DSDE-Homework_3_3_Object_detection_VOCDetection_yolov8_advanced

Input data เป็น Pascal VOC 2007
เป็น custom dataset ที่นำมา train YOLOv8 model

Setup Ultralytics

```
# Check for installed dependencies and available hardware
import ultralytics
ultralytics.checks()
```

function download และ extract

```
def download_file(url, dest_folder):
        if not os.path.exists(dest_folder):
                os.makedirs(dest_folder)
        filename = os.path.join(dest_folder, url.split('/')[-1])
                if os.path.isfile(filename):
                print(f"{filename} already exists. Skipping download.")
                return filename
        response = requests.get(url, stream=True)
        total_size = int(response.headers.get('content-length', 0))
        with open(filename, 'wb') as f, tqdm(
                desc=filename,
                total=total_size,
                unit='iB',
                unit_scale=True,
                unit_divisor=1024,
        )as bar:
                for chunk in response.iter_content(chunk_size=1024):
                        if chunk:
                                f.write(chunk)
                                bar.update(len(chunk))
        return filename
def extract_tar(tar_file, extract_to_folder):
        with tarfile.open(tar_file, 'r') as tar_ref:
                tar_ref.extractall(extract_to_folder)
import locale

locale.getpreferredencoding = lambda: "UTF-8"
```

Download ไฟล์

```
# Download and extract VOC 2007
base_url = "http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2007/"
files = {
'VOCtrainval_06-Nov-2007.tar': 'VOCtrainval_06-Nov-2007.tar',
'VOCtest_06-Nov-2007.tar': 'VOCtest_06-Nov-2007.tar'
}download_folder = '/content/voc2007'
for url, filename in files.items():
print(f"Downloading {filename}...")
tar_path = download_file(base_url + filename, download_folder)
print(f"Extracting {filename}...")
extract_tar(tar_path, download_folder)
print(f"{filename} extracted.")
```

สร้าง directory สำหรับเก็บ data YOLO format

```
# Define YOLO-compatible structure and directories
class_names = ["person", "car", "bus", "train", ... ]
voc_annot_folder = '/content/voc2007/VOCdevkit/VOC2007/Annotations'
image_folder = '/content/voc2007/VOCdevkit/VOC2007/JPEGImages'
yolo_annot_folder = '/content/yolo_annotations'
os.makedirs(yolo_annot_folder,exist_ok=True)
os.makedirs(os.path.join(yolo_annot_folder,"images"),exist_ok=True)
os.makedirs(os.path.join(yolo_annot_folder,"labels"),exist_ok=True)
```

function convert ให้เป็น yolo-text format

```
def process_annotations(file_list, annot_folder):
    for image_filename in file_list:
        xml_file = os.path.join(voc_annot_folder,
os.path.splitext(image_filename)[0] + '.xml')
        if not os.path.isfile(xml_file):
            continue
        # Parse XML, extract bounding box, convert to YOLO format
        tree = ET.parse(xml_file)
        root = tree.getroot()
        image = cv2.imread(os.path.join(image_folder, image_filename))
        h, w, _ = image.shape
        yolo_annot_file = os.path.join(annot_folder,
os.path.splitext(image_filename)[0] + '.txt')
        with open(yolo_annot_file, 'w') as f:
            for obj in root.findall('object'):
                cls = obj.find('name').text
                if cls not in class_names:
                    continue
                cls_id = class_names.index(cls)
                xml_box = obj.find('bndbox')
                b = [int(xml_box.find('xmin').text),
```

```
int(xml_box.find('ymin').text), int(xml_box.find('xmax').text),
int(xml_box.find('ymax').text)]
                # YOLO format conversion
                x_center = (b[0] + b[2]) / 2.0 / w
                y_center = (b[1] + b[3]) / 2.0 / h
                width = (b[2] - b[0]) / w
                height = (b[3] - b[1]) / h
                f.write(f"{cls_id} {x_center} {y_center} {width}
{height}\n")
```

train test split

```
from sklearn.model_selection import train_test_split
train_files, val_files = train_test_split(image_filenames, test_size=0.2,
random_state=42)
process_annotations(train_files, train_annot_folder)
process_annotations(val_files, val_annot_folder)
```

สร้าง data.yml

```
yaml_content = """
train: /content/yolo_annotations/images/train
val: /content/yolo_annotations/images/val
nc: 17  # Number of classes
names: ['person', 'car', 'bus', ... ]
"""
with open('/content/data.yaml', 'w') as f:
    f.write(yaml_content)
```

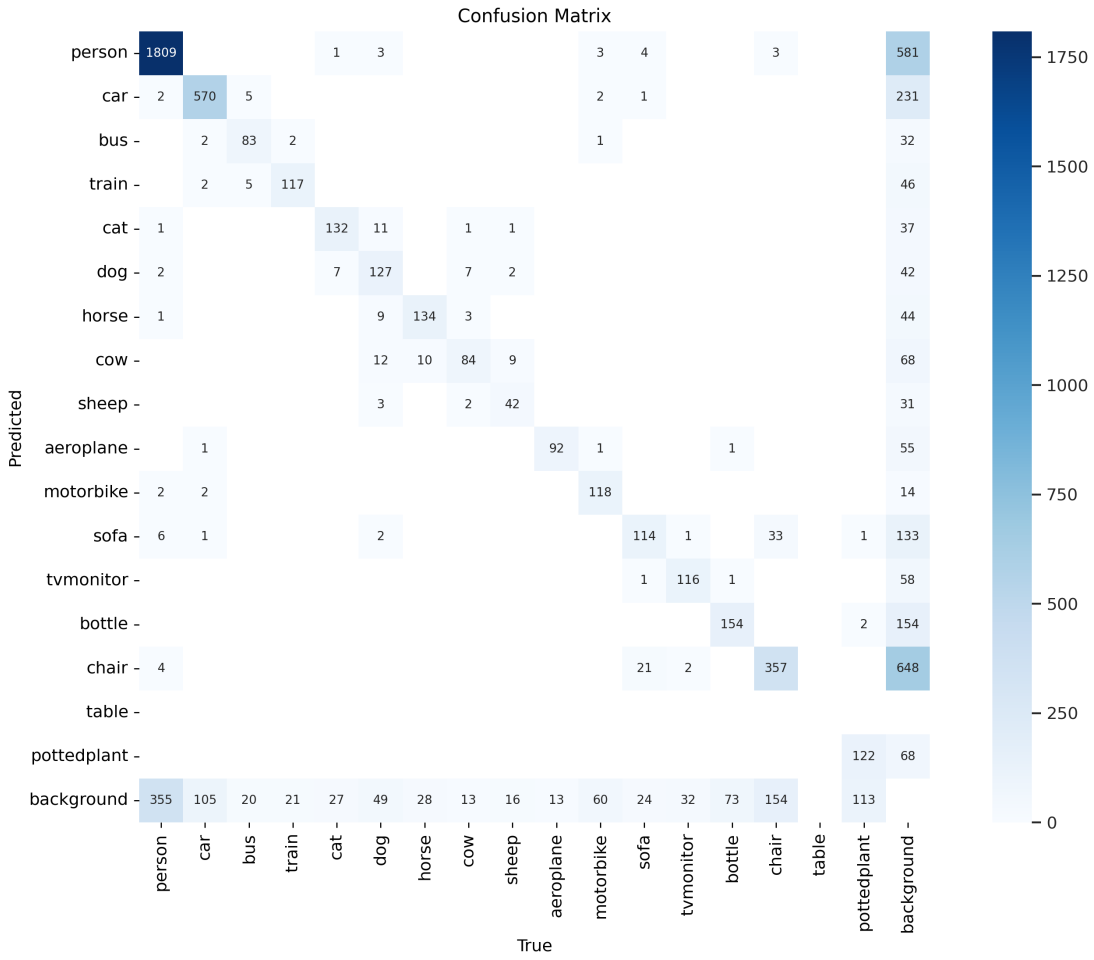
เริ่ม train model

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')  # Load pretrained model
model.to('cuda:0')  # Use GPU
results = model.train(data='data.yaml', epochs=3)  # Train for 3 epochs
```

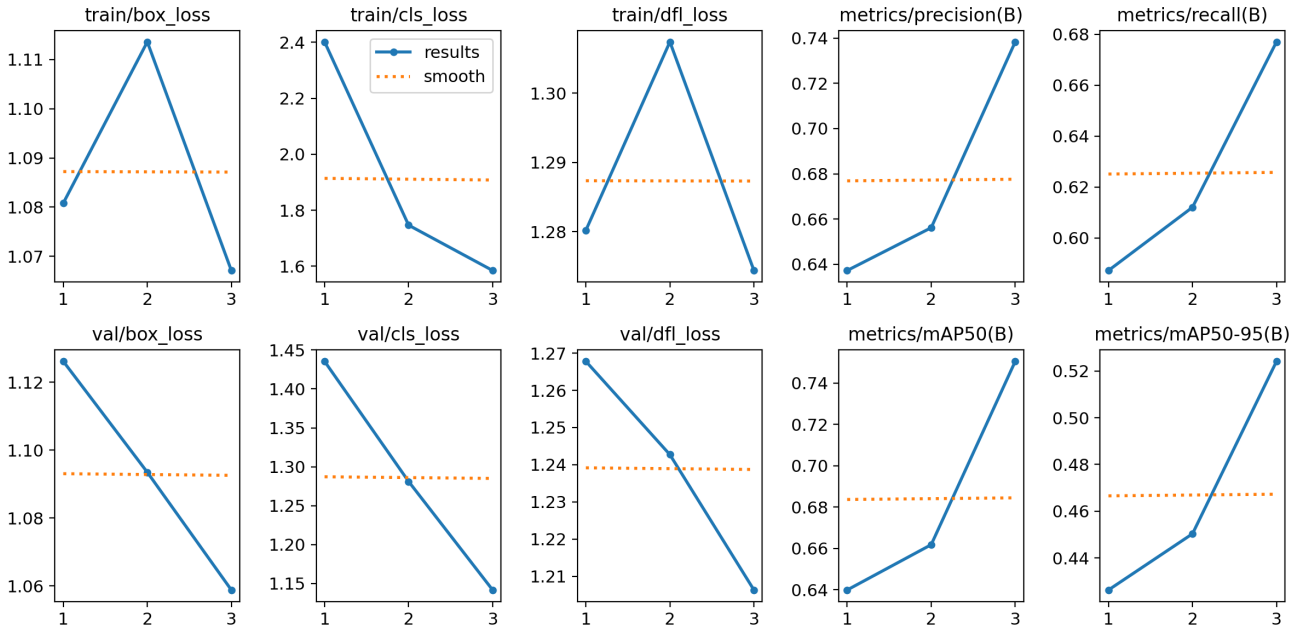
Evaluation

```
from PIL import Image
Image.open("/content/runs/detect/train/confusion_matrix.png")
Image.open("/content/runs/detect/train/results.png")
Image.open("/content/runs/detect/train/val_batch1_pred.jpg")
```

```
Image.open("/content/runs/detect/train/confusion_matrix.png")
```




```
Image.open("/content/runs/detect/train/results.png")
```




```
Image.open("/content/runs/detect/train/val_batch1_pred.jpg")
```


003321.jpg
car 0.
car 0.3
car 0.3 6
car 0.9
003786.jpg
bus 0.9
bus 0.3
person 0.3
009166.jpg
007297.jpg
aeroplane 0.9
006387.jpg
person 0.6
003806.jpg
car 1.0
car 1.0
sofa 1.0
003537.jpg
train 0.8
train 0.8
002920.jpg
train 0.7
person
car 0.6
001074.jpg
007097.jpg
000100.jpg
train 0.4 0.9
person 0.4
003198.jpg
person 0.7
person 0.3
person 0.3
person 0.9
person 0.8
chair 0.7
chair 0.6