Name : Panneerselvam N
Date : 21/06/2021

Task : Custom Yolo v4 Model Testing

System Specifications :(Colab)
Disk : 30 GB
RAM : 13 GB
GPU : Tesla K80

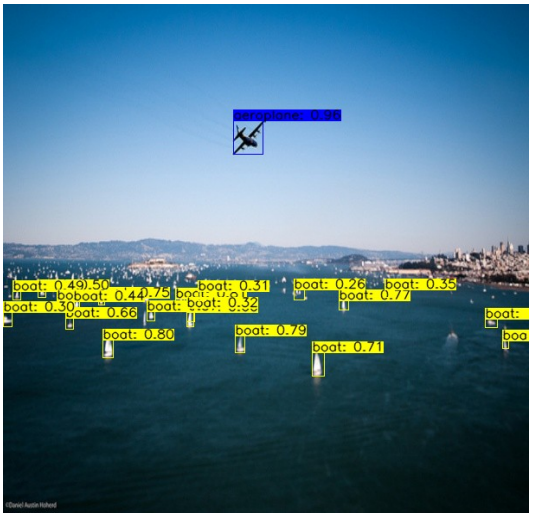
Testing method :
For image :
I have taken 150 images with various augmentations and calculated Mean Average Precision and Recall using Colab.

For video :
I have taken some video from youtube and tested in my custom model to calculate Frame per second.

Resize – (224x224):

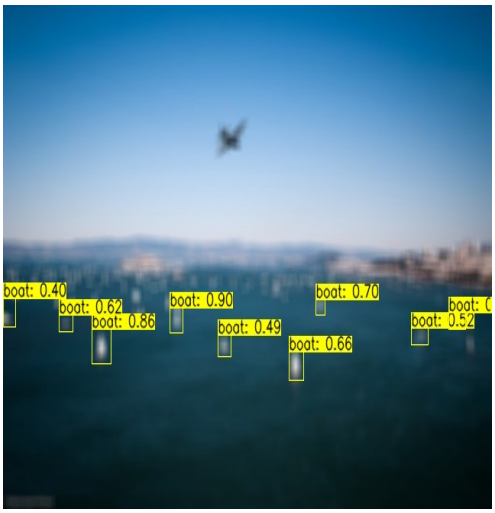

```
 detections_count = 1079, unique_truth_count = 326
class_id = 0, name = aeroplane, ap = 99.76%      (TP = 33, FP = 1)
class_id = 1, name = bird, ap = 84.57%           (TP = 41, FP = 11)
class_id = 2, name = boat, ap = 88.89%           (TP = 72, FP = 7)
class_id = 3, name = cow, ap = 91.79%            (TP = 59, FP = 8)
class_id = 4, name = motorcycle, ap = 98.39%     (TP = 42, FP = 3)
class_id = 5, name = umbrella, ap = 98.95%       (TP = 42, FP = 1)

 for conf_thresh = 0.25, precision = 0.90, recall = 0.89, F1-score = 0.89
 for conf_thresh = 0.25, TP = 289, FP = 31, FN = 37, average IoU = 68.77 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.937255, or 93.73 %
Total Detection Time: 11 Seconds
```

Resize – (608x608):




```
 detections_count = 1275, unique_truth_count = 326
class_id = 0, name = aeroplane, ap = 100.00%      (TP = 34, FP = 0)
class_id = 1, name = bird, ap = 90.95%            (TP = 44, FP = 14)
class_id = 2, name = boat, ap = 98.85%            (TP = 81, FP = 6)
class_id = 3, name = cow, ap = 95.85%             (TP = 66, FP = 14)
class_id = 4, name = motorcycle, ap = 98.59%      (TP = 44, FP = 2)
class_id = 5, name = umbrella, ap = 100.00%       (TP = 44, FP = 1)

 for conf_thresh = 0.25, precision = 0.89, recall = 0.96, F1-score = 0.93
 for conf_thresh = 0.25, TP = 313, FP = 37, FN = 13, average IoU = 70.33 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.973747, or 97.37 %
Total Detection Time: 11 Seconds
```

Blur with Kernel size (10x10):

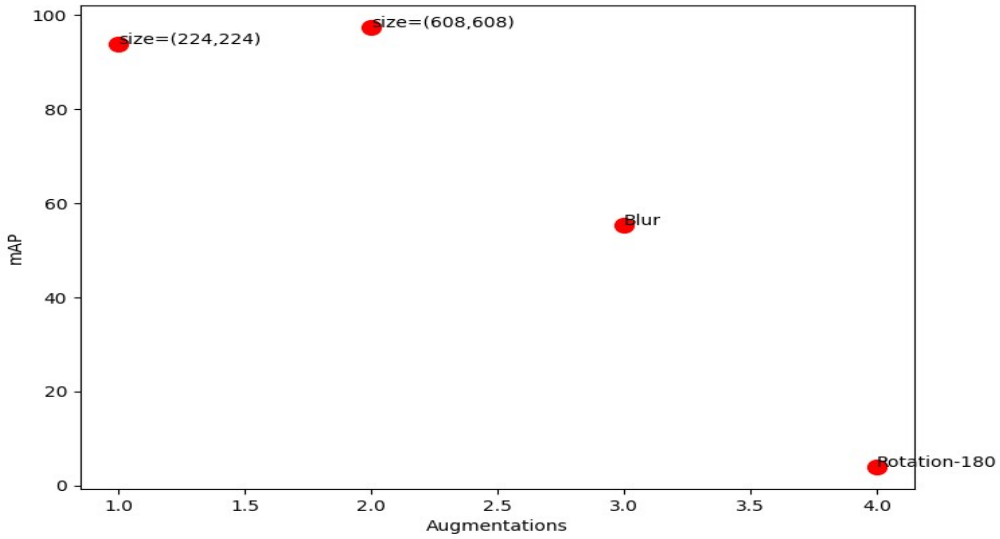



```
 detections_count = 744, unique_truth_count = 326
class_id = 0, name = aeroplane, ap = 73.22%       (TP = 24, FP = 4)
class_id = 1, name = bird, ap = 36.27%            (TP = 17, FP = 9)
class_id = 2, name = boat, ap = 36.21%            (TP = 29, FP = 13)
class_id = 3, name = cow, ap = 52.05%             (TP = 33, FP = 8)
class_id = 4, name = motorcycle, ap = 71.01%      (TP = 30, FP = 7)
class_id = 5, name = umbrella, ap = 63.88%        (TP = 26, FP = 5)

 for conf_thresh = 0.25, precision = 0.78, recall = 0.49, F1-score = 0.60
 for conf_thresh = 0.25, TP = 159, FP = 46, FN = 167, average IoU = 58.47 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.554401, or 55.44 %
Total Detection Time: 11 Seconds

Set -points flag:
```

Rotation – 180 degree:

```
 detections_count = 867, unique_truth_count = 326
class_id = 0, name = aeroplane, ap = 9.40%        (TP = 6, FP = 21)
class_id = 1, name = bird, ap = 2.74%             (TP = 5, FP = 48)
class_id = 2, name = boat, ap = 0.13%             (TP = 0, FP = 10)
class_id = 3, name = cow, ap = 0.27%      (TP = 2, FP = 36)
class_id = 4, name = motorcycle, ap = 5.60%       (TP = 5, FP = 16)
class_id = 5, name = umbrella, ap = 5.77%         (TP = 6, FP = 11)

 for conf_thresh = 0.25, precision = 0.14, recall = 0.07, F1-score = 0.10
 for conf_thresh = 0.25, TP = 24, FP = 142, FN = 302, average IoU = 9.44 %

 IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
 mean average precision (mAP@0.50) = 0.039861, or 3.99 %
Total Detection Time: 11 Seconds
```

Graph :



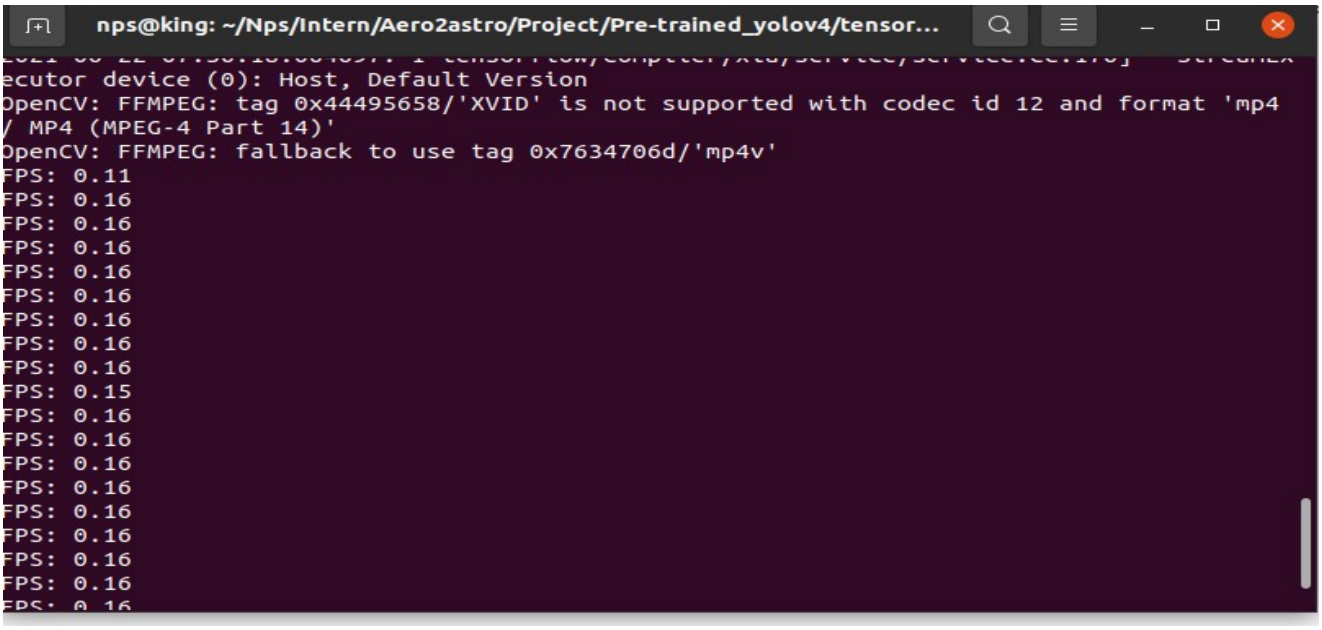
Above graph shows the Mean Average Precision with respect different image augmentation methods.

Video Testing :

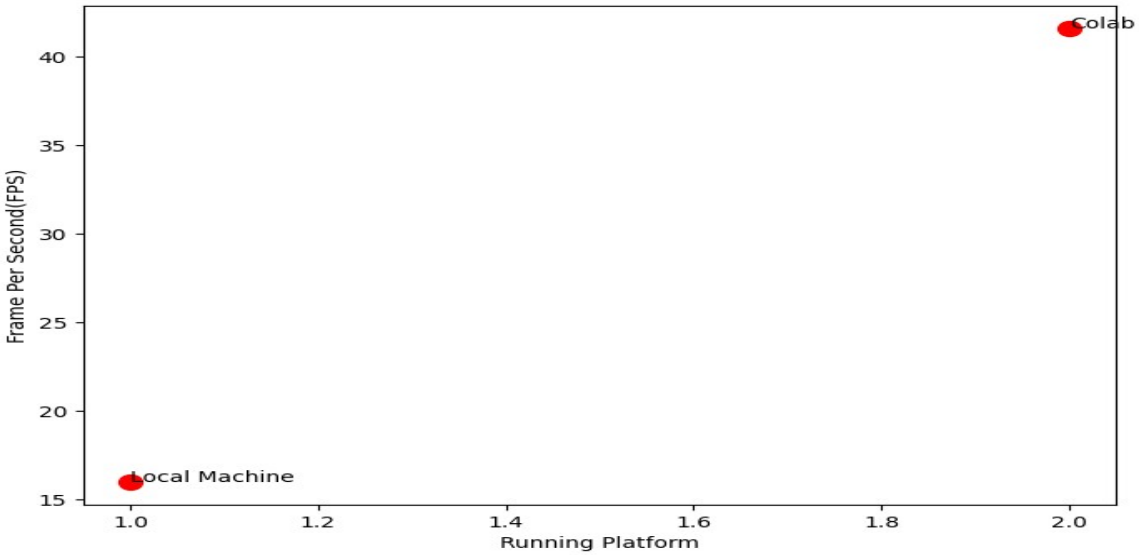
In Colab :

```
motorcycle: 98%
motorcycle: 80%
motorcycle: 77%

FPS:39.4         AVG_FPS:41.6

 cvWriteFrame
Objects:

motorcycle: 97%
motorcycle: 73%

FPS:41.2         AVG_FPS:41.6

 cvWriteFrame
Objects:

motorcycle: 96%
motorcycle: 71%

FPS:40.7         AVG_FPS:41.6

 cvWriteFrame
Objects:

motorcycle: 97%
motorcycle: 94%
motorcycle: 71%
```

In Local Machine :



Graph :



Above image shows the graph of FPS with respect different running machines.

Conclusion :

My model performs well in trained data, but when comes to new data, it’s mAP is much less compred to trained data mAP.

Model gives good FPS (41.6) in GPU colab .

Above analysis shows that my model was over-fitted . Because of less data and less augmentation.