Module: ITI107 Deep Learning Networks

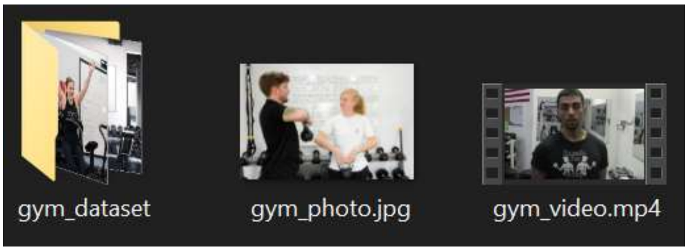
Name: Nursharinah Binte Sohaimi

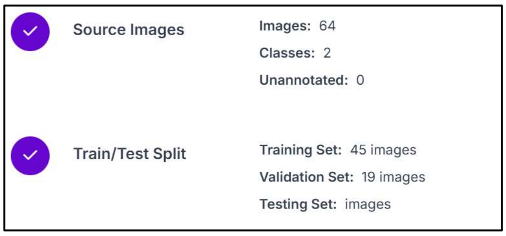
Student ID: 6422706H

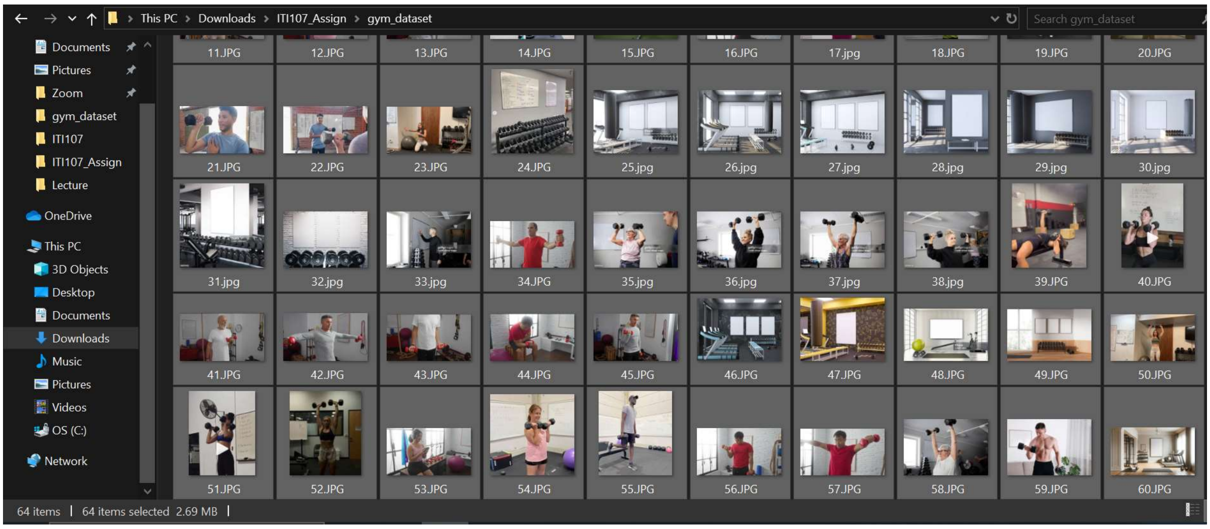
1. Data Collection & Annotation

Data Collection

The choice of data collection is based on an environment of a gym setting. The purpose is to use YOLO version 8 to identify and locate two classes of data: (1) dumbbell and (2) whiteboard that are commonly found in a gym. The custom gym dataset has a total of 64 images of jpeg format. It ranges from showcasing the gym environment and people exercising. These images are randomly found with Google Images search engine. For the test dataset, a separate gym photo and gym video (of 1 minute duration) are collected.







Annotation

For annotation of the images, Roboflow was used to label the two classes. The annotated gym dataset is then split into training & validation dataset with ratio 70:30. It is then exported to be placed on personal GitHub repository for ease of AI modelling using Google Collab platform.

Roboflow project ID: mygymproject
URL: https://universe.roboflow.com/shaiti107deep-learning-network/mygymproject/dataset/1

GitHub repository:
https://github.com/thmasquerade07/SpecialistDiploma_Repository/blob/c3e6ac76ecf193667d7f0a5d226111a6296845d6/SDip%20Applied%20AI/PDC2_Applied%20AI/ITI107_Assign/gym_annotated_dataset.zip

2. Fine-tuning process & Model Performance Comparison

<u>Fine-tuning process</u>

Setting	Initial	Scenario 1	Scenario 2
Epochs	30	50	30
Batch Size	16	16	32
Optimizer	SGD (default)	SGD (default)	AdamW
cos_lr	NA	True	True
lr0	NA	0.01	0.02
lrf	NA	0.0001	0.001

Recommendations:

- Scenario 1 is suitable if the current learning rate seems reasonable & the model benefits from longer training.
- Scenario 2 is recommended as suspect underfitting or want to faster converge while maintaining good generalization.

<u>Model Performance Comparison – Best checkpoint (weights & biases)</u>

i. Initial

```
Ultralytics 8.3.56 🚀 Python-3.10.12 torch-2.5.1+cu121 CPU (Intel Xeon 2.20GHz)
Model summary (fused): 168 layers, 11,126,358 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/valid/labels.cache... 19 images, 0 backgrounds, 0 corrupt: 100%|██████████| 19/19 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:15<00:00,  7.99s/it]
                   all         19        108       0.73      0.747      0.732      0.507
               dumbell         19         84      0.577      0.619      0.567      0.328
            whiteboard         19         24      0.882      0.875      0.897      0.686
Speed: 10.6ms preprocess, 817.1ms inference, 0.0ms loss, 2.1ms postprocess per image
Results saved to runs/detect/val
```

ii. Scenario 1 – Increase Training Time & Adjust Learning Rate Decay

```
Ultralytics 8.3.56 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11,126,358 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/valid/labels.cache... 19 images, 0 backgrounds, 0 corrupt: 100%|██████████| 19/19 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:00<00:00,  2.31it/s]
                   all         19        108      0.722      0.789      0.779      0.542
               dumbell         19         84      0.673      0.702      0.703      0.409
            whiteboard         19         24      0.772      0.875      0.856      0.675
Speed: 0.3ms preprocess, 26.0ms inference, 0.0ms loss, 2.8ms postprocess per image
Results saved to runs/detect/val
```

iii. Scenario 2 – Increase Batch Size & Use a Larger Initial Learning Rate

```
Ultralytics 8.3.56 🚀 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11,126,358 parameters, 0 gradients, 28.4 GFLOPs
val: Scanning /content/datasets/valid/labels.cache... 19 images, 0 backgrounds, 0 corrupt: 100%|██████████| 19/19 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:00<00:00,  2.91it/s]
                   all         19        108     0.0464     0.0536     0.0325     0.0129
               dumbell         19         84     0.0927      0.107     0.0521     0.0214
            whiteboard         19         24          0          0     0.0129    0.00454
Speed: 0.3ms preprocess, 23.1ms inference, 0.0ms loss, 3.5ms postprocess per image
Results saved to runs/detect/val2
```

iv. Conclusion

Model	mAP50 (Validation)	mAP50-95 (Validation)
Initial	0.732	0.507
Scenario 1	0.799	0.542
Scenario 2	0.0325	0.0129

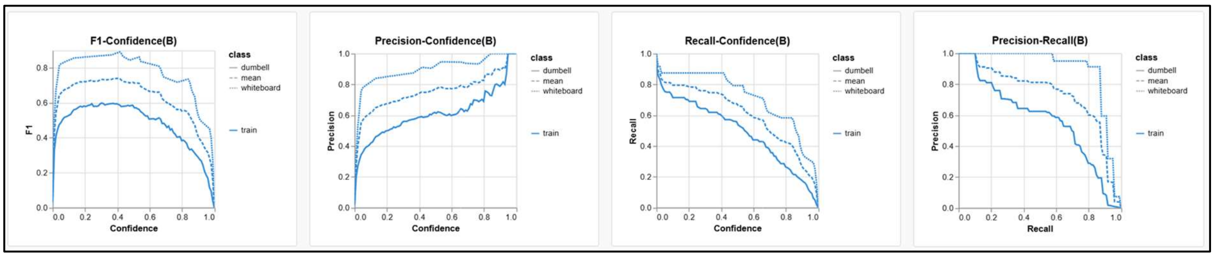
Results shows that Scenario 1 resulted in improvement while Scenario 2 detrimentally reduces the performance of the model, when compared to the initial mode. Note that values are compared using the model that evaluates against the validation dataset after training. Hence, the best model chosen has been tuned to have an increase in training time and adjusted learning rate decay.

Overall, the object identifier model works well but a suggested improvement would be to have images of different perspectives. Currently all the 64 images have both classes, Dumbbells and Whiteboard in it. But as usually Whiteboard is bigger scale than Dumbbells, the model might not be able to identify Dumbbells accurately. This can also be seen in the result where Whiteboard has generally higher Precision and Recall score than Dumbbells, for all scenarios tested. A better dataset that could produce a better model would probably have a ratio of 1:1:1 of images (consist of Dumbbells & Whiteboard) : images (consist of Dumbbells only) : images (consists of Whiteboards only).

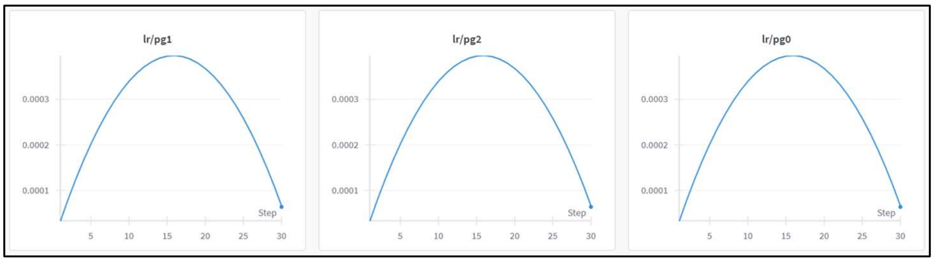
3. Experimental Log – Weights & Biases

Initial

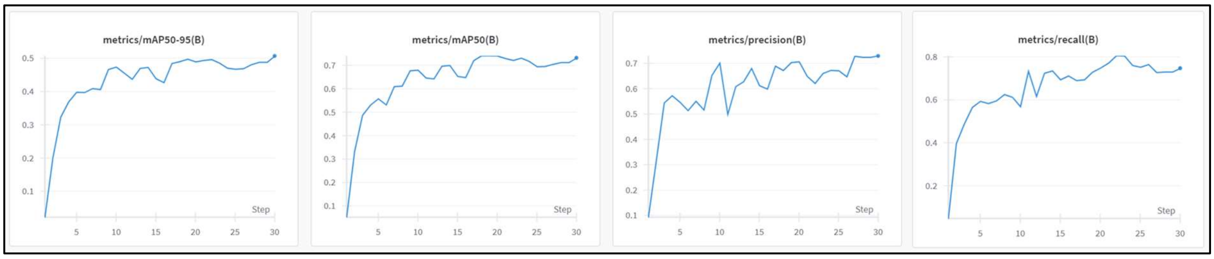
i. Confidence (F1, Precision, Recall, Precision-Recall)



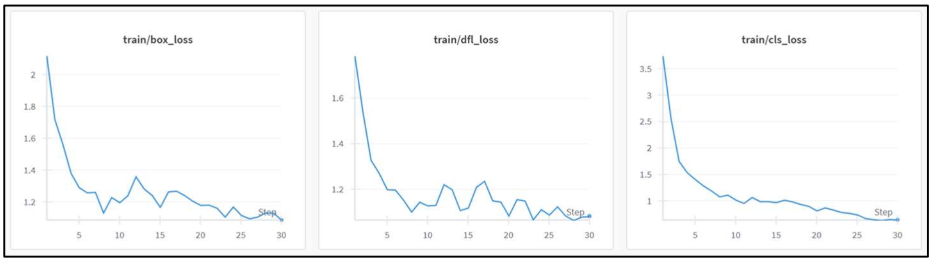
ii. Learning Rate



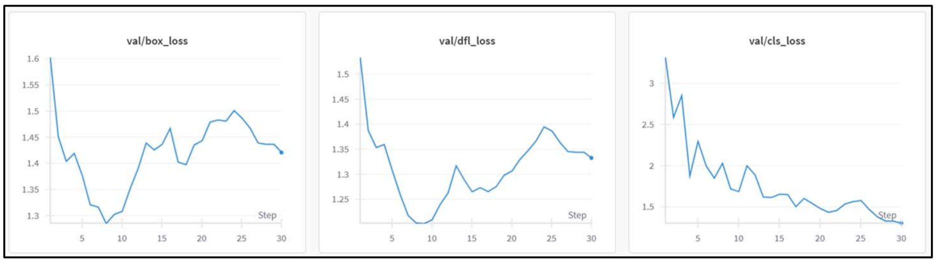
iii. Metric (Precision, Recall mAP50, mAP50-95)



iv. Train Loss

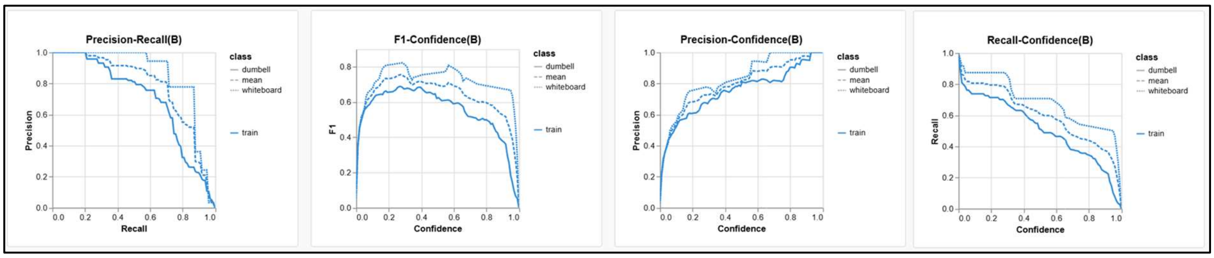


v. Validation Loss

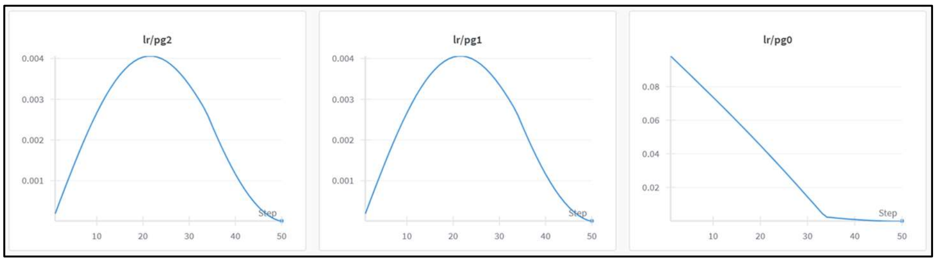


Scenario 1 – Increase Training Time & Adjust Learning Rate Decay

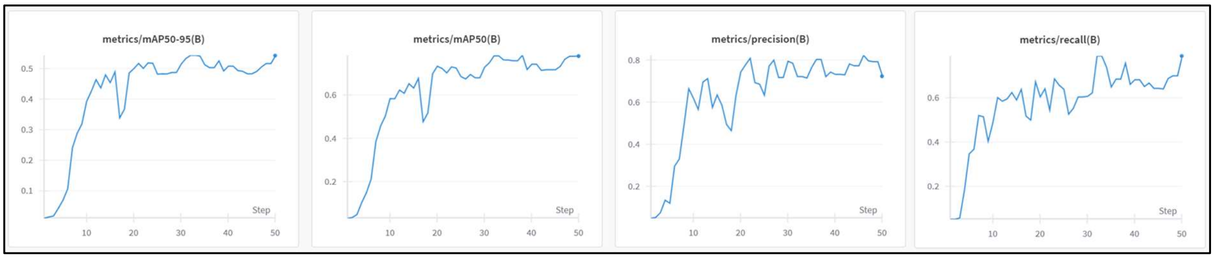
i. Confidence (F1, Precision, Recall, Precision-Recall)



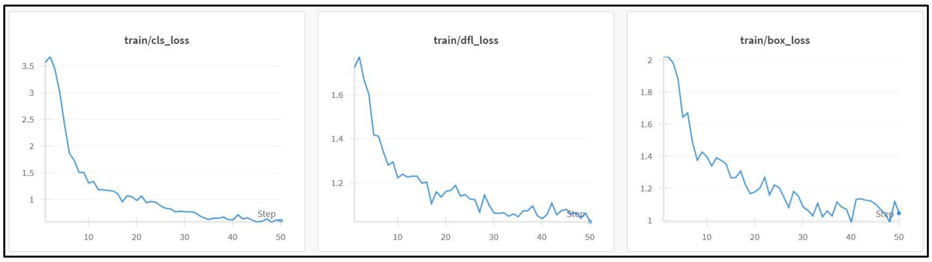
ii. Learning Rate



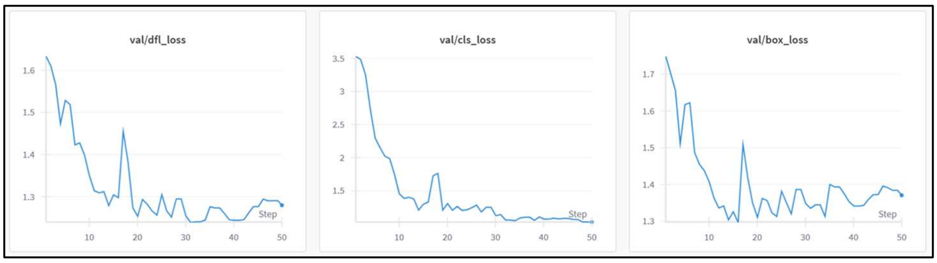
iii. Metric (Precision, Recall mAP50, mAP50-95)



iv. Train Loss

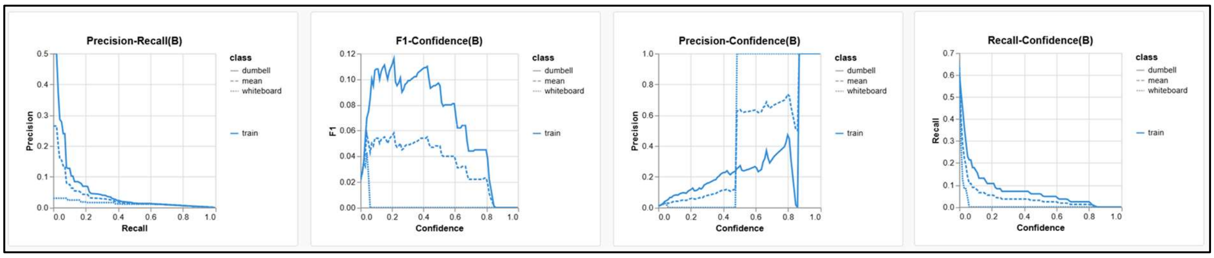


v. Validation Loss

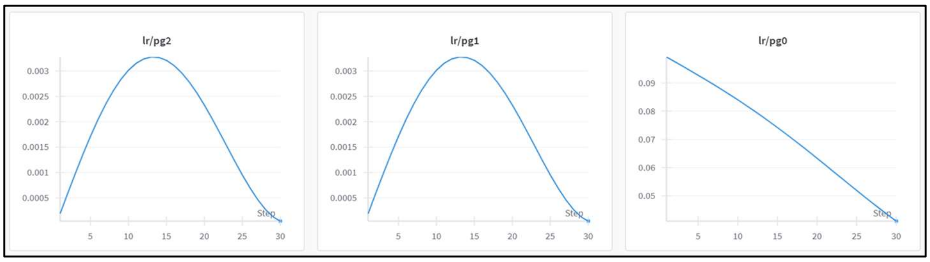


Scenario 2 – Increase Batch Size & Use a Larger Initial Learning Rate

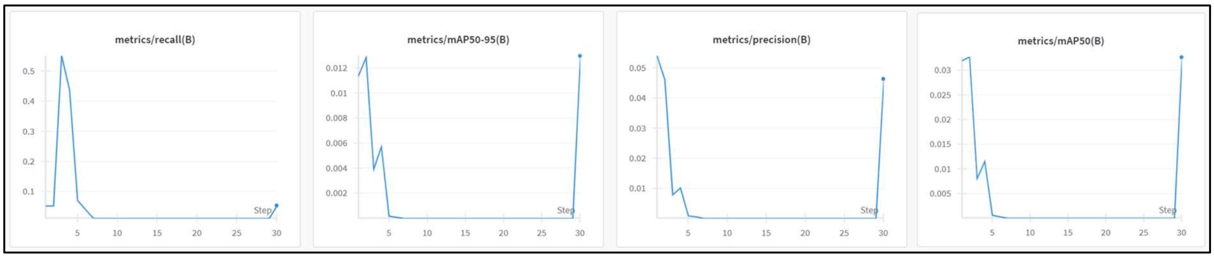
i. Confidence (F1, Precision, Recall, Precision-Recall)



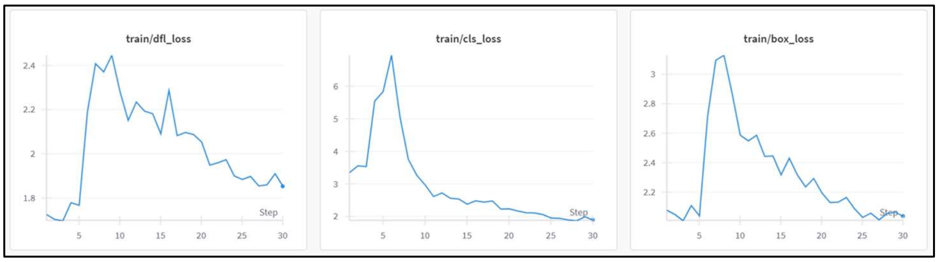
ii. Learning Rate



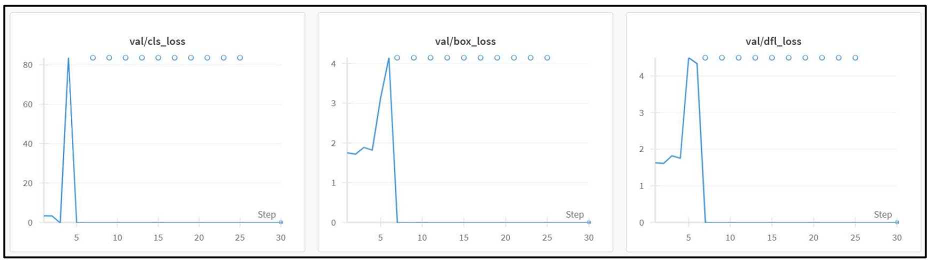
iii. Metric (Precision, Recall mAP50, mAP50-95)



iv. Train Loss



v. Validation Loss



4. Deployment – HuggingFace space

Space Link: git clone https://huggingface.co/spaces/ITI107-2024S2/6422706H

Model Link: https://huggingface.co/thmasquerade07/mygymproject

```
Command Prompt
Microsoft Windows [Version 10.0.19045.5247]
(c) Microsoft Corporation. All rights reserved.

C:\Users\SHARINAH>git clone https://huggingface.co/spaces/ITI107-2024S2/6422706H
'git' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\SHARINAH>
```

I don’t have any developer background so I’m not sure how to resolve this issue on my local laptop. But all the files needed to be created and edited accordingly have been established.