- Nama : Muhammad Taruna
- Email : muhamad.taruna@outlook.co.id

This notebook is to build YoloV8m model to detect custom dataset that include 5 class which is Boots, Glasses, Gloves, Helmet, and Vest. With result :

- Precision : 0.982
- Recall : 0.979

Set-up Notebook

- This notebook run on Google Colab service using GPU
- This notebook use ultralytics as library to handle the model and roboflow as library to take the dataset.

```
!nvidia-smi
```

```
Wed Mar  6 14:52:17 2024
+---------------------------------------------------------------------------------------+
| NVIDIA-SMI 535.104.05             Driver Version: 535.104.05   CUDA Version: 12.2     |
|-----------------------------------------+----------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |         Memory-Usage | GPU-Util  Compute M. |
|                                         |                      |               MIG M. |
|=========================================+======================+======================|
|   0  Tesla T4                       Off | 00000000:00:04.0 Off |                    0 |
| N/A   34C    P8               9W /  70W |      0MiB / 15360MiB |      0%      Default |
|                                         |                      |                  N/A |
+-----------------------------------------+----------------------+----------------------+

+---------------------------------------------------------------------------------------+
| Processes:                                                                            |
|  GPU   GI   CI        PID   Type   Process name                            GPU Memory |
|        ID   ID                                                             Usage      |
|=======================================================================================|
|  No running processes found                                                           |
+---------------------------------------------------------------------------------------+
```

```
!pip install ultralytics==8.0.196
!pip install roboflow
```

```
import os
from IPython import display
display.clear_output()

import ultralytics
from ultralytics import YOLO
from IPython.display import display, Image
ultralytics.checks()

from roboflow import Roboflow

from google.colab import files
```

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Setup complete ✅ (2 CPUs, 12.7 GB RAM, 26.5/78.2 GB disk)
```

```
HOME = os.getcwd()
!mkdir {HOME}/datasets
%cd {HOME}/datasets
```

```
# Censored because the rules got in roboflow
rf = Roboflow(api_key="")
project = rf.workspace("").project("")
version = project.version(1)
dataset = version.download("")
```

```
/content/datasets
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in synapsis-ppe-513 to yolov8:: 100%|██████████| 97235/97235 [00:05<00:00, 18499.40it/s]

Extracting Dataset Version Zip to synapsis-ppe-513 in yolov8:: 100%|██████████| 3128/3128 [00:00<00:00, 3472.72it/s]
```

Dataset Information

Contain 5 Class :

1. Boots
2. Glasses
3. Gloves
4. Helmet
5. Vest

Dissemination of Dataset

- Train Set : 1242 images
- Valid Set : 236 Images
- Tets Set : 80 Images

I apply this Preprocessing and Augmentations :

- Auto-Orientation Images
- Resize to 640 x 480 Pixel
- Noise Salt and Pepper 0.85%

Reason to chose the Dataset :

- Have high Variance images from any angles and point of view. This made model could learn from any angles to predict the class.
- Good Resolution and clear contour to make sure the model could capture the model of each class
- Adding some preprocessing and augmentation to make sure the model capture the pattern of each class even though the visibility not much clear or the change of the orientation images.

Training Model

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8m.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True
```

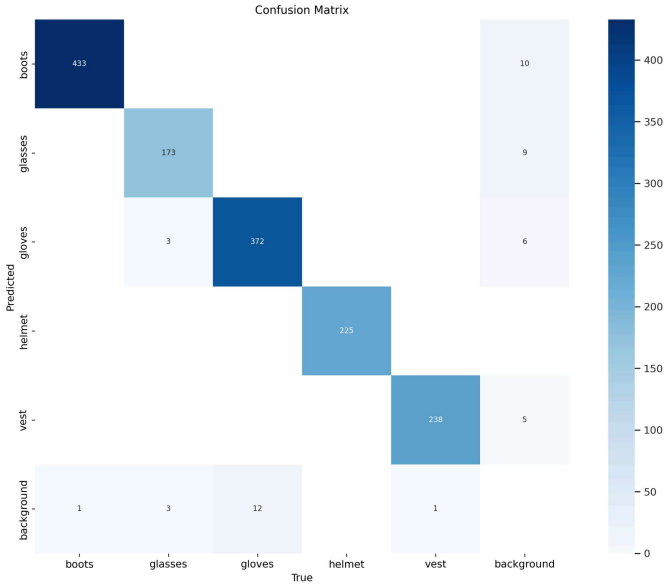
```
ng runs/detect/train/weights/best.pt...
ics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
mmary (fused): 218 layers, 25842655 parameters, 0 gradients, 78.7 GFLOPs
         Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 8/8 [00:11<00:00,  1.45s/it]
           all        236       1461      0.982      0.979      0.988      0.762
         boots        236        434      0.984      0.995      0.995      0.807
       glasses        236        179       0.96      0.943      0.977      0.548
        gloves        236        384      0.981      0.961       0.98      0.753
        helmet        236        225          1      0.999      0.995      0.798
          vest        236        239      0.985      0.996      0.995      0.903
.4ms preprocess, 16.4ms inference, 0.0ms loss, 5.1ms postprocess per image
saved to runs/detect/train
 more at https://docs.ultralytics.com/modes/train
```

```
!ls {HOME}/runs/detect/train/
```

```
args.yaml                                       R_curve.png          val_batch0_labels.jpg
confusion_matrix_normalized.png                 results.csv          val_batch0_pred.jpg
confusion_matrix.png                            results.png          val_batch1_labels.jpg
events.out.tfevents.1709736781.2beaa8d59e55.2321.0 train_batch0.jpg   val_batch1_pred.jpg
F1_curve.png                                    train_batch1170.jpg  val_batch2_labels.jpg
labels_correlogram.jpg                          train_batch1171.jpg  val_batch2_pred.jpg
labels.jpg                                      train_batch1172.jpg  weights
P_curve.png                                     train_batch1.jpg
PR_curve.png                                    train_batch2.jpg
```

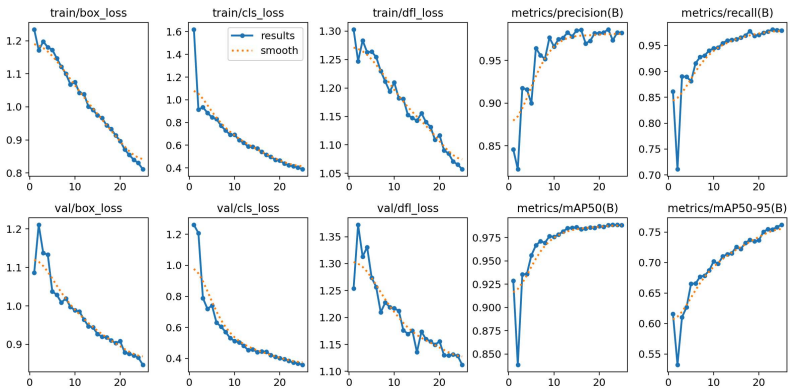
```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/confusion_matrix.png', width=600)
```

```
/content
```



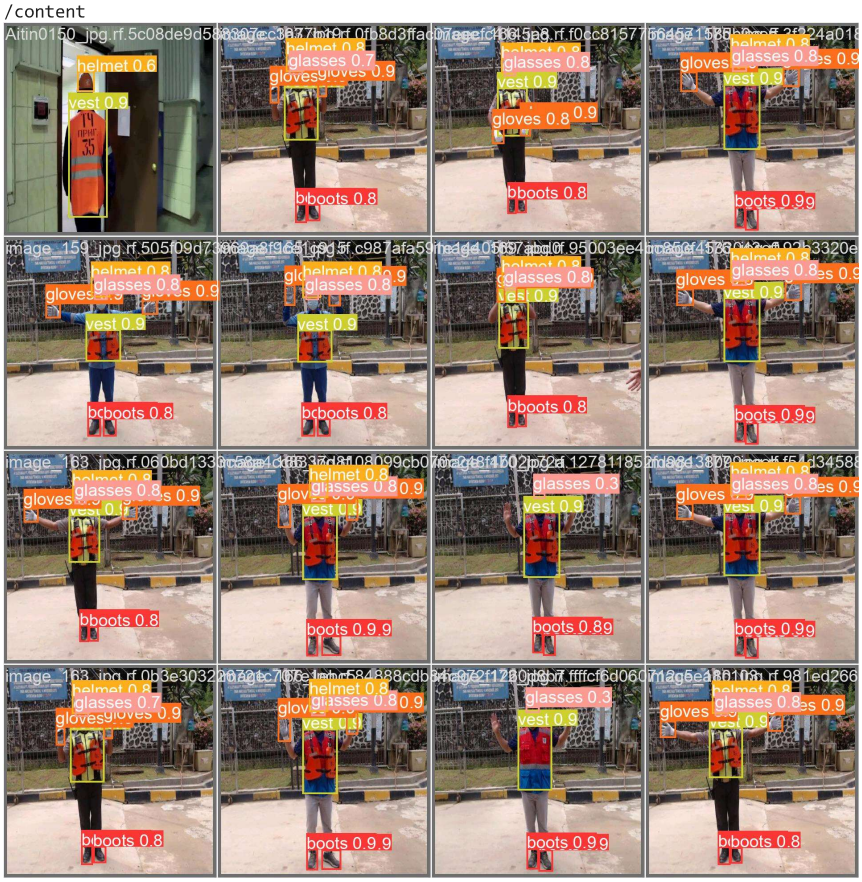

```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/results.png', width=600)
```

```
/content
```




```
%cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/val_batch0_pred.jpg', width=640)
```

```
/content
```



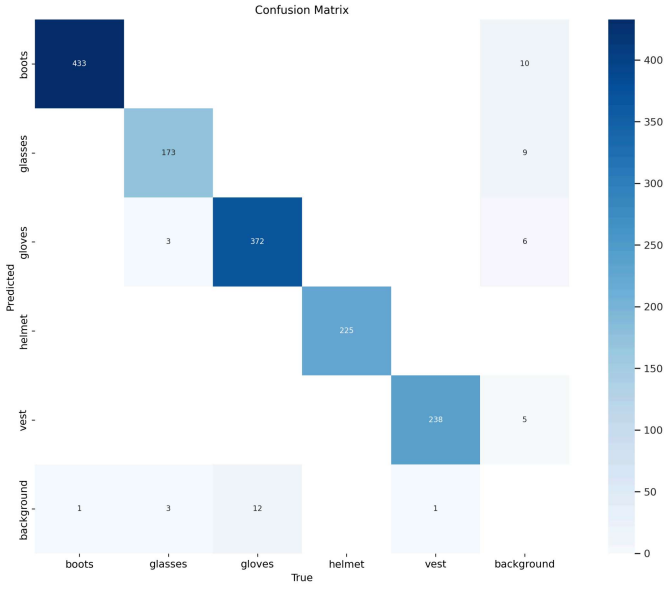
```
%cd {HOME}

!yolo task=detect mode=val model={HOME}/runs/detect/train/weights/best.pt data={dataset.location}/data.yaml
```

```
/content
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.1.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 218 layers, 25842655 parameters, 0 gradients, 78.7 GFLOPs
val: Scanning /content/datasets/synapsis-ppe-513/valid/labels.cache... 236 images, 0 backgrounds, 0 corrupt: 100% 236/236 [00:00<?,
val: WARNING ⚠️ /content/datasets/synapsis-ppe-513/valid/images/Video2_167_jpg.rf.fbdfe1b57d4492bd851c1a738e2d9ce1.jpg: 1 duplicate
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 15/15 [00:14<00:00,  1.00it/s]
                   all        236       1461      0.982      0.979      0.988      0.762
                 boots        236        434      0.982      0.995      0.995      0.807
               glasses        236        179       0.96      0.943      0.977      0.548
                gloves        236        384      0.981      0.961       0.98      0.755
                helmet        236        225          1      0.999      0.995      0.798
                  vest        236        239      0.985      0.996      0.995      0.903
Speed: 2.0ms preprocess, 35.0ms inference, 0.0ms loss, 5.1ms postprocess per image
Results saved to runs/detect/val
💡 Learn more at https://docs.ultralytics.com/modes/val
```

```
Image(filename=f'{HOME}/runs/detect/val/val_batch0_pred.jpg', width=640)
```

```
Image(filename=f'{HOME}/runs/detect/val/confusion_matrix.png', width=600)
```

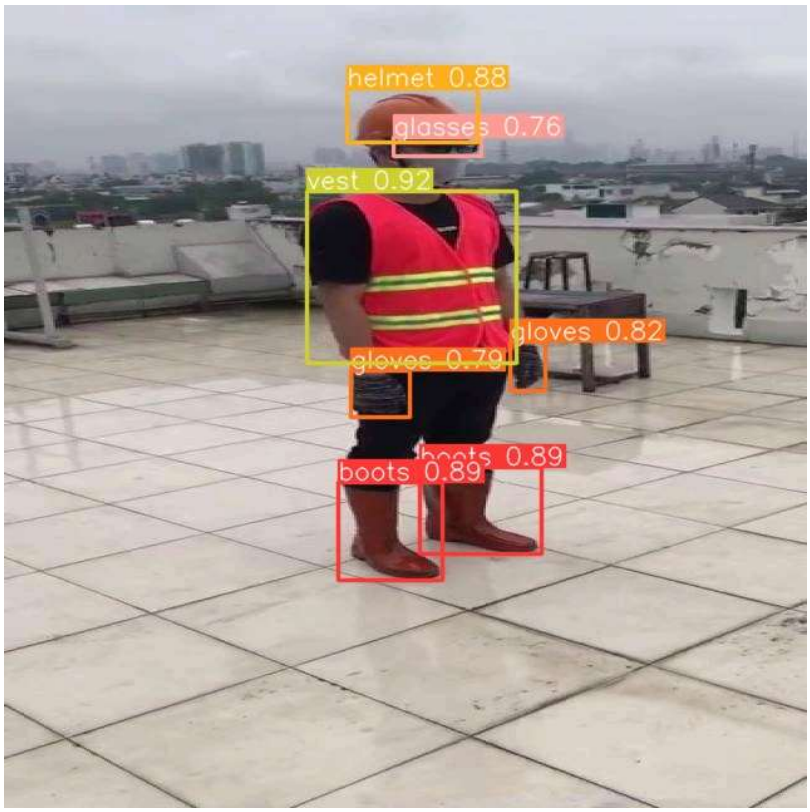


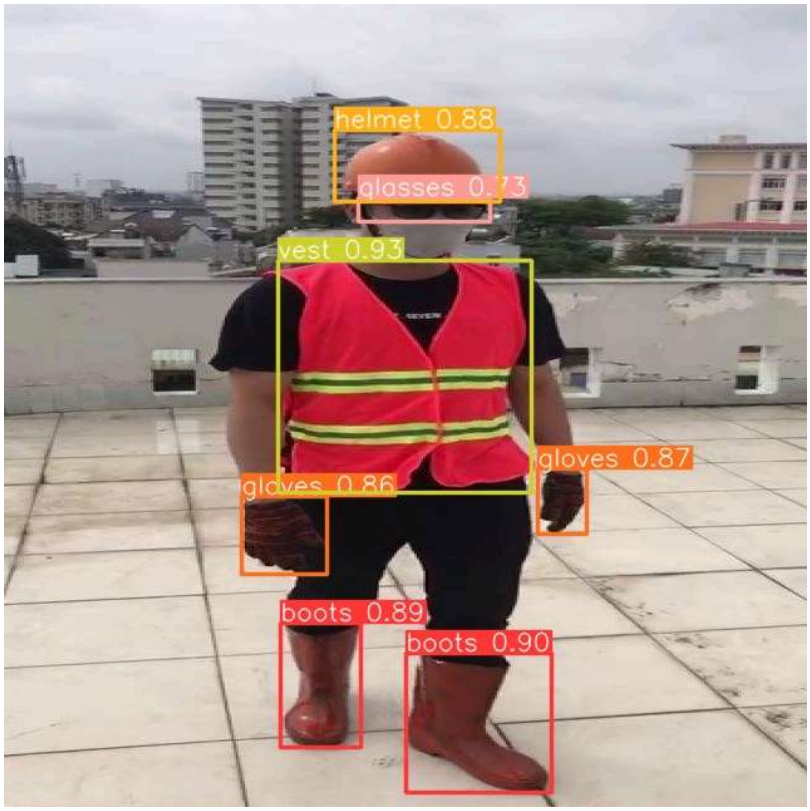
Validate the Model

```
%cd {HOME}
!yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source={dataset.location}/test/images save=True
```

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

for image_path in glob.glob(f'{HOME}/runs/detect/predict/*.jpg')[:3]:
      display(Image(filename=image_path, width=600))
      print("\n")
```


helmet 0.88
glasses 0.76
vest 0.92
gloves 0.82
gloves 0.79
boots 0.89
boots 0.89


helmet 0.88
vest 0.93
gloves 0.87
boots 0.89
boots 0.90


helmet 0.86
glasses 0.75
vest 0.93
gloves 0.85
gloves 0.83

Deploy and Save at External Env

```
# Save to Local Computer
from zipfile import ZipFile
import os

file_paths =[]

for root, directories, files in os.walk('runs'):
  for filename in files:
    filepath = os.path.join(root,filename)
    file_paths.append(filepath)


with ZipFile('sysnapsis-ppe-version2.zip','w') as zip:
  for file in file_paths:
    zip.write(file)

from google.colab import files
files.download('sysnapsis-ppe-version2.zip')
```