Object Detection in an Urban Environment

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Congratulations! You've done a splendid job on this project.
You've explored the data thoroughly. You've created a new config file and got some decent results. Thanks for sharing the results in the writeup file. You also provided some future improvements. Awesome!
Keep working hard and stay motivated. Good luck for your next projects.

Additional resources:
SSD is not the only single-stage encoder network. YOLO is another great single-stage network that is very popular in the automotive industry.

SSD is a good approach to balance between FPS and mAP performance. Whereas, YOLO focuses more on FPS performance
providing a better real-time method. You can find a good introduction to YOLO in the following blog or you can read the actual paper: You Only Look Once(Uni ed, Real-Time Object Detection) that introduced the idea for the first time.

Github and Code Quality

✓ All the code generated for this project lives in a Github repository and a link to this repository has been provided to the reviewer.

The student made at least five commits to this repository.

All the code generated for this project lives in this Github repository.
You also commit the code with a meaningful message, commit often, and single-purpose commit.

✓ All the code written for this project should follow the PEP 8 guidelines. Objects have meaningful names and syntax. Code is properly commented and organized. Imports are correctly ordered.

✅ Code written for this project mostly follows the PEP 8 guidelines.
✅ Objects have meaningful names and syntax.
✅ Code is properly commented and organized.
✅ Imports are correctly ordered.
Well done! pycodestyle is a good tool to check the code style.

✓ The project contains either a requirements.txt file or a Dockerfile. The instructions to install the dependencies are clear. The file contains a summary, a build section as well as a detailed description of the different functions and files. Someone should be able to run the code by reading the README.

✅ The project contains a requirements.txt file and a Dockerfile.
✅ The instructions to install the dependencies are clear.
✅ The file contains a summary, a build section as well as a detailed description of the different functions and files.

Well done! This course contains great examples to create a README.

Exploratory Data Analysis

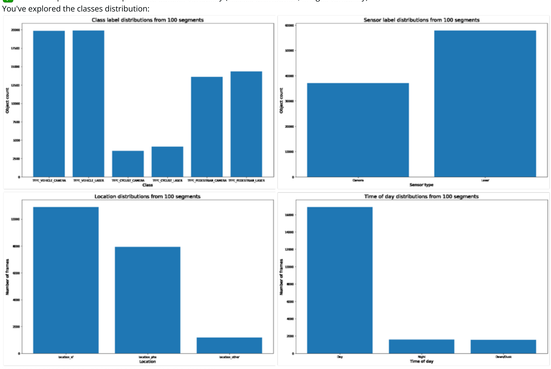
✓ The figures should display all the meaningful information. They are big enough, it is easy to grasp the message that the figure is conveying. When plotting charts, the axis should be labeled and the figure should be named. When plotting bounding boxes, the different classes should be color coded.

✅ The figures display all the meaningful information. They are big enough, it is easy to grasp the message that the figure is conveying.
✅ When plotting charts, the axis are labeled and the figure is named. When plotting bounding boxes, the different classes are color coded.

You are encouraged to display the class id next to the bounding boxes as presented in the following example!

✓ The write-up and the code capture the dataset variability (classes distribution, images variability).

✅ The write-up and the code capture the dataset variability (classes distribution, images variability).
You've explored the classes distribution:



You also looked at enough images to describe the dataset:

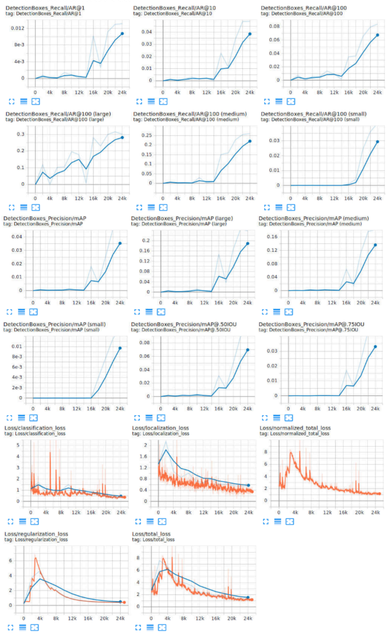
- Environmental: freeway, city, suburb, etc.
- Weather: day/night/foggy etc

Feel free to look at more images.

Training

✓ The write up contains a screenshot of the different Tensorboard charts and the write up describes these charts. For example, how does the validation loss compare to the training loss? Did you expect such behavior from the losses / metrics?

✅ The write up contains a screenshot of the different Tensorboard charts and the write up describes these charts.
Your Tensorboard statistics are clear:



Model Improvements

✓ A new version of the config file is created and contains modifications to improve the model performances. A new config file is created with meaningful modifications.

The write up details why these modifications were made. New augmentations are visualized and displayed in the writeup.

✅ A new version of the config file is created and contains modifications to improve the model performances.
✅ A new config file is created with meaningful modifications.
✅ The write up details why these modifications were made.
✅ New augmentations are visualized and displayed in the writeup.

Great job creating a new config file and add data augumentations like `RandomCropImage`.

✓ The write up demonstrates that the student investigated at least two modifications of the config that were not discussed in the course and referenced the corresponding Tf Object Detection API code documentation.

DOWNLOAD PROJECT

RETURN TO PATH