UDACITY

‹ Return to Classroom

Object Detection in an Urban Environment

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Added /Installation.ipynb. The training and evaluation was done on colab.

Great work Udacian! ⭐

Congratulations on passing the Object detection in an urban environment project by meeting all the required specifications. 🎉 🎉

It was nice going through your work, you have implemented required solutions nicely to analyze and explore the dataset, deriving required visualizations. The dataset variability has been calculated in one of your notebooks where you have plotted pie charts with different variations like time of day, location etc. 👌

Great work in performing different experiments and applying augmentations to base pipeline configuration, good job in sharing your observations as well in the write-up. 👏

Through this project, you have built a foundation knowledge for different challenges that may come up while working in this field and also learnt tools and techniques needed to apply experimentation to your model. ✌️

I wish you good luck for your future projects.

P.S. In this project you have developed an Object detection based on SSD. Remember that single-stage architectures

are preferred for autonomous vehicles because they provide a higher frame rate than other architectures (e.g. Dual-stage encoders such as R-CNN).
However, SSD is not the only single-stage encoder network. YOLO is another great single-stage network that is very popular in the automotive industry. For instance, you can see a comparison of both in the following paper from a couple of years ago. A relevant part of this paper is the comparison chart between YOLO and SSD.

Thus, in general, 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 here.

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.

Good work here!

✅The source code for this project is within a Github repository.
✅ The link to the Github repository was provided: https://github.com/pavanp5/2d_object_detection.
✅ Commits have meaningful messages.- There are total 35 commits.
✅ Separate folders are used to organize files. - Code is organized in separate build , experiment folders.

Note:

A very good tutorial on git control can be found in Udacity's free courses here.

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.

✅ The syntax/names are meaningful with respect to the context.
✅Code is properly commented with required imports in each of these files.
✅All imports are in order

References

Sharing few references for you to understand PEP 8 guidelines:

- Overview of PEP style - This gives overview of PEP styleguide.
- Pep.org - This is the official link to pep8 organisation website.

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.

✅Write-up is holding required information to run the solution and instructions are clearly written. You have prepared a detailed write-up and it reflects the work being done. ✌️

References

I am sharing few guidelines references for you to create a good README:

- README tutorial - Udacity's free course link which is very effective to learn about writing readmes.
- README best practices - A very nice document to know best practices of writing readme.

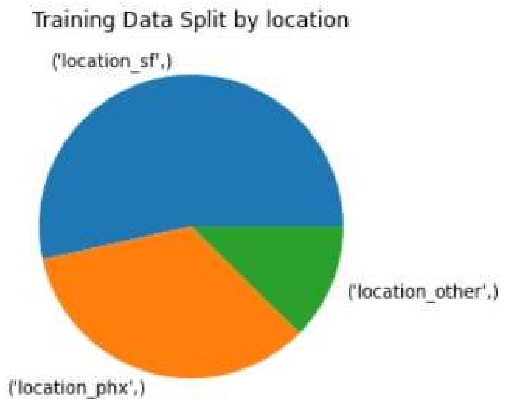
Exploratory Data Analysis

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.

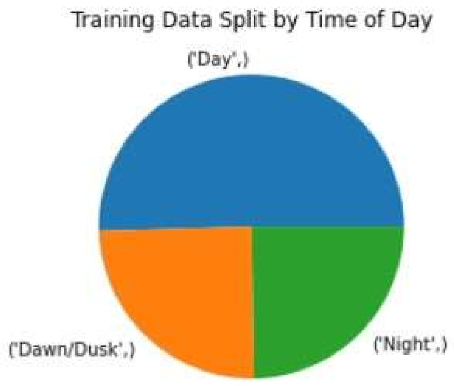
✔ Images size is big enough to visualize the data.
✔ Charts' axis should be labeled and the figure should be named.
✔ The bounding boxes must have different colors for classes (e.g Car=Red, Cyclist=Green, Pedestrian=Blue).

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

Well done, your write-up and code shows the distribution graphs generated with dataset classes and distribution. ✌️



In [47]:
```
df_plot(df_train_meta,'TimeofDay','Training Data Split by Time of Day')
```



I am sharing a very nice paper on multispectral object detection for your learning purpose.

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?

Nice work in adding the tensorboard images and comments in the `writeup.md`, you mentioned:

✓ The training and validation losses are decreasing.
✓ The overall mAP increases, but it remains low.
✓ The model is overfitting.

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.

Nice job in implementing several meaningful augmentations, you justified the reason for selecting each augmentation approach in the `writeup.md`. The `Augmentations.ipynb` contains the evidence of the augmentations applied to several sample images.

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.

Congratulations on investigating at least two meaningful augmentations and providing a good explanation of them. Data Augmentation techniques allow us to create more sample images (synthetic data) from existing images, thus it acts as a regularizer and helps to reduce the overfitting of the model.

DOWNLOAD PROJECT

RETURN TO PATH