📄 YOLOv8 Plastic Bottle Detection — Report

Video Link:

🎬 mainvideo.mp4

🧾1️⃣Overall Process

This project focuses on training a YOLOv8-based object detection model to detect plastic bottles.
The key stages of the project are:

🔷 Dataset Preparation

- Two datasets were downloaded from Kaggle:
    - **Dataset 1:** “Plastic Bottle Image Dataset”
    - **Dataset 2:** “Drinking Waste Classification”
- Dataset 1 already contained properly labeled bottle images.
- Dataset 2 was cleaned by:
    - Filtering only PET bottle images.
    - Renaming files to remove commas for compatibility.
    - Remapping their labels to `0` (bottle class).
- Both datasets were then merged into a single YOLOv8-compatible directory structure with `train` and `val` splits.

🔷 Configuration

- Created a YOLO data configuration file (`bottle.yaml`) which defines:
    - Dataset path.
    - Train and validation folders.
    - Number of classes (`nc=1`).
    - Class names: `['bottle']`.

🔷 Model Training

- Used the YOLOv8 nano pretrained weights (`yolov8n.pt`) for faster training.
- Trained the model for 20 epochs on the merged dataset.
- Saved the best-performing weights (`best.pt`) for later inference.

🔷 Model Validation & Testing

- Ran validation on the held-out validation set to measure metrics:
    - Precision, Recall, mAP@0.5, and mAP@0.5:0.95.
- Ran inference on a test image and displayed the results inline.

🧾2️⃣ Directory Structure

The final dataset and code are organized into a clear structure, as per YOLOv8 standards:

css
CopyEdit

```
/content/
├── bottle_data/
│   ├── images/
│   │   ├── train/
│   │   └── val/
```

```
|   └── labels/
|       ├── train/
|       └── val/
├── bottle.yaml          ← YOLO data configuration file
├── bottle-detect/
|   └── v8n-plastic/
|       └── weights/
|           ├── best.pt
|           └── last.pt
├── bottle_detection.py  ← Main code script
```

📁 **Dataset folders:**

- `images/train/` and `images/val/` contain the training and validation images.
- `labels/train/` and `labels/val/` contain corresponding YOLO-format annotation `.txt` files.

📁 **Config & Weights:**

- `bottle.yaml` defines the dataset for YOLOv8.
- Best-trained model is saved as `best.pt` in `weights/`.

🧾3️⃣Code and Logic

🔷 Install and Configure Kaggle

python
CopyEdit

```
!pip install kaggle
!mkdir -p ~/.kaggle
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
```

🎯 *Installs the Kaggle API and sets up credentials for downloading datasets.*

🔷 Clean Dataset 2

python
CopyEdit

```
# Filters PET images, cleans filenames, and rewrites labels as class 0
# Saves cleaned files in PET_only/images and PET_only/labels
```

🎯 *Keeps only plastic PET bottles from Dataset 2 and prepares YOLO-format labels.*

🔷 Merge Datasets

python
CopyEdit

```
# Copies Dataset 1 and cleaned Dataset 2 into /content/bottle_data
# Organizes into YOLOv8 structure: images/train, labels/train,
images/val, labels/val
```

🎯 *Creates the dataset folder YOLO expects.*

🔷 Create Config File

python
CopyEdit

```
yaml_content = """
path: /content/bottle_data
train: images/train
val: images/val

nc: 1
names: ['bottle']
"""
with open('/content/bottle.yaml', 'w') as f:
```

```
    f.write(yaml_content.strip())
```

🎯 *Tells YOLOv8 where to find data and what classes to expect.*

🔷 Train YOLOv8

python
CopyEdit

```
from ultralytics import YOLO
model = YOLO('yolov8n.pt')
results = model.train(
    data='/content/bottle.yaml',
    epochs=20,
    imgsz=640,
    batch=16,
    project='bottle-detect',
    name='v8n-plastic'
)
```

🎯 *Trains YOLOv8 for 20 epochs. Best weights are saved as `best.pt`.*

🔷 Validate & Test

python
CopyEdit

```
metrics = model.val(data='/content/bottle.yaml')

results = model('/content/test_image.jpg', save=True)

# Display result
from IPython.display import Image, display
import glob
output_dir = results[0].save_dir
predicted_image = glob.glob(f"{output_dir}/*.jpg")[0]
display(Image(filename=predicted_image))
```

🎯 *Evaluates model performance and shows predictions on a test image.*

📋 Summary

- Downloaded and prepared two datasets.
- Merged them into YOLOv8 format.
- Trained a YOLOv8-nano detector on plastic bottles.
- Evaluated its performance using standard detection metrics.
- Demonstrated detection on new images.