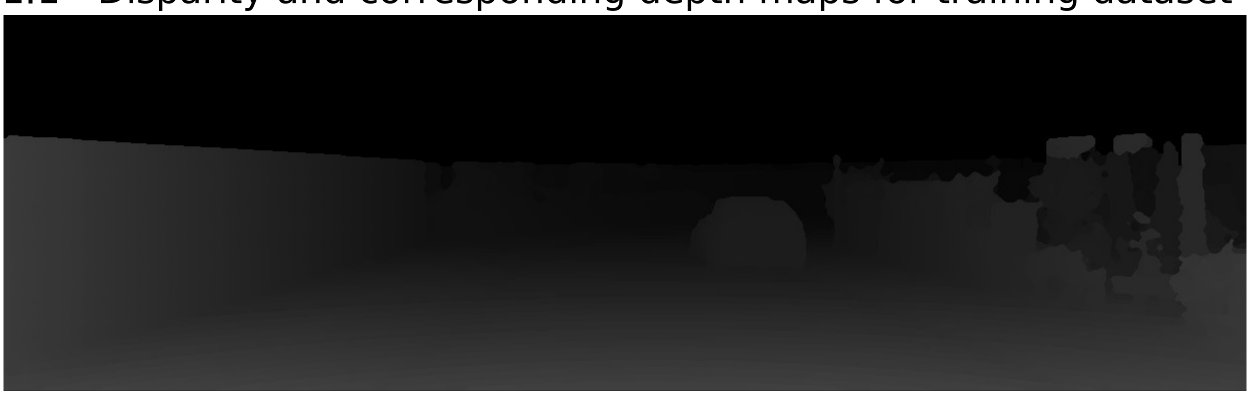

Figure 1: Disparity map for 000001

Figure 2: Depth map for 000001

Figure 3: Disparity map for 000002

Figure 4: Depth map for 000002

Figure 5: Disparity map for 000003

Figure 6: Depth map for 000003

Figure 7: Disparity map for 000004

Figure 8: Depth map for 000004

Figure 9: Disparity map for 000005

Figure 10: Depth map for 000005

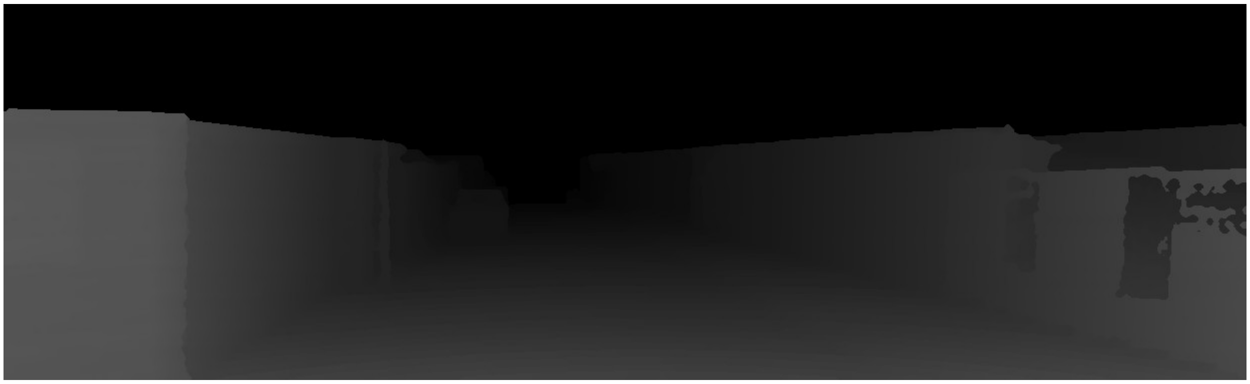

Figure 11: Disparity map for 000006

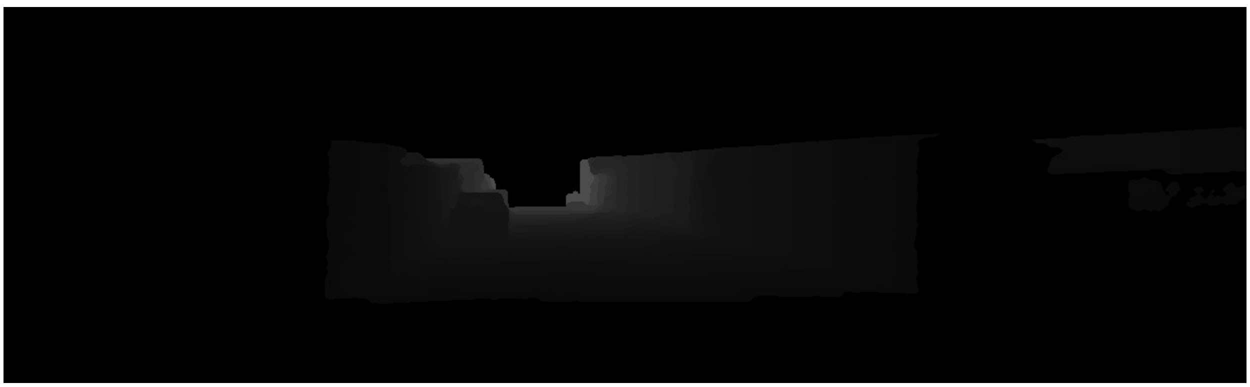

Figure 12: Depth map for 000006

Figure 13: Disparity map for 000007

Figure 14: Depth map for 000007

Figure 15: Disparity map for 000008

Figure 16: Depth map for 000008

Figure 17: Disparity map for 000009

Figure 18: Depth map for 000009

Figure 19: Disparity map for 000010

Figure 20: Depth map for 000010

1.2 Disparity and corresponding depth maps for test dataset

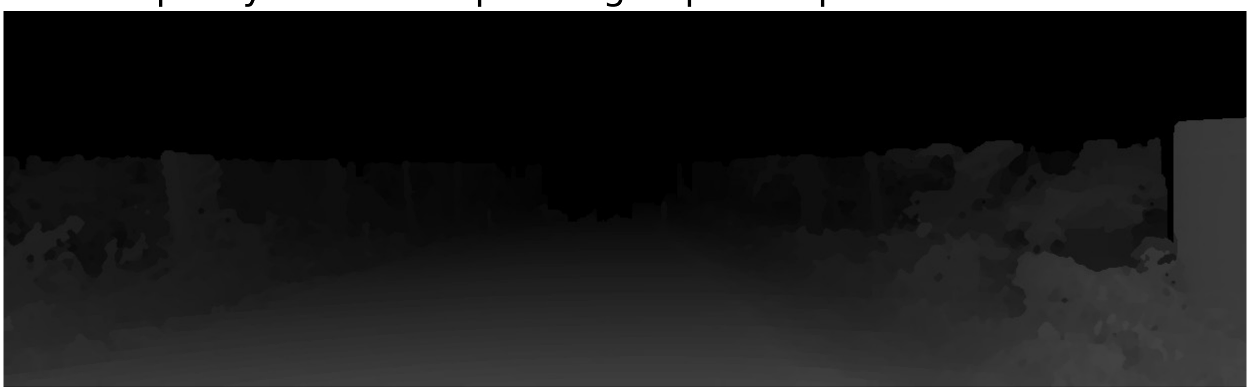

Figure 21: Disparity map for 000011

Figure 22: Depth map for 000011

Figure 23: Disparity map for 000012

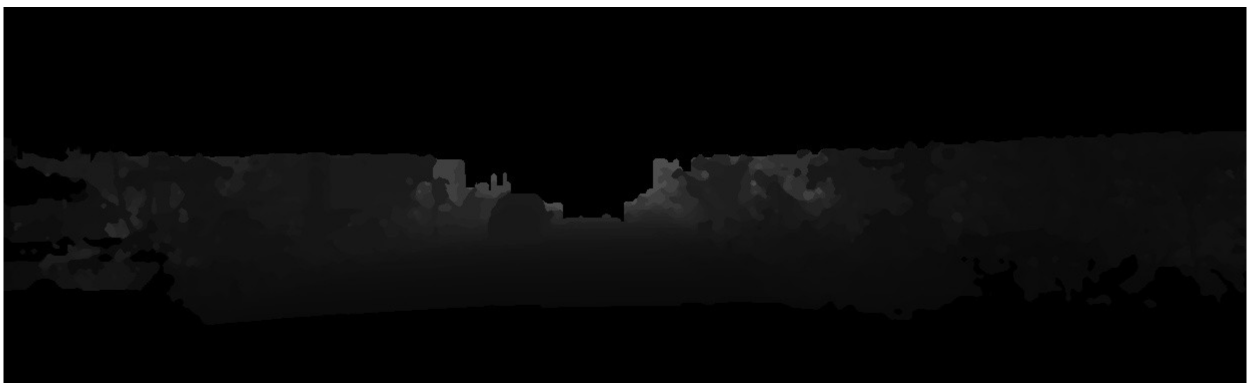

Figure 24: Depth map for 000012

Figure 25: Disparity map for 000013

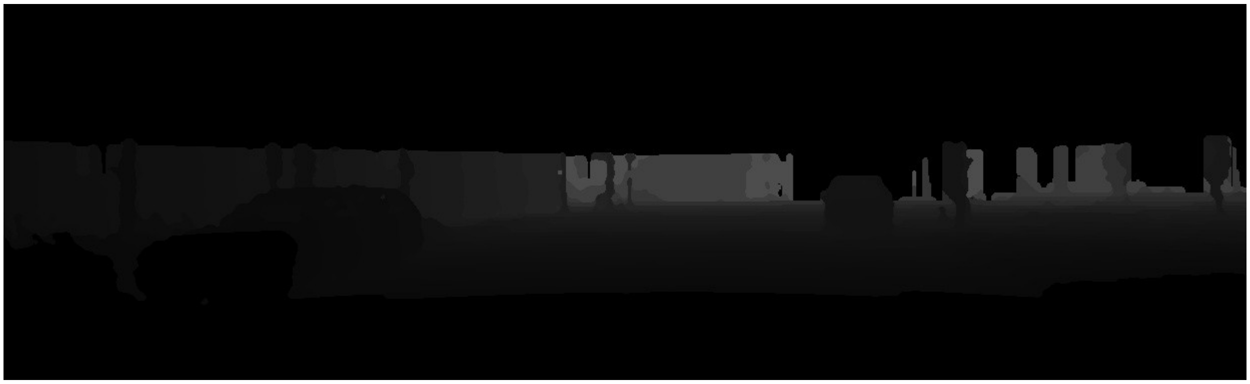

Figure 26: Depth map for 000013

Figure 27: Disparity map for 000014

Figure 28: Depth map for 000014

Figure 29: Disparity map for 000015

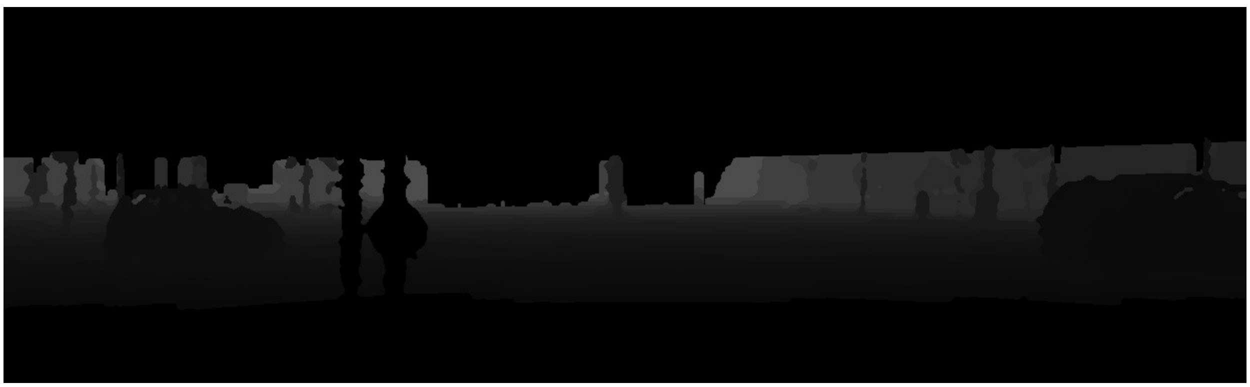

Figure 30: Depth map for 000015

2.0 2D Bounding Box Estimation

Implemented a YOLOv3 object detector to locate 2D bounding boxes for all cars in the left stereo image. The estimated location of the bounding box is represented as the pixel coordinates of the top left corner and the height and width of the box from the top left corner. The confidence and non-maximum suppression threshold parameters are tuned to the training set. After tuning, the optimal values were calculated to be 0.5 each.

2.1 2D bounding box estimate for training dataset

Figure 31: 2D bounding box estimate for 000001

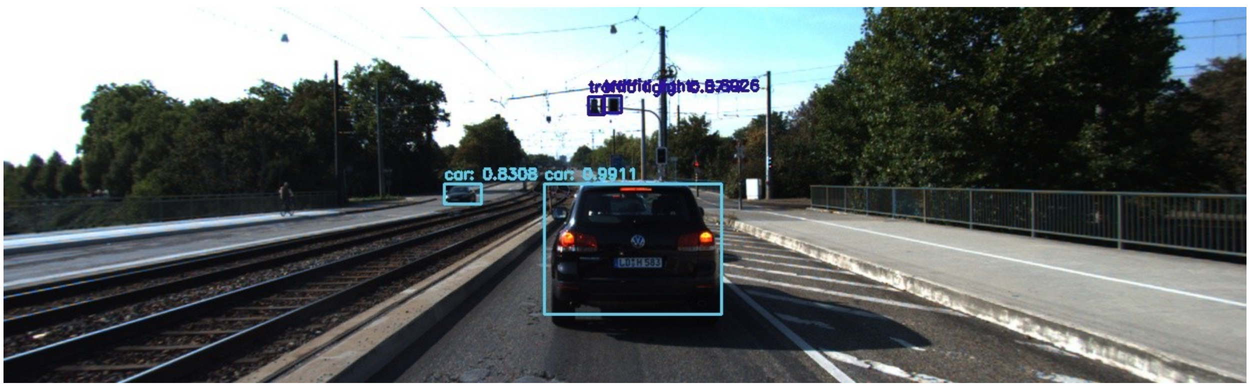

Figure 32: 2D bounding box estimate for 000002

Figure 33: 2D bounding box estimate for 000003

Figure 34: 2D bounding box estimate for 000004

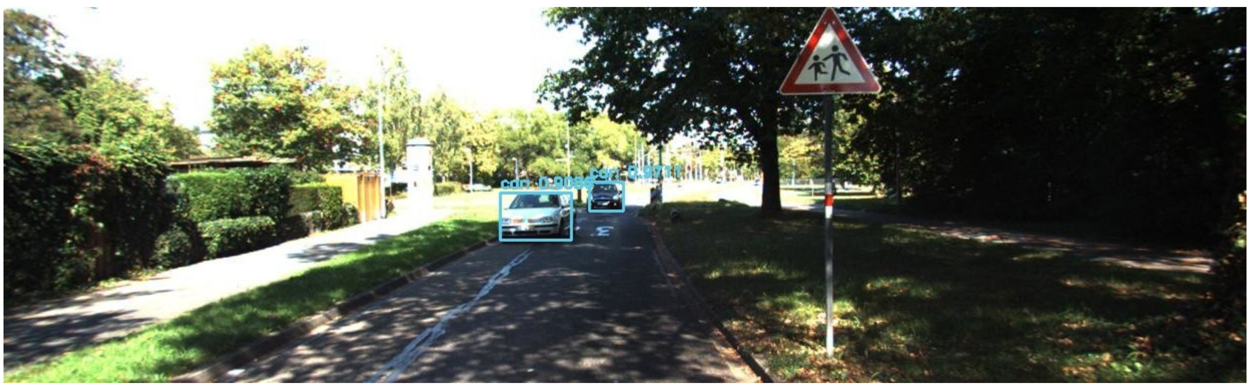

Figure 35: 2D bounding box estimate for 000005

Figure 36: 2D bounding box estimate for 000006

Figure 37: 2D bounding box estimate for 000007

Figure 38: 2D bounding box estimate for 000008

Figure 39: 2D bounding box estimate for 000009

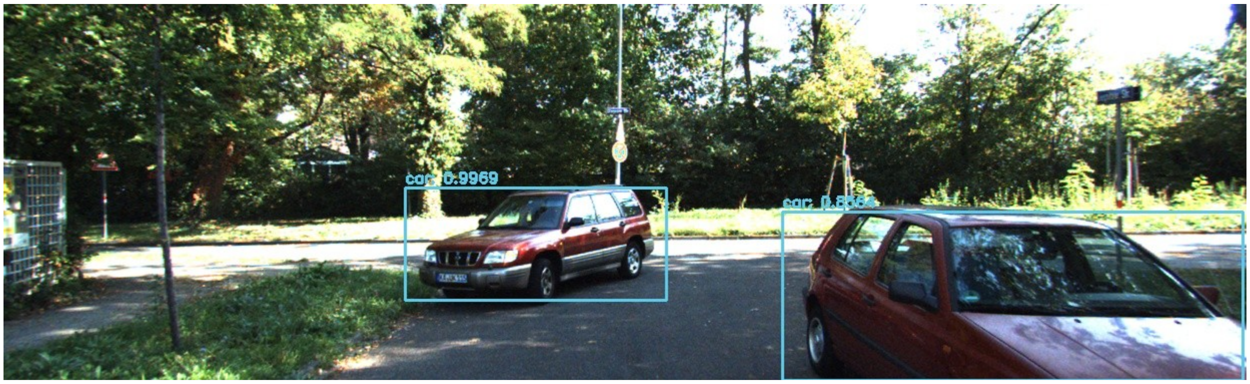

Figure 40: 2D bounding box estimate for 000010

2.2 2D bounding box estimate for test dataset

Figure 41: 2D bounding box estimate for 000011

Figure 42: 2D bounding box estimate for 000012

Figure 43: 2D bounding box estimate for 000013

Figure 44: 2D bounding box estimate for 000014

Figure 45: 2D bounding box estimate for 000015

3.0 Instance Segmentation

In this section, I have performed instance segmentation on the left image for every detected car from the YOLOv3 algorithm. Instance segmentation means you will attempt to find all pixels inside a detected bounding box that belongs to the object. The method I will implement is as follows:

1. For every detected bounding box, compute the average depth of the object

2. Find all pixels inside each bounding box that are within a certain distance (can be adjusted) from the computed average depth
    a. The threshold distance was tuned for the training dataset to have a value of 8.2 meters

For the segmentation mask, each pixel that corresponds to a car will be assigned a value of 0 and the non-car pixels will be assigned a value of 255.

3.1 Image segmentation for training dataset

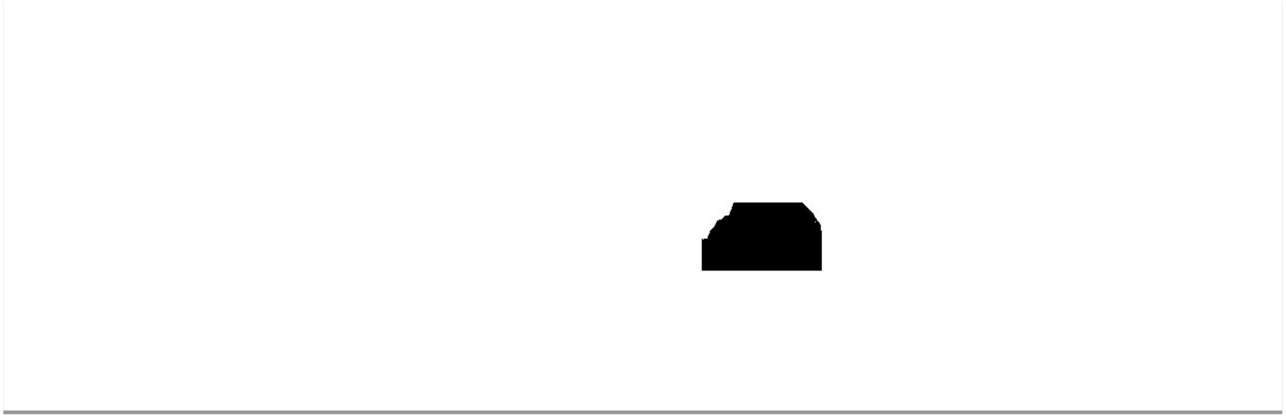

Figure 46: Image segmentation for 000001 (Precision=0.82, Recall=0.95)

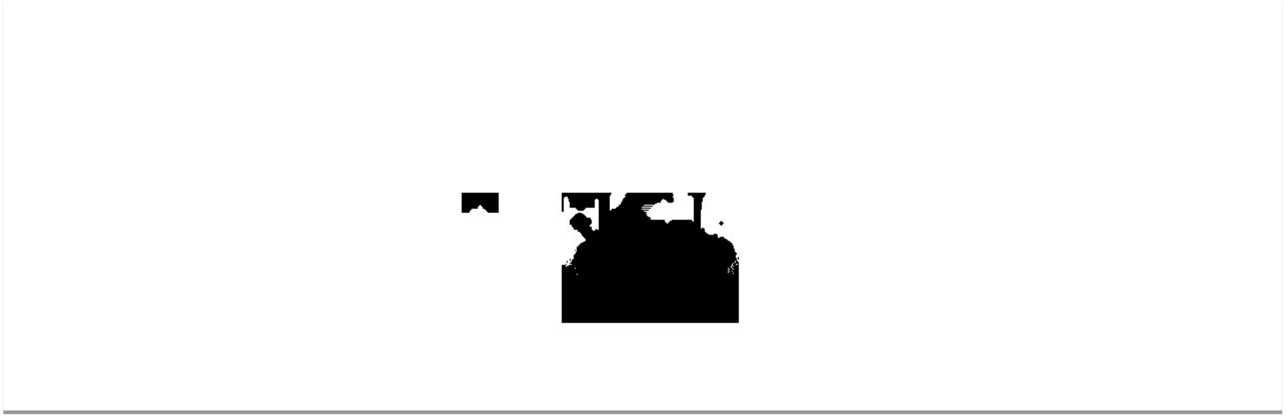

Figure 47: Image segmentation for 000002 (Precision=0.77, Recall=0.87)

Figure 48: Image segmentation for 000003 (Precision=0.78, Recall=0.94)

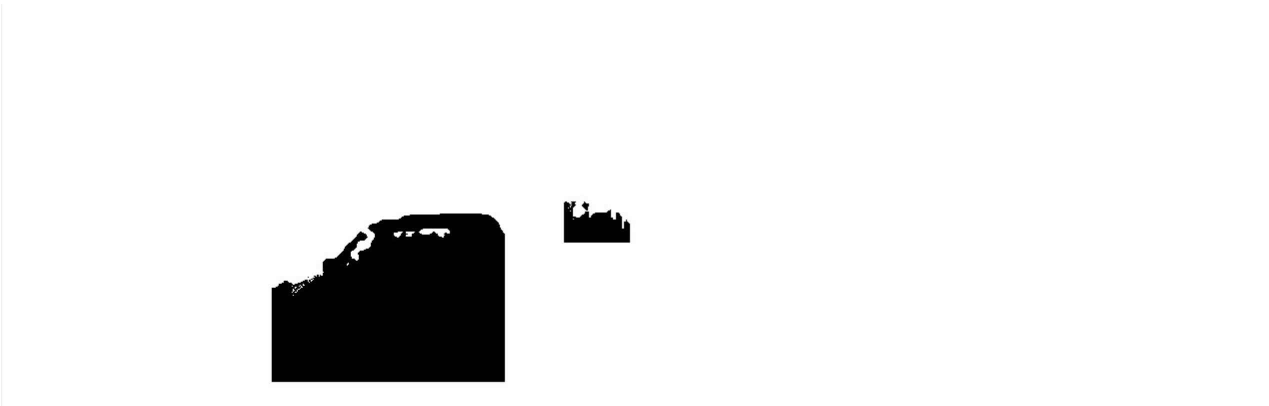

Figure 49: Image segmentation for 000004 (Precision=0.84, Recall=0.93)

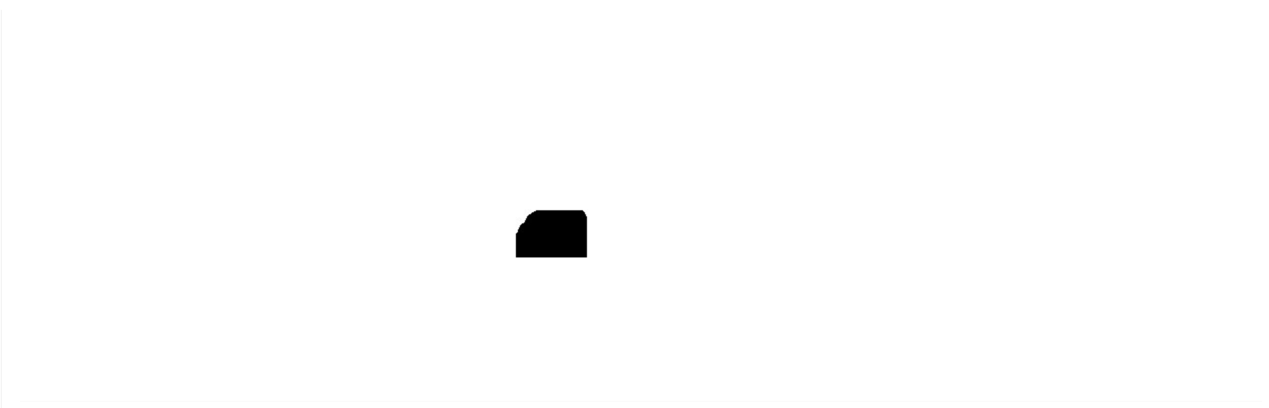

Figure 50: Image segmentation for 000005 (Precision=0.95, Recall=0.69)

Figure 51: Image segmentation for 000006 (Precision=0.82, Recall=0.93)

Figure 52: Image segmentation for 000007 (Precision=0.73, Recall=0.84)

Figure 53: Image segmentation for 000008 (Precision=0.68, Recall=0.84)

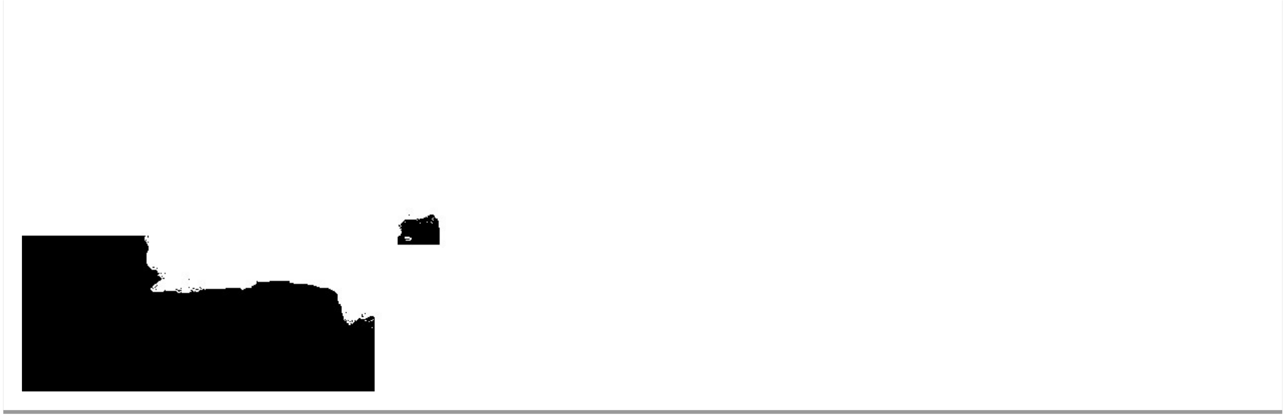

Figure 54: Image segmentation for 000009 (Precision=0.88, Recall=0.85)

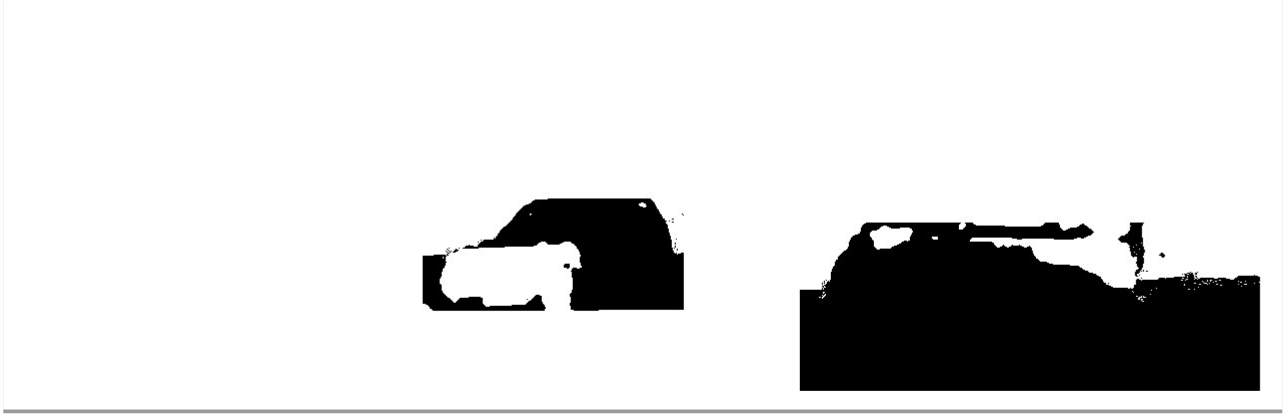

Figure 55: Image segmentation for 000010 (Precision=0.89, Recall=0.83)

The precision and recall values for each image in the training dataset is calculated using the ground truth segmentation map. The formulae for precision and recall are as follows:

- $precision = \frac{TP}{TP+FP}$
- $recall = \frac{TP}{TP+FN}$

Where,

- TP – number of true positives
- FP – number of false positives
- FN – number of false negatives

The respective precision and recall values for each image in the training set is as follows:

Training set image number	**Precision**	**Recall**
000001	0.82	0.95
000002	0.77	0.87

000003	0.78	0.94
000004	0.84	0.93
000005	0.95	0.69
000006	0.82	0.93
000007	0.73	0.84
000008	0.68	0.84
000009	0.88	0.85
000010	0.89	0.83
Average Values	**= 0.82**	**= 0.87**

3.2 Image segmentation for test dataset

Figure 56: Image segmentation for 000011

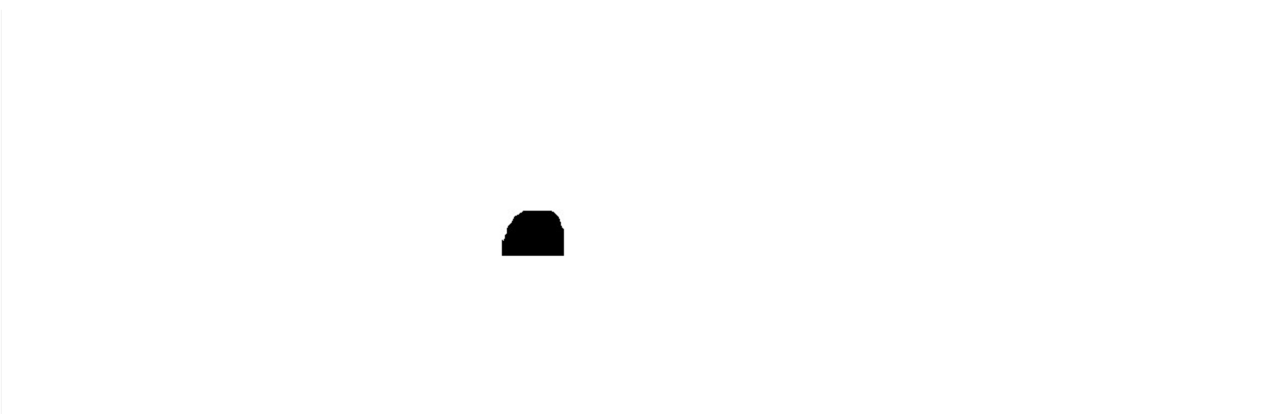

Figure 57: Image segmentation for 000012

Figure 58: Image segmentation for 000013

Figure 59: Image segmentation for 000014

Figure 60: Image segmentation for 000015

For image 000014, it is observed that the car isn’t detected at all in segmentation mask. The reason for this is possibly because the car in the image was really far away and the threshold value for rejecting pixels, rejected all the pixels. This is because all the pixels were close to each other based on distance.