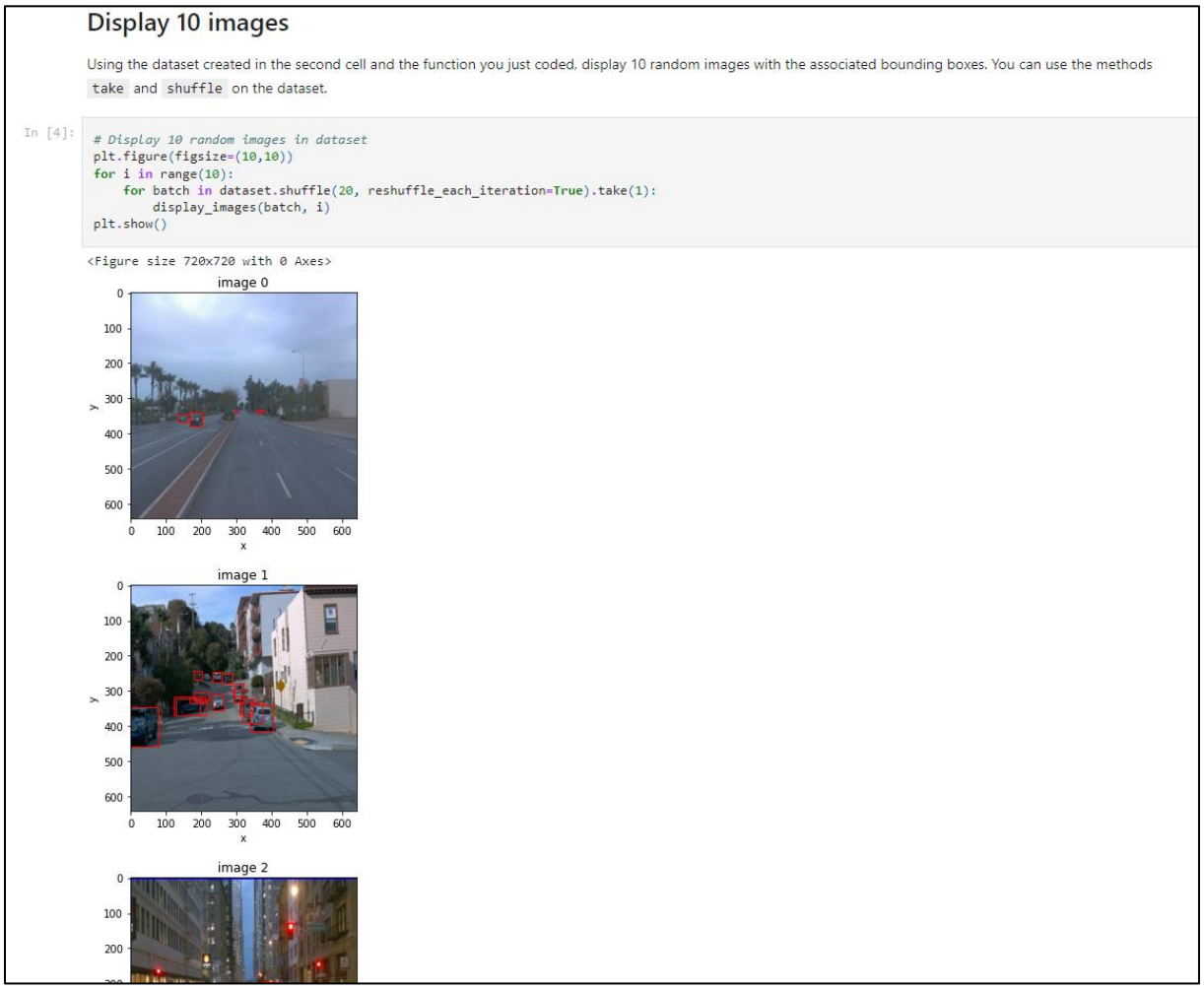
data.camp527@audi.de

Final project: Object Detection in an Urban Environment

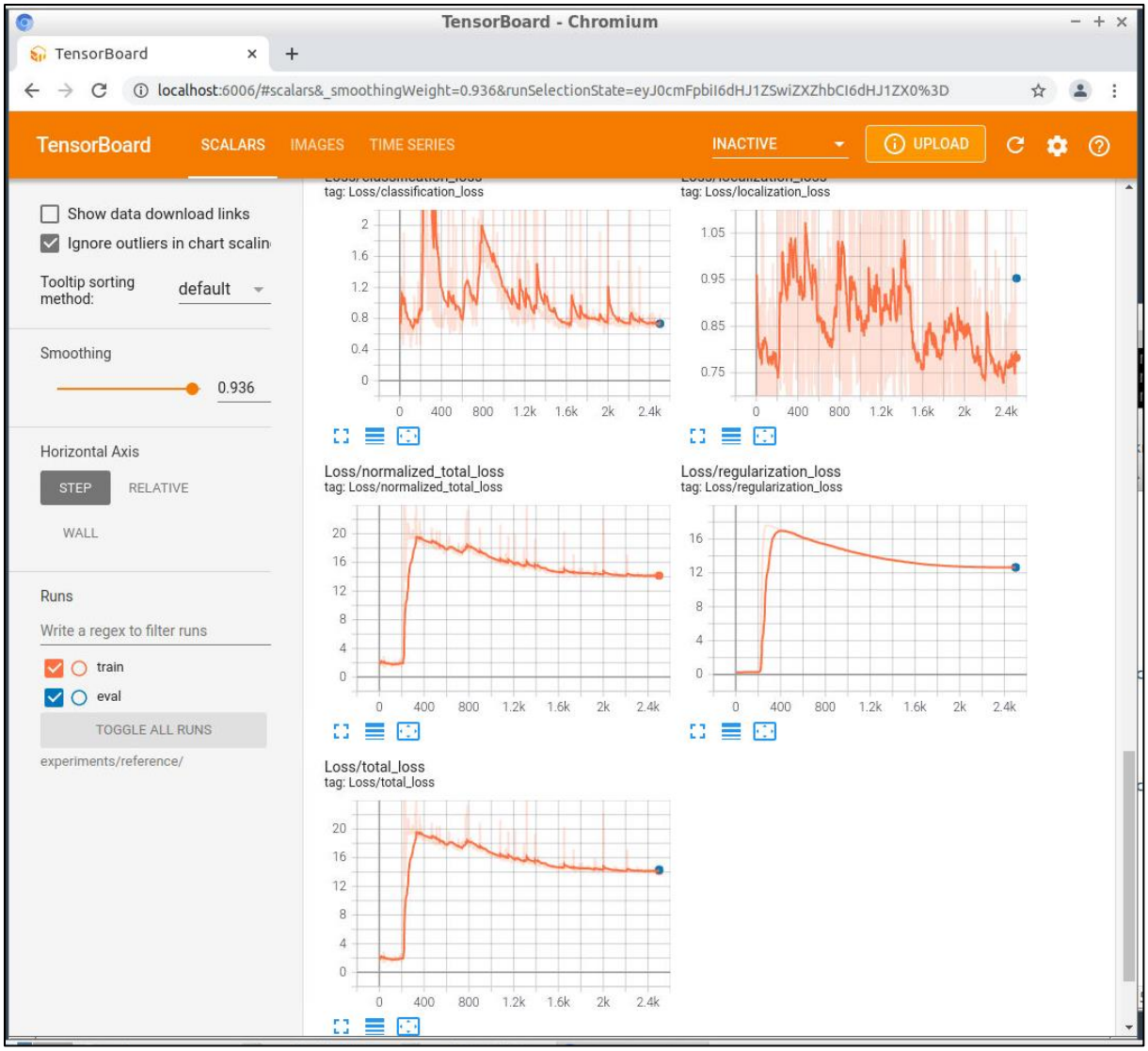
Step 1 – Exploratory Data Analysis

On my Github repository you can find my result for Exploratory Data Analysis.ipynb. It successfully displays 10 randomly chosen images from the training folder with the ground truth bounding boxes in separate colors according to the label classes.



Step 3 – Model Training and Evaluation

Here's the Tensorboard chart from the first training period:

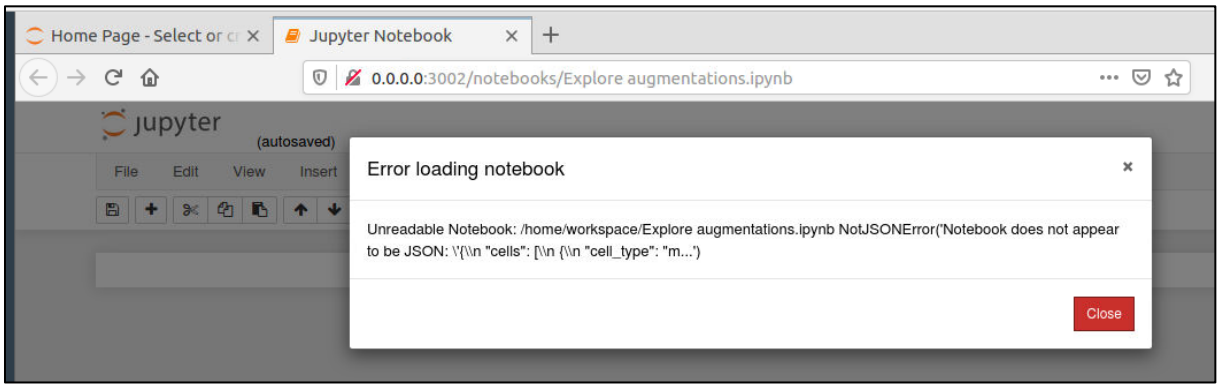


It shows that the loss is converging but at a very high value of about 14. The evaluation shows no sign of over- or underfitting, as the blue dot lies roughly on the orange graph.

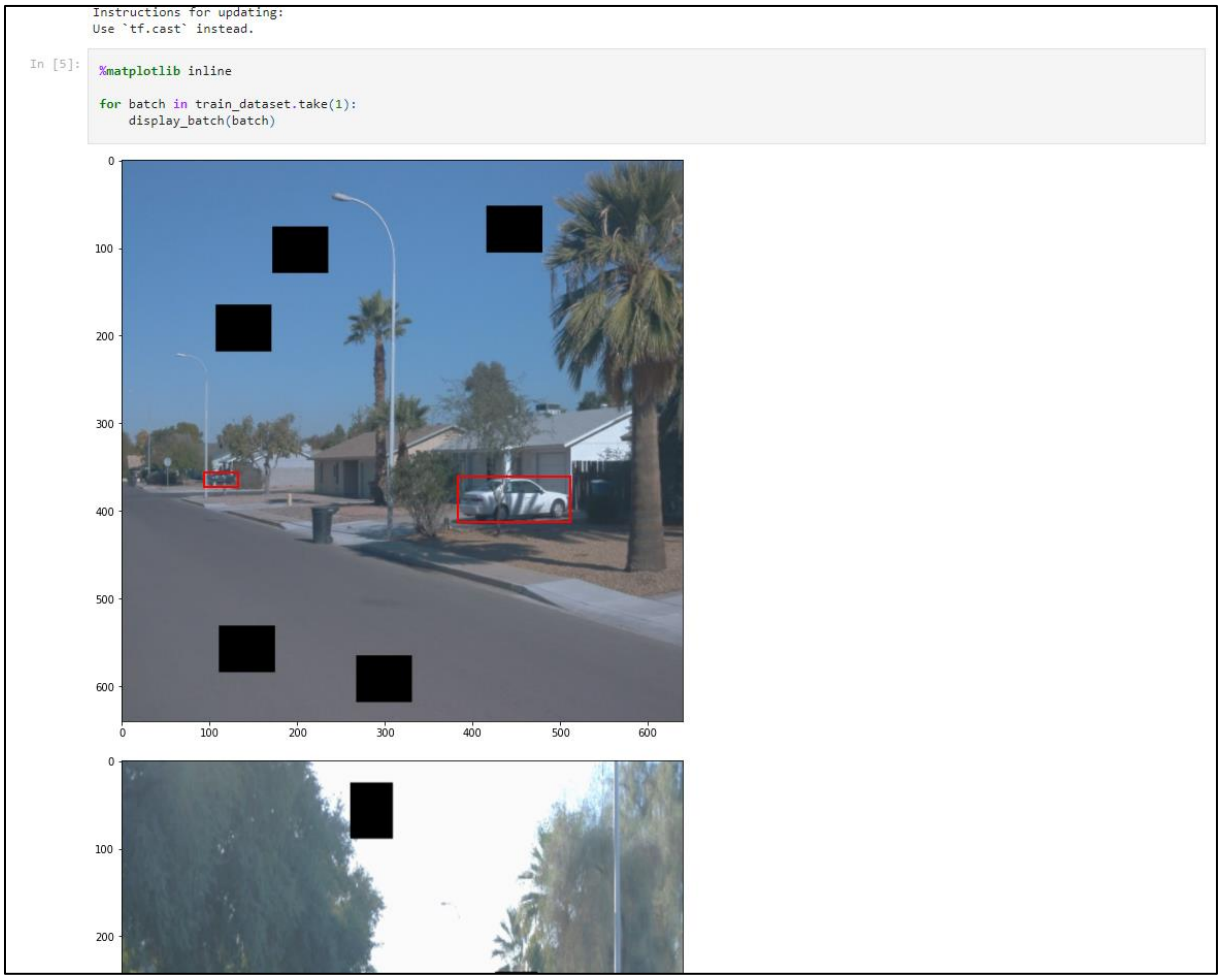
When I was running the animation, I randomly picked the night scenery and I thought I did completely wrong, because there was not asingle detection.

INTERNAL

Afterwards, I changed the data_augmentation_options in “pipeline_new.config” (also provided in the repository). The Explore augmentations.ipynb didn’t work in the beginning, I always received a certain error:



So, I started improving without that notebook. After the improvement step, I found a version of the notebook without that bug when I forked the repository, so I explored the augmentations afterwards:



For the improvement I chose certain additional augmentations of that I felt they were helpful:

- Random Image Scale
- Random Adjust Brightness
- Random Adjust contrast
- Random jitter boxes
- Random Black patches

In the first attempt all in default settings. The reasons I picked those were mainly to improve variety of training objects (Scale) and to improve performance for different weather conditions (brightness and contrast). To improve performance for occluded objects I chose Black patches.

Others I didn't find useful like flipping vertical or rotating 90° because it's not a natural case. It could help to prevent some emergency situations, like cars lying on their roof or people lying on the ground. But for the scenarios in our course, I found that too specific.

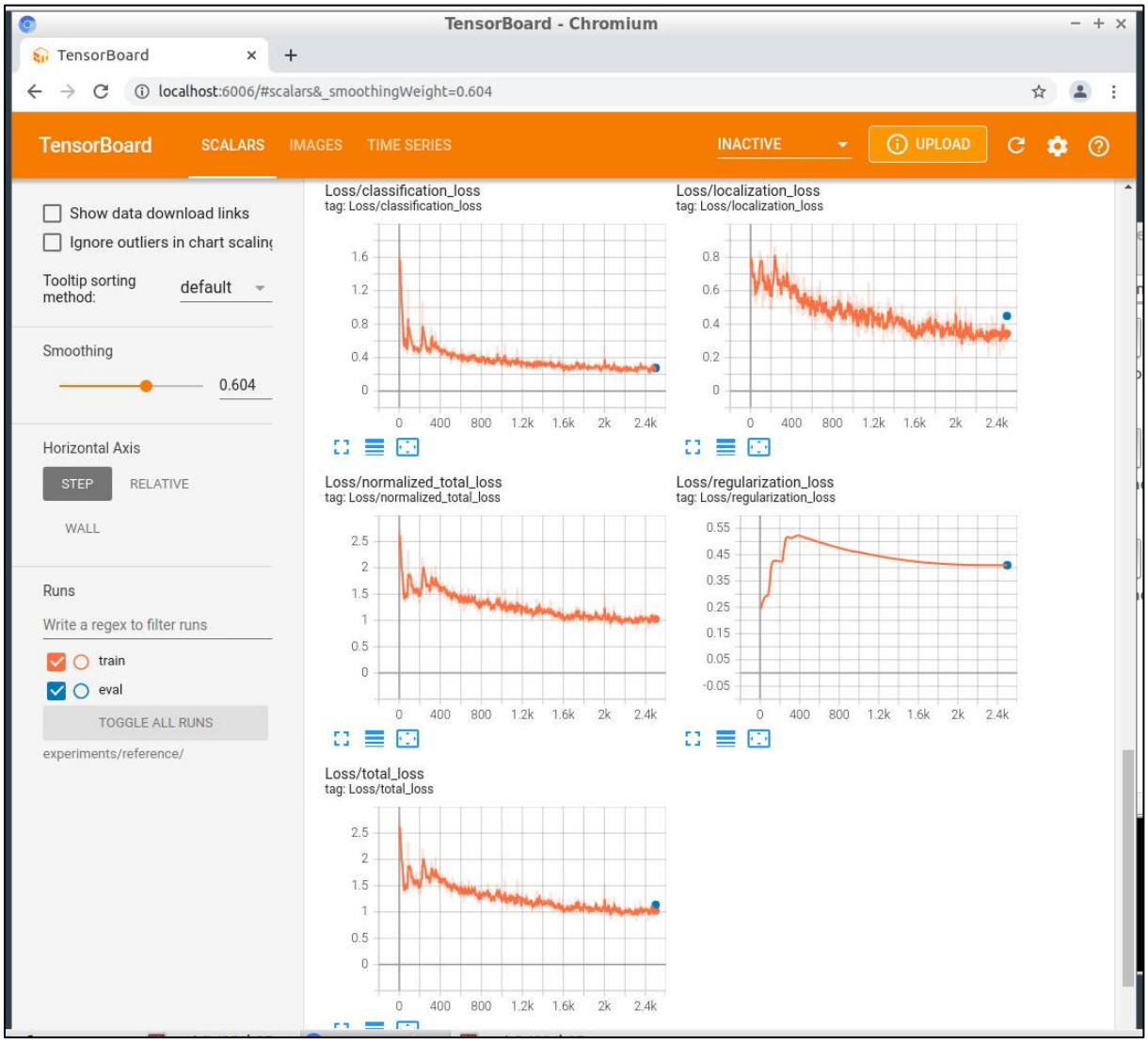
As already written, pipeline_new.config" is also part of the repository.

```
129  }
130  train_config {
131    batch_size: 7
132    data_augmentation_options {
133      random_horizontal_flip {
134      }
135    }
136    data_augmentation_options {
137      random_crop_image {
138        min_object_covered: 0.0
139        min_aspect_ratio: 0.75
140        max_aspect_ratio: 3.0
141        min_area: 0.75
142        max_area: 1.0
143        overlap_thresh: 0.0
144      }
145    }
146    data_augmentation_options {
147      random_image_scale{
148      }
149    }
150    data_augmentation_options {
151      random_adjust_brightness{
152      }
153    }
154    data_augmentation_options {
155      random_adjust_contrast {
156      }
157    }
158    data_augmentation_options {
159      random_jitter_boxes {
160      }
161    }
162    data_augmentation_options {
163      random_black_patches {
164      }
165    }
166    sync_replicas: true
167    optimizer {
```

INTERNAL

The result of my second training was surprisingly good. That's why I didn't do any further improvement loops. The Training time raised naturally (1:30h) but it was still not too extreme.

My final Tensorboard chart looked like that:



The total loss was converging towards 1.0, so a lot better than in the first loop. Also, the evaluation (blue dot) showed a loss of about the same dimension. So, there's hopefully no over- or underfitting.

The animations ran pretty well although the classification of the objects is displayed with a lower accuracy than I expected (mostly around 30%). In the night scenery there was a high amount of false negative detections. Shows my just that it's not so easy and that a lot more training data is needed.

You can find my animations also in the repository. But I had to reduce the size of the gifs through a limitation to 25MB from github. Unfortunately, one of the three animations was still too big.

Kind regards,

Chris

INTERNAL