INNOVATION LAB

(TITLE:GROCERY OBJECT DETECTION: IMPROVING MODEL PERFORMANCE THROUGH DATA AUGMENTATION)

NAME: HIMANI YADAV
ROLL NO: 2201AI16
UNDER - DR.JIMSON MATHEW

OBJECTIVE:

- BUILD A GROCERY OBJECT DETECTION MODEL.
- ENHANCE PERFORMANCE USING DATA AUGMENTATION TECHNIQUES.

CHALLENGES IDENTIFIED:

- POOR MODEL PERFORMANCE ON INITIAL TRAINING.

OBJECT DETECTION USING YOLOV7

-DATASET ANNOTATION
-MODEL TRAINING
-MODEL TESTING

INTRODUCTION TO OBJECT DETECTION

- OBJECT DETECTION IS THE TASK OF IDENTIFYING OBJECTS WITHIN AN IMAGE OR VIDEO.
- IT OUTPUTS BOTH CLASS LABELS AND BOUNDING BOXES FOR EACH OBJECT DETECTED.
- WIDELY USED IN APPLICATIONS LIKE AUTONOMOUS DRIVING, SECURITY, MEDICAL IMAGING, ETC.

INTRODUCTION TO YOLOV7

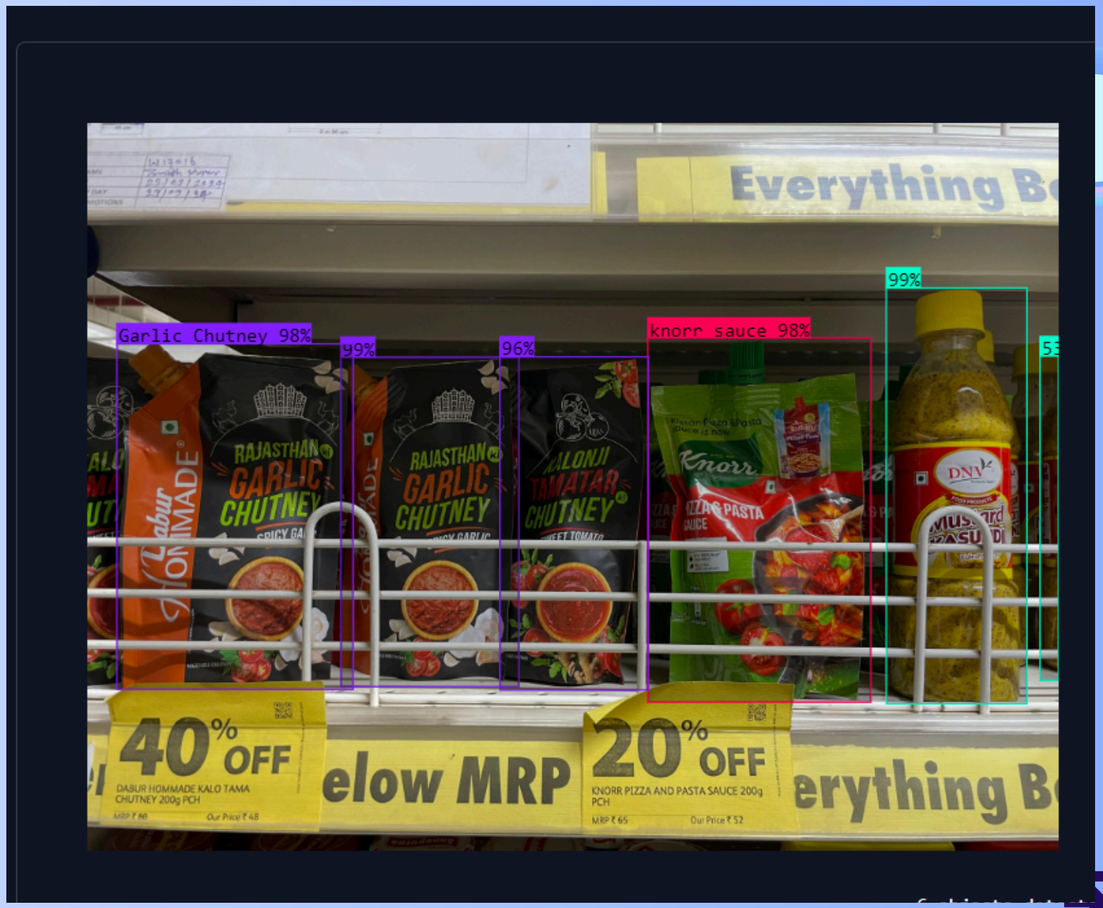
- RELEASED IN 2022, YOLOV7 IS OPTIMIZED FOR BOTH SPEED AND ACCURACY, MAKING IT ONE OF THE MOST EFFICIENT REAL-TIME OBJECT DETECTION MODELSACHIEVES BETTER PERFORMANCE THAN PREVIOUS YOLO VERSIONS WITH FEWER PARAMETERS WHILE MAINTAINING HIGH ACCURACY.
- YOLOV7 IS DESIGNED WITH AN IMPROVED BACKBONE AND HEAD ARCHITECTURE, FOCUSING ON DETECTING SMALLER OBJECTS MORE EFFECTIVELY AND INCREASING DETECTION PRECISION.

LOSS FUNCTION USED IN YOLOV7:

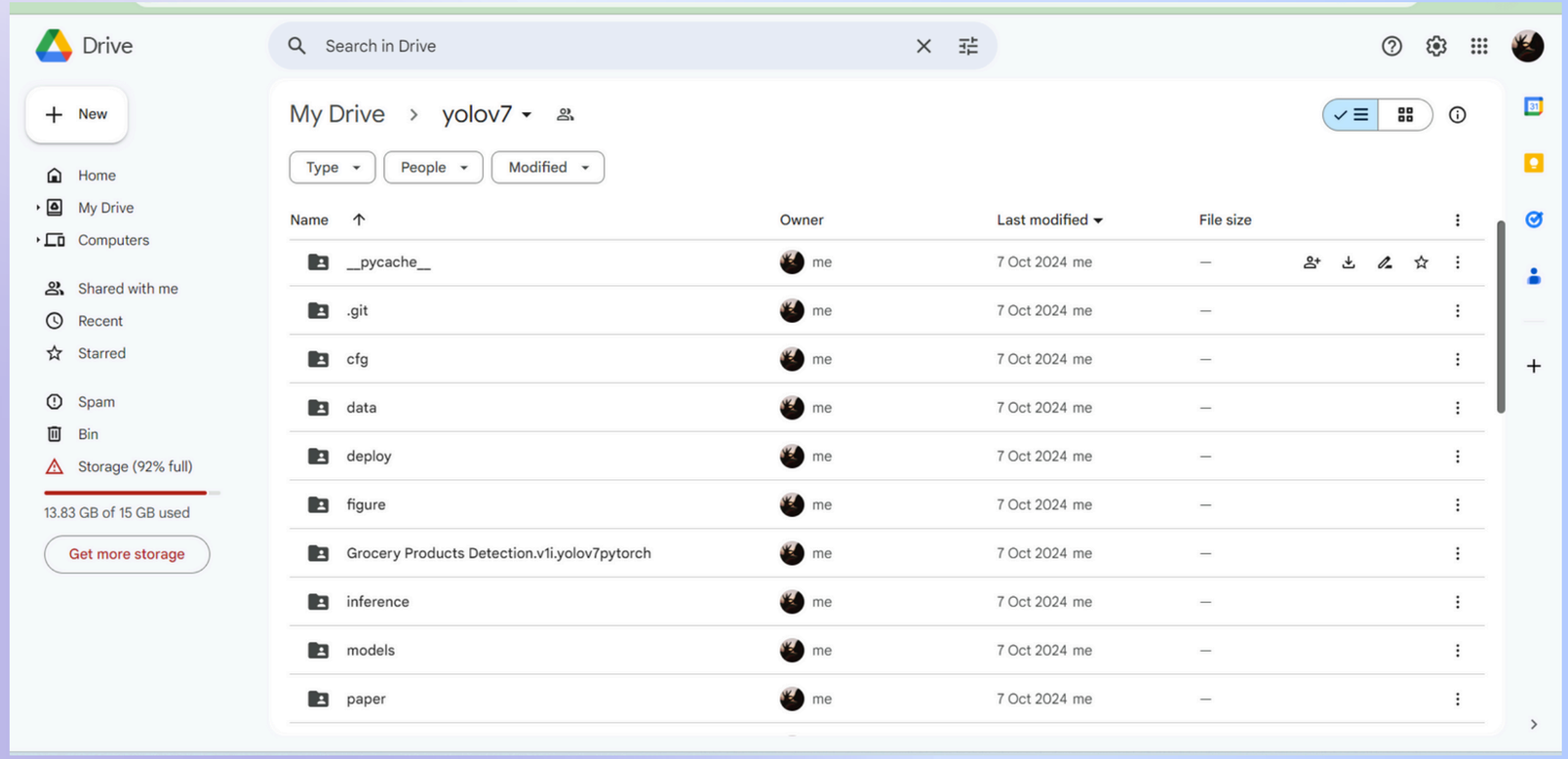
- BOUNDING BOX REGRESSION LOSS (CIOU): CALCULATES HOW WELL THE PREDICTED BOUNDING BOXES MATCH THE GROUND TRUTH.
- OBJECTNESS LOSS: DETERMINES IF AN OBJECT IS PRESENT IN A GRID CELL.
- CLASSIFICATION LOSS: HANDLES THE ACCURACY OF PREDICTING THE CORRECT CLASS LABEL FOR THE DETECTED OBJECTS.

DATASET CREATION & ANNOTATION

- DATASET CREATED USING IMAGES OF SPECIFIC OBJECTS.
- ANNOTATION PLATFORM: ROBOFLOW USED FOR ANNOTATION.
- ANNOTATION INCLUDED DRAWING BOUNDING BOXES AROUND OBJECTS AND LABELING THEM

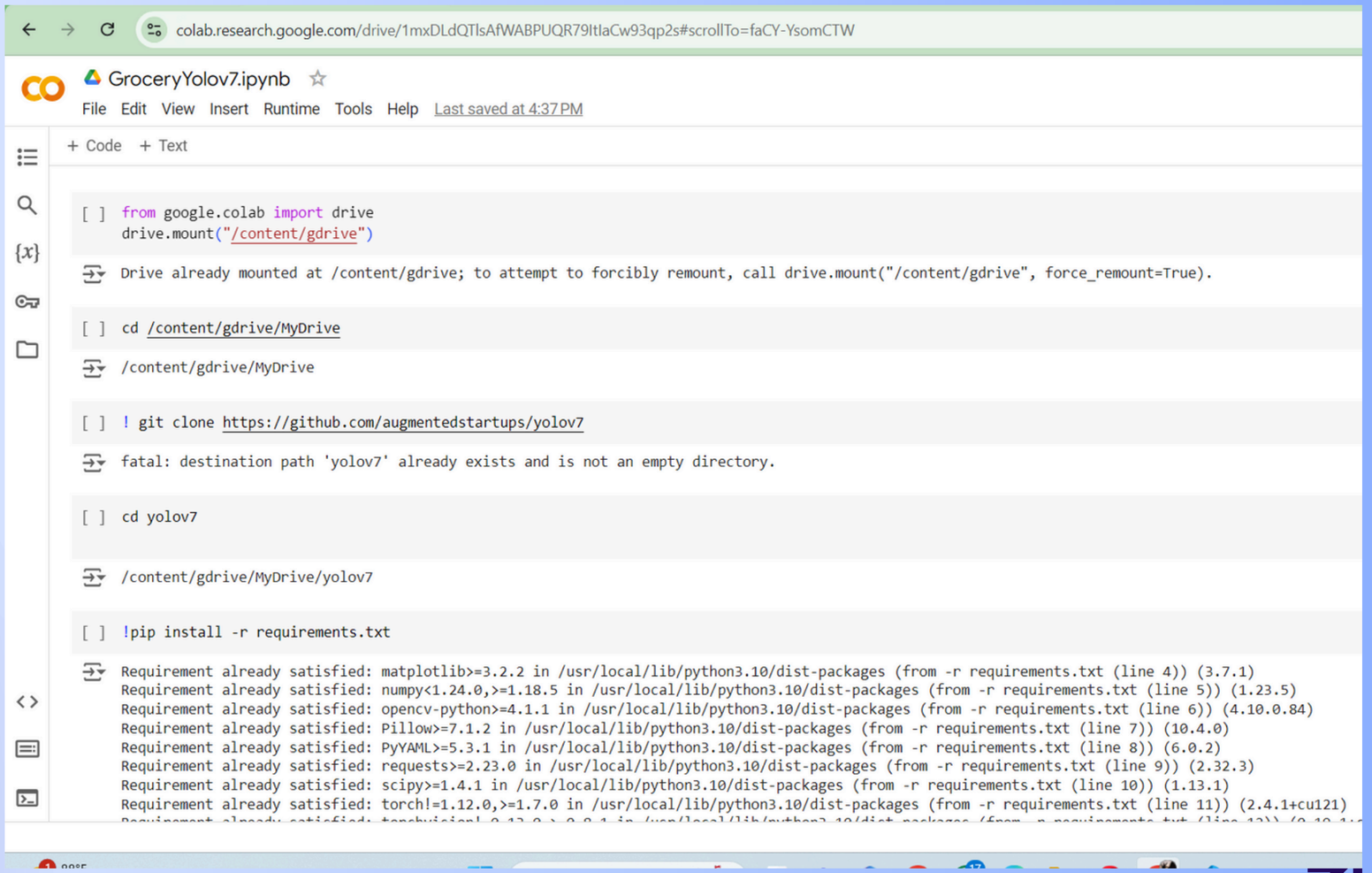


TRAINING YOLOV7 MODEL



- DATASET LOADED FROM ROBOFLOW, TO GOOGLE DRIVE WITH NECESSARY PRE-PROCESSING.

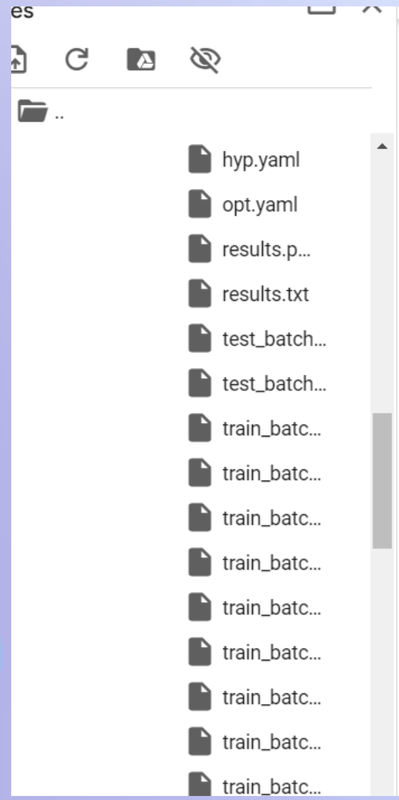
- PLATFORM: GOOGLE COLAB (LEVERAGE GPUS FOR FASTER TRAINING).
- FRAMEWORK: PYTORCH IMPLEMENTATION OF YOLOV7.



- **STARTED THE TRAINING PROCESS WITH PRE-TRAINED WEIGHTS**

TTPS://GITHUB.COM/WONGKINYIU/YOLOV7/RELEASES/DOWNLOAD/V0.1/YOLOV7.PT

```
#!python train.py --img-size 640 --cfg cfg/training/yolov7.yaml --hyp data/hyp.scratch.custom.yaml --batch 8 --epochs 100 --data
#!python train.py --workers 8 --device 0 --batch-size 8 --data data/custom_train.yaml --img 640 640 --cfg cfg/training/yolov7.ya
!python train.py --batch 16 --cfg cfg/training/yolov7.yaml --epochs 50 --data "data/data.yaml" --weights 'yolov7.pt'
```




```
      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      27/49       14G    0.0526    0.0356   0.05215    0.1404        78       640: 100% 5/5 [00:04<00:00,  1.13it/s]
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.04it/s]
                  all         20        101       0.11     0.0752     0.0403      0.017

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      28/49       14G    0.0525   0.03132   0.05139    0.1352        45       640: 100% 5/5 [00:05<00:00,  1.10s/it]
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.60it/s]
                  all         20        101       0.104     0.0775     0.0402      0.0166

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      29/49       14G   0.05302    0.0401   0.05124    0.1444       119       640: 100% 5/5 [00:04<00:00,  1.17it/s]
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.98it/s]
                  all         20        101       0.109     0.0486     0.0412      0.0215

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      30/49       14G   0.05189    0.0364   0.05157    0.1399        55       640: 100% 5/5 [00:06<00:00,  1.27s/it]
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.83it/s]
                  all         20        101       0.109      0.063     0.0436      0.0188

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      31/49       14G   0.05008   0.03124   0.05321    0.1345        52       640: 100% 5/5 [00:05<00:00,  1.02s/it]
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.63it/s]
                  all         20        101      0.0886      0.101     0.0472      0.0224

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      32/49       14G   0.05032   0.03624   0.05117    0.1377        93       640: 100% 5/5 [00:04<00:00,  1.14it/s]
                Class     Images     Labels          P          R     mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.54it/s]
                  all         20        101      0.0381      0.106     0.0494      0.0211
```

AFTER THE FIRST TRAINING PROCESS WE GOT THE UPDATED WEIGHTS

olov7.ipynb

Insert Runtime Tools Help All changes saved

Comment Share Gemini

+ Code + Text

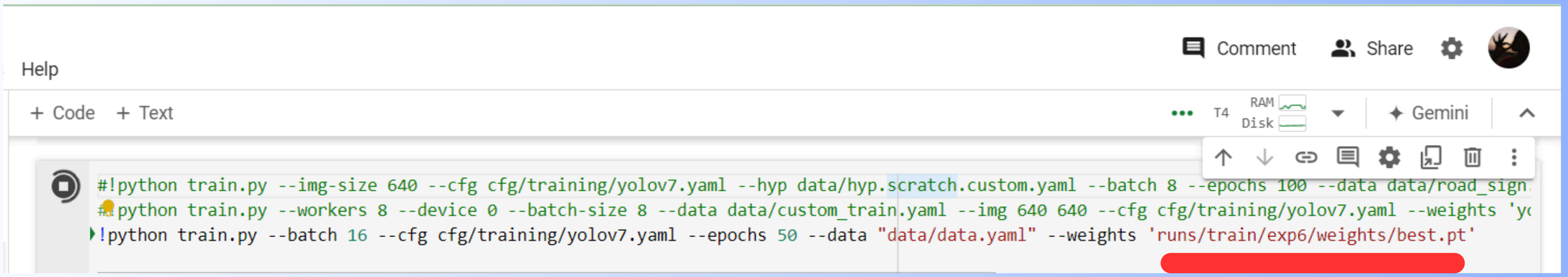
T4 RAM Disk

hyp.yaml
opt.yaml
results.p...
results.txt
test_batch...
test_batch...
train_batc...
train_batc...
train_batc...
train_batc...
train_batc...
train_batc...
train_batc...
train_batc...
train_batc...

71.69 GB available

```
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<0
                 all          20         101      0.0253       0.315      0.0631      0.0325

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      45/49       14G   0.04759   0.04035   0.05139    0.1393       115       640: 100% 5/5 [00:04<00:00,  1.03it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.58it/s]
                 all          20         101      0.0265       0.201      0.0595      0.0325

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      46/49       14G   0.04699   0.03576   0.05037    0.1331       145       640: 100% 5/5 [00:04<00:00,  1.15it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  2.19it/s]
                 all          20         101      0.0255       0.451      0.0634      0.0314

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      47/49       14G   0.04669   0.03969   0.05107    0.1375       114       640: 100% 5/5 [00:05<00:00,  1.15s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:01<00:00,  1.06s/it]
                 all          20         101      0.0257       0.451      0.0631      0.0336

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      48/49       14G   0.04691   0.03625   0.05127    0.1344        61       640: 100% 5/5 [00:04<00:00,  1.12it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.99it/s]
                 all          20         101        0.029        0.23      0.0643      0.0342

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      49/49       14G   0.04651   0.03642   0.05049    0.1334        81       640: 100% 5/5 [00:05<00:00,  1.16s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:01<00:00,  1.23s/it]
                 all          20         101      0.0309       0.234      0.0653      0.0354
50 epochs completed in 0.147 hours.

/content/gdrive/MyDrive/yolov7/utils/general.py:802: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default
  x = torch.load(f, map_location=torch.device('cpu'))
Optimizer stripped from runs/train/exp6/weights/last.pt, 75.3MB
Optimizer stripped from runs/train/exp6/weights/best.pt, 75.3MB
```

- **BY REPLACING THE OLD WEIGHTS WITH UPDATED WEIGHTS WE HAVE TRAINED THE MODEL MULTIPLE TIMES TO INCREASE THE MEAN AVERAGE PRECISION**



- **UPDATED WEIGHTS**

```
      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      48/49     14.6G   0.03903   0.03722   0.04958    0.1258        61       640: 100% 5/5 [00:05<00:00,  1.04s/it]
                Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:01<00:00,  1.11s/it]
                  all          20         101      0.0456       0.341      0.0855      0.0576

      Epoch   gpu_mem       box       obj       cls     total    labels  img_size
      49/49     14.6G   0.03828   0.03702   0.04875    0.1241        81       640: 100% 5/5 [00:04<00:00,  1.14it/s]
                Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:01<00:00,  1.43s/it]
                  all          20         101       0.239       0.184      0.0881      0.0613
50 epochs completed in 0.150 hours.

/content/gdrive/MyDrive/yolov7/utils/general.py:802: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default
  x = torch.load(f, map_location=torch.device('cpu'))
Optimizer stripped from runs/train/exp7/weights/last.pt, 75.3MB
Optimizer stripped from runs/train/exp7/weights/best.pt, 75.3MB
```

TRAINING WITH UPDATED WEIGHTS

```
#!python train.py --img-size 640 --cfg cfg/training/yolov7.yaml --hyp data/hyp.scratch.custom.yaml --batch 8 --epochs 100 --data data/road_sign
#!python train.py --workers 8 --device 0 --batch-size 8 --data data/custom_train.yaml --img 640 640 --cfg cfg/training/yolov7.yaml --weights 'yo
!python train.py --batch 16 --cfg cfg/training/yolov7.yaml --epochs 50 --data "data/data.yaml" --weights 'runs/train/exp11/weights/best.pt'
```

```
               all          20         101      0.0598       0.426      0.0995      0.0724

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     34/49     15.1G    0.0362   0.03474   0.04856    0.1195        85       640: 100% 5/5 [00:06<00:00,  1.20s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.06it/s]
                 all          20         101      0.0586        0.44      0.0957      0.0676

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     35/49     15.1G   0.03682   0.03022   0.04729    0.1143        35       640: 100% 5/5 [00:04<00:00,  1.00it/s]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.09it/s]
```

NEW UPDATED WEIGHTS

```
     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     48/49     15.1G   0.03422   0.03678   0.04826    0.1193        61       640: 100% 5/5 [00:05<00:00,  1.02s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:00<00:00,  1.86it/s]
                 all          20         101      0.0582       0.329       0.105      0.0819

     Epoch   gpu_mem       box       obj       cls     total    labels  img_size
     49/49     15.1G   0.03361   0.03636    0.0472    0.1172        81       640: 100% 5/5 [00:06<00:00,  1.36s/it]
               Class      Images      Labels           P           R      mAP@.5  mAP@.5:.95: 100% 1/1 [00:01<00:00,  1.25s/it]
                 all          20         101        0.06       0.325       0.105       0.077
50 epochs completed in 0.149 hours.

/content/gdrive/MyDrive/yolov7/utils/general.py:802: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default
  x = torch.load(f, map_location=torch.device('cpu'))
Optimizer stripped from runs/train/exp12/weights/last.pt, 75.3MB
Optimizer stripped from runs/train/exp12/weights/best.pt, 75.3MB
```

• USING THE FINAL WEIGHTS WE HAVE PREDICTED OUTPUTS FOR OUR IMAGES

```
[ ] !python detect.py --weights '/content/gdrive/MyDrive/yolov7/runs/train/exp13/weights/last.pt' --conf 0.1 --source '/content/gdrive/MyDrive/yolov7/Grocery Products Detection.v1i.yolov7

Namespace(weights=['/content/gdrive/MyDrive/yolov7/runs/train/exp13/weights/last.pt'], source='/content/gdrive/MyDrive/yolov7/Grocery Products Detection.v1i.yolov7pytorch/test/images',
YOLOR 🚀 v0.1-121-g2fdc7f1 torch 2.4.1+cu121 CUDA:0 (Tesla T4, 15102.0625MB)

/content/gdrive/MyDrive/yolov7/models/experimental.py:252: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickl
  ckpt = torch.load(w, map_location=map_location)  # load
Fusing layers...
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
RepConv.fuse_repvgg_block
IDetect.fuse
/usr/local/lib/python3.10/dist-packages/to
```

T4 RAM Disk ✦ Gemini

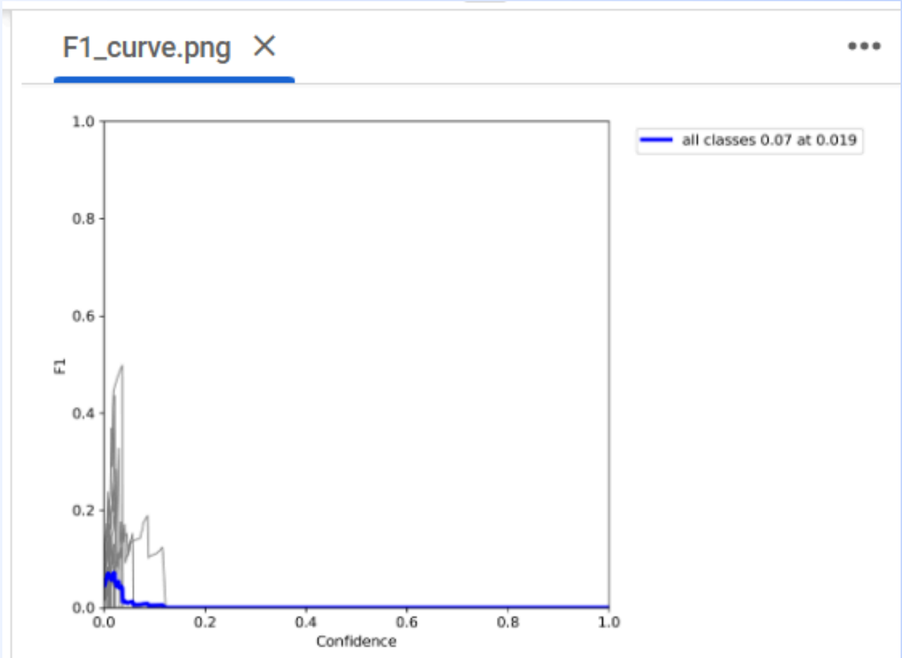
IMG20241003103119_jpg.rf.4cc03dc16f586a2606fad ••• l be required to pass the indexing argument. (Triggered inte

MODEL PERFORMANCE AND EVALUATION

- MAP (MEAN AVERAGE PRECISION): MEASURES THE ACCURACY OF THE MODEL IN DETECTING AND CLASSIFYING OBJECTS.

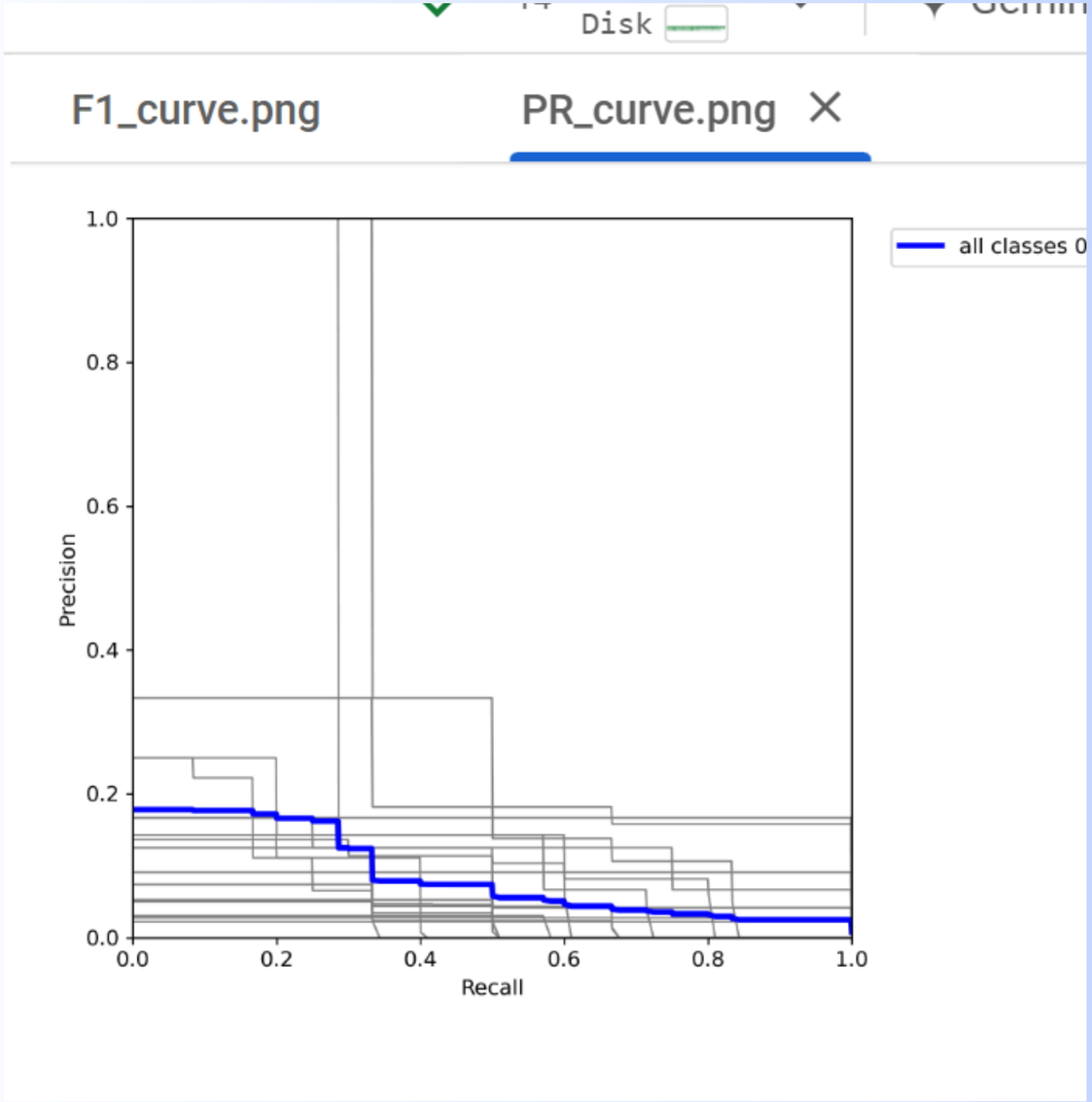
P	R	mAP@.5
0.222	0.327	0.115

- F1 CURVE



OBSERVATION:
F1-SCORE ACROSS ALL CLASSES IS 0.07 WHEN THE CONFIDENCE THRESHOLD IS SET AT 0.019. THIS VALUE SEEMS LOW, WHICH MAY INDICATE THAT YOUR MODEL IS NOT PERFORMING WELL IN TERMS OF BALANCING PRECISION AND RECALL.

PRECISION-RECALL CURVE



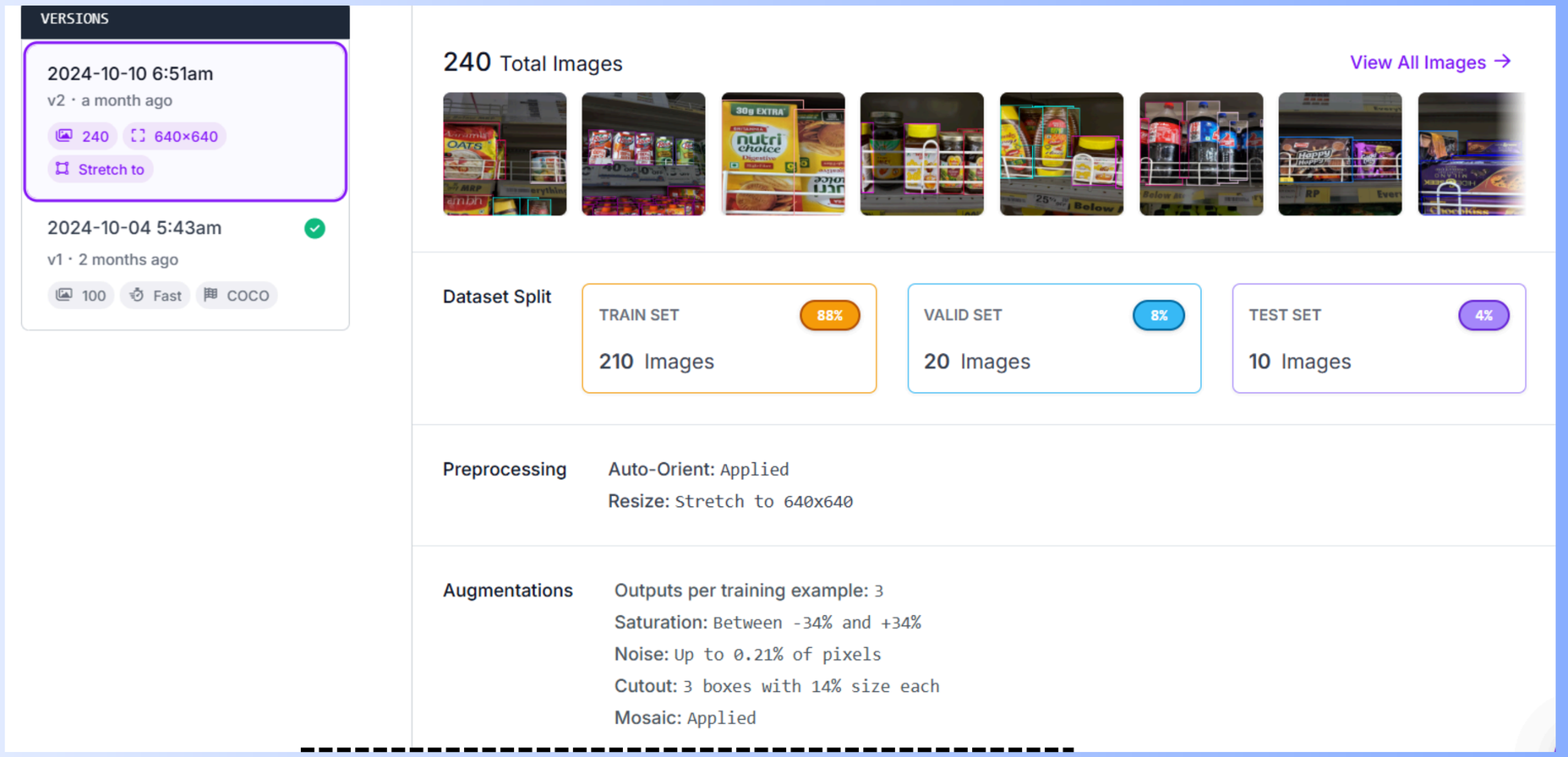
OBSERVATION:

FROM THE PLOT, THE PRECISION DROPS RAPIDLY AS RECALL INCREASES. THIS SUGGESTS THAT AS THE MODEL BECOMES MORE LENIENT (I.E., INCREASES RECALL), IT STARTS MAKING MORE FALSE POSITIVE DETECTIONS, LEADING TO LOWER PRECISION.

CONCLUSION

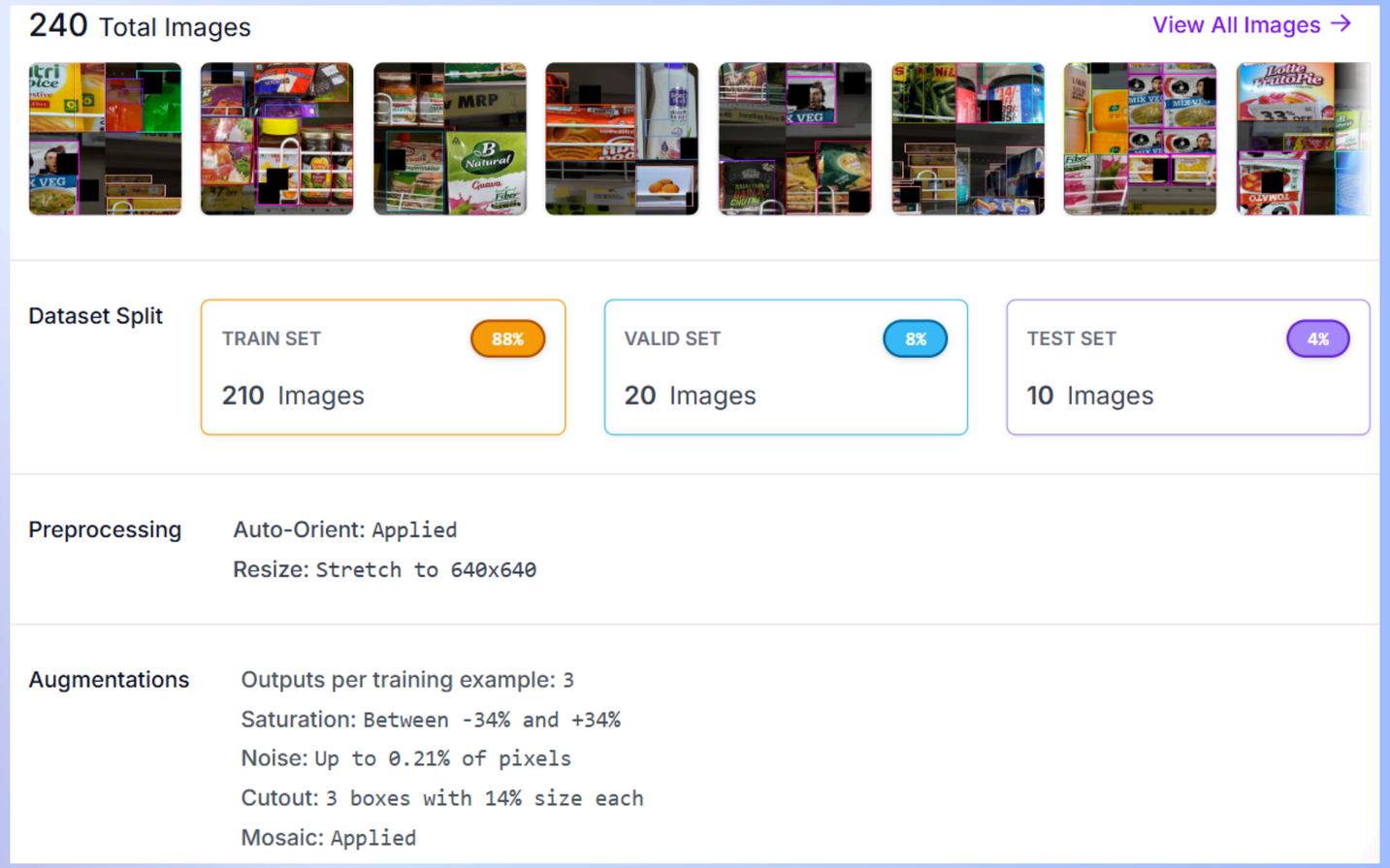
- THE MODEL STRUGGLES TO MAINTAIN PRECISION AS RECALL INCREASES.
- MOST OF THE PRECISION SCORES ARE BELOW 0.4, SUGGESTING A HIGH FALSE POSITIVE RATE.
- LOW F1 SCORES INDICATE A LACK OF BALANCE BETWEEN PRECISION AND RECALL, MEANING THE MODEL MAY EITHER MISS OBJECTS (LOW RECALL) OR MISCLASSIFY NON-OBJECTS AS OBJECTS (LOW PRECISION).
- THE FACT THAT F1 SCORES DON’T RISE SIGNIFICANTLY AT ANY CONFIDENCE LEVEL SUGGESTS THAT OVERALL DETECTION QUALITY IS POOR .

USING DATA AUGMENTATION : TO IMPROVE THE MODEL



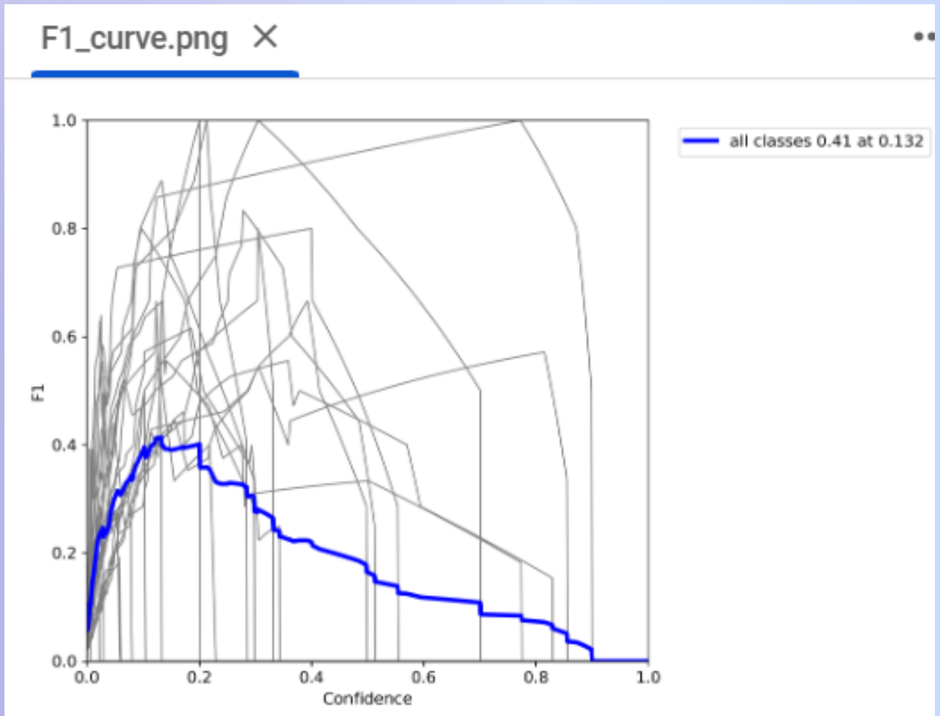
CREATING DIFFERENT VERSIONS OF THE MODEL USING DATA AUGMENTATION:

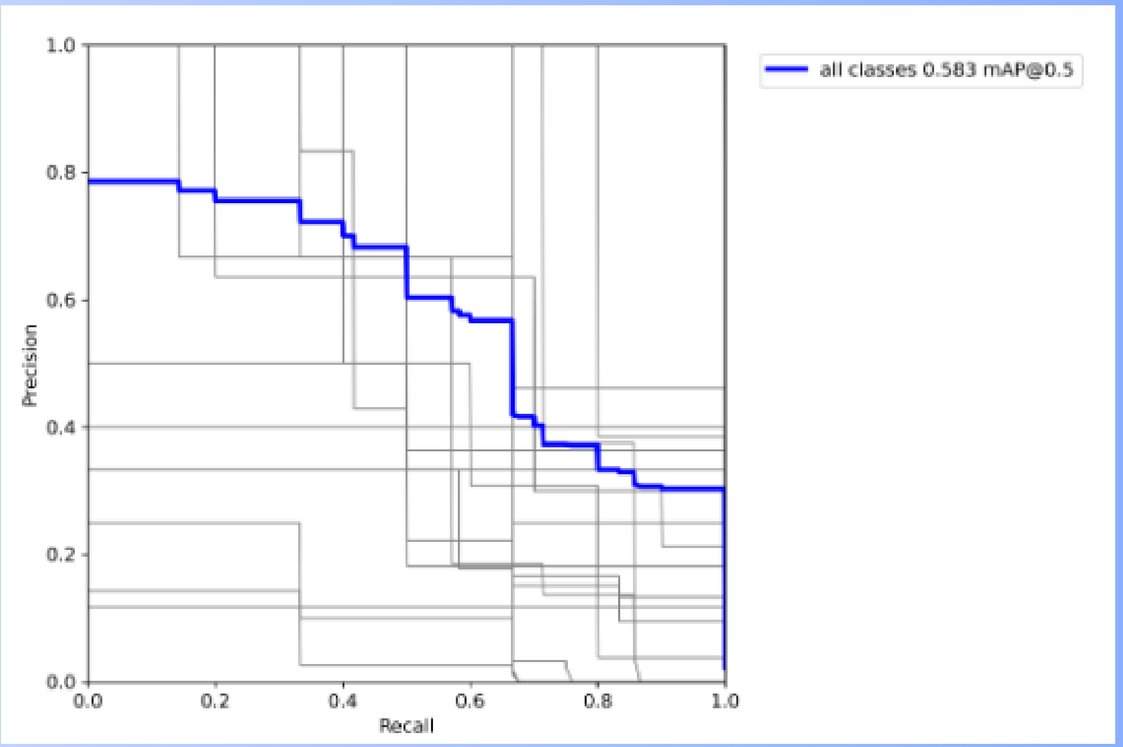
V2:



RESULTS OBTAINED FROM V2:

P	R	mAP@.5	mAP@.5:.95:
0.524	0.547	0.578	0.388





V3:

121 Total Images

View All Images →

Dataset Split

TRAIN SET	VALID SET	TEST SET
75%	17%	8%
91 Images	**20** Images	**10** Images

Preprocessing

Auto-Orient: `Applied`

Resize: `Stretch to 640x640`

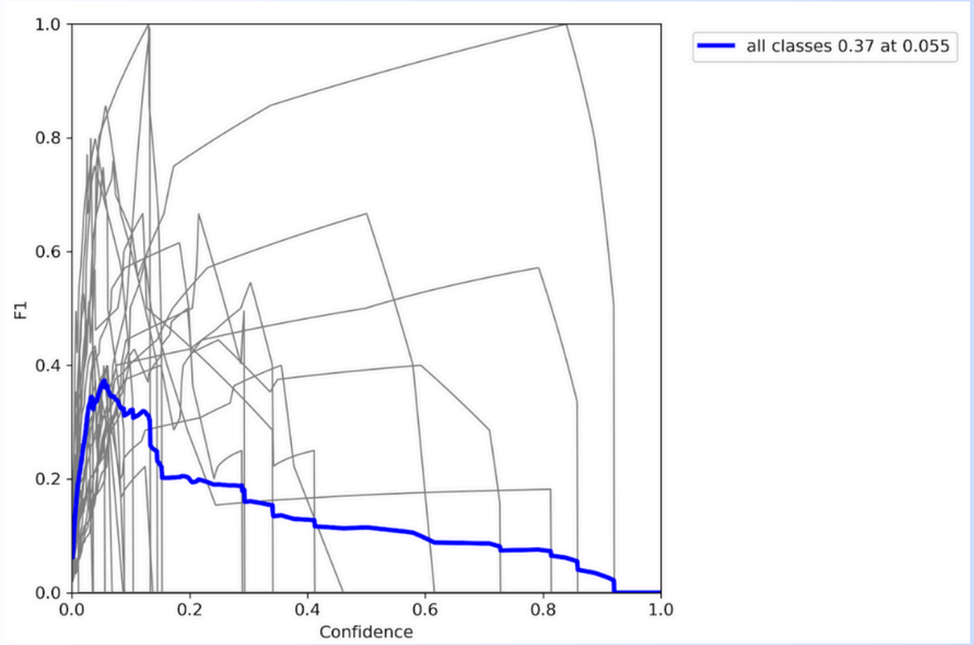
Augmentations

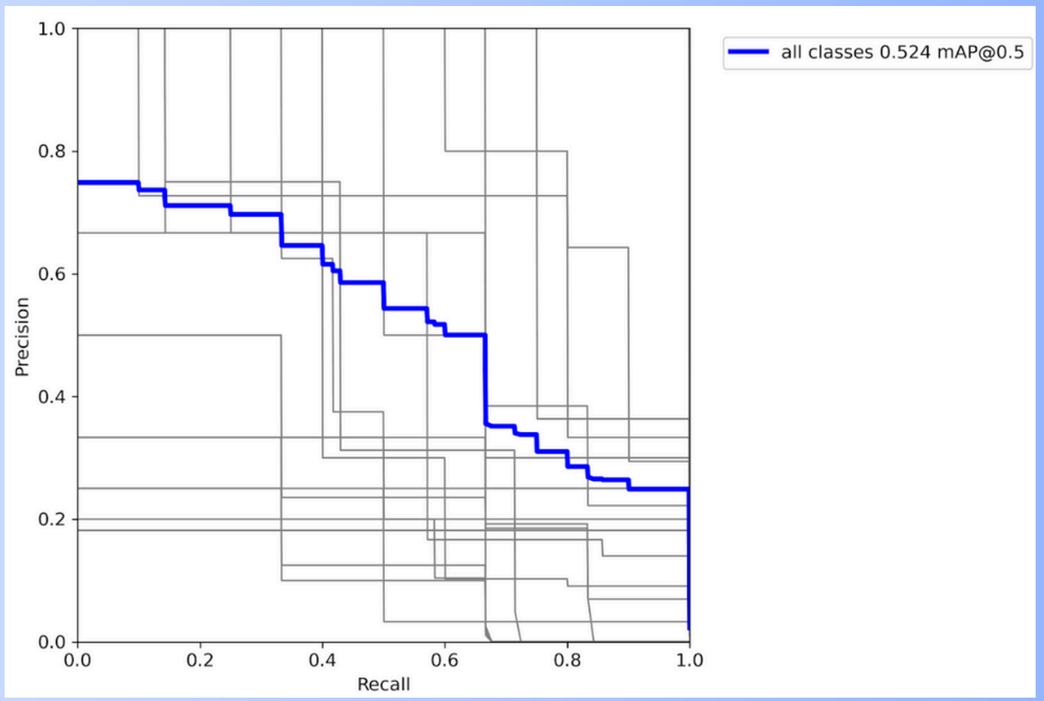
Outputs per training example: `3`

Grayscale: `Apply to 15% of images`

RESULTS OBTAINED FROM V3:

P	R	mAP@.5	mAP@.5:.95:
0.771	0.435	0.584	0.396





V4:

240 Total Images

View All Images →

Dataset Split

TRAIN SET	VALID SET	TEST SET
88%	8%	4%
210 Images	20 Images	10 Images

Preprocessing

Auto-Orient: `Applied`

Resize: `Stretch to 640x640`

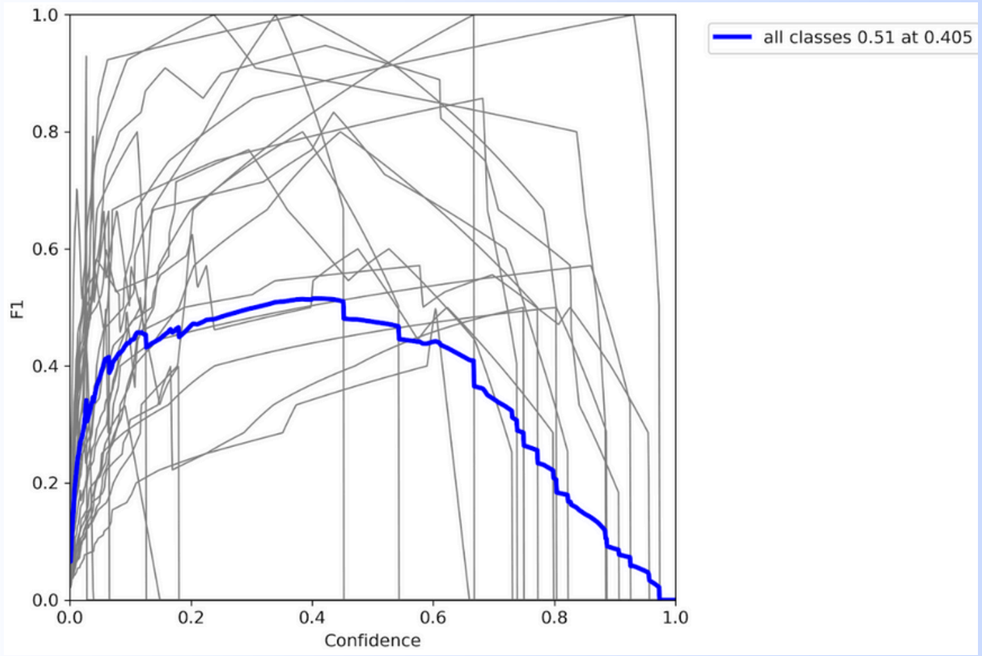
Augmentations

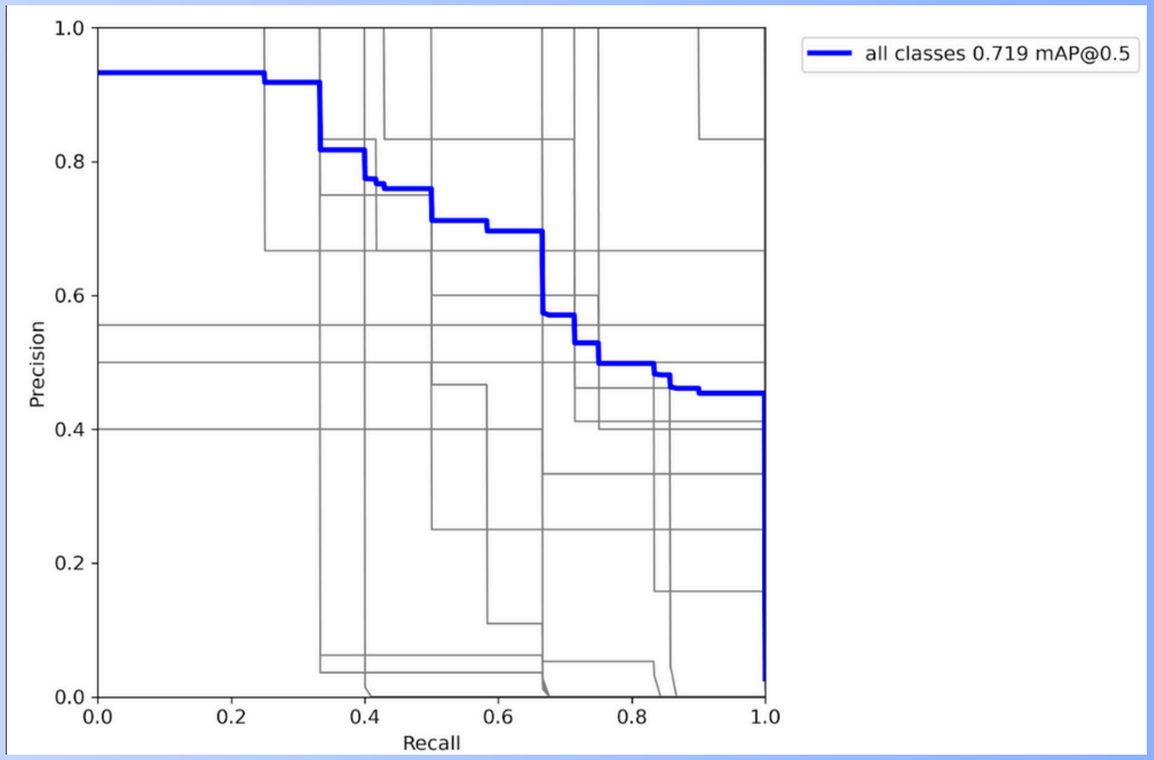
Outputs per training example: `3`

Mosaic: `Applied`

RESULTS OBTAINED FROM V4:

P	R	mAP@.5	mAP@.5:.95:
0.792	0.634	0.746	0.531





CONCLUSION

- SUCCESSFULLY ENHANCED OBJECT DETECTION MODEL THROUGH DATA AUGMENTATION.
- AMONG ALL THE MODELS V4 IS PERFORMING BEST WHICH CAN BE USED FOR OBJECT DETECTION TASK

REAL-LIFE USE CASE OF THIS MODEL:

RETAIL INVENTORY MANAGEMENT

- *USE CASE*: RETAIL STORES CAN USE THE MODEL TO AUTOMATE STOCK COUNTING AND INVENTORY MANAGEMENT. THE MODEL CAN SCAN SHELVES OR STORAGE AREAS TO IDENTIFY AND COUNT THE NUMBER OF PRODUCTS, DETECTING LOW STOCK OR MISPLACED ITEMS.

- *IMPACT*: REDUCES HUMAN LABOR, MINIMIZES ERRORS, AND ENSURES MORE ACCURATE INVENTORY RECORDS, LEADING TO BETTER STOCK MANAGEMENT, REORDERING, AND AVOIDING STOCKOUTS

THANK YOU