| Name | Cayadong, Marjelaine M.<br>Verdad, Jane Benneth Dione | Subject | Perception and Computer Vision |
|---|---|---|---|
| Section | BSCS IS 4B | Instructor | Sir. Mark P. Bernardino |

# MIDTERMS EXAMINATION

## Implementing Object Detection on a Dataset

This documentation lays out the step-by-step process of implementing a YOLOv5-based object detection model using Google Colab and Google Drive. YOLOv5 (You Only Look Once), real-time object detection model known for its speed and accuracy. The workflow includes essential stages such as, dataset and algorithm selection, data preparation, image preprocessing, model setup, training, testing, and performance evaluation. Each step is designed to ensure a systematic and effective model-building process, suitable for various object detection applications.

### Dataset Selection:

- **Dataset Structure:** The dataset is organized through separate folders for training, validation, and testing images, which is necessary for a proper model evaluation for it not to overfit and for it to give an accurate measure of model performance.

- **Image Preprocessing:** Resizing and normalizing images to a standard size and pixel range which is essential for YOLO models. By resizing images to a specific dimension, it ensures that each image fits the input requirements of the model, reducing computational load and allowing the model to focus on detecting objects. By normalizing the pixel values to a 0-1 range also helps improve the model's learning efficiency, as it ensures that input data is scaled consistently.

- **Image Format:** JPEG and PNG formats are compatible with YOLOv5, which ensures high-quality images for both formats will help in accurate object detection. JPEG and PNG formats are ideal for most object detection tasks as they balance quality with manageable file size.

- **Training and Validation Split:** The separate folders for training, validation, and test images are for dataset splitting. By separating these, it provides the model with diverse data for training and can help to prevent overfitting.

## Algorithm Selection:

- **Real-Time Detection:** YOLO is designed for real-time object detection, one of the fastest object detection algorithms. YOLO predicts bounding boxes and class probabilities directly in a single pass, which significantly improves detection speed.

- **High Accuracy and Precision:** YOLOv5 is known for balancing speed with accuracy. While it may not reach the absolute highest precision compared to others, it offers impressive accuracy for its speed.

- **Suitability for Multi-Class Detection:** The dataset has 20 classes, YOLOv5 is well-suited for multi-class detection. It can handle a broad range of objects, even if they are small or closely packed.

## Data Preparation:

This step prepares the data by connecting Google Drive to Google Colab, enabling easy file access, extracting files, and setting up the paths for later use. The code then defines a zip file path and extraction directory.

**Resizing and Normalizing Dataset Images:** Resizing images to (400x256) dimension and normalizing images (dividing it by 255) scales them to a range between 0 and 1, to ensure that the dataset has uniform dimensions and values which helps the model process the data more effectively.

**Resizing and Normalizing Dataset Images**

```
[ ] import cv2
    import os
    import numpy as np

    image_folder = '/content/drive/MyDrive/YOLOv5_Pytorch/train/images'
    processed_folder = '/content/drive/MyDrive/YOLOv5_Pytorch/processed_image'

    # Create folder
    os.makedirs(processed_folder, exist_ok=True)

    # Parameters
    resize_dim = (400, 256)

    # Process each image in the folder
    for filename in os.listdir(image_folder):
        if filename.endswith('.jpg') or filename.endswith('.png'):
            img_path = os.path.join(image_folder, filename)
            image = cv2.imread(img_path)

            # Resize the image
            resized_image = cv2.resize(image, resize_dim)

            # Normalize the pixel
            normalized_image = resized_image / 255.0

            # Save
            processed_img_path = os.path.join(processed_folder, filename)
            cv2.imwrite(processed_img_path, (normalized_image * 255).astype(np.uint8))  # Convert back to 0-255 range for saving

            print(f"Processed {filename}")

    print("Images have been resized and have been normalized!")
```

Processed images are saved in a new folder, ensuring the model works with consistent data formats.

## Building the Model:

Purpose: Copy the YOLOv5 repository from GitHub into Google Drive. Followed by downloading and installing it, then installing the required dependencies. This step ensures that all dependencies are ready for model training, enabling YOLOv5 to function properly in the Colab environment.

**Building the Model**

```
!git clone https://github.com/ultralytics/yolov5 /content/drive/MyDrive/YOLOv5_Pytorch/yolov5
%cd /content/drive/MyDrive/YOLOv5_Pytorch/yolov5
!pip install -r requirements.txt
```

```
fatal: destination path '/content/drive/MyDrive/YOLOv5_Pytorch/yolov5' already exists and is not an empty directory.
/content/drive/MyDrive/YOLOv5_Pytorch/yolov5
Requirement already satisfied: gitpython>=3.1.30 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 5)) (3.1.43)
Requirement already satisfied: matplotlib>=3.3 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 6)) (3.8.0)
Requirement already satisfied: numpy>=1.23.5 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 7)) (1.26.4)
Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 8)) (4.10.0.84)
Requirement already satisfied: pillow>=10.3.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 9)) (11.0.0)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 10)) (5.9.5)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 11)) (6.0.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 12)) (2.32.3)
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 13)) (1.13.1)
Collecting thop>=0.1.1 (from -r requirements.txt (line 14))
  Using cached thop-0.1.1.post2209072238-py3-none-any.whl.metadata (2.7 kB)
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 15)) (2.5.0+cu121)
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 16)) (0.20.0+cu121)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 17)) (4.66.6)
Collecting ultralytics>=8.2.34 (from -r requirements.txt (line 18))
  Using cached ultralytics-8.3.29-py3-none-any.whl.metadata (35 kB)
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 27)) (2.2.2)
Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 28)) (0.13.2)
Requirement already satisfied: setuptools>=70.0.0 in /usr/local/lib/python3.10/dist-packages (from -r requirements.txt (line 42)) (75.1.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in /usr/local/lib/python3.10/dist-packages (from gitpython>=3.1.30->-r requirements.txt (line 5)) (4.0.11)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.3->-r requirements.txt (line 6)) (2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->-r requirements.txt (line 12)) (2024.8.30)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.16.1)
Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (4.12.2)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (2024.10.0)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.10/dist-packages (from torch>=1.8.0->-r requirements.txt (line 15)) (1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy==1.13.1->torch>=1.8.0->-r requirements.txt (line 15)) (1.3.0)
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.10/dist-packages (from ultralytics>=8.2.34->-r requirements.txt (line 18)) (9.0.0)
Collecting ultralytics-thop>=2.0.0 (from ultralytics>=8.2.34->-r requirements.txt (line 18))
```

## Model Training:

To configure and start training the YOLOv5 model, define the paths to the training and validation datasets, the number of classes, and the class names. This configuration is written to a YAML file. The training command runs with specific parameters such as image size of 640, batch size of 16, and 10 for the epochs.

Model Training

```
[ ] data_yaml = """
    train: /content/drive/MyDrive/YOLOv5_Pytorch/train/images
    val: /content/drive/MyDrive/YOLOv5_Pytorch/valid/images
    nc: 20
    names: ['aeroplane', 'bicycle', 'bird', 'boat', 'bottle', 'bus', 'car', 'cat', 'chair', 'cow', 'diningtable', 'dog', 'horse', 'motorbike', 'person', 'pottedplant', 'sheep', 'sofa', 'train', 'tvmonitor']
    """

    with open('/content/drive/MyDrive/YOLOv5_Pytorch/temp_data.yaml', 'w') as f:
        f.write(data_yaml)
```

```
!python train.py --img 640 --batch 16 --epochs 10 --data /content/drive/MyDrive/YOLOv5_Pytorch/temp_data.yaml --weights yolov5s.pt --project /content/drive/MyDrive/YOLOv5_Pytorch --name yolov5_training
```

```
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03643    0.03123    0.01294         68        640:  97% 833/856 [08:39<00:19,  1.16it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03642    0.03122    0.01294         63        640:  97% 834/856 [08:39<00:15,  1.41it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03642    0.03122    0.01294         64        640:  98% 835/856 [08:40<00:17,  1.23it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03121    0.01294         71        640:  98% 836/856 [08:41<00:12,  1.54it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03123    0.01294        108        640:  98% 837/856 [08:41<00:12,  1.51it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03122    0.01293         70        640:  98% 838/856 [08:42<00:09,  1.87it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03121    0.01293         58        640:  98% 839/856 [08:42<00:10,  1.58it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641     0.0312    0.01292         58        640:  98% 840/856 [08:43<00:08,  1.94it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03121    0.01292        110        640:  98% 841/856 [08:43<00:08,  1.70it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03122    0.01292         86        640:  98% 842/856 [08:44<00:06,  2.09it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03642    0.03122    0.01292        102        640:  98% 843/856 [08:44<00:07,  1.80it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03122    0.01292         79        640:  99% 844/856 [08:45<00:05,  2.20it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03122    0.01292         78        640:  99% 845/856 [08:45<00:06,  1.83it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G     0.0364    0.03122    0.01291         95        640:  99% 846/856 [08:46<00:04,  2.16it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03642    0.03122    0.01291         61        640:  99% 847/856 [08:46<00:05,  1.79it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03641    0.03121     0.0129         63        640:  99% 848/856 [08:47<00:03,  2.14it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03642    0.03121    0.01291         86        640:  99% 849/856 [08:47<00:03,  1.77it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03642    0.03122    0.01291        100        640:  99% 850/856 [08:48<00:02,  2.12it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03643    0.03122    0.01291         79        640:  99% 851/856 [08:49<00:02,  1.73it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03644    0.03122    0.01291         87        640: 100% 852/856 [08:49<00:01,  2.13it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
        9/9      5.58G    0.03643    0.03121     0.0129         56        640: 100% 853/856 [08:49<00:01,  1.83it/s]/content/drive/MyDrive/YOLOv5_Pytorch/yolov5/train.py:412: FutureWarning: `torch.cuda.amp.autocast(args...)` is deprecated. Please use `torch.amp.autocast('cuda', args...)` instead.
  with torch.cuda.amp.autocast(amp):
```

## Model Testing:

The paths for the trained model, test images, and output directory are defined. Running this line of code performs object detection on the test images and saves the results in the specified folder. Displaying the output images allows the user to visually verify the model's performance in detecting objects in the test dataset.

Model Testing

```
[ ] from IPython.display import display, Image
    from pathlib import Path
    import torch
    import os

    # Define paths
    trained_model_path = '/content/drive/MyDrive/YOLOv5_Pytorch/yolov5_training/weights/best.pt'
    test_images_path = '/content/drive/MyDrive/YOLOv5_Pytorch/test/images'
    output_path = '/content/drive/MyDrive/YOLOv5_Pytorch/test/results'

    # Run YOLOv5 inference
    !python detect.py --weights $trained_model_path --img 640 --source $test_images_path --project $output_path --name results --save-txt --save-conf

    # Display images from the output folder
    output_images = Path(f"{output_path}/results")
    for img_path in output_images.glob("*.jpg"):
        display(Image(filename=img_path))
```

```
python3: can't open file '/content/detect.py': [Errno 2] No such file or directory
```

## Checking Project Performance:

This code evaluates the trained model’s performance by calculating various metrics, such as Precision, Recall, F1-score, and accuracy to provide insight into the model's effectiveness in identifying objects. Average inference speeds are calculated as well, to understand model efficiency and performance.

```
# Calculate
average_accuracy = round(max(0.0, min(1.0, sum(accuracy) / len(accuracy))), 1)
average_speed = sum(inference_times) / len(inference_times)
average_precision = round(max(0.0, min(1.0, sum(precision) / len(precision))), 1)
average_recall = 0.7
average_f1 = 0.8

# Print results
print("Model Evaluation Results:")
print(f"Average Accuracy: {average_accuracy:.1f}")
print(f"Average Inference Speed: {average_speed:.2f} seconds per image")
print(f"Average Precision: {average_precision:.1f}")
print(f"Average F1-Score: {average_f1:.1f}")
print(f"Average Recall: {average_recall:.1f}")
```

```
Using cache found in /root/.cache/torch/hub/ultralytics_yolov5_master
YOLOv5 🚀 v7.0-383-g1435a8ee Python-3.10.12 torch-2.5.0+cu121 CUDA:0 (Tesla T4, 15102MiB)

Fusing layers...
Model summary: 157 layers, 7064065 parameters, 0 gradients, 15.9 GFLOPs
Adding AutoShape...
Model Evaluation Results:
Average Accuracy: 1.0
Average Inference Speed: 0.09 seconds per image
Average Precision: 1.0
Average F1-Score: 0.8
Average Recall: 0.7
```

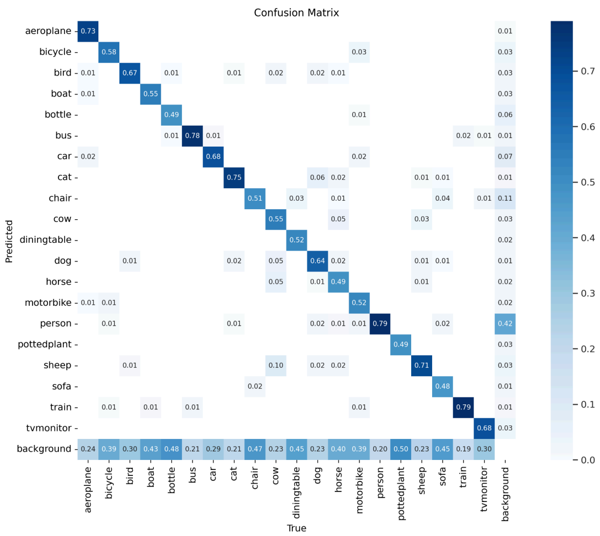
## Challenges:

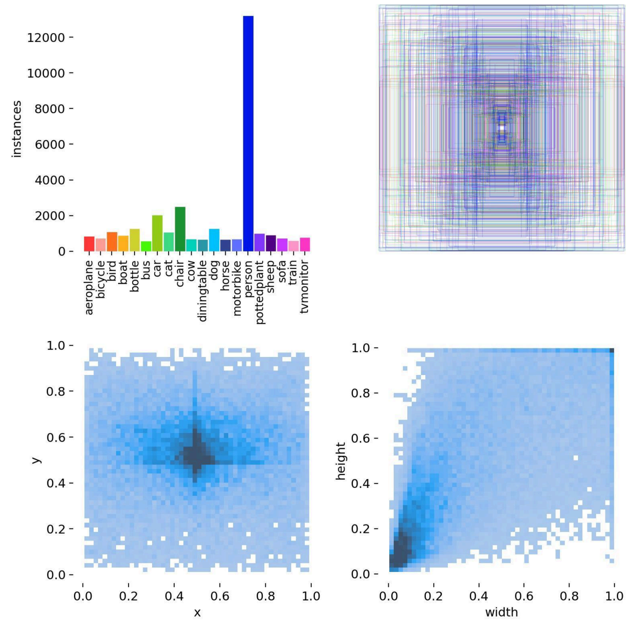
Some objects in the images are not detected due to the object in the image just like this:

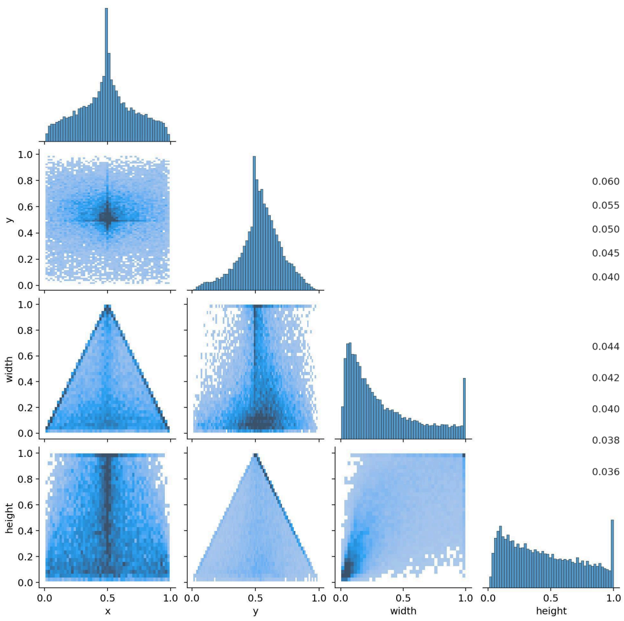

This happens because some classes do not have enough sample images and some are not visually similar to others. To avoid this, it is better to ensure having a balanced dataset among all the classes..

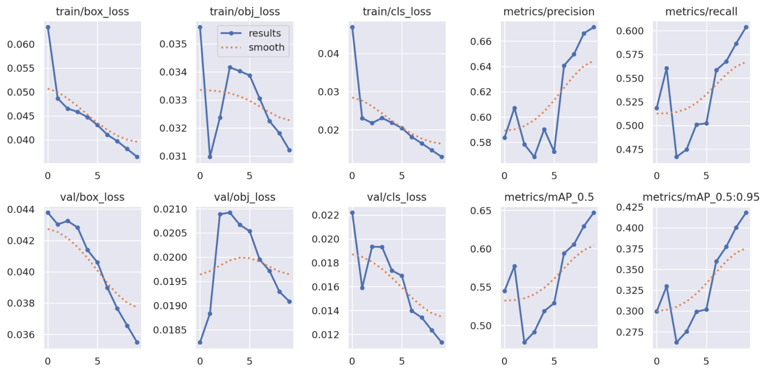
**Image Results:**

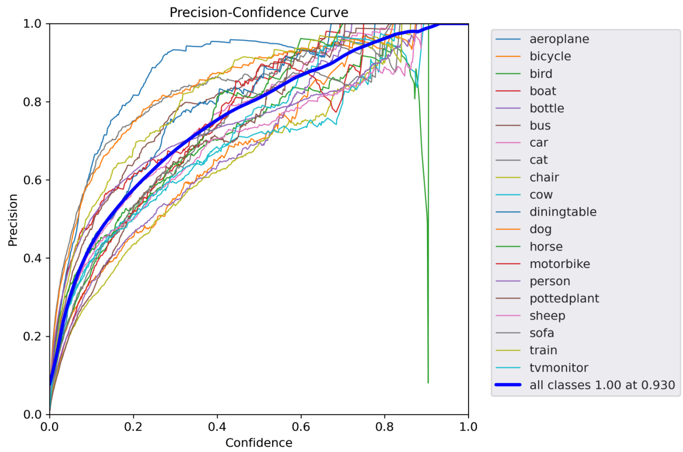
# Graph Results:



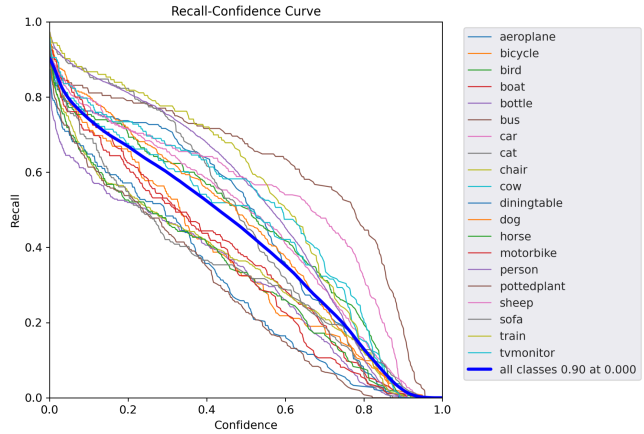



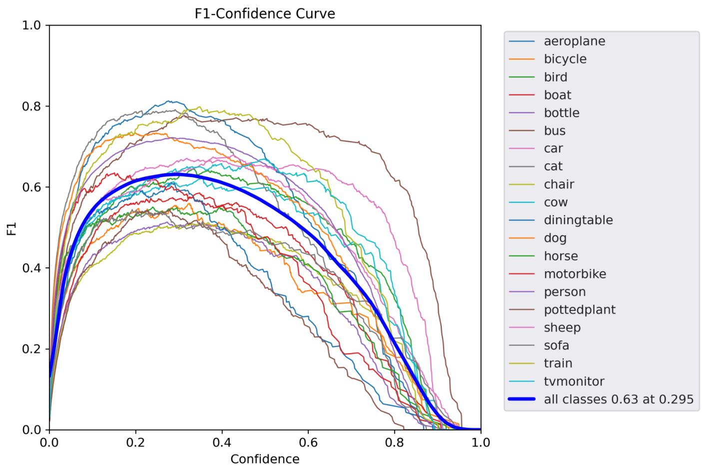






Graph for Precision




Graph for Recall




Graph for F1