COMSATS UNIVERSITY ISLAMABAD,
WAH CAMPUS

Group-Members:

FA21-BCS-066	Hussain Ali
FA21-BCS-080	Ali Shan

Section:

BCS-8A

Project-Title:

Real-time Weapon Detection

Table:

Learning Outcomes:

Hussain Ali

(FA21-BCS-066):

Model Training

Model Validation

Flow-Diagrams

Ali Shan

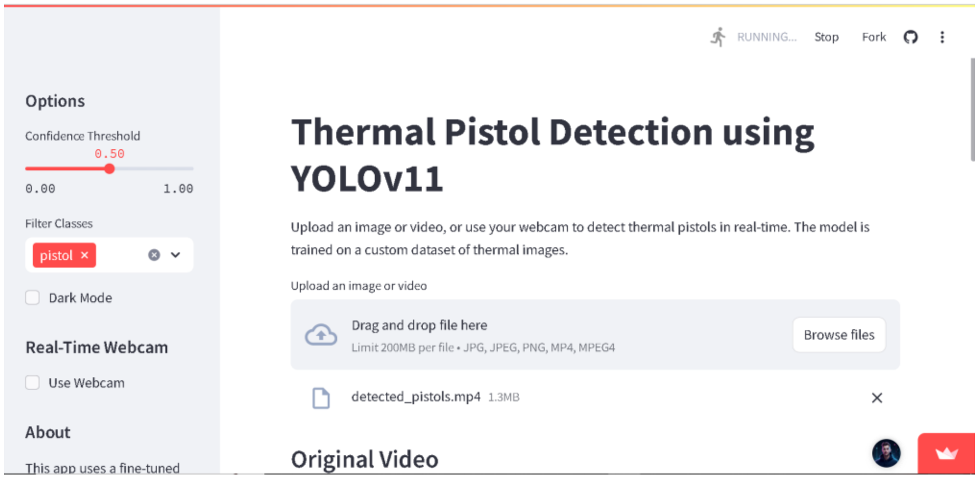
(FA21-BCS-080):

Interface

Model Training

Model Validation

Interface:



RUNNING... Stop Fork

Options

Confidence Threshold 0.50

0.00 1.00

Filter Classes

pistol ×

Dark Mode

Real-Time Webcam

Use Webcam

About

This app uses a fine-tuned

Thermal Pistol Detection using YOLOv11

Upload an image or video, or use your webcam to detect thermal pistols in real-time. The model is trained on a custom dataset of thermal images.

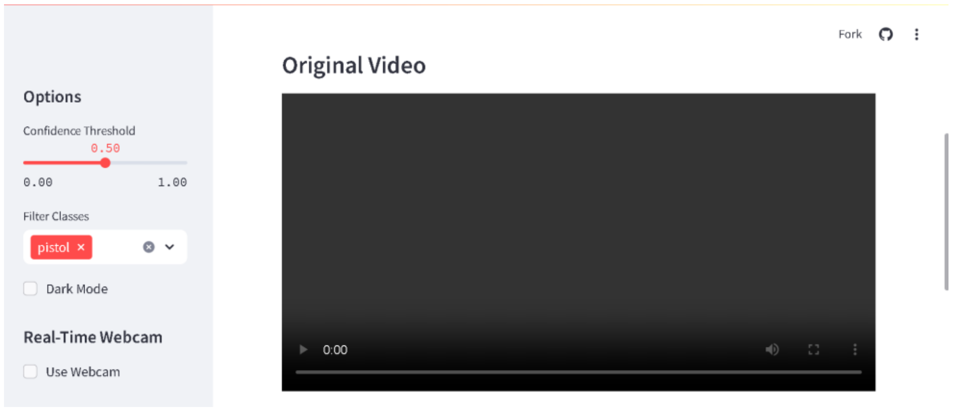
Upload an image or video

Drag and drop file here
Limit 200MB per file • JPG, JPEG, PNG, MP4, MPEG4

Browse files

detected_pistols.mp4 1.3MB

Original Video



Fork

Options

Confidence Threshold 0.50

0.00 1.00

Filter Classes

pistol ×

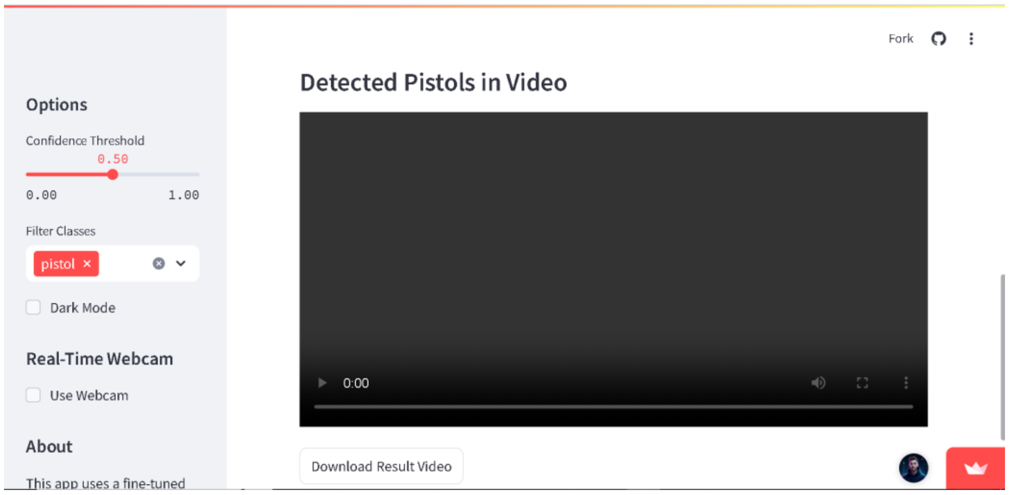
Dark Mode

Real-Time Webcam

Use Webcam

Original Video

0:00



Fork

Options

Confidence Threshold 0.50

0.00 1.00

Filter Classes

pistol ×

Dark Mode

Real-Time Webcam

Use Webcam

About

This app uses a fine-tuned

Detected Pistols in Video

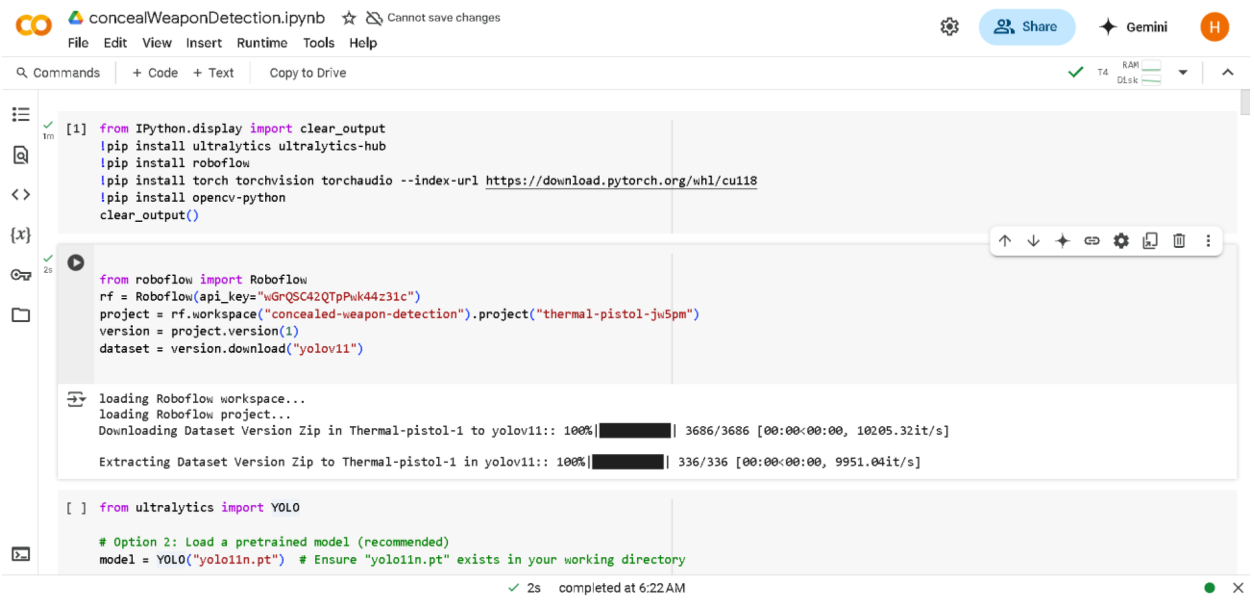
0:00

Download Result Video

Video:

detected_pistols (2).mp4

Model Training



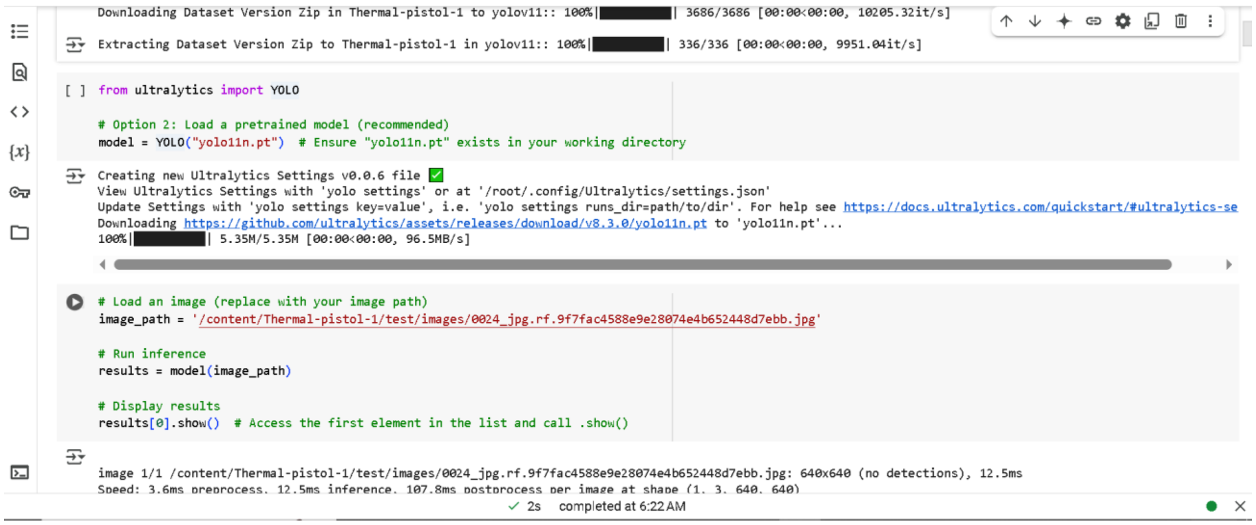
```
[1] from IPython.display import clear_output
    !pip install ultralytics ultralytics-hub
    !pip install roboflow
    !pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu118
    !pip install opencv-python
    clear_output()
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="wGrQSC42QTpPwk44z31c")
project = rf.workspace("concealed-weapon-detection").project("thermal-pistol-jw5pm")
version = project.version(1)
dataset = version.download("yolov11")
```

```
loading Roboflow workspace...
loading Roboflow project...
Downloading Dataset Version Zip in Thermal-pistol-1 to yolov11:: 100%|██████████| 3686/3686 [00:00<00:00, 10205.32it/s]

Extracting Dataset Version Zip to Thermal-pistol-1 in yolov11:: 100%|██████████| 336/336 [00:00<00:00, 9951.04it/s]
```

```
[ ] from ultralytics import YOLO

    # Option 2: Load a pretrained model (recommended)
    model = YOLO("yolo11n.pt")  # Ensure "yolo11n.pt" exists in your working directory
```

✓ 2s completed at 6:22 AM



```
Downloading Dataset Version Zip in Thermal-pistol-1 to yolov11:: 100%|██████████| 3686/3686 [00:00<00:00, 10205.32it/s]

Extracting Dataset Version Zip to Thermal-pistol-1 in yolov11:: 100%|██████████| 336/336 [00:00<00:00, 9951.04it/s]
```

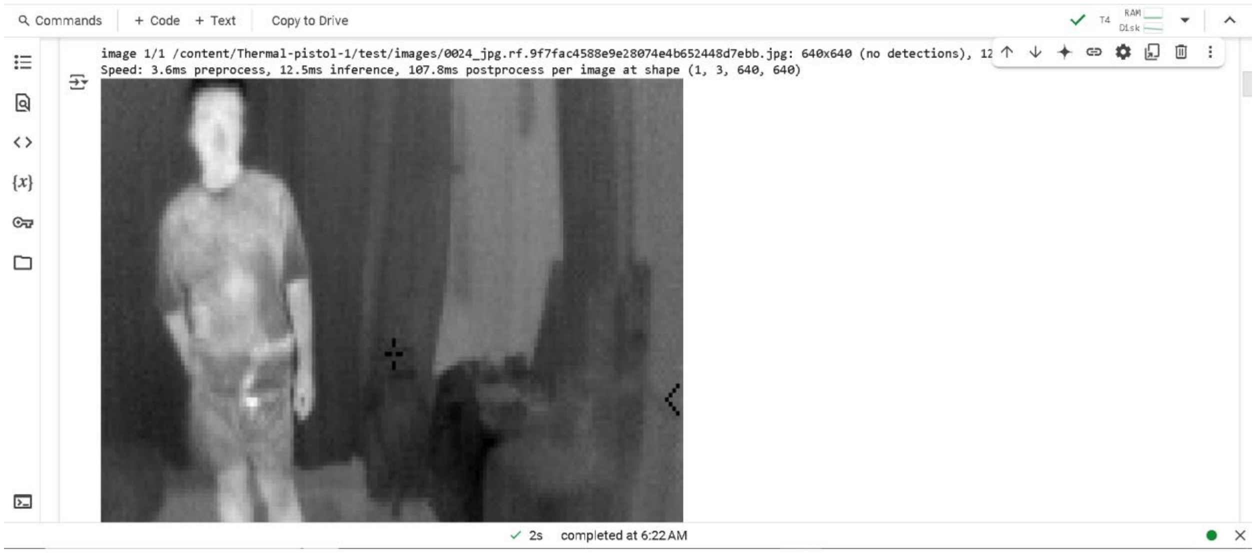
```
[ ] from ultralytics import YOLO

    # Option 2: Load a pretrained model (recommended)
    model = YOLO("yolo11n.pt")  # Ensure "yolo11n.pt" exists in your working directory
```

```
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-se
Downloading https://github.com/ultralytics/assets/releases/download/v8.3.0/yolo11n.pt to 'yolo11n.pt'...
100%|██████████| 5.35M/5.35M [00:00<00:00, 96.5MB/s]
```

```
# Load an image (replace with your image path)
image_path = '/content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg'

# Run inference
results = model(image_path)

# Display results
results[0].show()  # Access the first element in the list and call .show()
```

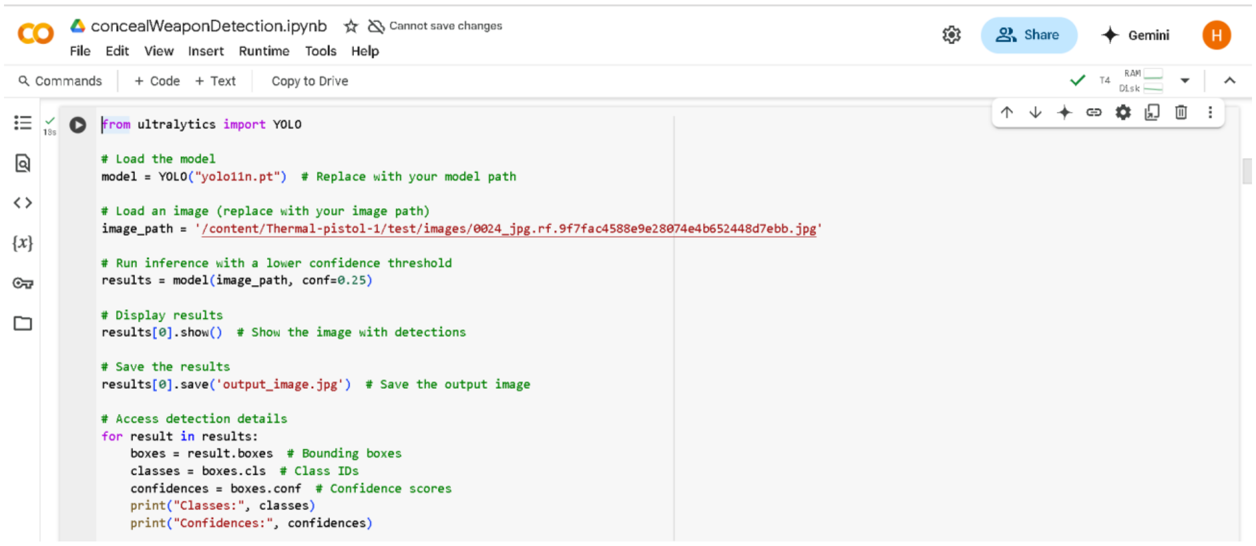
```
image 1/1 /content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg: 640x640 (no detections), 12.5ms
Speed: 3.6ms preprocess, 12.5ms inference, 107.8ms postprocess per image at shape (1, 3, 640, 640)
```

✓ 2s completed at 6:22 AM



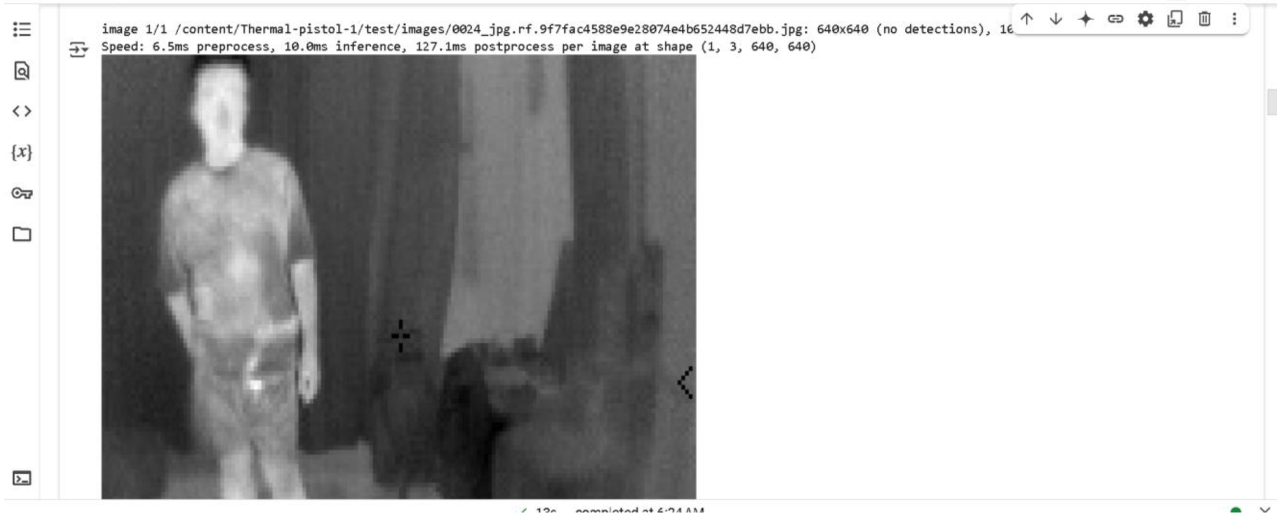
```
image 1/1 /content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg: 640x640 (no detections), 12
Speed: 3.6ms preprocess, 12.5ms inference, 107.8ms postprocess per image at shape (1, 3, 640, 640)
```

✓ 2s completed at 6:22 AM

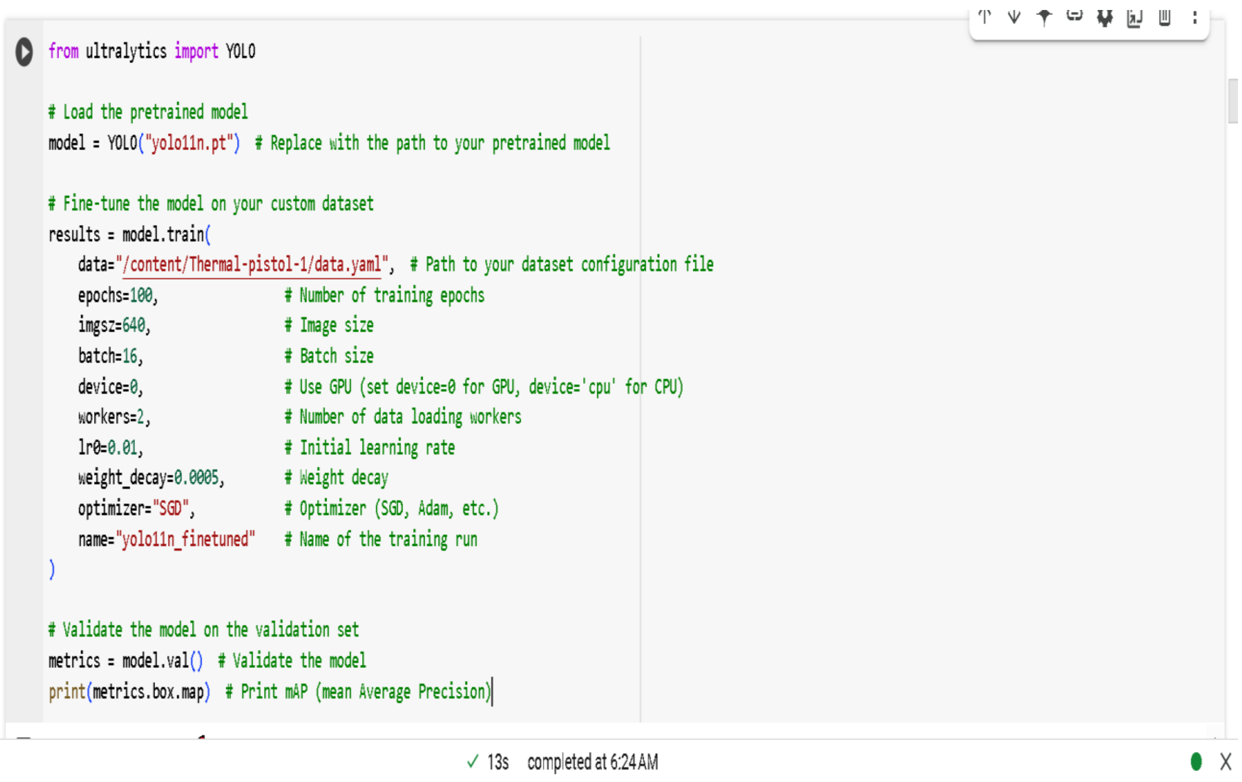


conceaIWeaponDetection.ipynb
Cannot save changes
Share
Gemini
File Edit View Insert Runtime Tools Help
Commands
+ Code
+ Text
Copy to Drive

```
from ultralytics import YOLO

# Load the model
model = YOLO("yolo11n.pt")  # Replace with your model path

# Load an image (replace with your image path)
image_path = '/content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg'

# Run inference with a lower confidence threshold
results = model(image_path, conf=0.25)

# Display results
results[0].show()  # Show the image with detections

# Save the results
results[0].save('output_image.jpg')  # Save the output image

# Access detection details
for result in results:
    boxes = result.boxes  # Bounding boxes
    classes = boxes.cls  # Class IDs
    confidences = boxes.conf  # Confidence scores
    print("Classes:", classes)
    print("Confidences:", confidences)
```



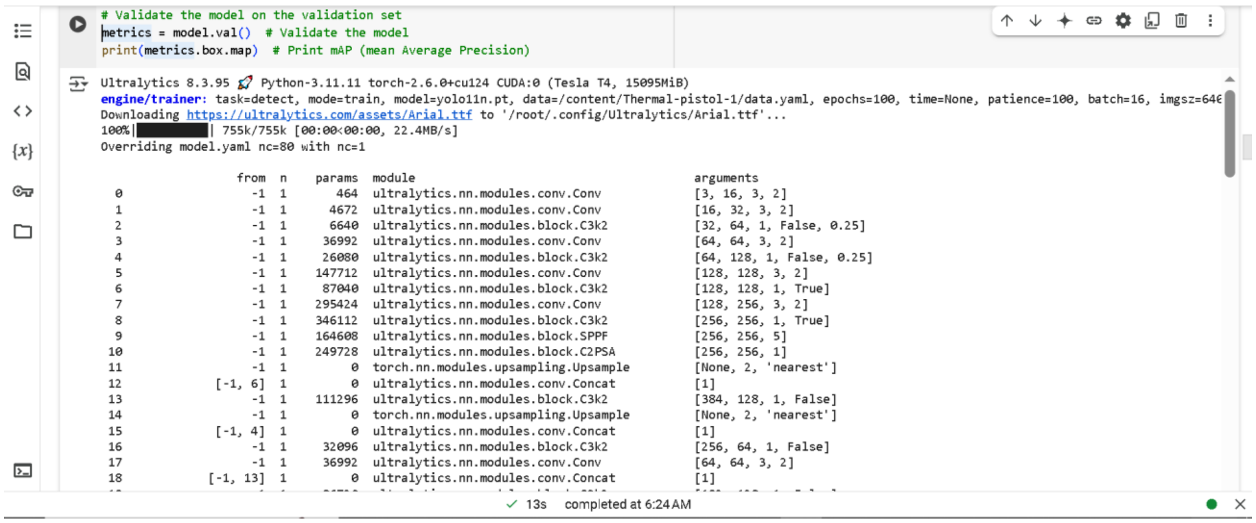
```
image 1/1 /content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg: 640x640 (no detections), 16
Speed: 6.5ms preprocess, 10.0ms inference, 127.1ms postprocess per image at shape (1, 3, 640, 640)
```



```python
from ultralytics import YOLO

# Load the pretrained model
model = YOLO("yolo11n.pt")  # Replace with the path to your pretrained model

# Fine-tune the model on your custom dataset
results = model.train(
    data="/content/Thermal-pistol-1/data.yaml",  # Path to your dataset configuration file
    epochs=100,               # Number of training epochs
    imgsz=640,                # Image size
    batch=16,                 # Batch size
    device=0,                 # Use GPU (set device=0 for GPU, device='cpu' for CPU)
    workers=2,                # Number of data loading workers
    lr0=0.01,                 # Initial learning rate
    weight_decay=0.0005,      # Weight decay
    optimizer="SGD",          # Optimizer (SGD, Adam, etc.)
    name="yolo11n_finetuned"   # Name of the training run
)

# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

✓ 13s   completed at 6:24 AM

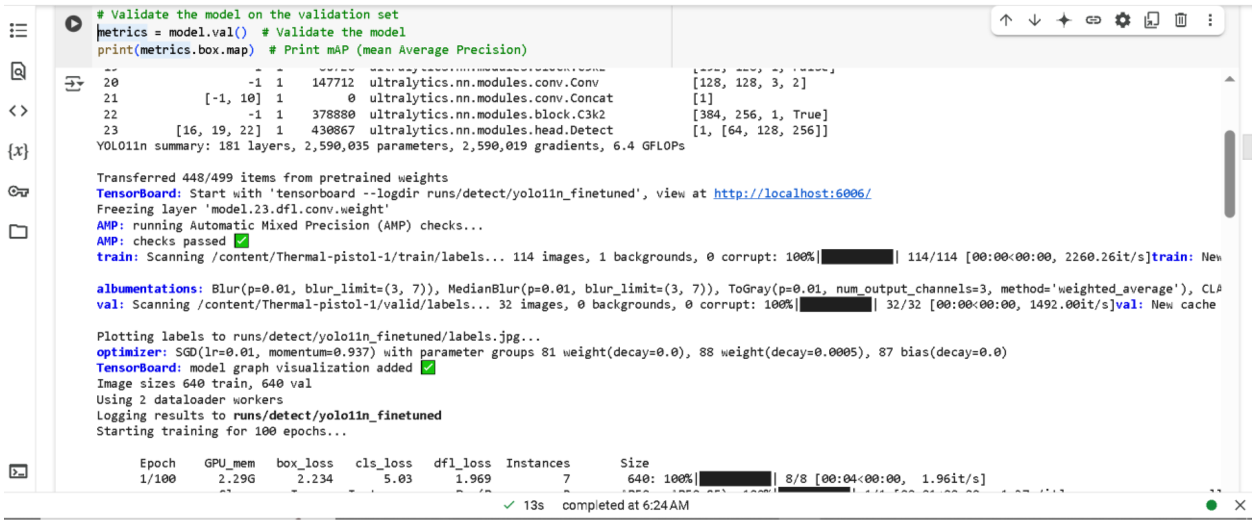
Model Validation



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
Ultralytics 8.3.95 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
engine/trainer: task=detect, mode=train, model=yolo11n.pt, data=/content/Thermal-pistol-1/data.yaml, epochs=100, time=None, patience=100, batch=16, imgsz=64
Downloading https://ultralytics.com/assets/Arial.ttf to '/root/.config/Ultralytics/Arial.ttf'...
100%|██████████| 755k/755k [00:00<00:00, 22.4MB/s]
Overriding model.yaml nc=80 with nc=1

                   from  n    params  module                                       arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv             [3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv             [16, 32, 3, 2]
  2                  -1  1      6640  ultralytics.nn.modules.block.C3k2            [32, 64, 1, False, 0.25]
  3                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
  4                  -1  1     26080  ultralytics.nn.modules.block.C3k2            [64, 128, 1, False, 0.25]
  5                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
  6                  -1  1     87040  ultralytics.nn.modules.block.C3k2            [128, 128, 1, True]
  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv             [128, 256, 3, 2]
  8                  -1  1    346112  ultralytics.nn.modules.block.C3k2            [256, 256, 1, True]
  9                  -1  1    164608  ultralytics.nn.modules.block.SPPF            [256, 256, 5]
 10                  -1  1    249728  ultralytics.nn.modules.block.C2PSA           [256, 256, 1]
 11                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 12             [-1, 6]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 13                  -1  1    111296  ultralytics.nn.modules.block.C3k2            [384, 128, 1, False]
 14                  -1  1         0  torch.nn.modules.upsampling.Upsample         [None, 2, 'nearest']
 15             [-1, 4]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 16                  -1  1     32096  ultralytics.nn.modules.block.C3k2            [256, 64, 1, False]
 17                  -1  1     36992  ultralytics.nn.modules.conv.Conv             [64, 64, 3, 2]
 18            [-1, 13]  1         0  ultralytics.nn.modules.conv.Concat           [1]
```

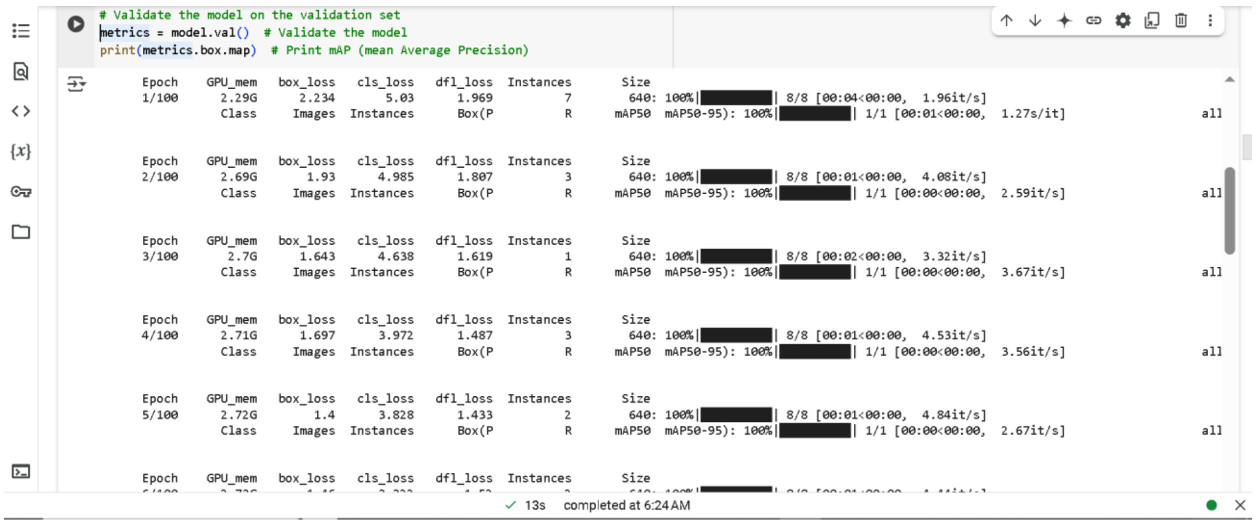
✓ 13s completed at 6:24 AM



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
 20                  -1  1    147712  ultralytics.nn.modules.conv.Conv             [128, 128, 3, 2]
 21            [-1, 10]  1         0  ultralytics.nn.modules.conv.Concat           [1]
 22                  -1  1    378880  ultralytics.nn.modules.block.C3k2            [384, 256, 1, True]
 23        [16, 19, 22]  1    430867  ultralytics.nn.modules.head.Detect           [1, [64, 128, 256]]
YOLO11n summary: 181 layers, 2,590,035 parameters, 2,590,019 gradients, 6.4 GFLOPs

Transferred 448/499 items from pretrained weights
TensorBoard: Start with 'tensorboard --logdir runs/detect/yolo11n_finetuned', view at http://localhost:6006/
Freezing layer 'model.23.dfl.conv.weight'
AMP: running Automatic Mixed Precision (AMP) checks...
AMP: checks passed ✅
train: Scanning /content/Thermal-pistol-1/train/labels... 114 images, 1 backgrounds, 0 corrupt: 100%|██████████| 114/114 [00:00<00:00, 2260.26it/s]train: New

albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01, num_output_channels=3, method='weighted_average'), CLA
val: Scanning /content/Thermal-pistol-1/valid/labels... 32 images, 0 backgrounds, 0 corrupt: 100%|██████████| 32/32 [00:00<00:00, 1492.00it/s]val: New cache

Plotting labels to runs/detect/yolo11n_finetuned/labels.jpg...
optimizer: SGD(lr=0.01, momentum=0.937) with parameter groups 81 weight(decay=0.0), 88 weight(decay=0.0005), 87 bias(decay=0.0)
TensorBoard: model graph visualization added ✅
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to runs/detect/yolo11n_finetuned
Starting training for 100 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      1/100      2.29G      2.234       5.03      1.969          7        640: 100%|██████████| 8/8 [00:04<00:00,  1.96it/s]
```

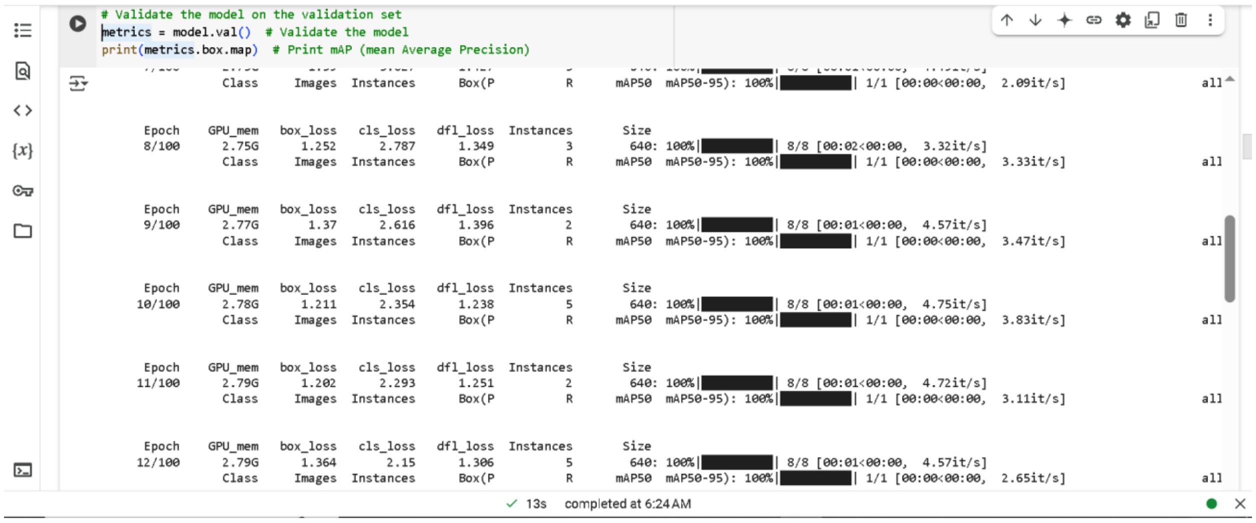
✓ 13s completed at 6:24 AM



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      1/100      2.29G      2.234       5.03      1.969          7        640: 100%|██████████| 8/8 [00:04<00:00,  1.96it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:01<00:00,  1.27s/it]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      2/100      2.69G       1.93      4.985      1.807          3        640: 100%|██████████| 8/8 [00:01<00:00,  4.08it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.59it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      3/100       2.7G      1.643      4.638      1.619          1        640: 100%|██████████| 8/8 [00:02<00:00,  3.32it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.67it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      4/100      2.71G      1.697      3.972      1.487          3        640: 100%|██████████| 8/8 [00:01<00:00,  4.53it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.56it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      5/100      2.72G        1.4      3.828      1.433          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.84it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.67it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
```

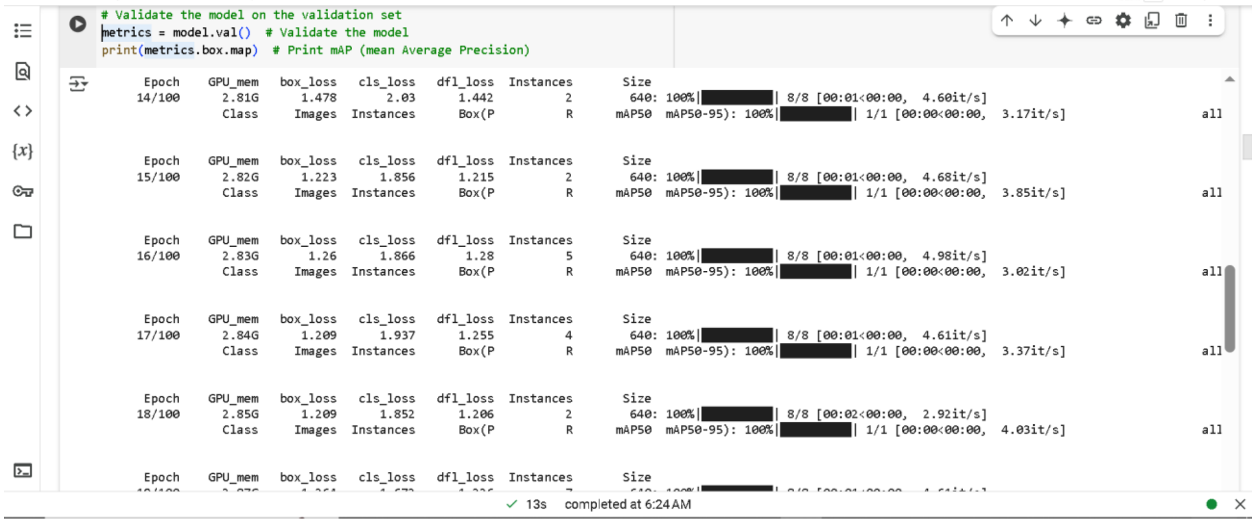
✓ 13s completed at 6:24 AM



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.09it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      8/100      2.75G      1.252      2.787      1.349          3        640: 100%|██████████| 8/8 [00:02<00:00,  3.32it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.33it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      9/100      2.77G       1.37      2.616      1.396          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.57it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.47it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     10/100      2.78G      1.211      2.354      1.238          5        640: 100%|██████████| 8/8 [00:01<00:00,  4.75it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.83it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     11/100      2.79G      1.202      2.293      1.251          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.72it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.11it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     12/100      2.79G      1.364       2.15      1.306          5        640: 100%|██████████| 8/8 [00:01<00:00,  4.57it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.65it/s]                    all
```

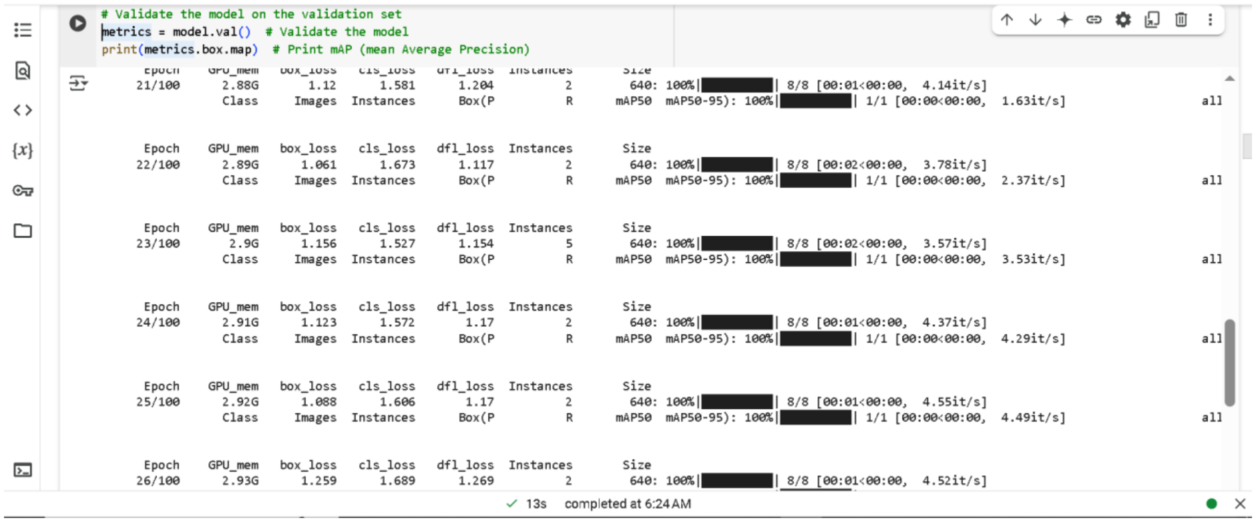
✓ 13s completed at 6:24 AM



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     14/100      2.81G      1.478       2.03      1.442          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.60it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.17it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     15/100      2.82G      1.223      1.856      1.215          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.68it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.85it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     16/100      2.83G       1.26      1.866       1.28          5        640: 100%|██████████| 8/8 [00:01<00:00,  4.98it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.02it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     17/100      2.84G      1.209      1.937      1.255          4        640: 100%|██████████| 8/8 [00:01<00:00,  4.61it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.37it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     18/100      2.85G      1.209      1.852      1.206          2        640: 100%|██████████| 8/8 [00:02<00:00,  2.92it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  4.03it/s]                    all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
```

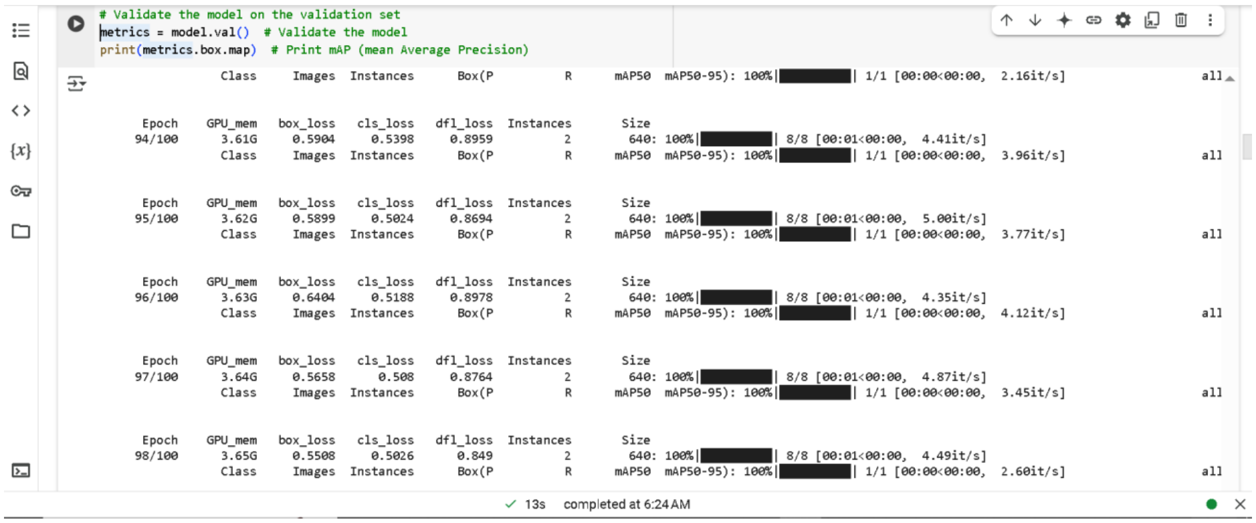
✓ 13s completed at 6:24 AM



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     21/100      2.88G       1.12      1.581      1.204          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.14it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  1.63it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     22/100      2.89G      1.061      1.673      1.117          2        640: 100%|██████████| 8/8 [00:02<00:00,  3.78it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.37it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     23/100       2.9G      1.156      1.527      1.154          5        640: 100%|██████████| 8/8 [00:02<00:00,  3.57it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.53it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     24/100      2.91G      1.123      1.572       1.17          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.37it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  4.29it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     25/100      2.92G      1.088      1.606       1.17          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.55it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  4.49it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     26/100      2.93G      1.259      1.689      1.269          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.52it/s]
```

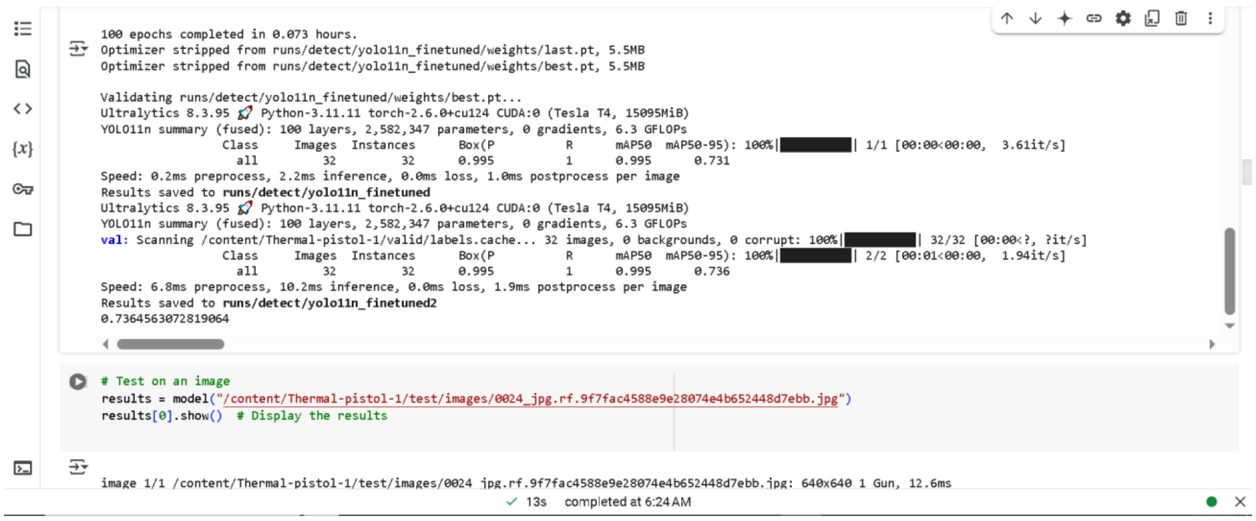
✓ 13s completed at 6:24 AM



```
# Validate the model on the validation set
metrics = model.val()  # Validate the model
print(metrics.box.map)  # Print mAP (mean Average Precision)
```

```
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.16it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     94/100      3.61G     0.5904     0.5398     0.8959          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.41it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.96it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     95/100      3.62G     0.5899     0.5024     0.8694          2        640: 100%|██████████| 8/8 [00:01<00:00,  5.00it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.77it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     96/100      3.63G     0.6404     0.5188     0.8978          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.35it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  4.12it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     97/100      3.64G     0.5658      0.508     0.8764          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.87it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.45it/s]                   all

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     98/100      3.65G     0.5508     0.5026      0.849          2        640: 100%|██████████| 8/8 [00:01<00:00,  4.49it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  2.60it/s]                   all
```

✓ 13s completed at 6:24 AM

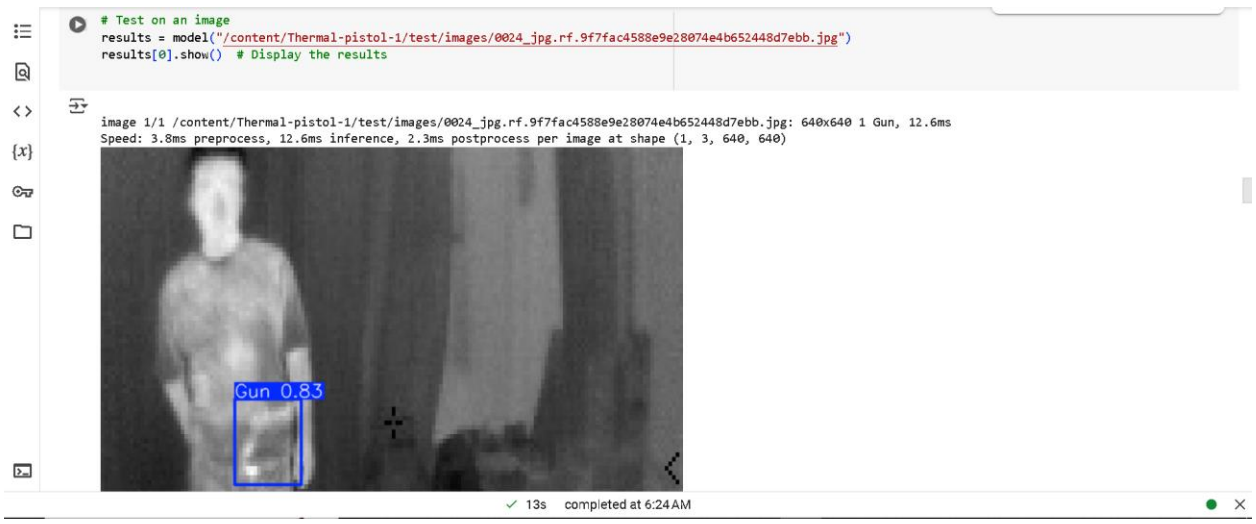


```
100 epochs completed in 0.073 hours.
Optimizer stripped from runs/detect/yolo11n_finetuned/weights/last.pt, 5.5MB
Optimizer stripped from runs/detect/yolo11n_finetuned/weights/best.pt, 5.5MB

Validating runs/detect/yolo11n_finetuned/weights/best.pt...
Ultralytics 8.3.95 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:00<00:00,  3.61it/s]
                   all         32         32      0.995          1      0.995      0.731
Speed: 0.2ms preprocess, 2.2ms inference, 0.0ms loss, 1.0ms postprocess per image
Results saved to runs/detect/yolo11n_finetuned
Ultralytics 8.3.95 🚀 Python-3.11.11 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs
val: Scanning /content/Thermal-pistol-1/valid/labels.cache... 32 images, 0 backgrounds, 0 corrupt: 100%|██████████| 32/32 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 2/2 [00:01<00:00,  1.94it/s]
                   all         32         32      0.995          1      0.995      0.736
Speed: 6.8ms preprocess, 10.2ms inference, 0.0ms loss, 1.9ms postprocess per image
Results saved to runs/detect/yolo11n_finetuned2
0.7364563072819064
```

```
# Test on an image
results = model("/content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg")
results[0].show()  # Display the results
```

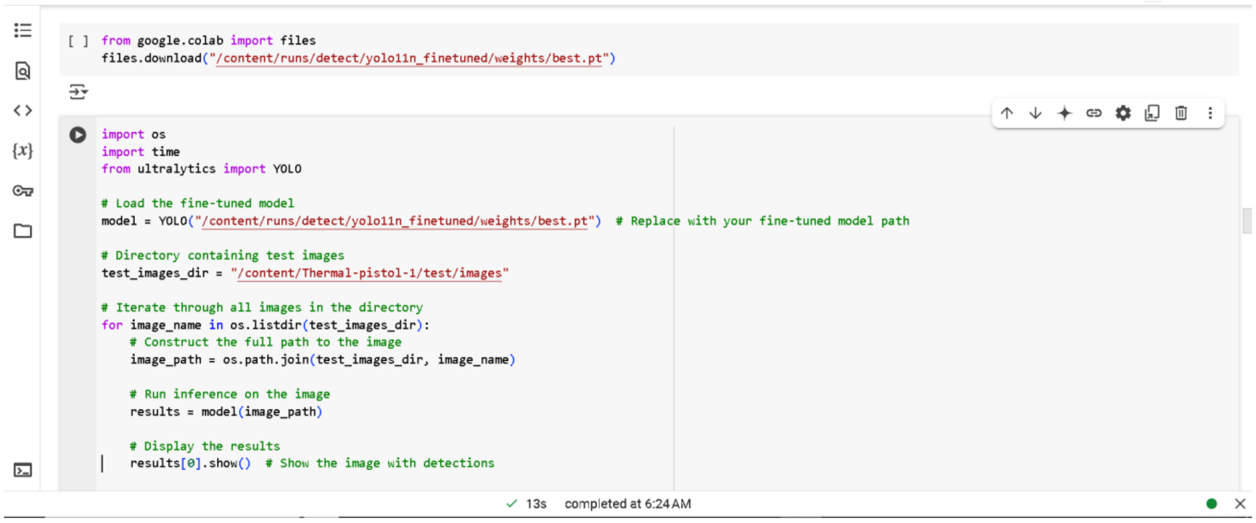
```
image 1/1 /content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg: 640x640 1 Gun, 12.6ms
```

✓ 13s completed at 6:24 AM



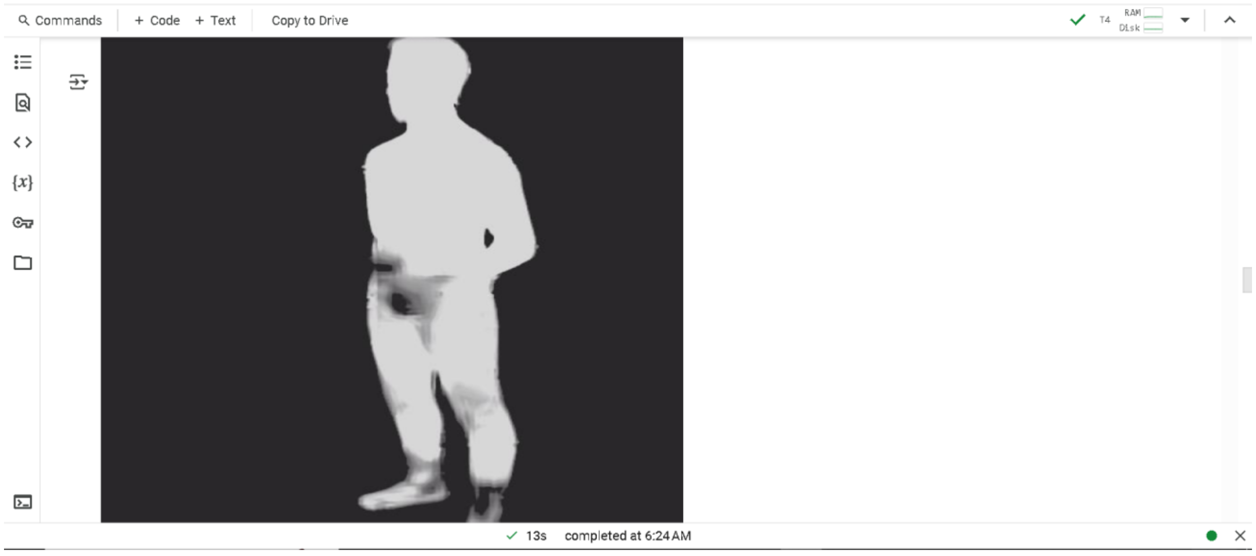
```
# Test on an image
results = model("/content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg")
results[0].show()  # Display the results
```

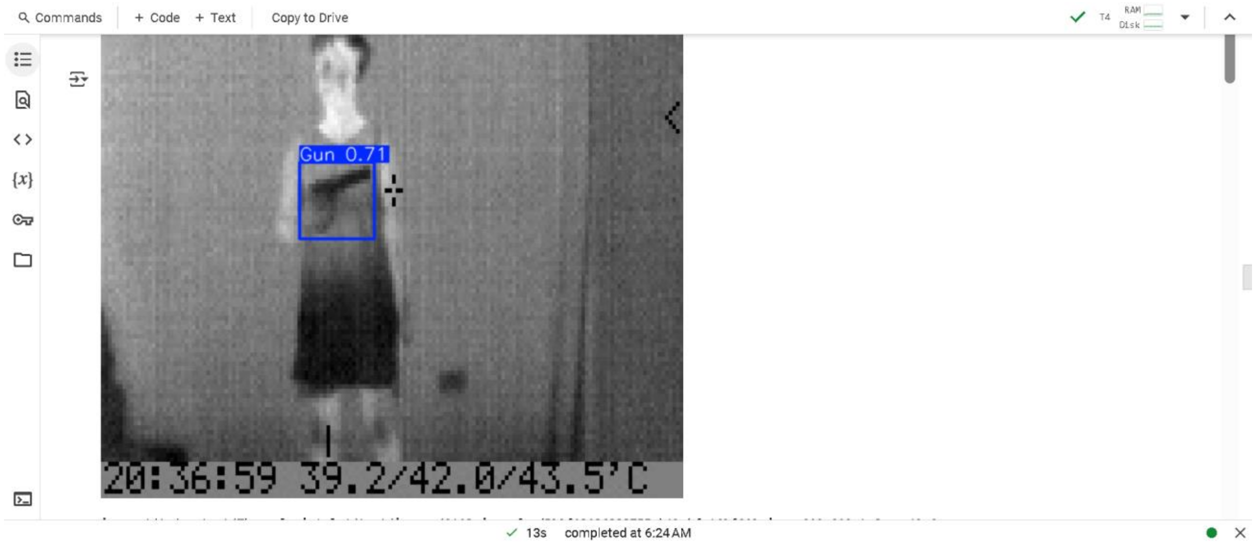
```
image 1/1 /content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg: 640x640 1 Gun, 12.6ms
Speed: 3.8ms preprocess, 12.6ms inference, 2.3ms postprocess per image at shape (1, 3, 640, 640)
```

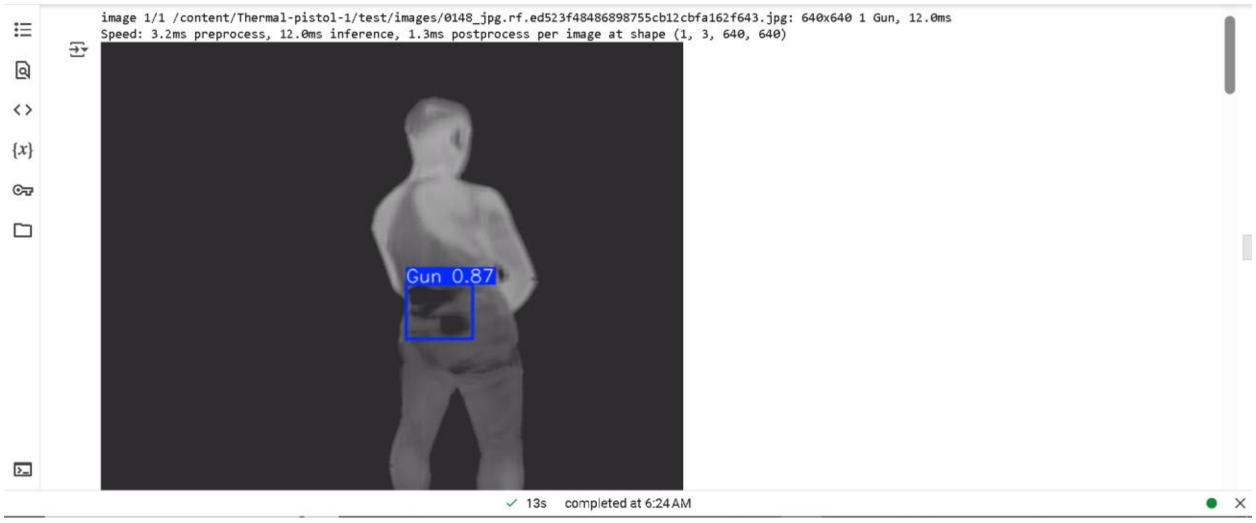


```
from google.colab import files
files.download("/content/runs/detect/yolo11n_finetuned/weights/best.pt")
```

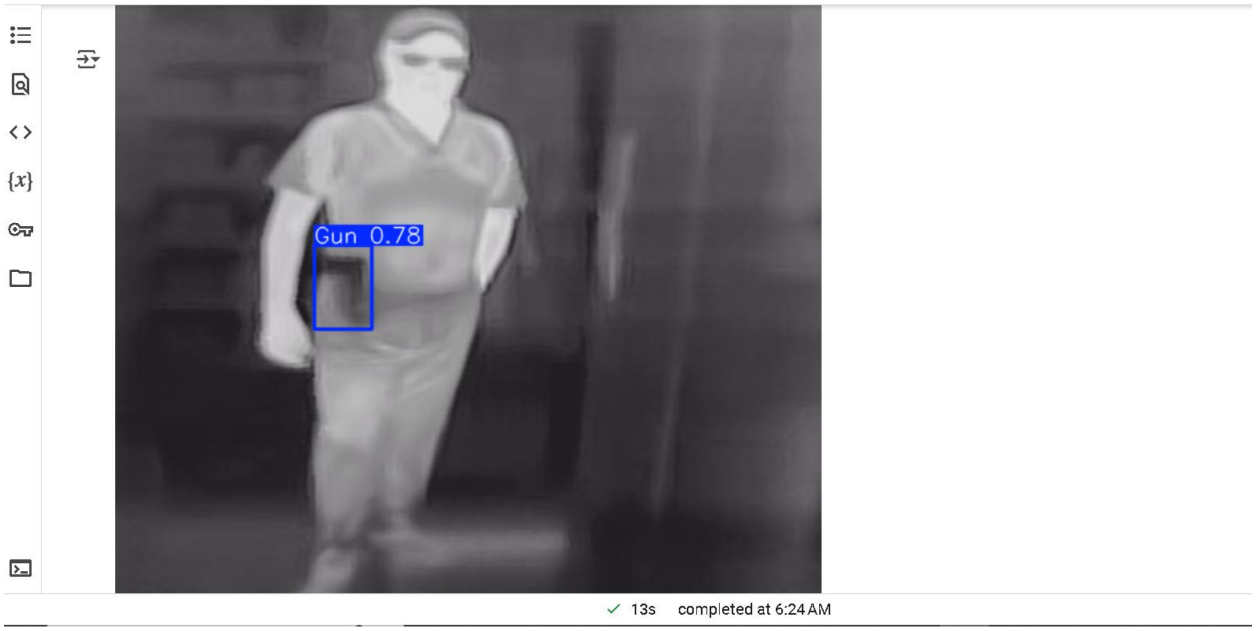
```
import os
import time
from ultralytics import YOLO

# Load the fine-tuned model
model = YOLO("/content/runs/detect/yolo11n_finetuned/weights/best.pt")  # Replace with your fine-tuned model path

# Directory containing test images
test_images_dir = "/content/Thermal-pistol-1/test/images"

# Iterate through all images in the directory
for image_name in os.listdir(test_images_dir):
    # Construct the full path to the image
    image_path = os.path.join(test_images_dir, image_name)

    # Run inference on the image
    results = model(image_path)

    # Display the results
    results[0].show()  # Show the image with detections
```

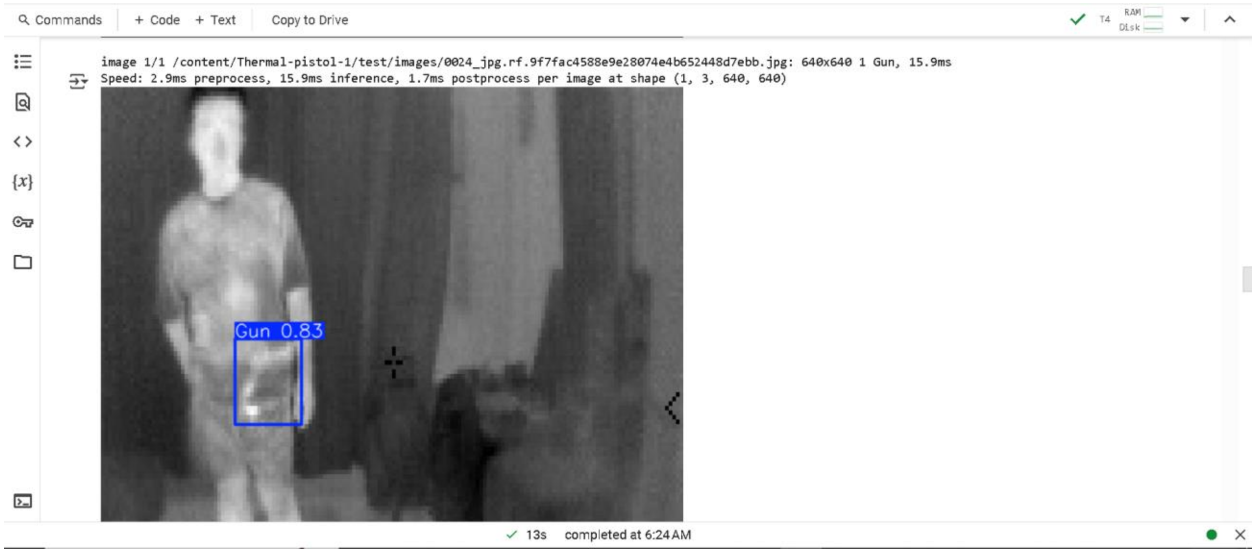




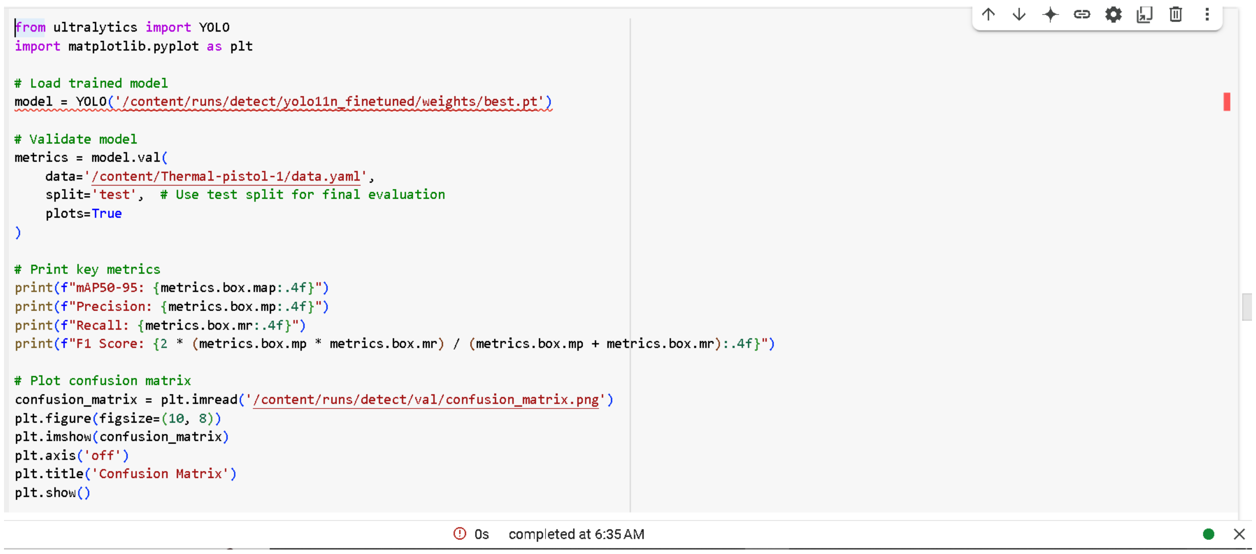


```
image 1/1 /content/Thermal-pistol-1/test/images/0148_jpg.rf.ed523f48486898755cb12cbfa162f643.jpg: 640x640 1 Gun, 12.0ms
Speed: 3.2ms preprocess, 12.0ms inference, 1.3ms postprocess per image at shape (1, 3, 640, 640)
```



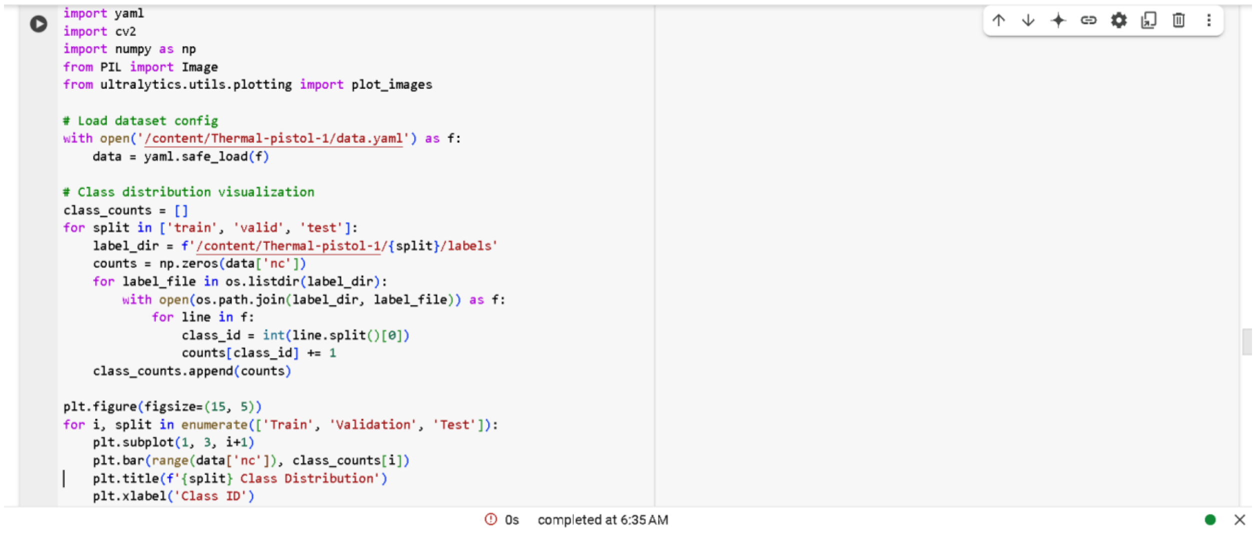


```
image 1/1 /content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg: 640x640 1 Gun, 15.9ms
Speed: 2.9ms preprocess, 15.9ms inference, 1.7ms postprocess per image at shape (1, 3, 640, 640)
```



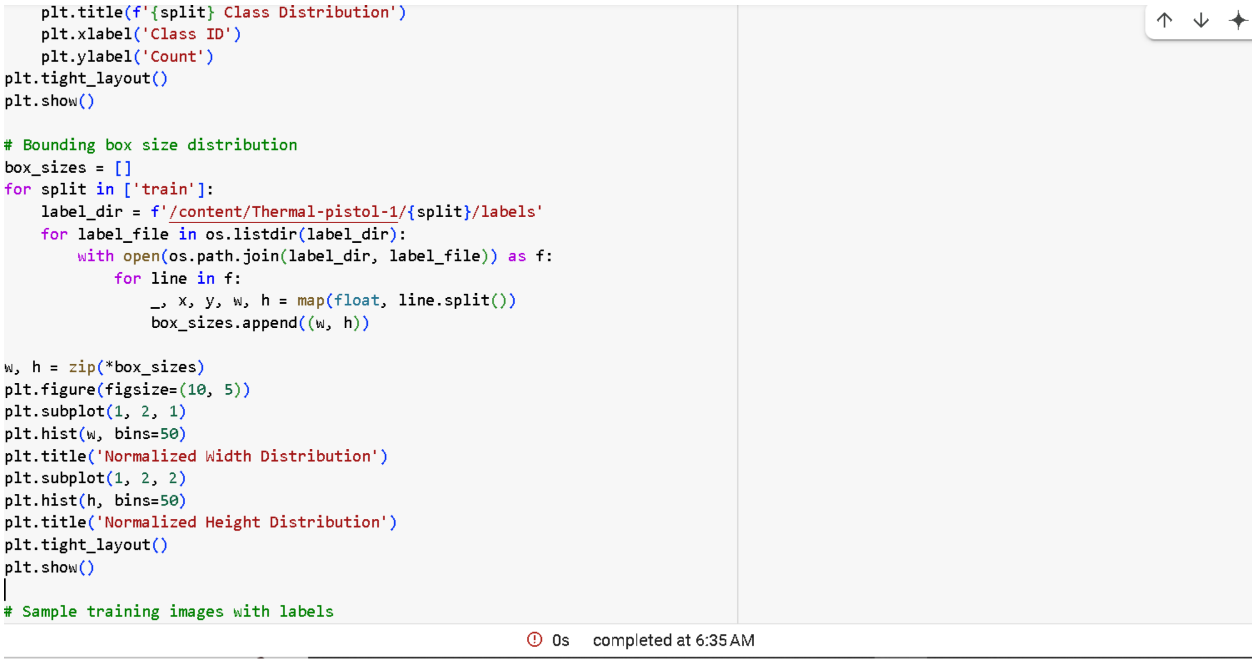
```python
from ultralytics import YOLO
import matplotlib.pyplot as plt

# Load trained model
model = YOLO('/content/runs/detect/yolo11n_finetuned/weights/best.pt')

# Validate model
metrics = model.val(
    data='/content/Thermal-pistol-1/data.yaml',
    split='test',  # Use test split for final evaluation
    plots=True
)

# Print key metrics
print(f"mAP50-95: {metrics.box.map:.4f}")
print(f"Precision: {metrics.box.mp:.4f}")
print(f"Recall: {metrics.box.mr:.4f}")
print(f"F1 Score: {2 * (metrics.box.mp * metrics.box.mr) / (metrics.box.mp + metrics.box.mr):.4f}")

# Plot confusion matrix
confusion_matrix = plt.imread('/content/runs/detect/val/confusion_matrix.png')
plt.figure(figsize=(10, 8))
plt.imshow(confusion_matrix)
plt.axis('off')
plt.title('Confusion Matrix')
plt.show()
```

0s completed at 6:35 AM



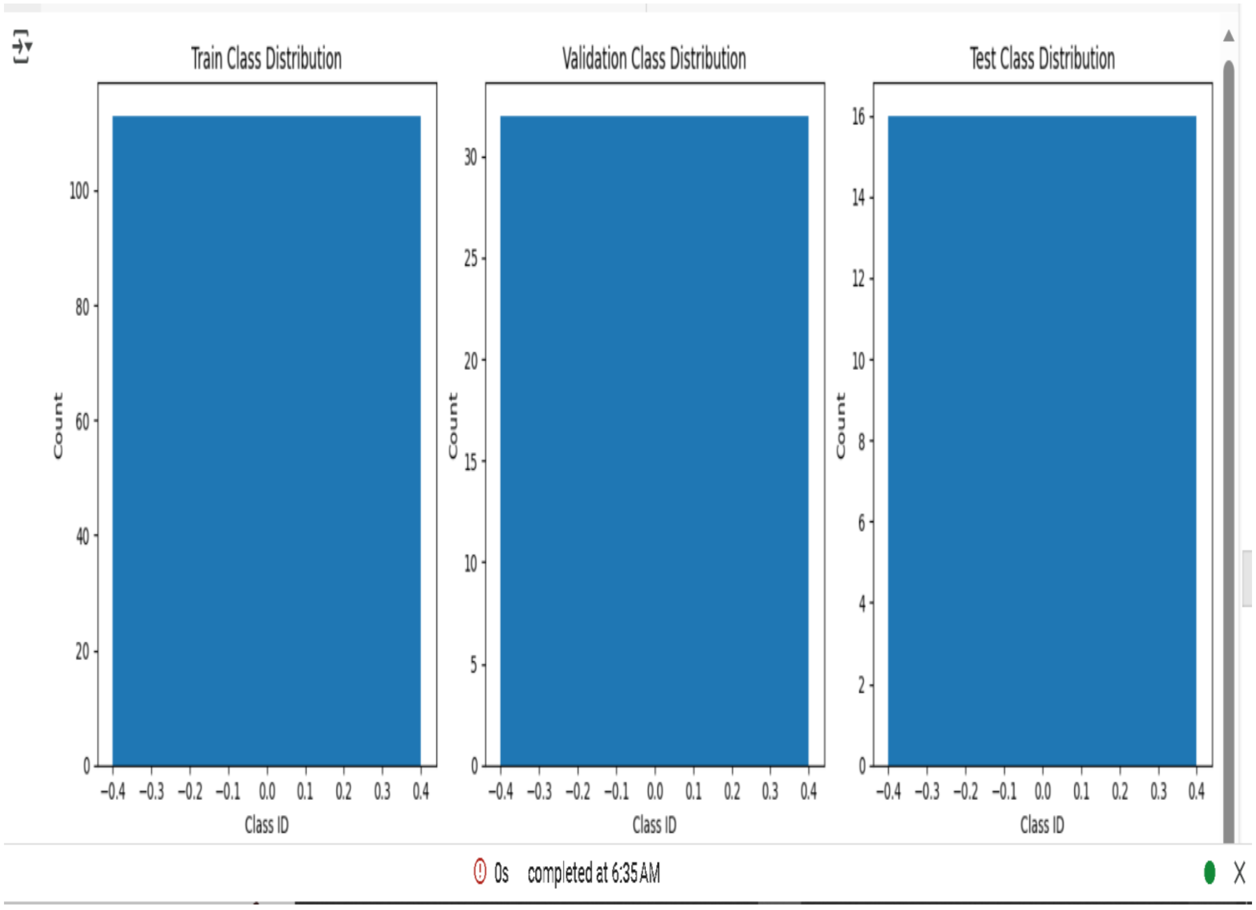
```python
import yaml
import cv2
import numpy as np
from PIL import Image
from ultralytics.utils.plotting import plot_images

# Load dataset config
with open('/content/Thermal-pistol-1/data.yaml') as f:
    data = yaml.safe_load(f)

# Class distribution visualization
class_counts = []
for split in ['train', 'valid', 'test']:
    label_dir = f'/content/Thermal-pistol-1/{split}/labels'
    counts = np.zeros(data['nc'])
    for label_file in os.listdir(label_dir):
        with open(os.path.join(label_dir, label_file)) as f:
            for line in f:
                class_id = int(line.split()[0])
                counts[class_id] += 1
    class_counts.append(counts)

plt.figure(figsize=(15, 5))
for i, split in enumerate(['Train', 'Validation', 'Test']):
    plt.subplot(1, 3, i+1)
    plt.bar(range(data['nc']), class_counts[i])
    plt.title(f'{split} Class Distribution')
    plt.xlabel('Class ID')
```

0s completed at 6:35 AM

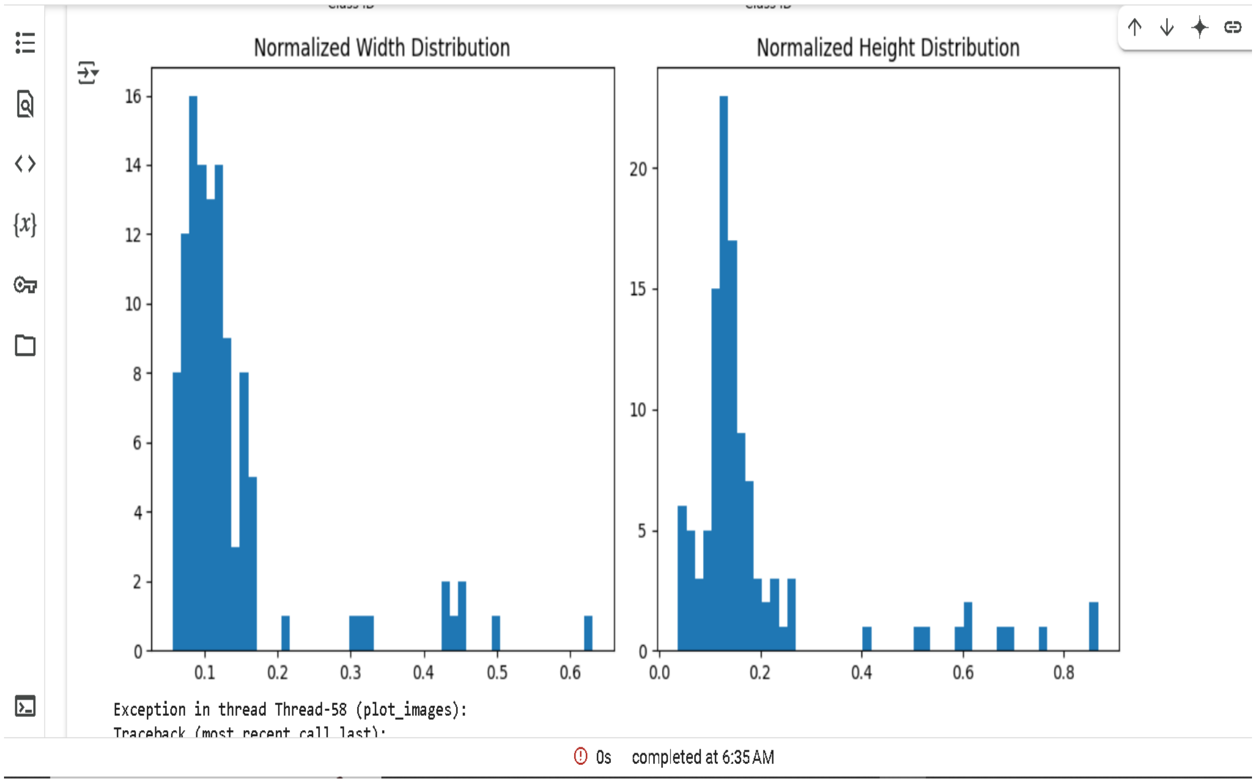


```
    plt.title(f'{split} Class Distribution')
    plt.xlabel('Class ID')
    plt.ylabel('Count')
plt.tight_layout()
plt.show()

# Bounding box size distribution
box_sizes = []
for split in ['train']:
    label_dir = f'/content/Thermal-pistol-1/{split}/labels'
    for label_file in os.listdir(label_dir):
        with open(os.path.join(label_dir, label_file)) as f:
            for line in f:
                _, x, y, w, h = map(float, line.split())
                box_sizes.append((w, h))

w, h = zip(*box_sizes)
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.hist(w, bins=50)
plt.title('Normalized Width Distribution')
plt.subplot(1, 2, 2)
plt.hist(h, bins=50)
plt.title('Normalized Height Distribution')
plt.tight_layout()
plt.show()

# Sample training images with labels
```

0s completed at 6:35 AM

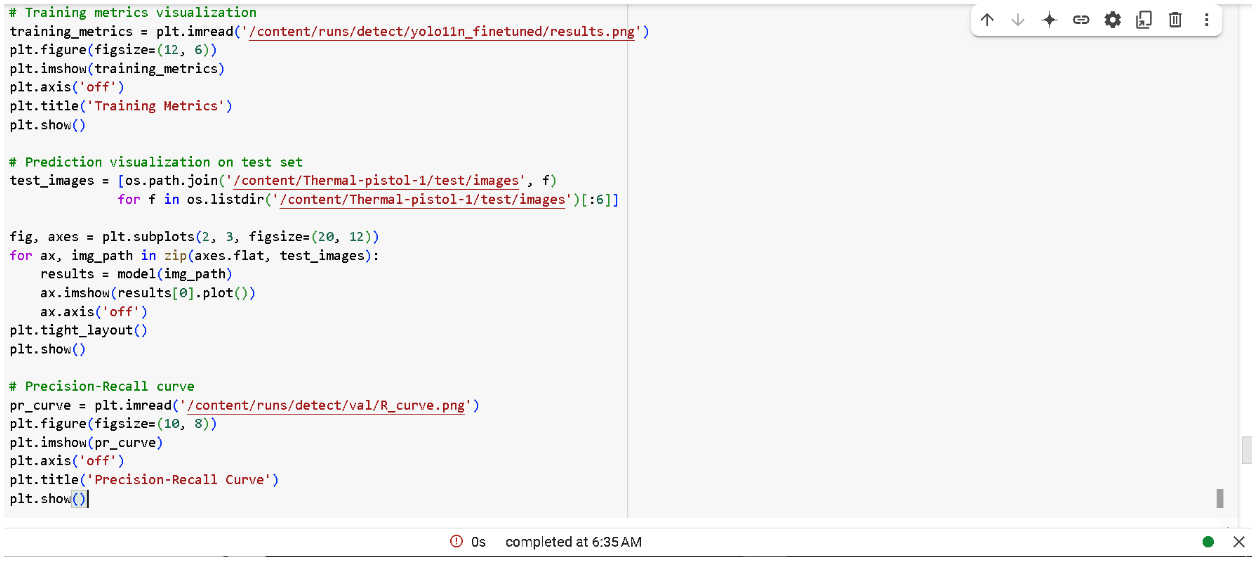


0s completed at 6:35 AM



```
Exception in thread Thread-58 (plot_images):
Traceback (most recent call last):
```

0s completed at 6:35 AM

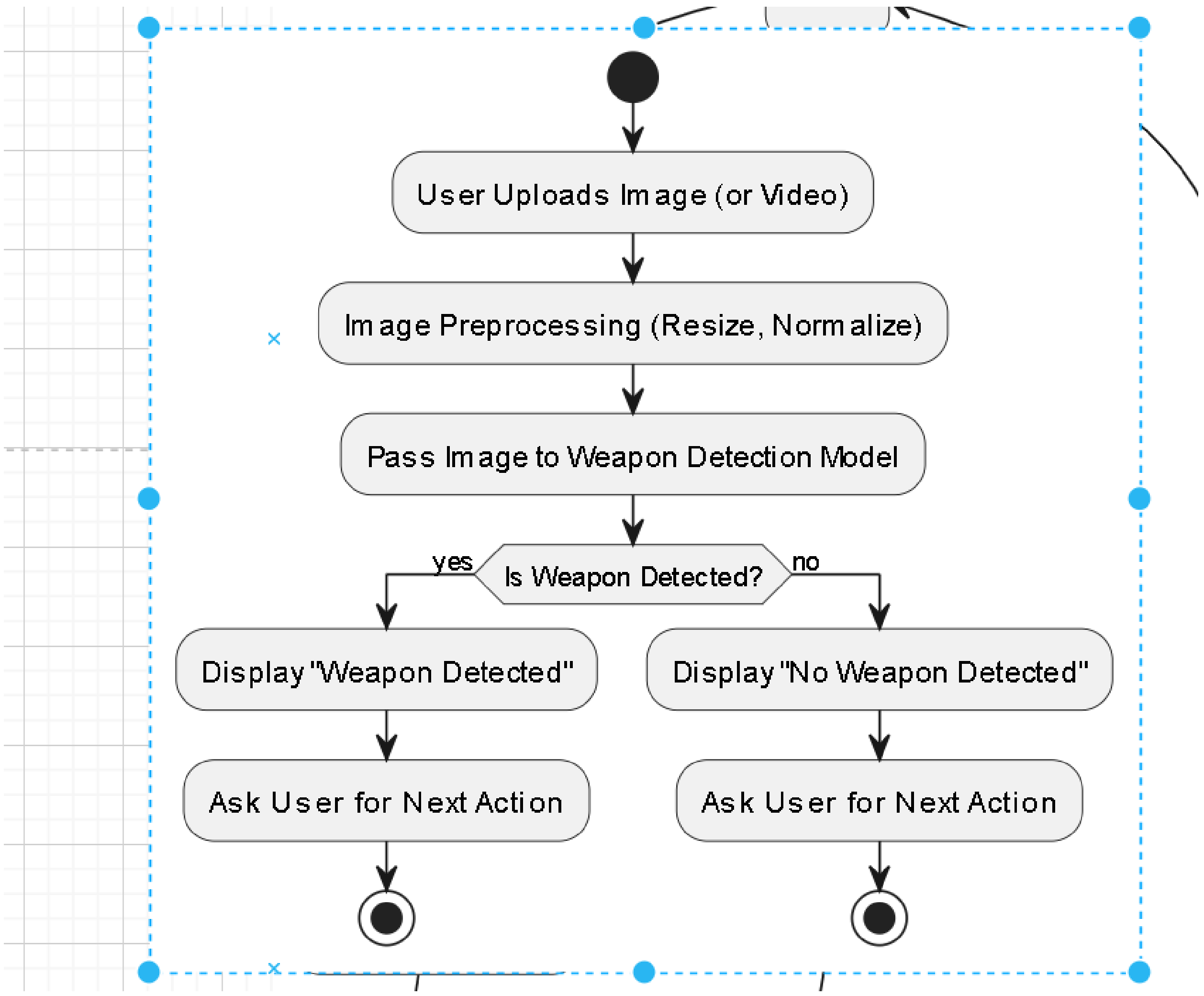


```python
# Training metrics visualization
training_metrics = plt.imread('/content/runs/detect/yolo11n_finetuned/results.png')
plt.figure(figsize=(12, 6))
plt.imshow(training_metrics)
plt.axis('off')
plt.title('Training Metrics')
plt.show()

# Prediction visualization on test set
test_images = [os.path.join('/content/Thermal-pistol-1/test/images', f)
              for f in os.listdir('/content/Thermal-pistol-1/test/images')[:6]]

fig, axes = plt.subplots(2, 3, figsize=(20, 12))
for ax, img_path in zip(axes.flat, test_images):
    results = model(img_path)
    ax.imshow(results[0].plot())
    ax.axis('off')
plt.tight_layout()
plt.show()

# Precision-Recall curve
pr_curve = plt.imread('/content/runs/detect/val/R_curve.png')
plt.figure(figsize=(10, 8))
plt.imshow(pr_curve)
plt.axis('off')
plt.title('Precision-Recall Curve')
plt.show()
```

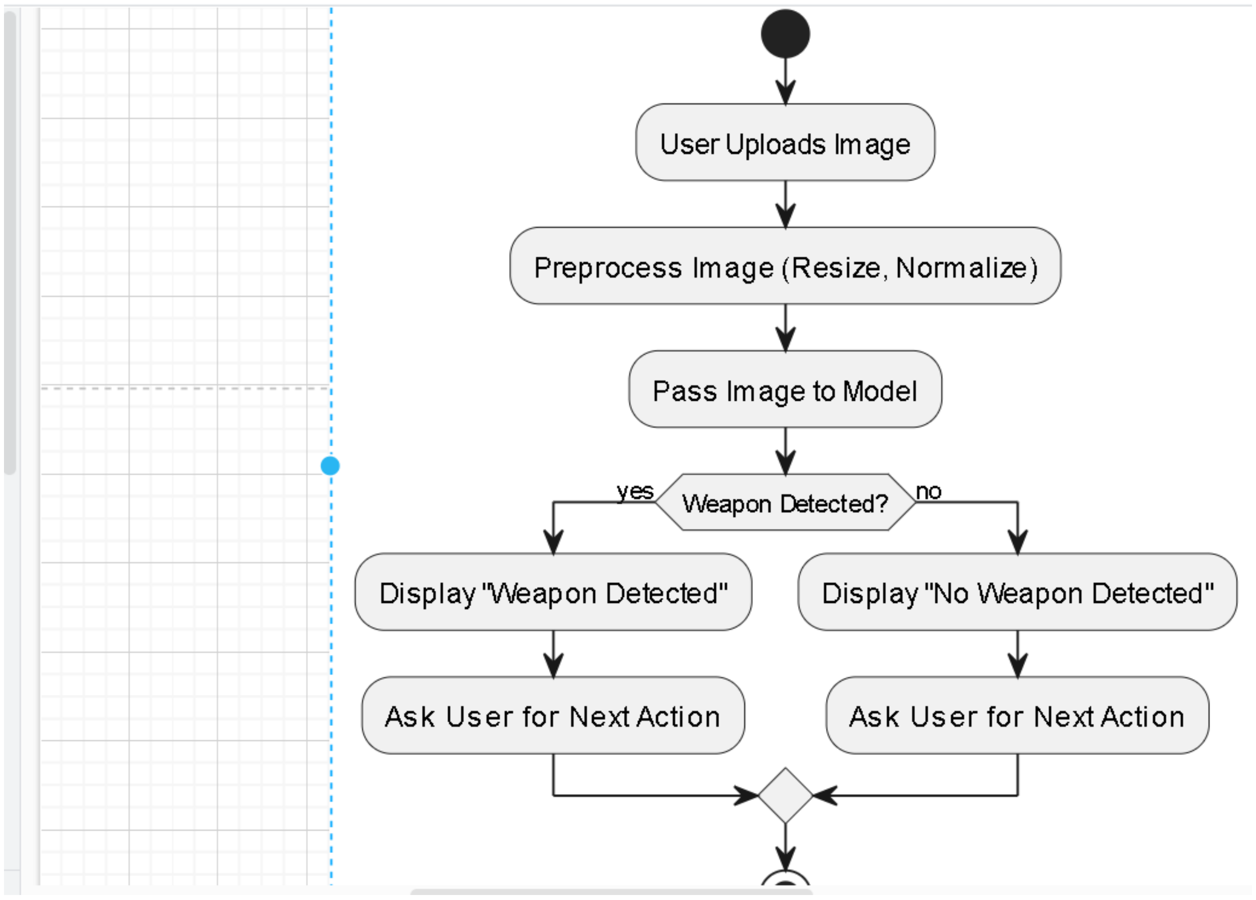
0s completed at 6:35 AM

Flow-Diagrams

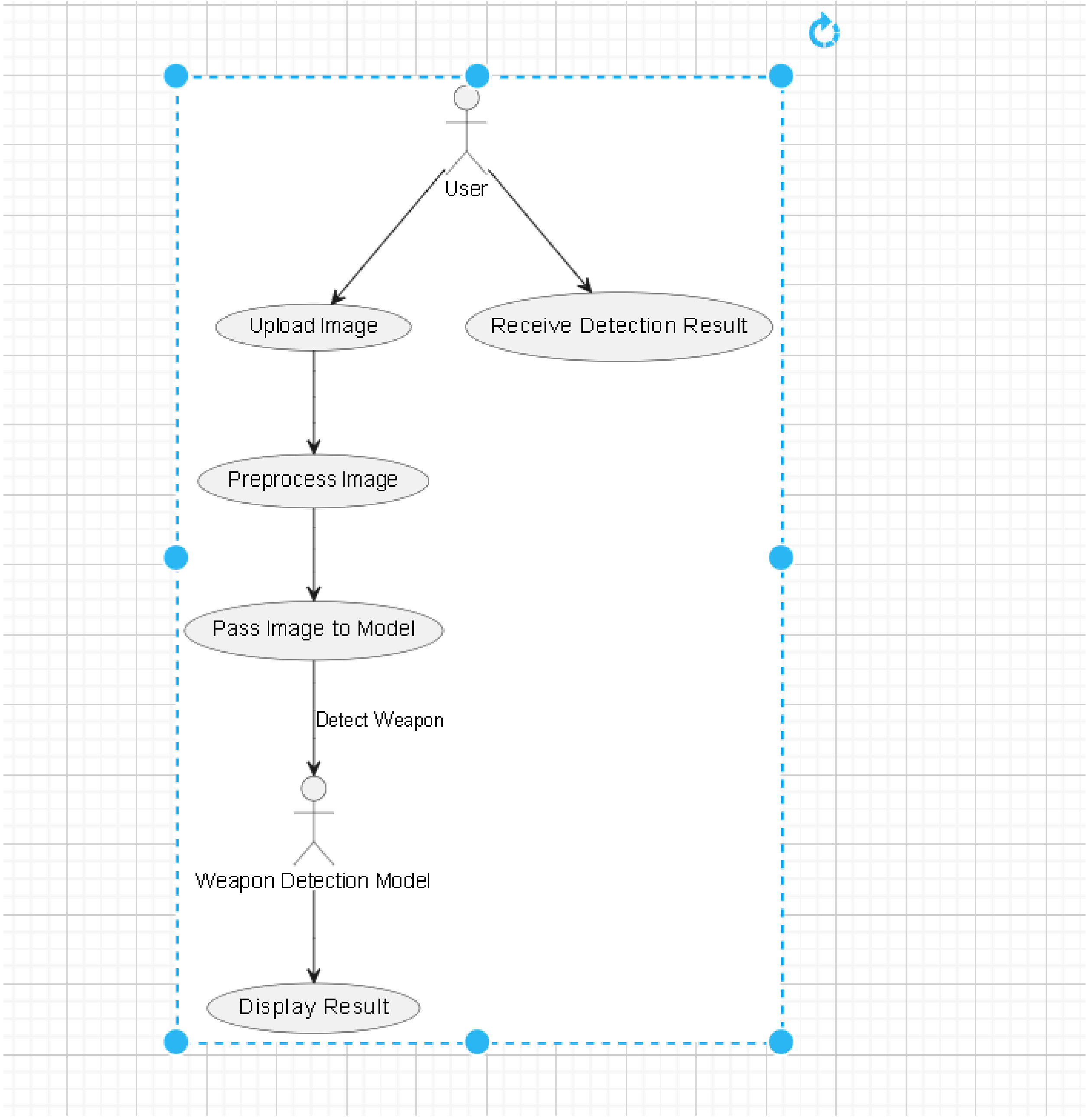
1. Flow-chart


User Uploads Image (or Video)
Image Preprocessing (Resize, Normalize)
Pass Image to Weapon Detection Model
Is Weapon Detected?
yes
no
Display "Weapon Detected"
Display "No Weapon Detected"
Ask User for Next Action
Ask User for Next Action

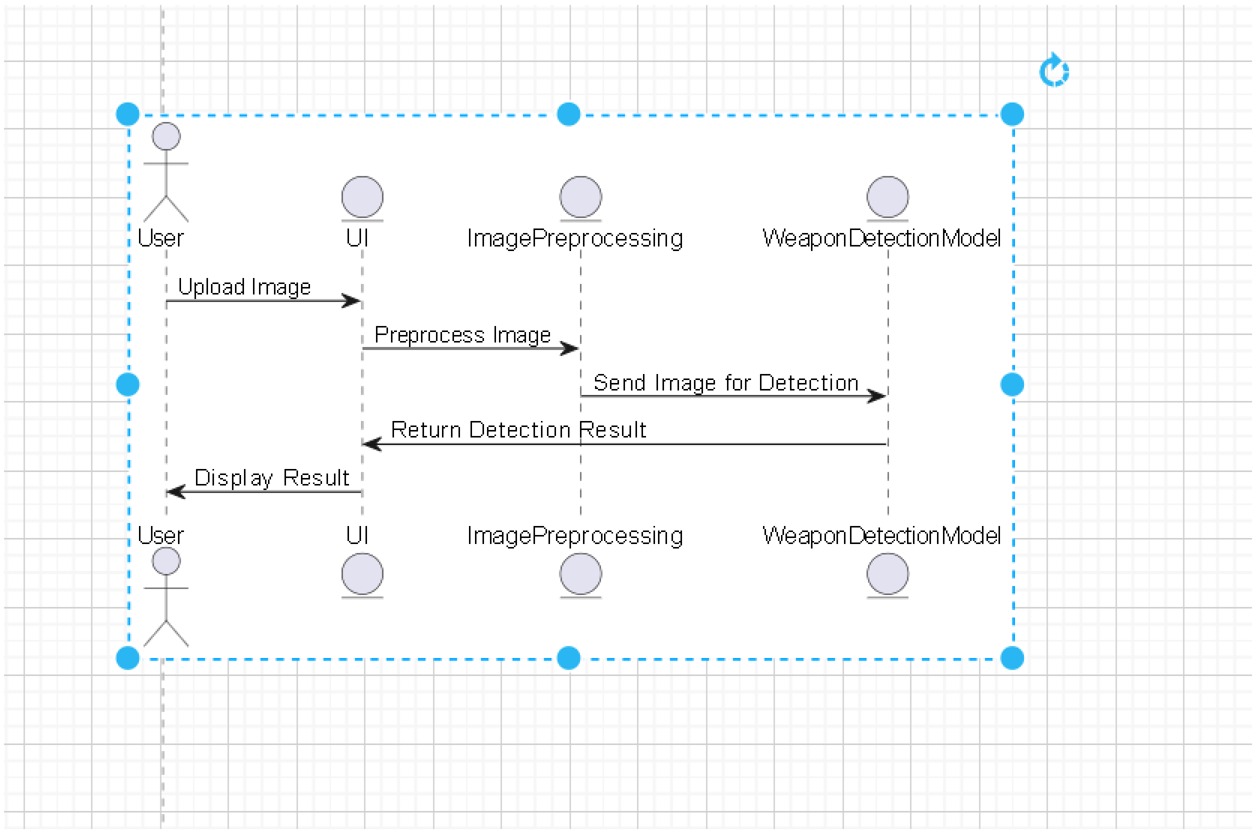
2. Activity-Diagram



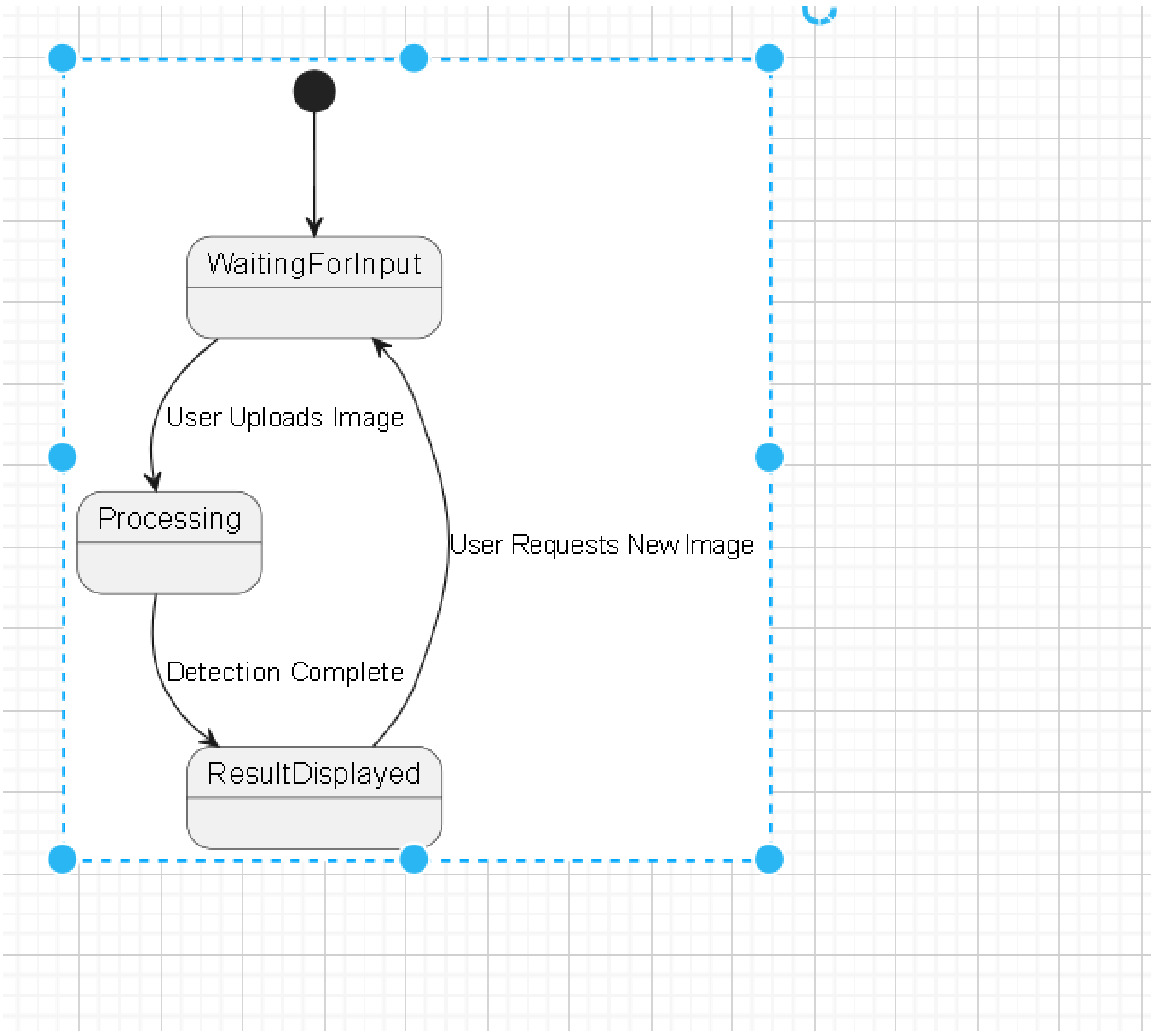
3. Use-Case Diagram



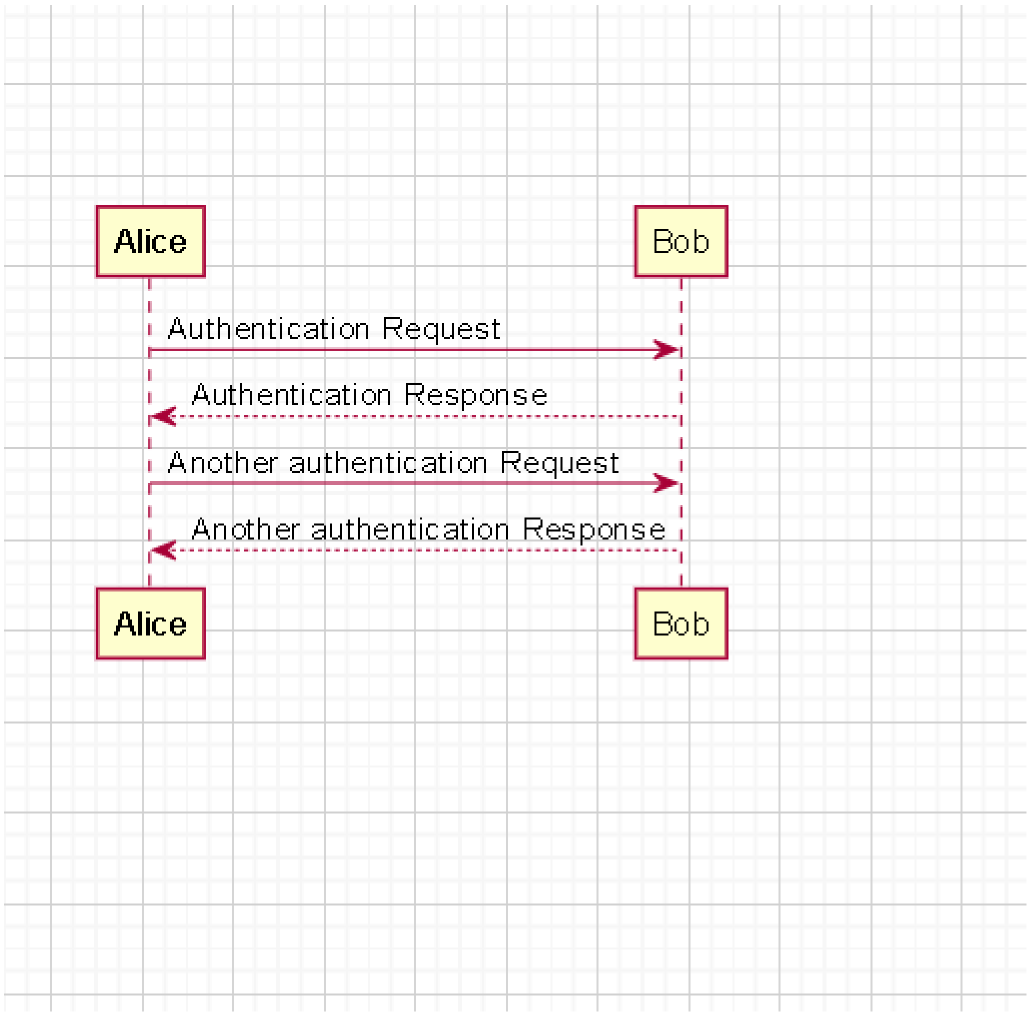
4. Sequence-Diagram



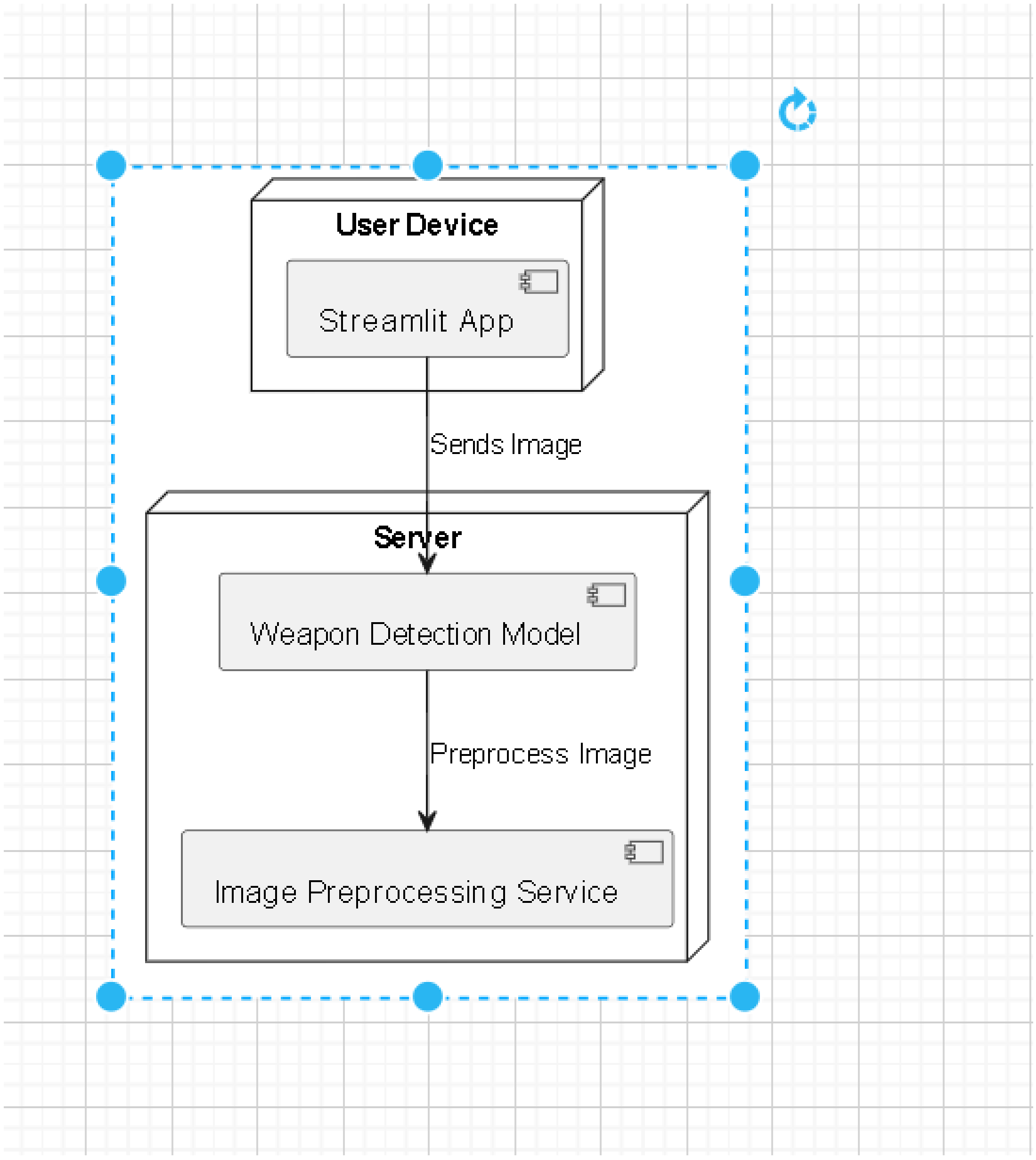
5. State-Diagram



6. Component-Diagram


Alice
Bob
Authentication Request
Authentication Response
Another authentication Request
Another authentication Response
Alice
Bob

7. Deployment-Diagram



*****THE END *****