École Polytechnique Fédérale de Lausanne

Milestone 2 - Project 8

3D Object Detection using Monocular Camera

Deep learning for autonomous vehicles (CIVIL-459)

Timothée Hirt
Loïc Von Deschwanden
- Group 12 -

Supervisor:
Ahmad Rahimi

Professor:
Alexandre Alahi

EPFL
May 31, 2023

Table of contents

Network execution

1 Problem definition

As part of the course *deep learning for autonomous vehicles*, the class as a whole creates the visualization interface of an autonomous vehicle. The project is broken down into subsections addressing the different topics. This project focuses on 3D object detection using a monocular camera. The goal is to recognize agent categories like cars, pedestrians and cycles from a RGB monocular camera and identifying their size and location in space with 3D bounding boxes.

2 Dataset

2.1 Dataset description

KITTI is an autonomous driving dataset created in 2012 and updated in 2017 for stereo, optical flow, visual odometry, 3D object detection and 3D tracking. Until the recent growth of the new NuScenes dataset, KITTI has been a reference for the research around autonomous driving.
We chose this dataset over NuScenes for the relevance of the German scene which is representative of the European and Swiss landscape. However, the photographic quality of the NuScenes is undeniable.

2.2 How to get KITTI

The dataset is available at

https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d.

The files can be downloaded after creation of an account. The following files should be downloaded:

- Left color images (label_2)
- Camera calibration matrices (calib)
- Training labels (image_2)

The data must be rearranged following the structure and naming scheme of the bellow section 2.4. All files must be unzipped in their respective directory.

2.3 Samples format

The dataset contains 7481 train images and 7518 test images (without labels provided). Eight different classes are labeled (table 1). The dimension are $1242px \times 375px$. and the format is `.png`. The sky is mainly cropped, only the useful front view is kept.

2.4 File structure

```
-- KITTI/
      -- ImageSets/
            -- test.txt
            -- train.txt
            -- val.txt
      -- data/
            -- calib/
            -- image_2/
            -- label_2/
            -- predicted/
```

The `label_2` folder contains the train images.
The `predicted` folder is automatically created during inference.
The `ImageSets` folder contains the samples split indexes. Those files are given in the GitHub.

2.5 Label format

Each image sample is paired with a `.txt` file containing the description of the objects in the scene. Each object is represented by a line. The values are described in table 1.

#Values	Name	Description
1	type	Describes the type of object: ’Car’, ’Van’, ’Truck’, ’Pedestrian’, ’Person_sitting’, ’Cyclist’, ’Tram’, ’Misc’ or ’DontCare’
1	truncated	Float from 0 (non-truncated) to 1 (truncated), where truncated refers to the object leaving image boundaries
1	occluded	Integer (0,1,2,3) indicating occlusion state: 0 = fully visible, 1 = partly occluded 2 = largely occluded, 3 = unknown
1	alpha	Observation angle of object, ranging [-pi..pi]
4	bbox	2D bounding box of object in the image (0-based index): contains left, top, right, bottom pixel coordinates
3	dimensions	3D object dimensions: height, width, length (in meters)
3	location	3D object location x,y,z in camera coordinates (in meters)
1	rotation_y	Rotation ry around Y-axis in camera coordinates [-pi..pi]
1	score	Only for results: Float, indicating confidence in detection, needed for p/r curves, higher is better.

Table 1: Label format

2.6 Label indexes files

The samples of the dataset are numbered in random order, mixing all video sequences. The KITTI website also provide informations to get the samples belonging to the sequence. We created the `train.txt`, `test.txt` and `val.txt` index files by dividing by sequence with ratios 0.6 - 0.2 - 0.2 respectively.

3 CONTRIBUTION

3.1 Curriculum learning

This contribution leverage the classification of KITTI's objects into three categories: easy, moderate and hard, depending on the occlusion level, the truncation and the pixel size. From that we apply a curriculum learning during the training epochs. First only the easy objects are trained for a few epochs, then only the moderates ones, then only the hard ones and finally all of them are used for the remaining epochs.

To do so we implemented the curriculum in the code and did a grid search on 40 epochs with different number of epochs thresholds on the easy, medium and hard labels and looked at the effect on the losses (see fig.1 and tab.2).

Table 2: configuration of the curriculum for the different tests

test nb.	easy	moderate	hard
test 1	3	6	9
test 2	5	10	15
test 3	8	16	24
test 4	10	20	30
test 5	10	15	20
test 6	5	10	20
test 7	0	0	0

All these tests had similar shaped losses with varying accuracies. We chose to continue with the third test because it has one of the best descending curve on the heading loss without going back up too much on the 2d offset loss, as well as encouraging accuracy results.

The second step was to test the curriculum against the warm-up technique already implemented. Warm-up is a strategy to start the training with a very low learning-rate and to rapidly increase its value over the five first epochs. This prevent a quick overfit of the initial data. Therefore two test were performed.

The first one was to train the curriculum and then restart the training from scratch with the whole dataset (140 epochs) using the weight obtained after the hard level epochs, which means that the warm-up will be used again and the learning rate restart.

The second one was to to continue the training after the curriculum for 140 epochs without resetting the learning rate. As we expected, we obtained better results for the second one, which can be because the warm-up learning rate destroy what has been learned by the curriculum.

We didn't achieve better results for all the categories. The cyclists and pedestrians are better (+1.3% easy, +0.9% moderate & hard) but not the cars (-0.9% easy, -0.5% moderate & hard).

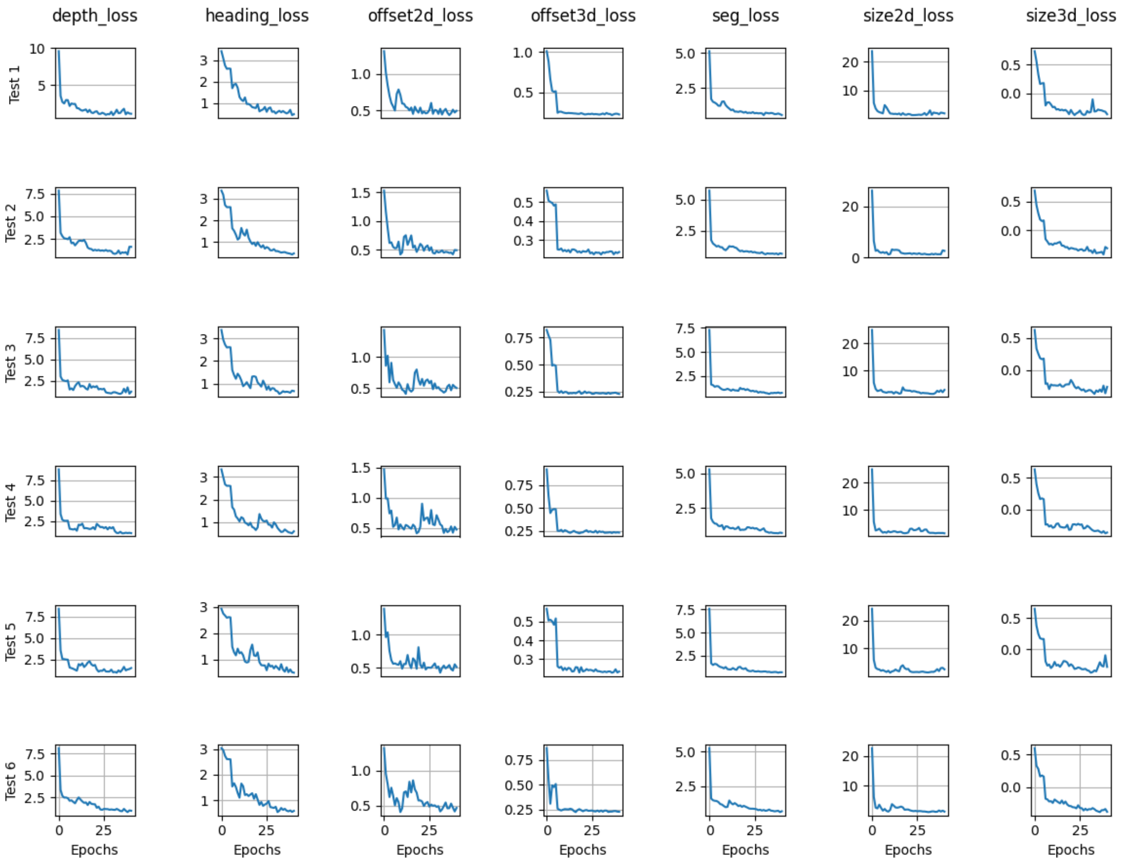

Figure 1: plot of the losses for different curriculum configuration

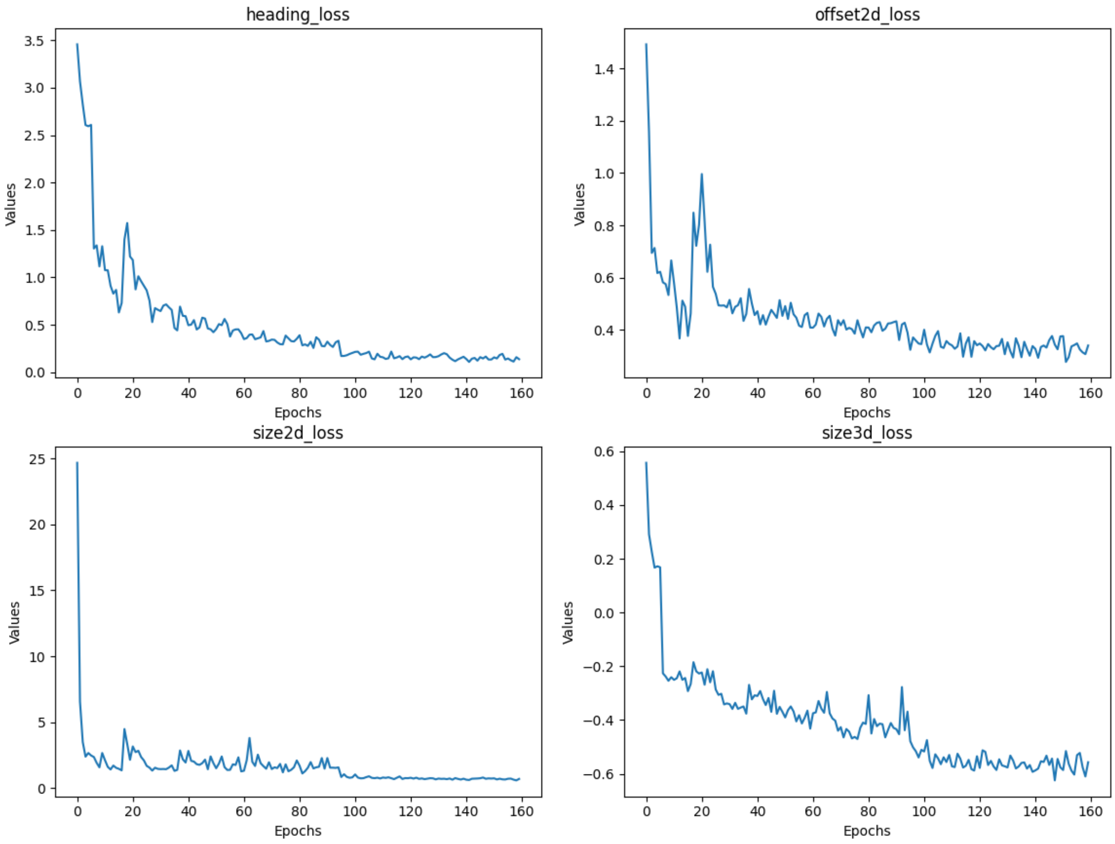

Figure 2: plot of some losses for best curriculum threshold [8,16,24]

3.2 Multi-class detection

The GUPnet source code detect the cars, pedestrians and cyclists, but the KITTI dataset provide labels for other items as trucks, vans, trams and person sitting (why not). We decided to improve the code by adding all these other items for the detection. Unfortunately after training the code to detect all the labeled items the model couldn't detect more than cars, pedestrians and cyclists. We supposed that it is due to the fact that these other items appears much less in the dataset and are harder to detect or differentiate from cars (like vans). However it improved the detection of vans and trucks but it labeled them as cars as shown on fig.7.

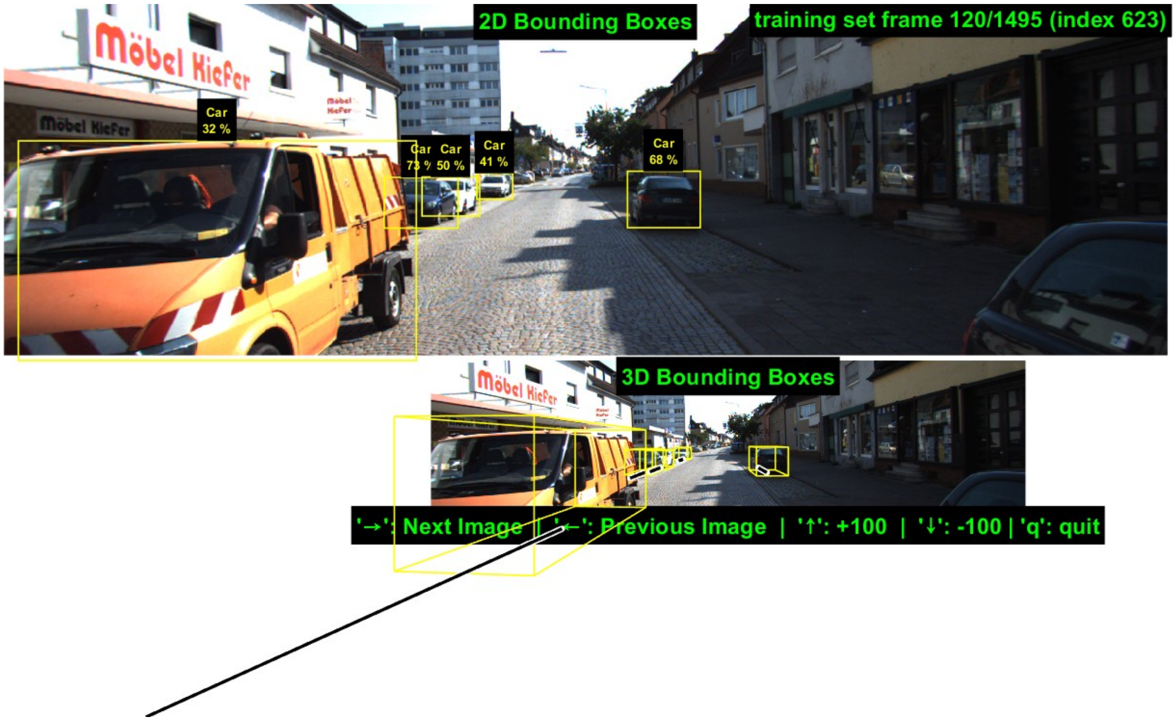
2D Bounding Boxes
training set frame 120/1495 (index 623)
Car 32 %
Car 73 %
Car 50 %
Car 41 %
Car 68 %
3D Bounding Boxes
'→': Next Image | '←': Previous Image | '↑': +100 | '↓': -100 | 'q': quit

Figure 3: example of van detected as car

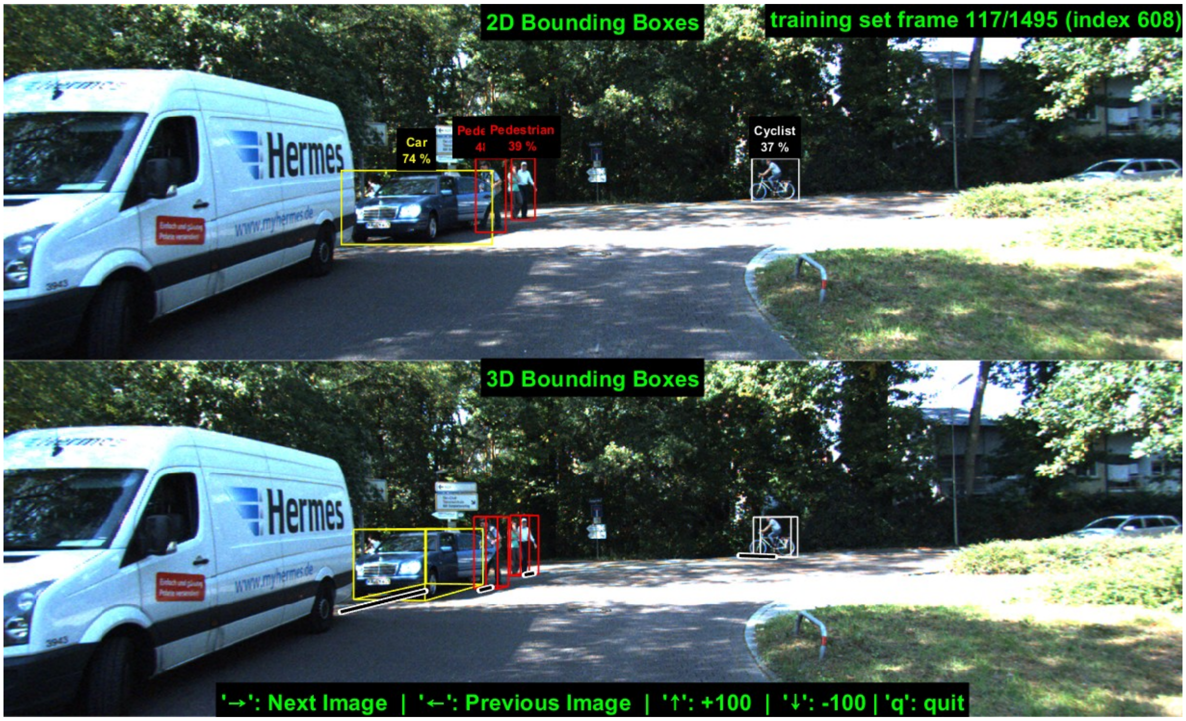
2D Bounding Boxes
training set frame 117/1495 (index 608)
Car
74 %
Pede
4(
Pedestrian
39 %
Cyclist
37 %
3D Bounding Boxes
'→': Next Image | '←': Previous Image | '↑': +100 | '↓': -100 | 'q': quit

Figure 4

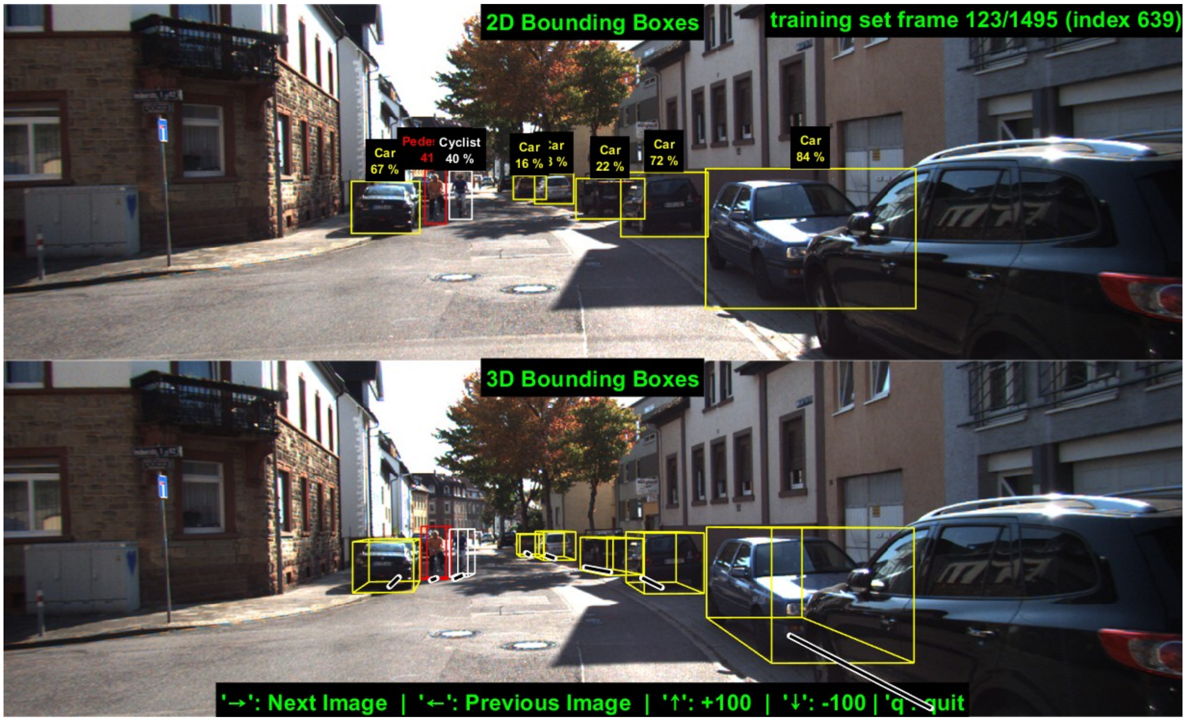
2D Bounding Boxes
training set frame 123/1495 (index 639)
Car
67 %
Pede
41
Cyclist
40 %
Car
16 %
ar
3 %
Car
22 %
Car
72 %
Car
84 %
3D Bounding Boxes
'→': Next Image | '←': Previous Image | '↑': +100 | '↓': -100 | 'q': quit

Figure 5

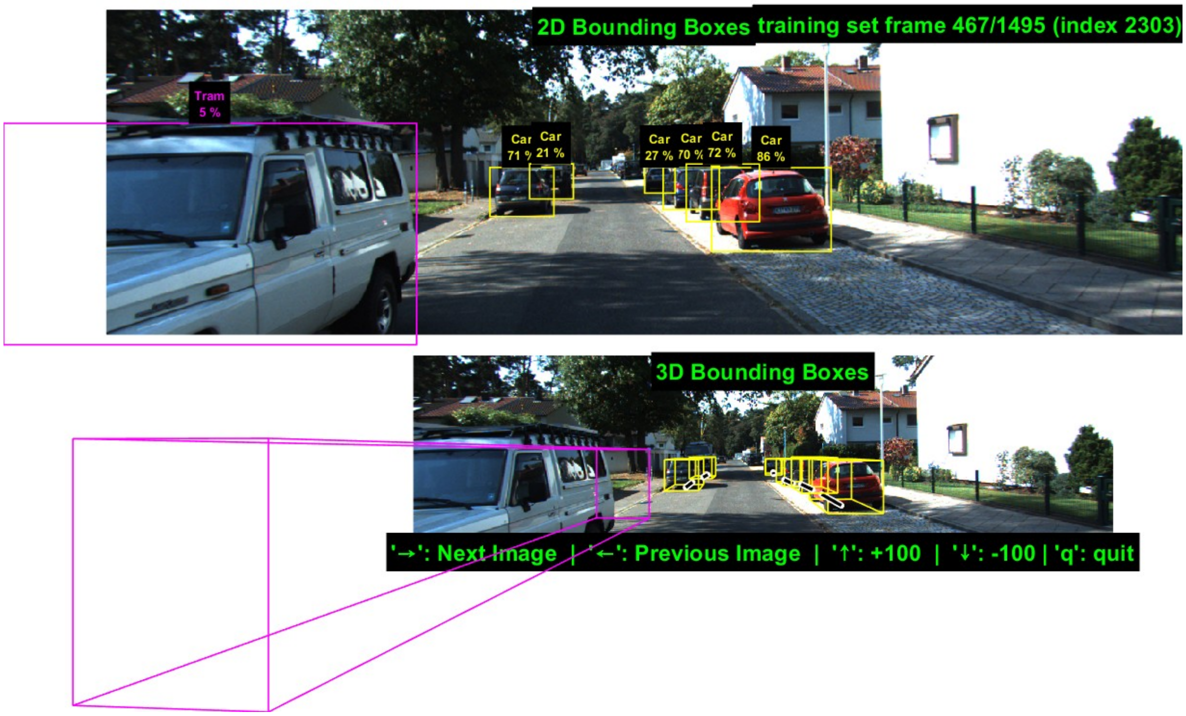
2D Bounding Boxes
training set frame 467/1495 (index 2303)
Tram
5 %
Car
71 %
Car
21 %
Car
27 %
Car
70 %
Car
72 %
Car
86 %
3D Bounding Boxes
'→': Next Image | '←': Previous Image | '↑': +100 | '↓': -100 | 'q': quit

Figure 6

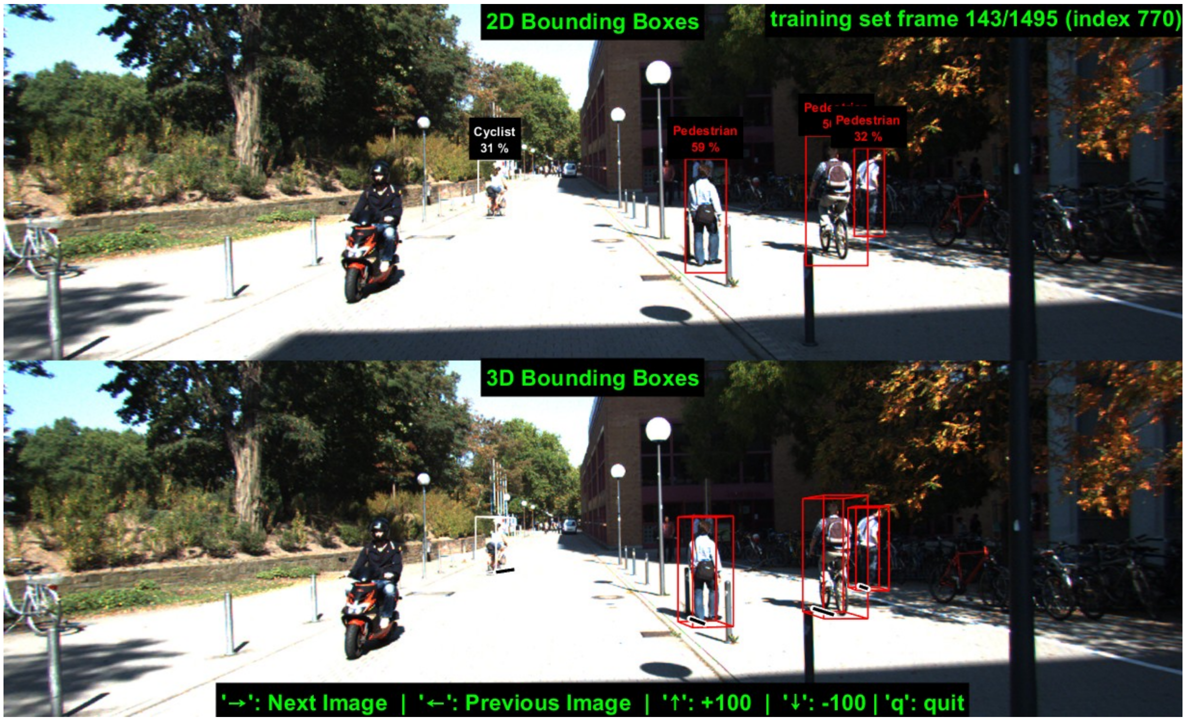
2D Bounding Boxes
training set frame 143/1495 (index 770)
Cyclist
31 %
Pedestrian
59 %
Pedestrian
32 %
3D Bounding Boxes
'→': Next Image | '←': Previous Image | '↑': +100 | '↓': -100 | 'q': quit

Figure 7