YOLO v5 Modeling

Cara Rono

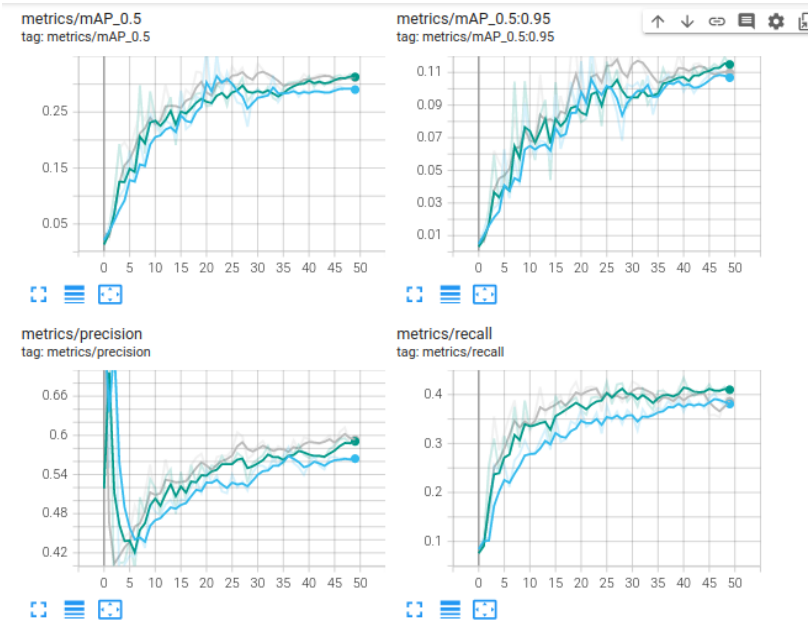
December 2, 2022

In this project, a cctv video is divided into three to get data from different time frames; this is to compare YOLOv5's detection performance in settings with different lighting. Part 1 (5-6pm), Part 2 (6-7pm), Part 3 (7-8pm).

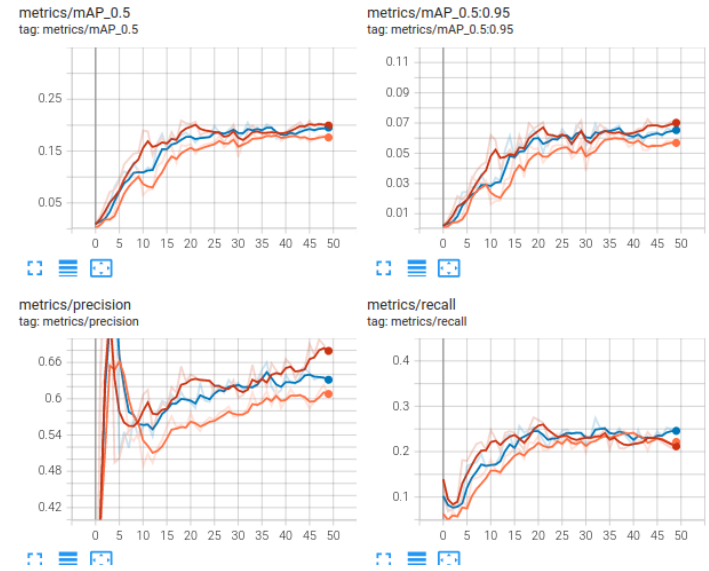
Contents:

I. Scalar Data

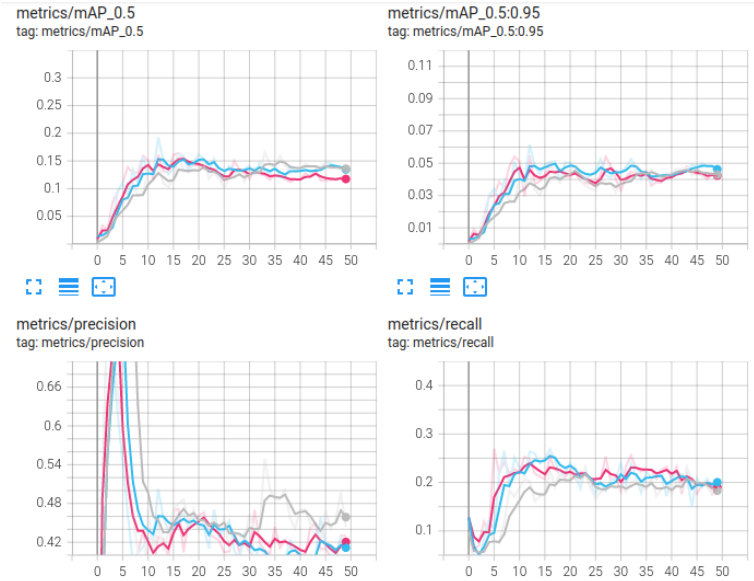
Shows mAP, precision, and recall



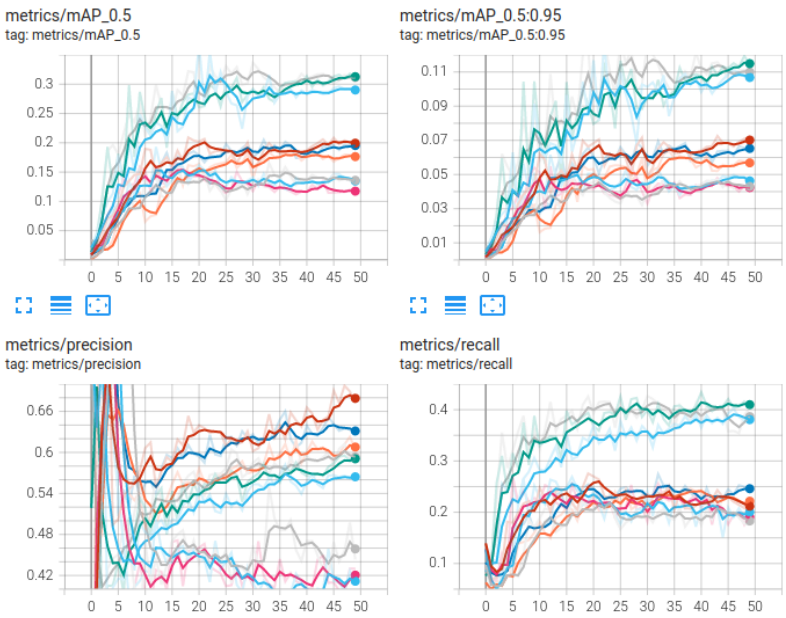
Part 1 (5-6 pm)



Part 2 (6-7 pm)



Part 3 (7-8 pm)



Comparison of metrics for the three parts (1-3) and their scales(small, medium, large).

As seen on the metrics, the model with the large scales has the highest mAP and precision score. For the recall score the model with medium and large scale are a tie.

For the comparison of different time frames, the data from 5-6 pm has the highest metric scores compared to the other two (6-8pm). This is because video data from part 1 is clearer because there is still sunlight, not like the other video data which are shot in the dark.

II. Confusion Matrix

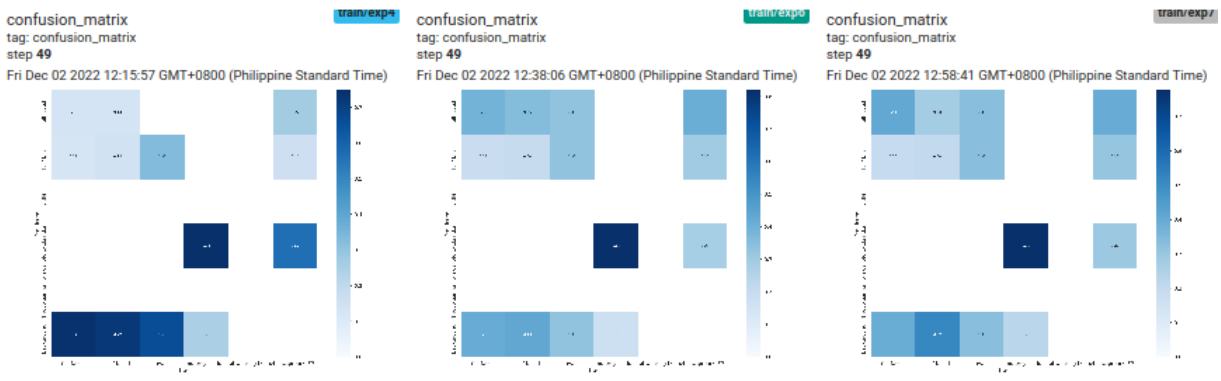
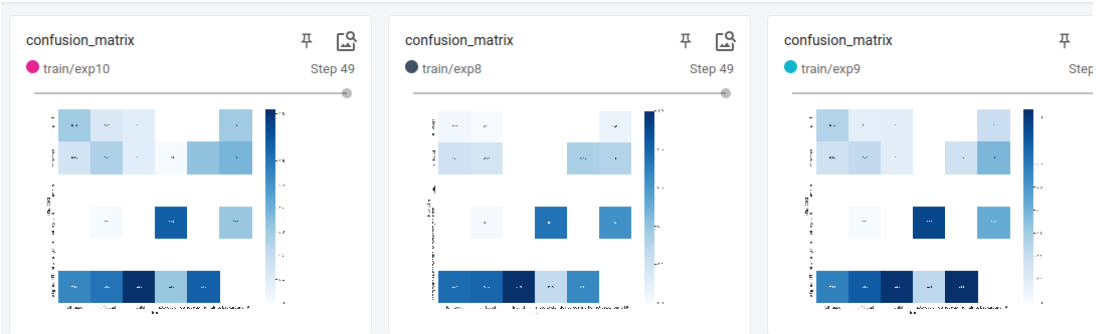
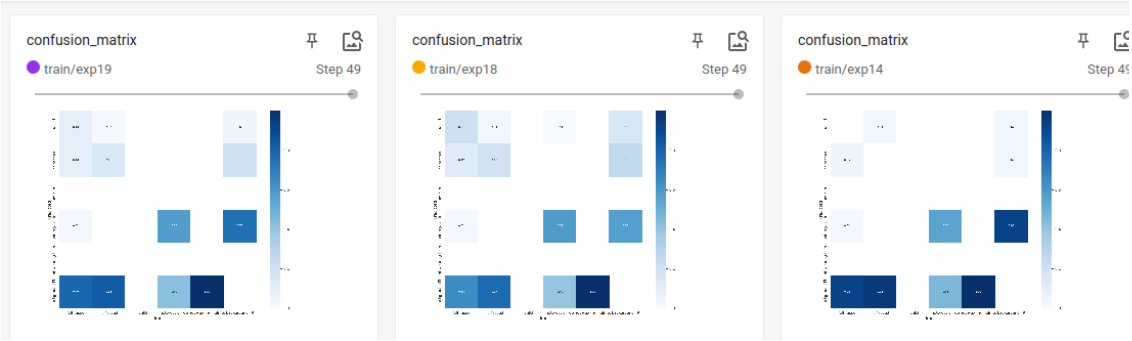


*image small to fit three models into one image
Confusion Matrix Part 1 (5-6pm) for all scales



Confusion Matrix Part 2 (6-7pm) for all scales

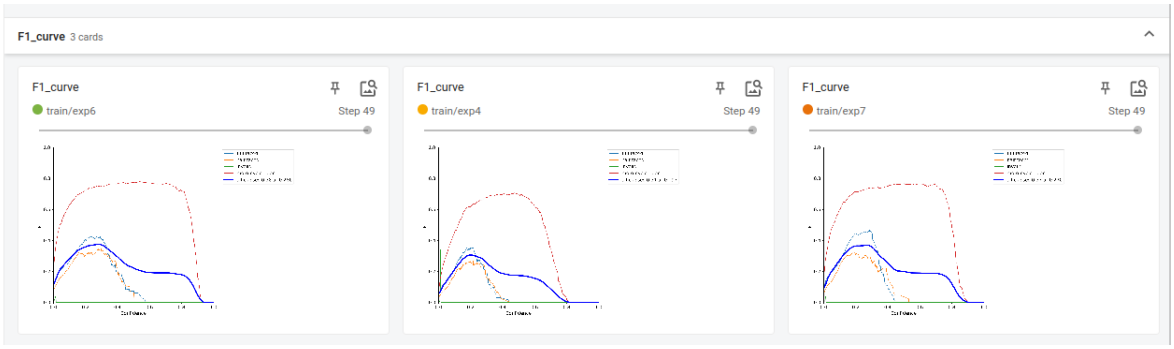


Confusion Matrix Part 3 (7-8pm) for all scales

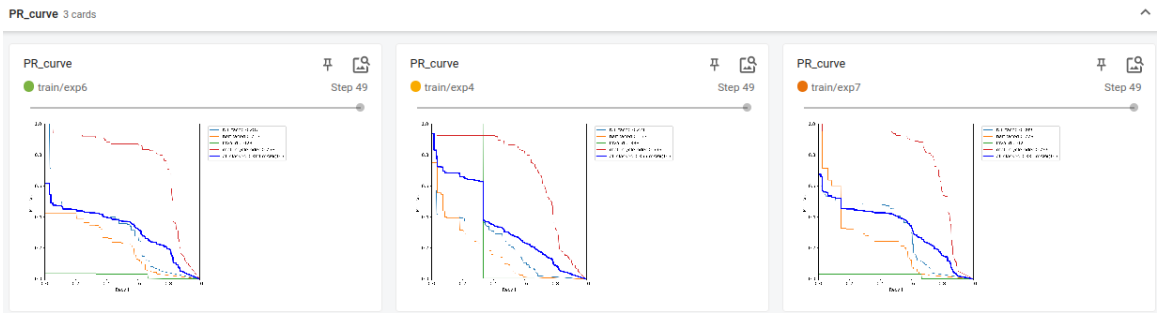
Based on the results of the confusion matrix of the different time frames, it shows that part 1 was able to detect motorcycle riders most accurately compared to the other time frames. The same also goes to 'background FN', as time progresses more objects are missed by the detector and considered as some other helmet types. Overall, the data shows that as time progresses the less the model is able to detect the motorcycle riders, this is because the video data is getting darker.

III. Time Series

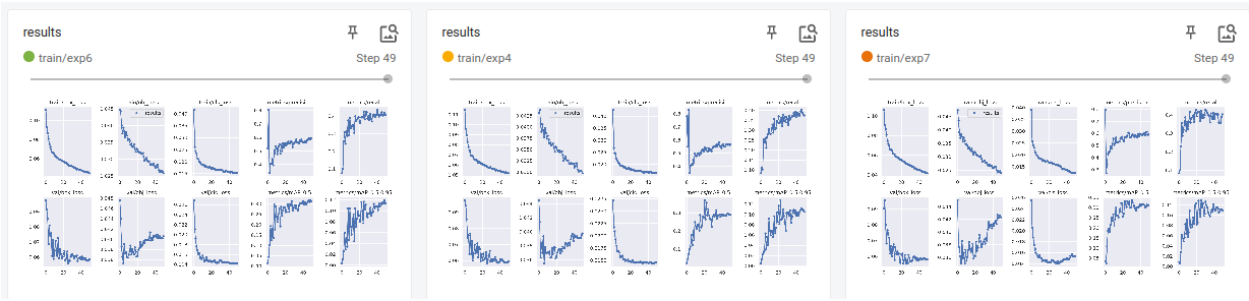
Part 1



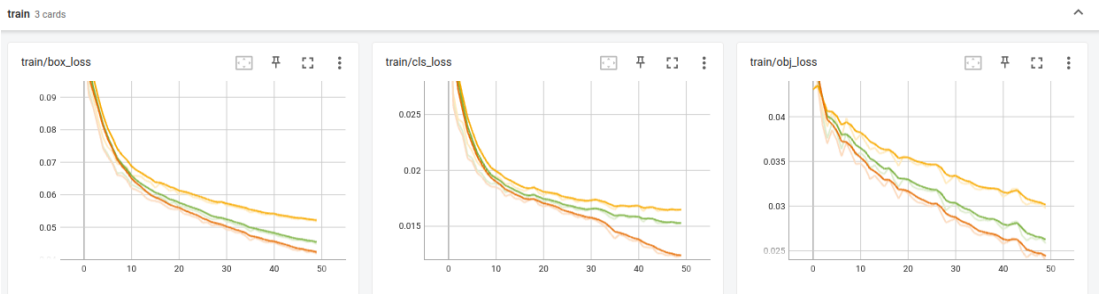
F1_curve



PR_curve

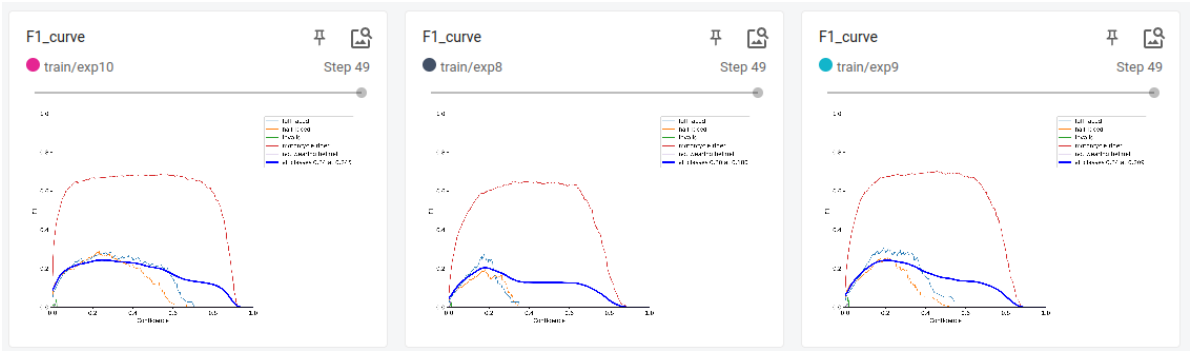


Results

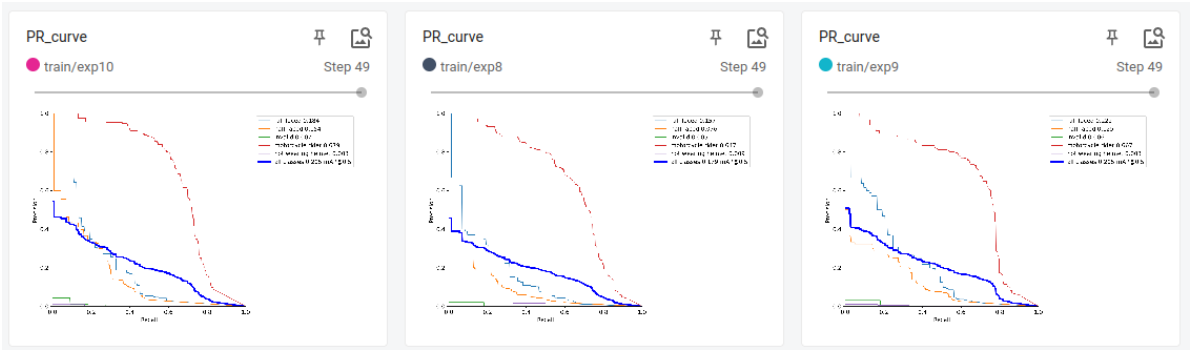


Train tab

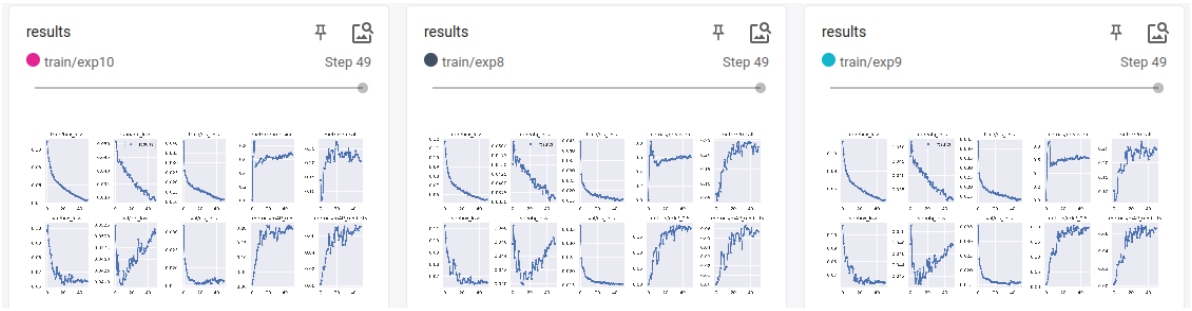
Part 2



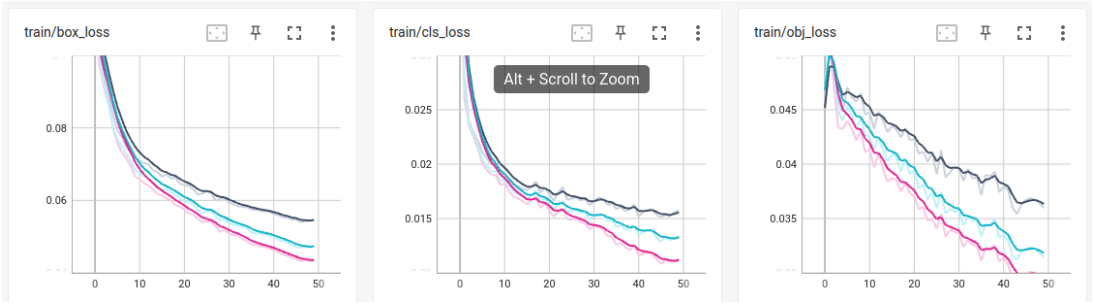
F1_curve



PR_curve

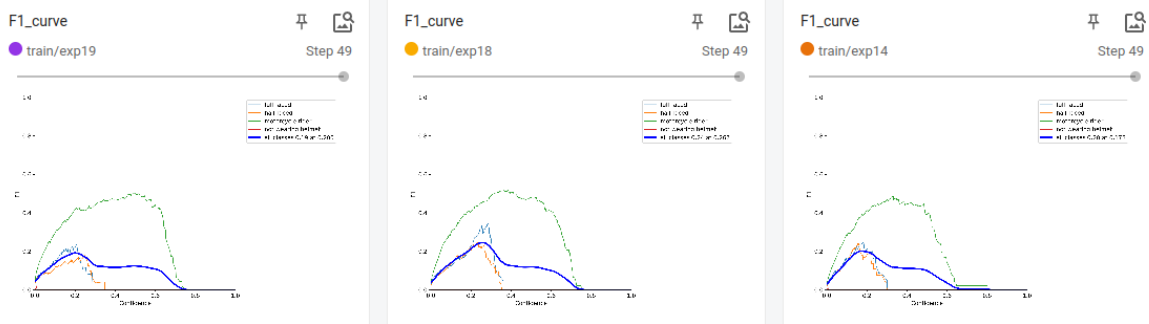


Results

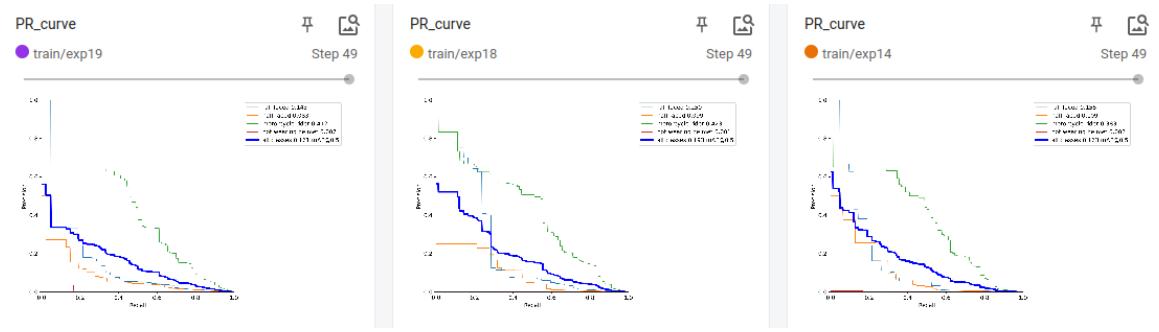


Train

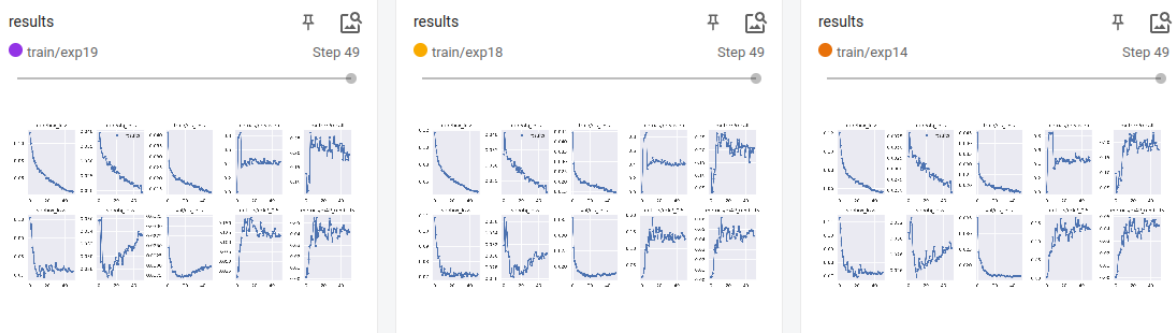
Part 3



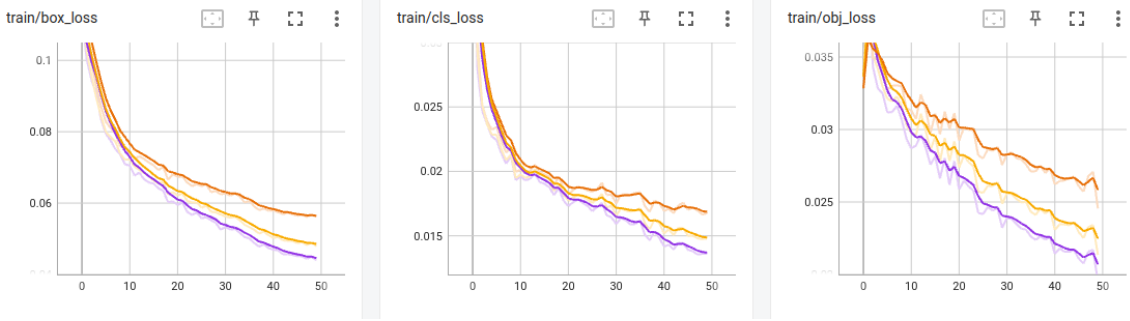
F1 Curve



PR Curve



Results



Train

The following insights are retrieved from the time series data:

- The three different time frames have almost the same results, but results per different scales differ.

- Based on the f1 curve of the three different time frames, they were all able to accurately detect motorcycle riders. The confidence score for Full-face helmet detection are higher compared to half-faced helmets

- Based on the PR curve, detection of motorcycle riders have the highest precision-recall relationship score followed by full-faced and half-faced helmets.

- Based on the Train curve, on all three time frames, it shows that the ‘small’ model takes the fastest time to train, followed by medium and large models.

IV: Testing the Model

Displaying inference on all test images.Video on a separate link.

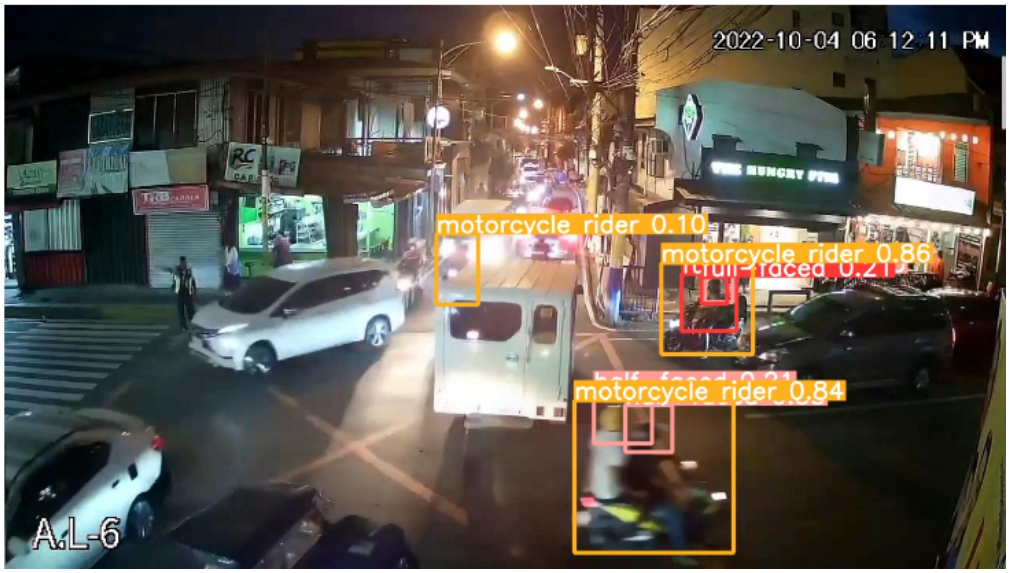
```
#detecting and predicting using the test dataset
#part 1 - medium
!python /content/yolov5/detect.py --weights /content/yolov5/runs/train/exp9/weights/best.pt --img 416 --conf 0.1 --source /

detect: weights=['/content/yolov5/runs/train/exp9/weights/best.pt'], source=/content/datasets/Helmets-1/part1/test/images, d
YOLOv5 🚀 v6.1-355-gd3761df Python-3.8.15 torch-1.12.1+cu113 CUDA:0 (Tesla T4, 15110MiB)

Fusing layers...
Model summary: 290 layers, 20869098 parameters, 0 gradients, 47.9 GFLOPs
image 1/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-102_jpg.rf.7732a1d9b4e7055be194bcad5b516fec.jpg: 2
image 2/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-106_jpg.rf.dd2a4020ca4edf27865a64a4a37f6c6e.jpg: 2
image 3/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-137_jpg.rf.f554e8d2b9eef42fa9a5d1f817cbb94f.jpg: 2
image 4/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-141_jpg.rf.6dfa4b454a08ffd1b796da5abe608f92.jpg: 2
image 5/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-153_jpg.rf.b75024f1f19f3419a955a9cafa71906e.jpg: 2
image 6/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-159_jpg.rf.d25ba167266c8203b1015a85100e3390.jpg: 2
image 7/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-168_jpg.rf.ff6ac5bcb2829ae7a12792815b9f00ad.jpg: 2
image 8/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-183_jpg.rf.8e4423c03b41cdb2482520a33d94cc80.jpg: 2
image 9/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-18_jpg.rf.85dc094de93c0637d676ad20cb0a1e73.jpg: 25
image 10/39 /content/datasets/Helmets-1/part1/test/images/part1_Trim-2_mp4-191_jpg.rf.bf6a6d36f24dc1aa69d501ed4c56ec84.jpg:
```

Detecting and predicting using the test dataset - part 1 (medium scale)
This will be done for all parts and each scale.

```
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("\n")
```

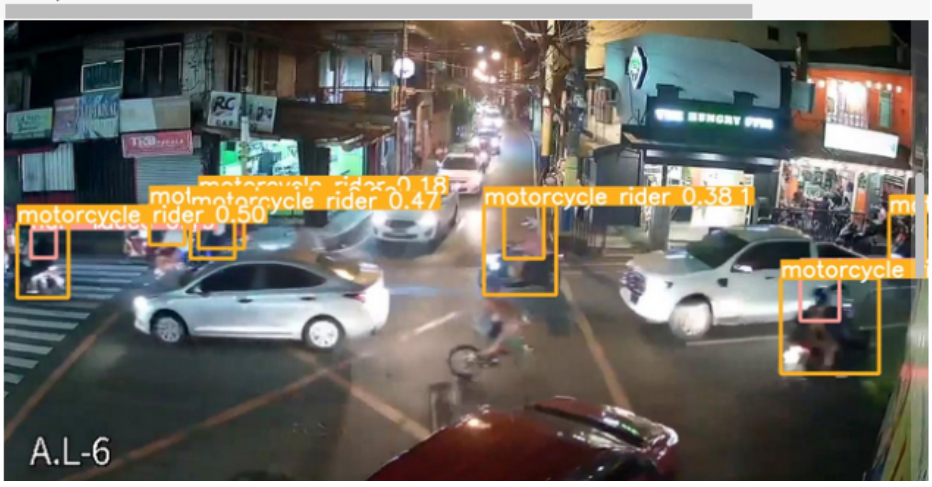
(part 1: 5-6pm)

```
#display inference on ALL test images
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp5/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("\n")
```



(part 2: 6-7pm)

```
#display inference on ALL test images
import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp6/*.jpg'): #ass
    display(Image(filename=imageName))
    print("\n")
```



(part 3: 7-8pm)