EECS 581/582 Final Project

Team 18

Project Artifacts

The following diagram shows what the flow of our project would look like:

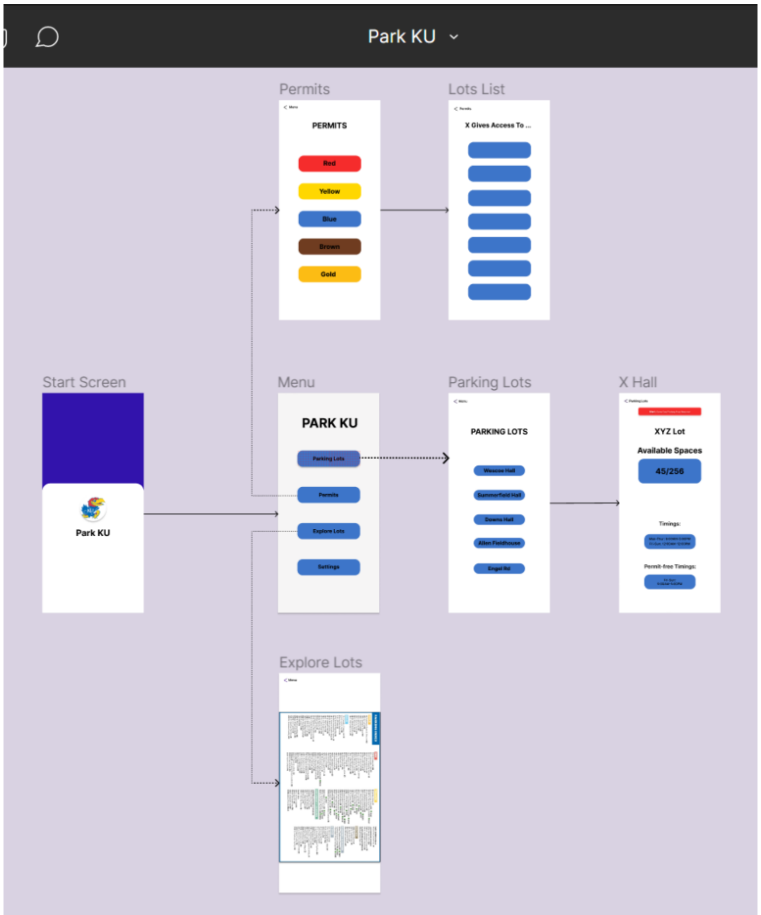


Fig5: The flow of our project; ParkKU

The tentative flow of our app can be viewed on the following link:

https://github.com/hzahid99/EECS-Final-Project/blob/main/Demo%20of%20the%20app.mp4

The following GitHub repository holds our progress so far:

https://github.com/hzahid99/EECS-Final-Project

Below are some of the figures that show how we are training our machine learning component of the app to detect cars. This will enable us to increment/decrement the cars as they enter/leave the lot.

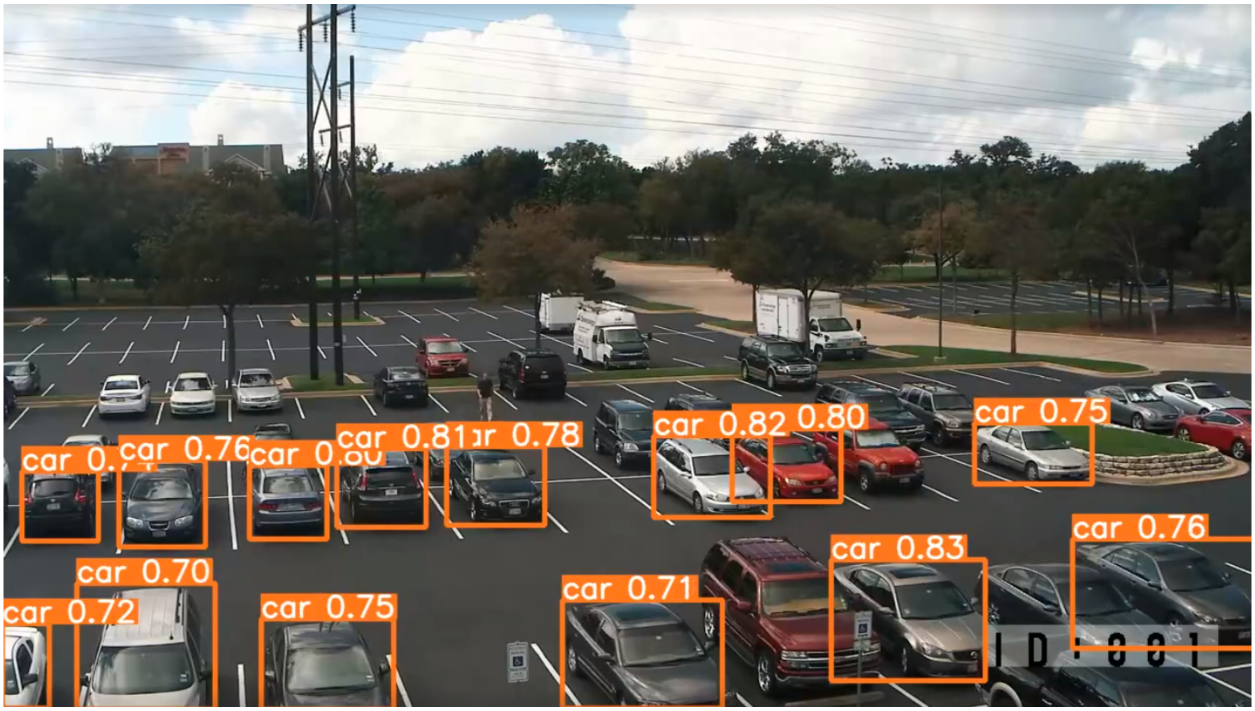
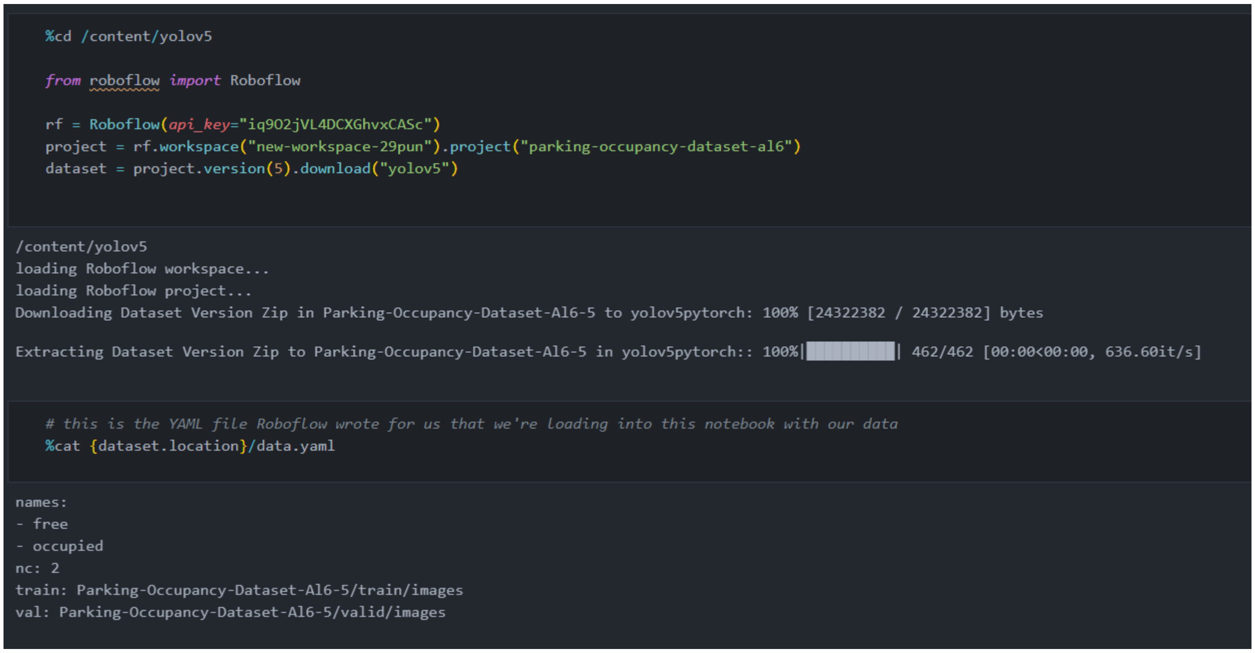


Fig1: Detection of cars



```
    %cd /content/yolov5

    from roboflow import Roboflow

    rf = Roboflow(api_key="iq9O2jVL4DCXGhvxCASc")
    project = rf.workspace("new-workspace-29pun").project("parking-occupancy-dataset-al6")
    dataset = project.version(5).download("yolov5")
```

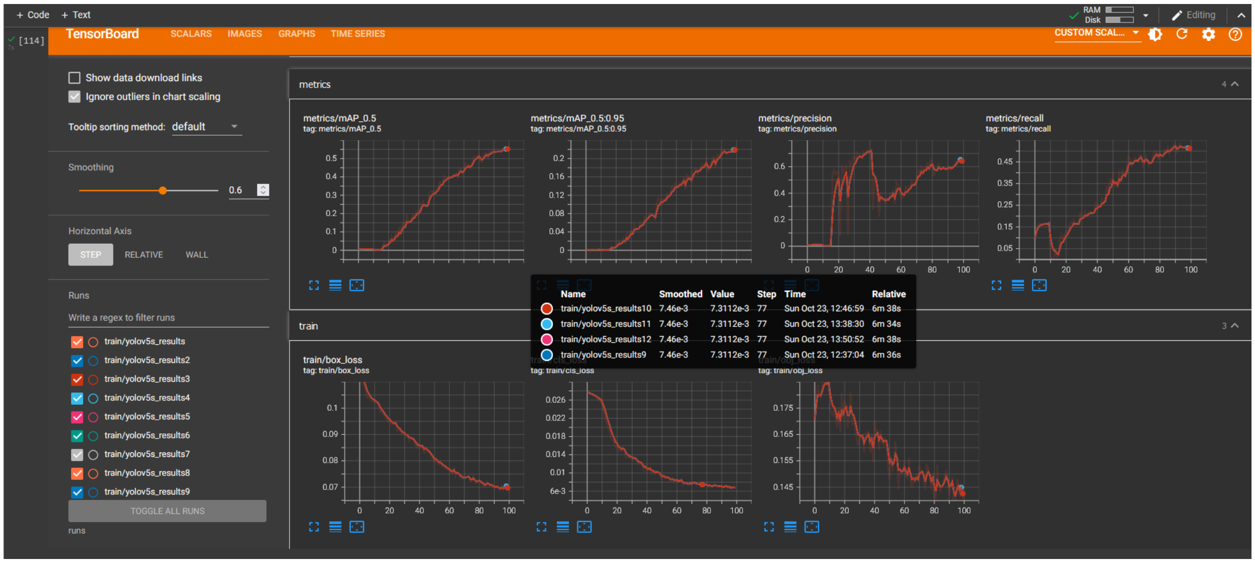
```
/content/yolov5
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Parking-Occupancy-Dataset-Al6-5 to yolov5pytorch: 100% [24322382 / 24322382] bytes

Extracting Dataset Version Zip to Parking-Occupancy-Dataset-Al6-5 in yolov5pytorch:: 100%|██████████| 462/462 [00:00<00:00, 636.60it/s]
```

```
    # this is the YAML file Roboflow wrote for us that we're loading into this notebook with our data
    %cat {dataset.location}/data.yaml
```

```
names:
- free
- occupied
nc: 2
train: Parking-Occupancy-Dataset-Al6-5/train/images
val: Parking-Occupancy-Dataset-Al6-5/valid/images
```

Fig2: Loading dataset and checking the classes



□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

- ☑ train/yolov5s_results
- ☑ train/yolov5s_results2
- ☑ train/yolov5s_results3
- ☑ train/yolov5s_results4
- ☑ train/yolov5s_results5
- ☑ train/yolov5s_results6
- ☑ train/yolov5s_results7
- ☑ train/yolov5s_results8
- ☑ train/yolov5s_results9

TOGGLE ALL RUNS

runs

	Name	Smoothed	Value	Step	Time	Relative
●	train/yolov5s_results10	7.46e-3	7.3112e-3	77	Sun Oct 23, 12:46:59	6m 38s
●	train/yolov5s_results11	7.46e-3	7.3112e-3	77	Sun Oct 23, 13:38:30	6m 34s
●	train/yolov5s_results12	7.46e-3	7.3112e-3	77	Sun Oct 23, 13:50:52	6m 38s
●	train/yolov5s_results9	7.46e-3	7.3112e-3	77	Sun Oct 23, 12:37:04	6m 36s

Fig3: Metrics on the model

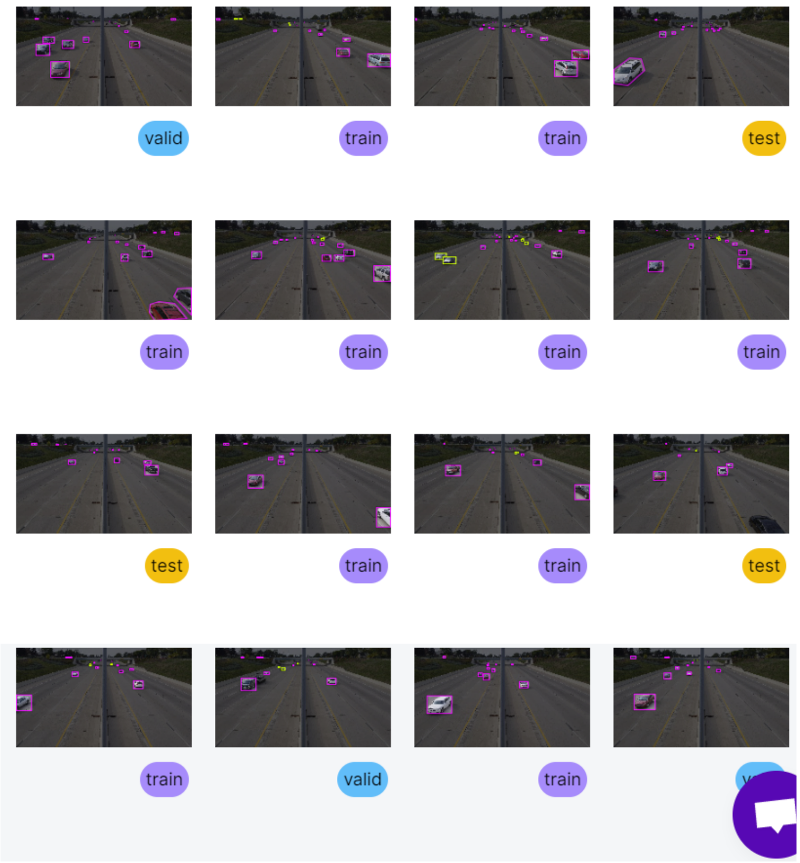


Fig4: Example of Images we used to manually annotate cars on Roboflow

Following are the screenshots of the screens we have made so far for the app:

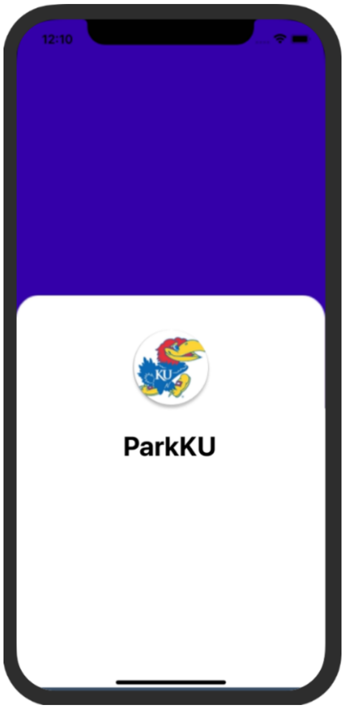
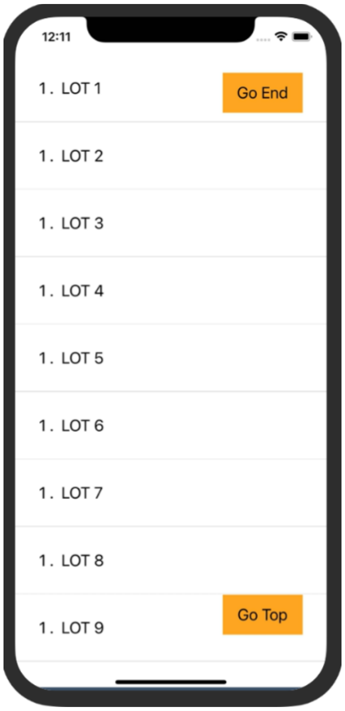


Fig5: Start Screen of the app



1. LOT 1

1. LOT 2

1. LOT 3

1. LOT 4

1. LOT 5

1. LOT 6

1. LOT 7

1. LOT 8

1. LOT 9

Fig6: Screen that will list down all the lots and permits they require

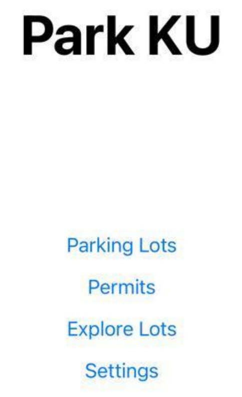


Fig7: Menu Screen

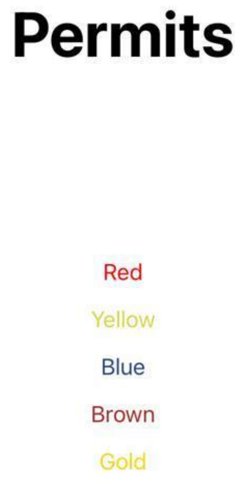


Fig8: Permits Screen

Worked on the recognition model and increased the epochs to 170. Ran the model over the video collected from our camera. The video is inside an engineering building. The model detects some cars for some reason inside, especially when the camera is being moved. But, we will have to debug the situation more. The camera still doesn't connect to the WIFI and we may need to take it to the IT so that they can help us connect it.

The validation error is going down which is a good indicator for the model accuracy. We will still keep working on it as we collect video from our camera.The test video has been uploaded on github.

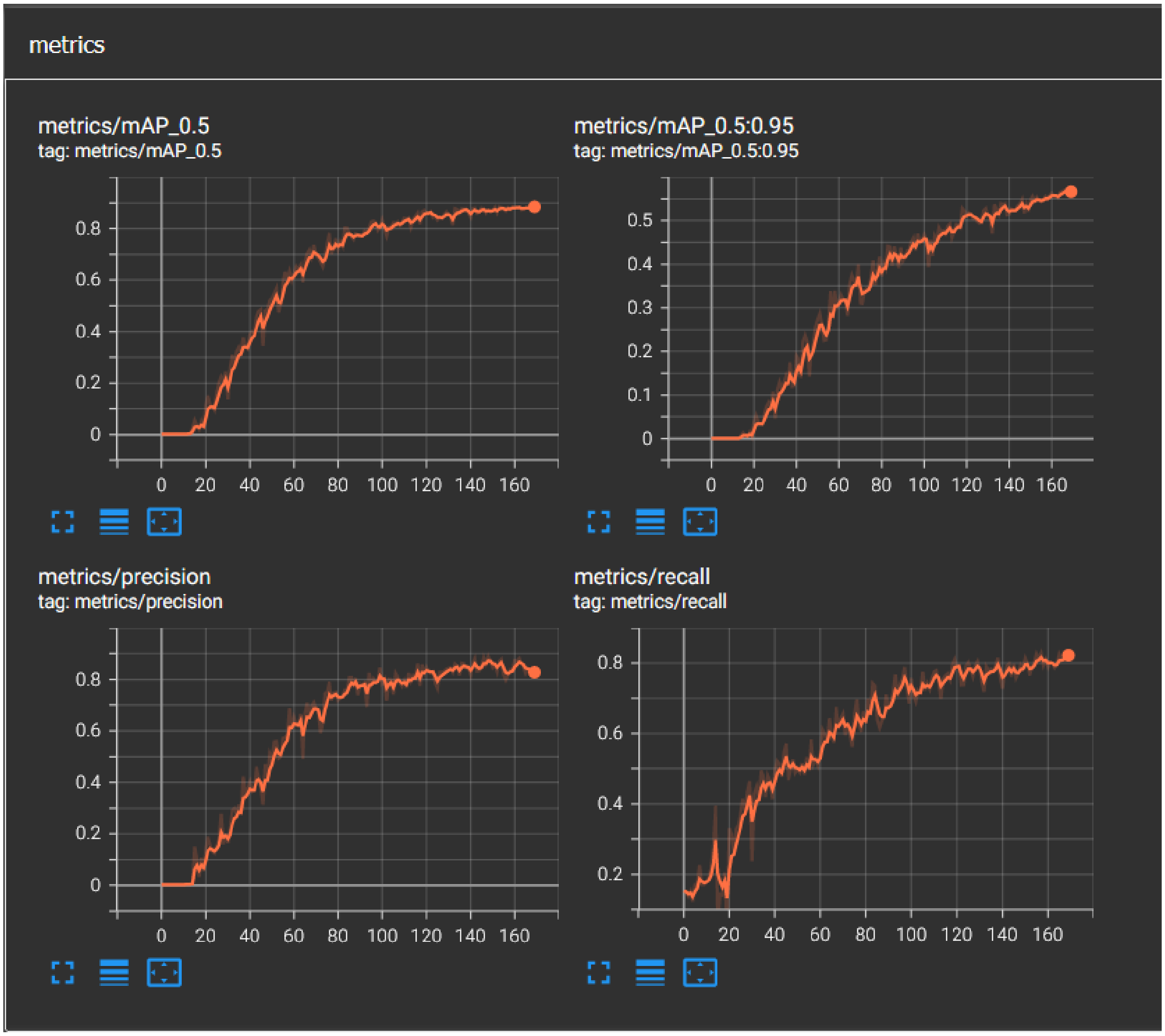


Fig9: Metrics on the model

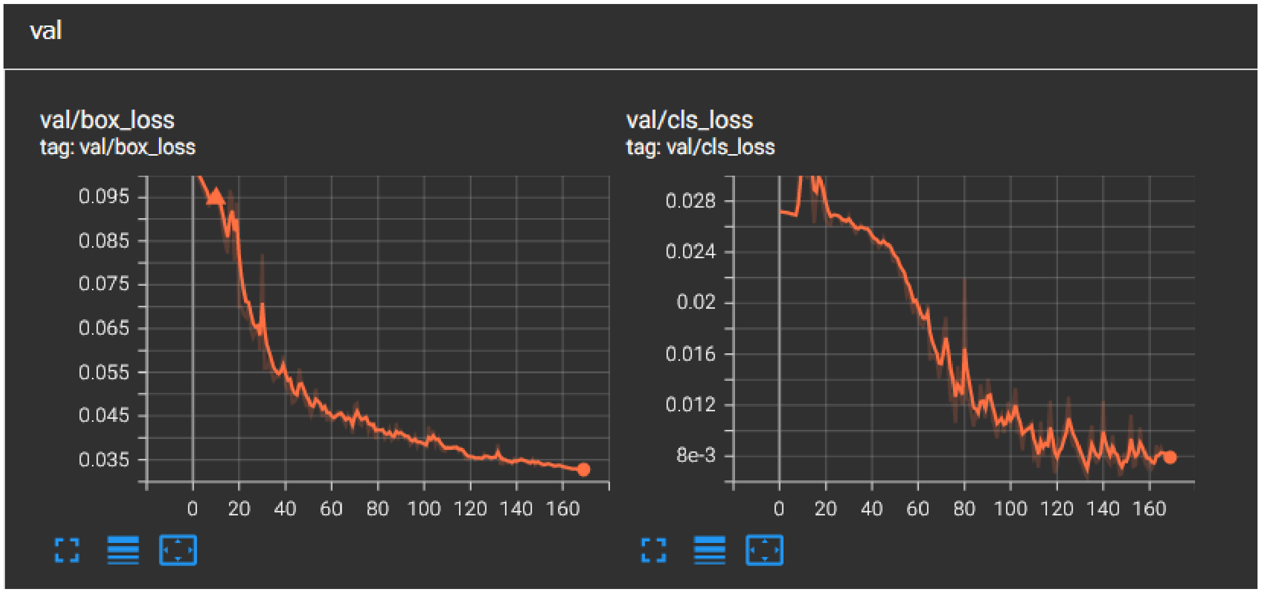


Fig9: Validation metric of the model