Comsats Islamabad, Wah Cantt

Reg# & Name:

FA21-BCS-080(Ali Shan)

Course title:

Topics in computer Science

Submitted to:

Dr.Saeed-ur-rehman

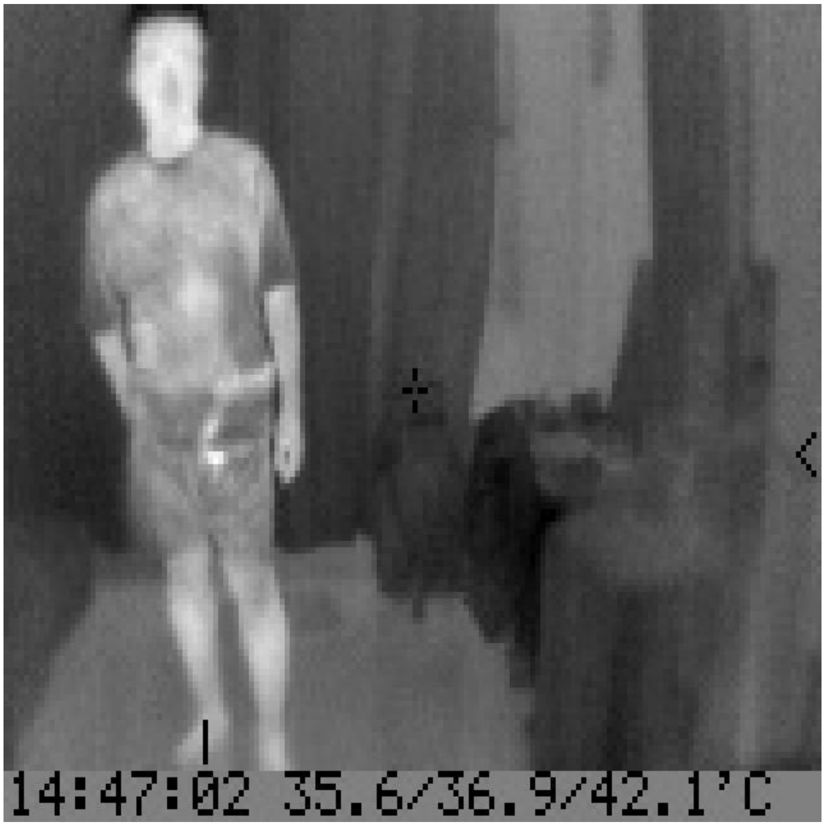
Google Colab Link:

https://colab.research.google.com/drive/1qSBC3GYZnbKQaPrb1RxN8yUUVM05dgbr?usp=sharing

Task1: *Understanding Yolov11*

```
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")
from ultralytics import YOLO


# Load a pretrained model (recommended)
model = YOLO("yolo11n.pt")  # Ensure "yolo11n.pt" exists in your working directory
# Load an image (replace with your image path)
image_path = '/content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg'
```

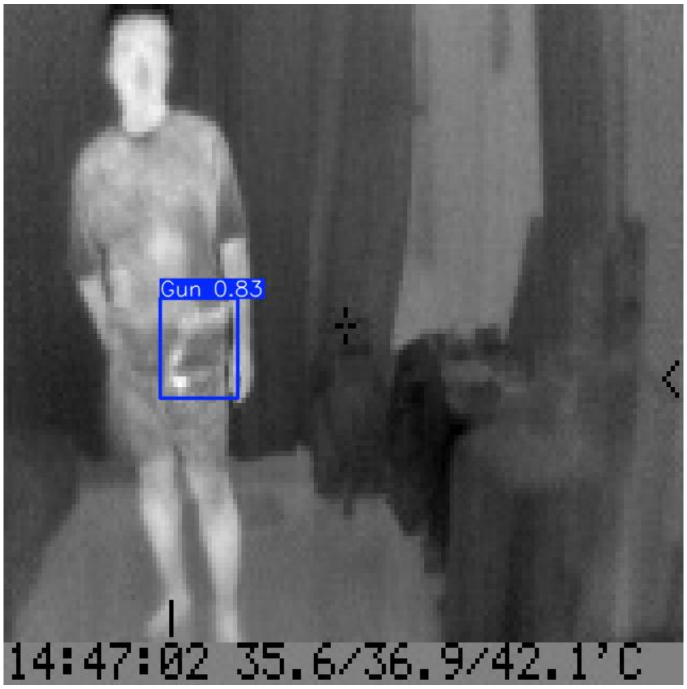
```
# Run inference
results = model(image_path)
# Display results
results[0].show()  # Access the first element in the list and call .show()
```



Task2: *Fine-tuning it on our dataset*

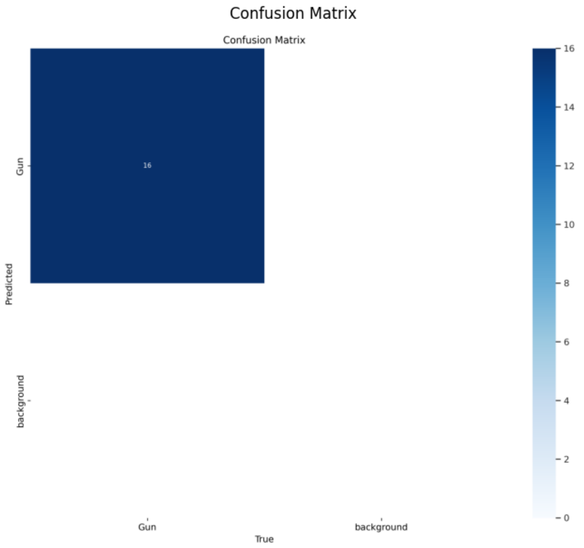
```
from ultralytics import YOLO
# Load the pretrained model
model = YOLO("yolo11n.pt")  # Replace with the path to your pretrained model
# Fine-tune the model on your custom dataset
results = model.train(
   data="/content/Thermal-pistol-1/data.yaml",  # Path to your dataset configuration file
   epochs=100,              # 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)
# Test on an image
results = model("/content/Thermal-pistol-1/test/images/0024_jpg.rf.9f7fac4588e9e28074e4b652448d7ebb.jpg")
results[0].show()  # Display result
```



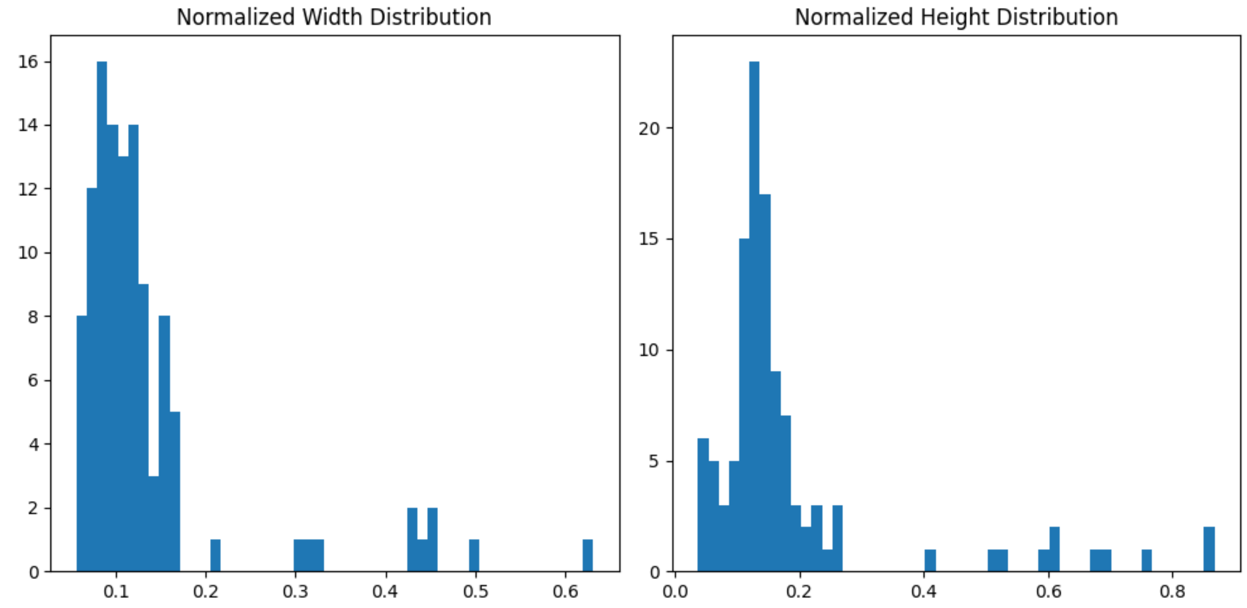
```
from ultralytics import YOLO
```

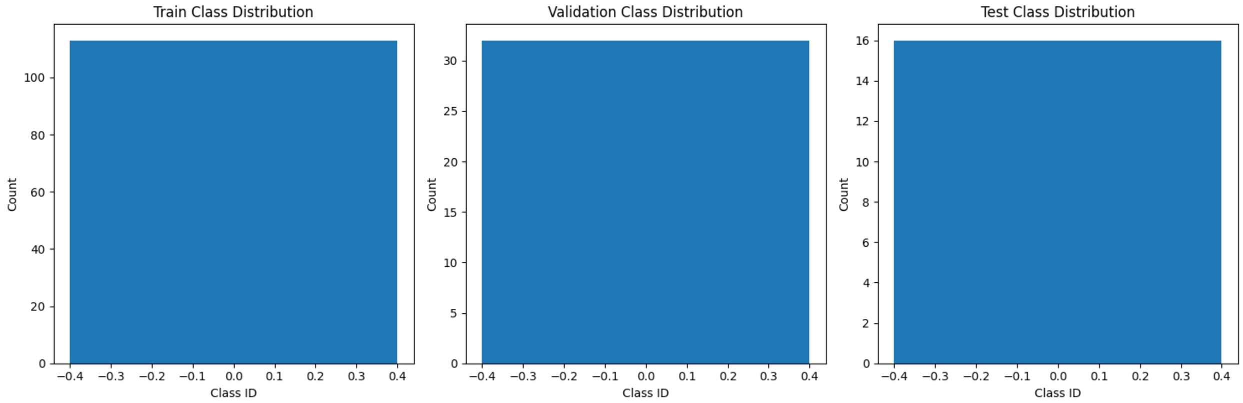
```
import matplotlib.pyplot as plt
# Load trained model
model = YOLO('/content/runs/detect/yolo11n_finetuned/weights/best.pt')

# 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()
```



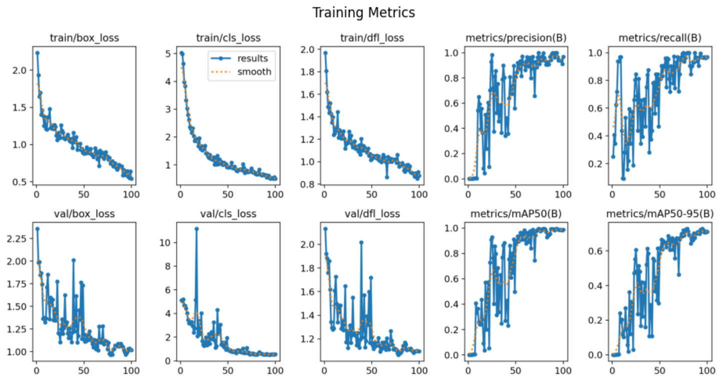
```
import yaml
import cv2
import numpy as np
from PIL import Image
from ultralytics.utils.plotting import plot_images

# 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')
    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
sample_images = [os.path.join('/content/Thermal-pistol-1/train/images', f)
```

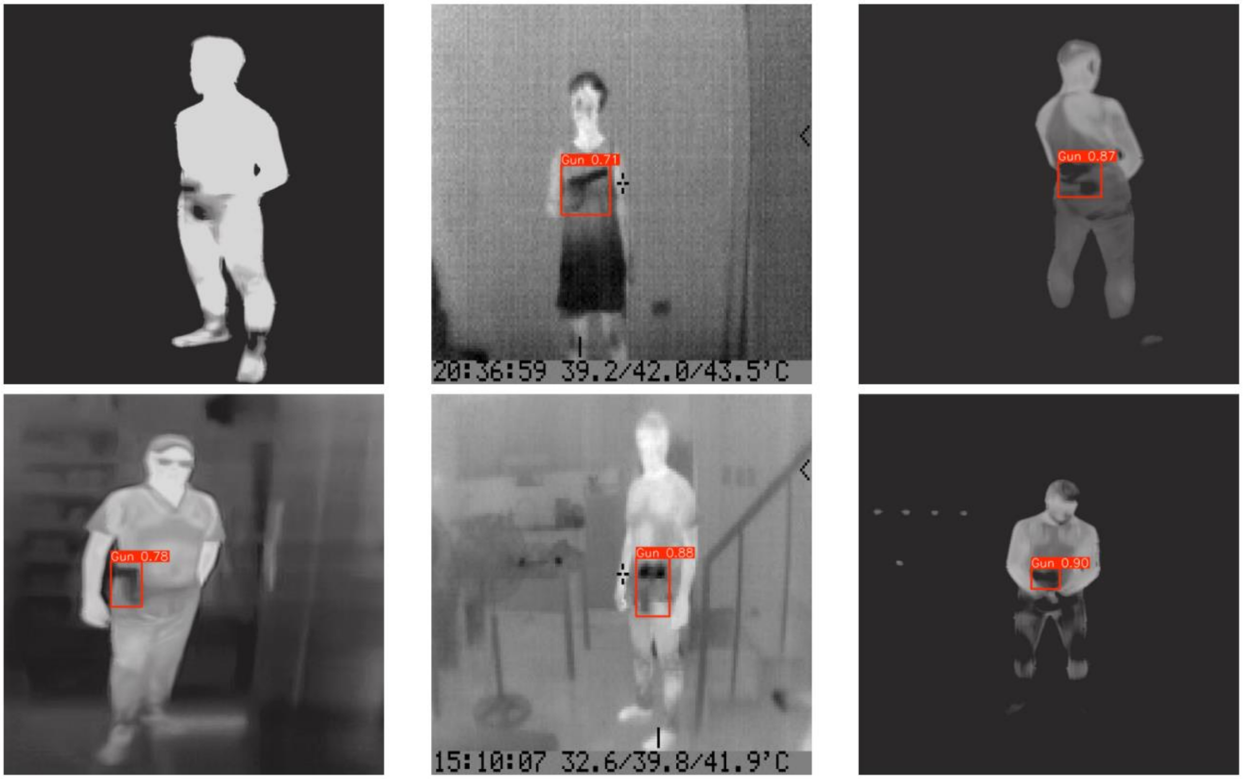
```
              for f in os.listdir('/content/Thermal-pistol-1/train/images')[:4]]
plot_images([cv2.imread(img) for img in sample_images],
        [Image.open(img).size for img in sample_images],
        data['names'])
```



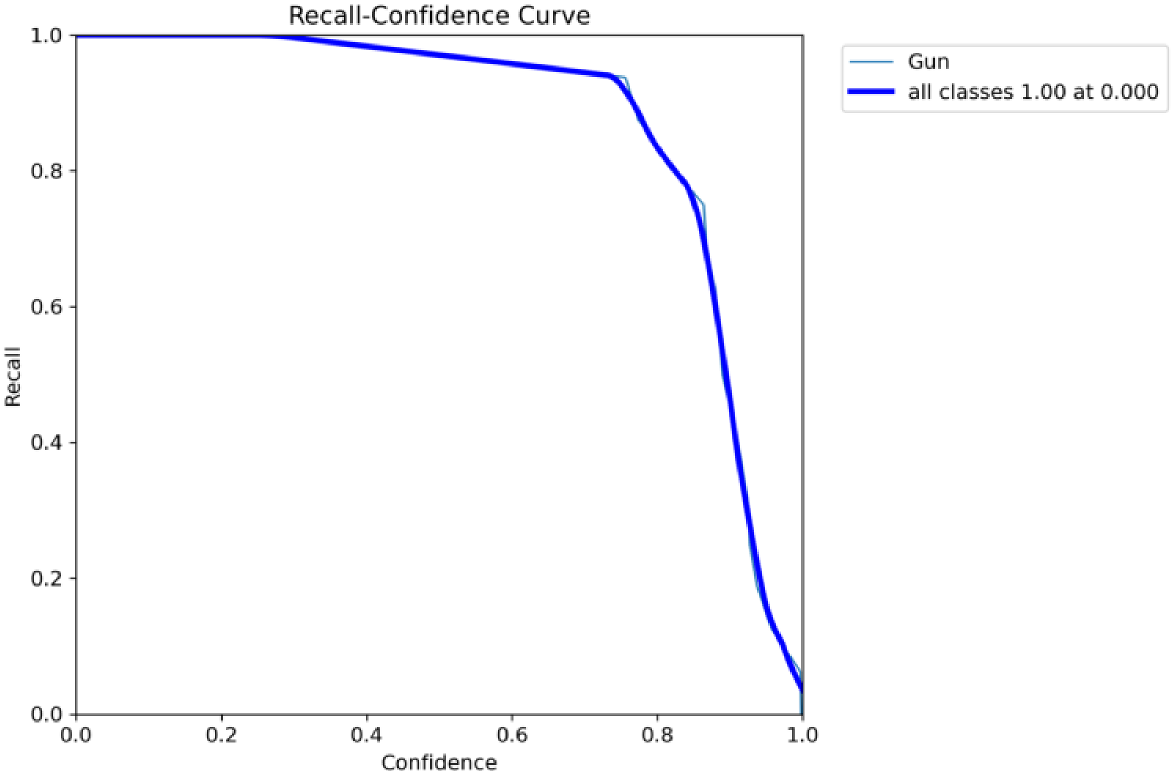


```
# Training metrics visualization
training_metrics = plt.imread('/content/runs/detect/yolo11n_finetuned/results.png')
plt.figure(figsize=(12, 6))
plt.imshow(training_metrics)
plt.axis('off')
plt.title('Training Metrics')

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()
```





Precision-Recall Curve


Recall-Confidence Curve
Recall
Confidence
Gun
all classes 1.00 at 0.000