MTH423 Project

İsmail Berkay Yücel
Onat Özgen
Ömer Başar Sönmez

Augmentation & Preprocessing

Preprocessing	`No preprocessing steps were applied.`
Augmentations	**Outputs per training example:** `3` **Rotation:** `Between -10° and +10°` **Saturation:** `Between -25% and +25%` **Blur:** `Up to 2.5px` **Noise:** `Up to 0.1% of pixels` **Bounding Box: Rotation:** `Between -7° and +7°` **Bounding Box: Brightness:** `Between -10% and +10%` **Bounding Box: Blur:** `Up to 0.9px` **Bounding Box: Noise:** `Up to 0.1% of pixels`

Profile A

Profile A – “Classic Blur/Noise”		
Step	**Range**	**Why we use it**
Rotation	±10 °	Camera tilt robustness
Saturation	–25 % … +25 %	Colour-cast & weather variation
Blur	≤ 2.5 px Gaussian	Motion / focus blur tolerance
Noise	0.1 % pixels	Sensor / JPEG artefacts
BBox aug. Rotation / Blur / Noise	Same ranges but inside boxes only	Hardens classifier inside small objects

Augmentation & Preprocessing

Preprocessing	**Auto-Orient:** `Applied` **Resize:** `Stretch to 640x640` **Auto-Adjust Contrast:** `Using Adaptive Equalization`
Augmentations	**Outputs per training example:** `3` **Rotation:** `Between -10° and +10°` **Brightness:** `Between -11% and +11%` **Exposure:** `Between -8% and +8%` **Mosaic:** `Applied` **Bounding Box: Shear:** `±10° Horizontal, ±10° Vertical` **Bounding Box: Brightness:** `Between -10% and +10%` **Bounding Box: Exposure:** `Between -10% and +10%`

Profile B

Profile B – "CLAHE + Mosaic + Shear" *(current best)*

Step	Range / Type	Benefit
Auto-Orient	Fix EXIF	Removes up-side-down frames
CLAHE (adaptive contrast)	—	Recovers dark shadows & night shots
Resize (stretch)	640 × 640	Uniform tensor, upsizes tiny objects
Rotation	±10 °	Same robustness
Brightness	–11 % … +11 %	Day / night simulation
Exposure	–8 % … +8 %	Head-light / tunnel adaptation
Mosaic	4-image collage	Boosts small-object recall (traffic signs far away)
BBox Shear	±10 ° H & V	Perspective distortion for moving camera
BBox Brightness / Exposure	±10 %	Local glare & shadow on object

Comparison of YOLO Models

Dataset = Augmented Epoch = 15 İmage size = 640 Batch = 16 Confidence 0.001

YOLOv5	YOLOv8
mAP@0.5	mAP@0.5
0.18022	0.42194

Comparison of Batch Size

Dataset = Augmented Epoch = 15 İmage size = 640 Confidence 0.10

16 Batch	64 Batch
mAP@0.5	mAP@0.5
0.42194	0.56402

Comparison of Confidence

Dataset = Augmented Epoch = 100 İmage size = 640 Batch = 16

0.20 conf	0.10 conf
mAP@0.5	mAP@0.5
0.58083	0.59781

Impact of Epoch Count

Dataset = Not Augmented İmage size = 640 Batch = 64 Confidence 0.001

15 Epoch	100 Epoch	200 Epoch
mAP@0.5	mAP@0.5	mAP@0.5
0.59128	0.61243	0.61852

Our Best

Dataset = Not Augmented Epoch = 100 İmage size = 896 Batch = 32 Confidence 0.001

YOLOv8m

mAP@0.5

0.64047

Which Model?

YOLOv8

YOLO v8 delivers the highest mAP at real-time speed; a simple pip install plus its anchor-free design lets us fine-tune with a single line of code.

Competing options (YOLO v5/v7, Faster R-CNN, DETR) are either slower or less accurate on small objects such as traffic signs and distant pedestrians.

YOLOV8

Özellik	Açıklama
Anchor-Free	Önceki sürümlerdeki anchor kutular kaldırıldı. Her grid hücresi doğrudan kutu üretir.
Backbone	C2f blokları + SPPF (Fast Spatial Pyramid Po
Neck	PAN-FPN yapısı (çok ölçekli çıktı: P3, P4.
Başlık (Head)	Decoupled Head + Distribution Focal Loss (D
Görev Türleri	detect, segment, classify, pose, obb
Export Formatları	ONNX, TFLite, CoreML, OpenVINO, TF.js, Paddle, NC

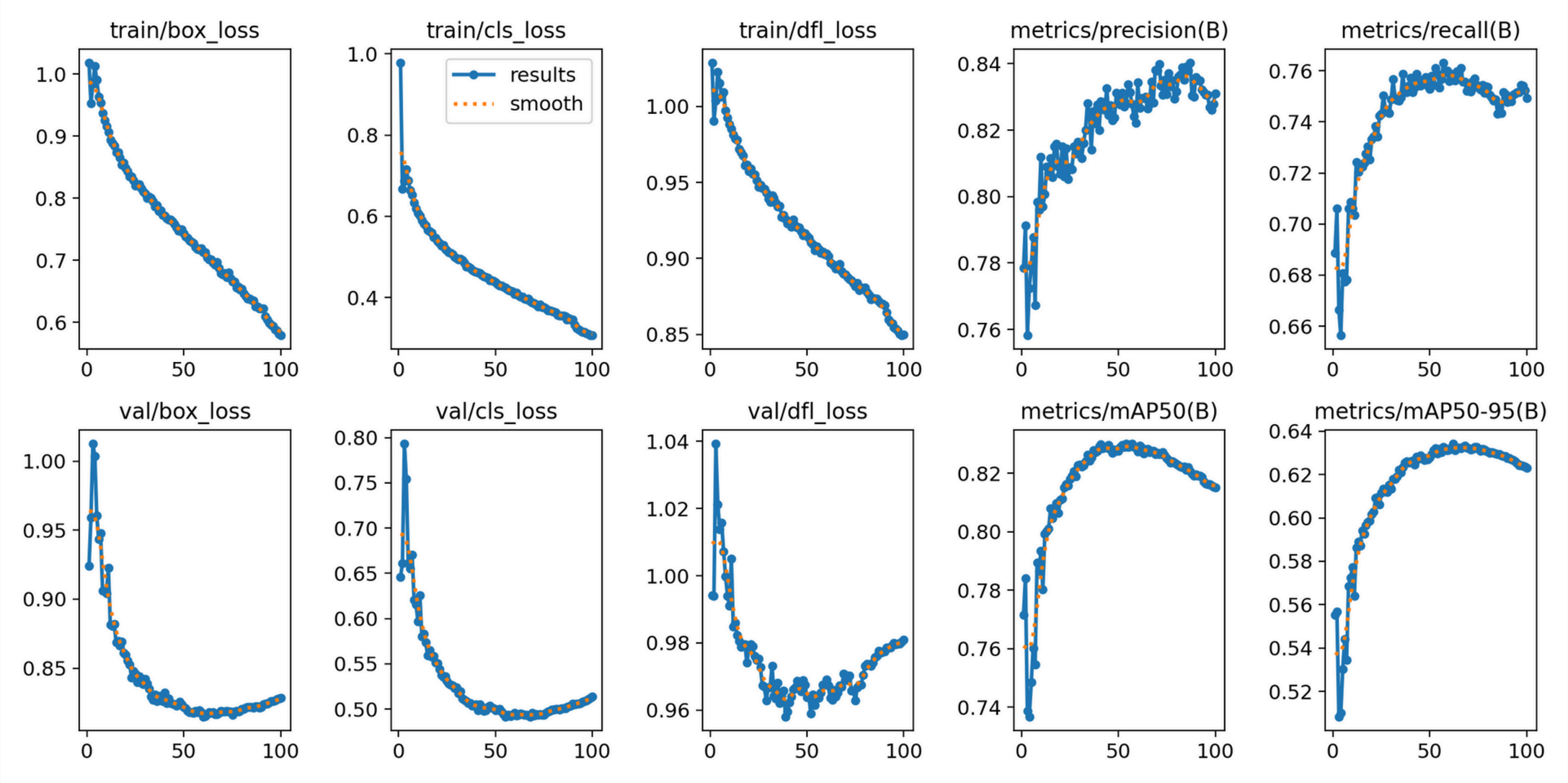
In our project we decided to use YOLOV8m which include 25m parameters. It is ideal for medium scale projects.

Training

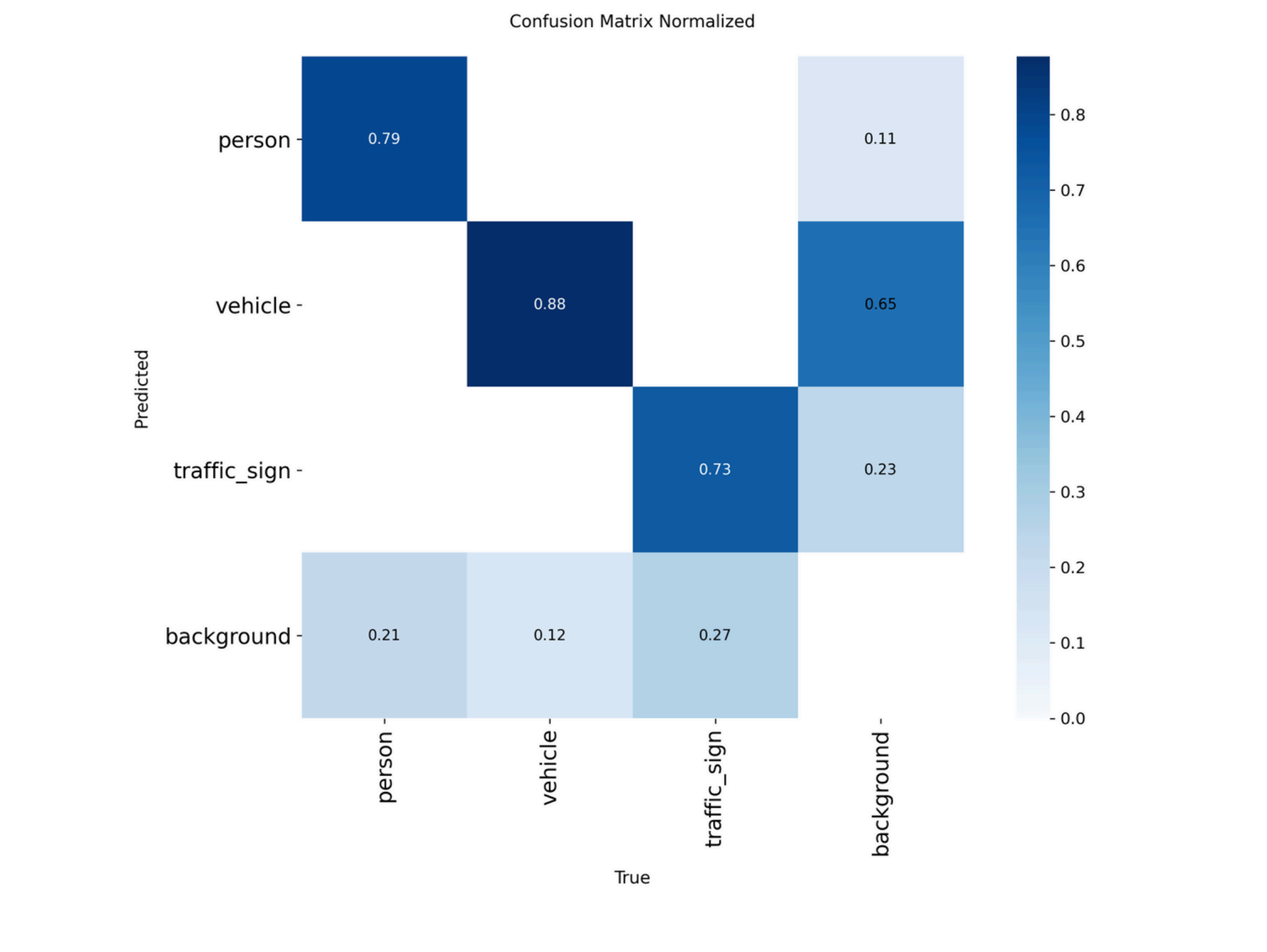
```python
from ultralytics import YOLO
model = YOLO('yolov8m.pt')

model.train(
    data='/content/data.yaml',
    epochs=200,
    imgsz=640,
    batch=64,
    device=0,
    project='/content/runs',
    name='yolov8m_200ep',
    save_period=10
)

metrics = YOLO('/content/runs/yolov8m_200ep/weights/best.pt').val(data='/content/data.yaml')
print(f"mAP@0.5 : {metrics.box.map50:.3f}")
print(f"mAP@0.5:.95 : {metrics.box.map:.3f}")

# Drive'a taşı
DEST = f'{ROOT}/yolo_outputs/yolov8m_200ep'
shutil.copytree('/content/runs/yolov8m_200ep', DEST, dirs_exist_ok=True)
print('✓ outputs copied to Drive →', DEST)
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     87/200      24.2G     0.7038     0.4064     0.8812       1032        640: 100%|██████████| 126/126 [00:50<00:00,  2.47it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:08<00:00,  1.83it/s]

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     88/200      24.4G     0.7081     0.4076     0.8803       1218        640: 100%|██████████| 126/126 [00:50<00:00,  2.47it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:08<00:00,  1.86it/s]

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     89/200      24.4G     0.7033     0.4059     0.8802       1051        640: 100%|██████████| 126/126 [00:50<00:00,  2.47it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:08<00:00,  1.87it/s]

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
     90/200      23.5G     0.6964     0.4025     0.8793       1090        640: 100%|██████████| 126/126 [00:50<00:00,  2.47it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 16/16 [00:08<00:00,  1.86it/s]
```

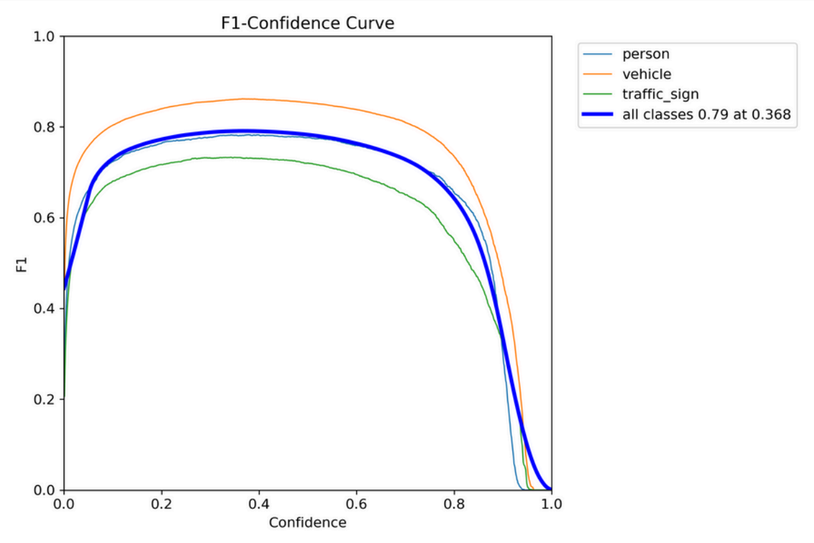
Model Training Process

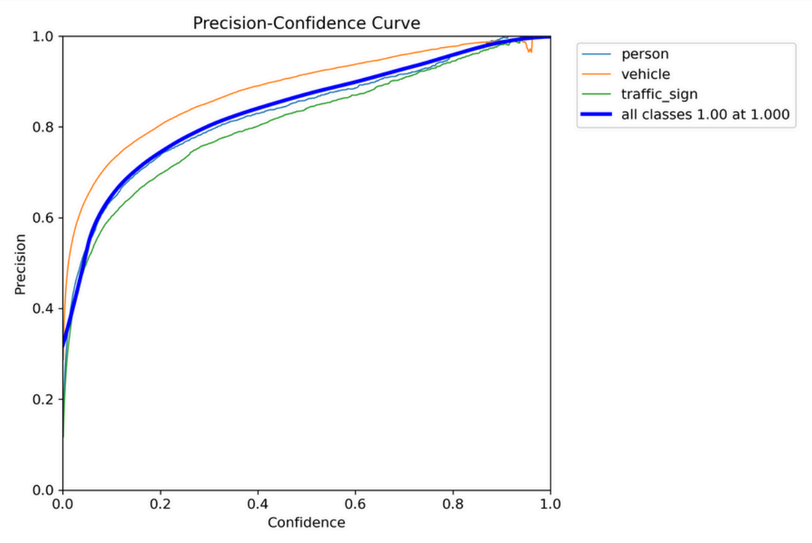


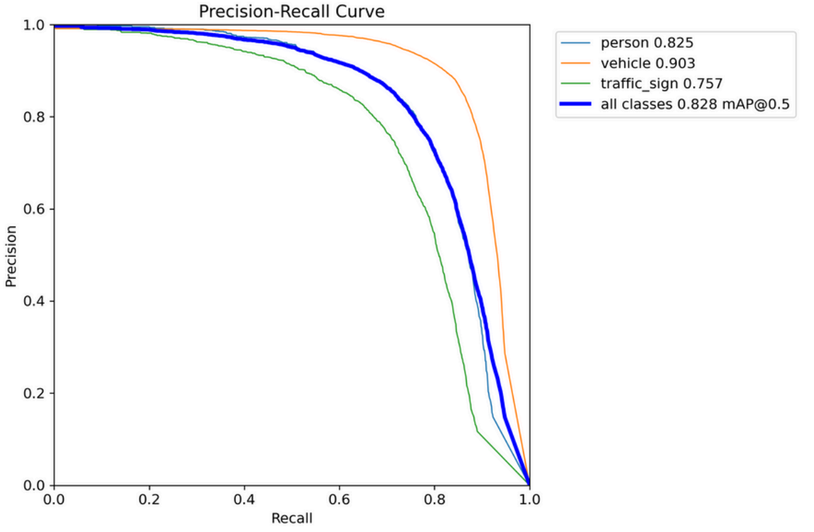
Confusion Matrix

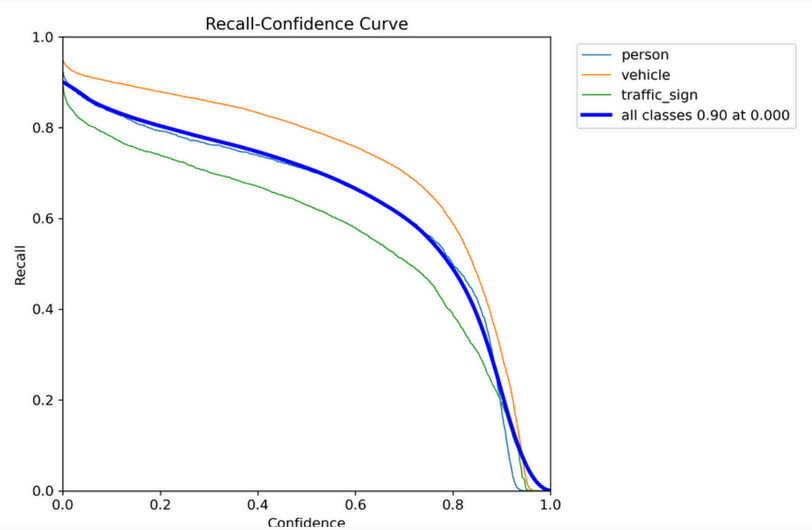


Per-Class Detection Curves









Submission

Elden Burros 0.64047

```python
TEMPLATE_CSV = '/content/drive/MyDrive/MTH_proje/mth-423-computer-vision-for-smart-vehicles-sp-25/yolo_outputs/test_empty_submission_spring25.csv'
WEIGHTS_PT   = '/content/drive/MyDrive/MTH_proje/mth-423-computer-vision-for-smart-vehicles-sp-25/yolo_outputs/yolov8m_200ep/weights/best.pt'
TEST_DIR     = '/content/drive/MyDrive/MTH_proje/mth-423-computer-vision-for-smart-vehicles-sp-25/test/images'
OUT_CSV      = '/content/drive/MyDrive/MTH_proje/mth-423-computer-vision-for-smart-vehicles-sp-25/yolo_outputs/submission_200ep1.csv'

from ultralytics import YOLO
import pandas as pd, cv2, glob, os
from tqdm.auto import tqdm

class_map = {
    0: 'pedestrian',
    1: 'car',
    2: 'traffic sign'
}

template_df = pd.read_csv(TEMPLATE_CSV)
print(f'Şablon satır sayısı: {len(template_df)}')


slots = {}
for idx, row in template_df.iterrows():
    slots.setdefault(row['img_name'], []).append(idx)

model = YOLO(WEIGHTS_PT)
```

```python
image_paths = sorted(glob.glob(os.path.join(TEST_DIR, '*')))
for img_path in tqdm(image_paths, desc='Predicting'):
    img_name = os.path.basename(img_path)
    img      = cv2.imread(img_path)
    if img is None:
        continue
    h, w = img.shape[:2]

    result = model(img_path, conf=0.001, iou=0.5, verbose=False)[0]

    for box, cid in zip(result.boxes.xyxy.cpu().numpy(),
                        result.boxes.cls.cpu().numpy().astype(int)):

        if not slots.get(img_name):
            continue

        idx = slots[img_name].pop(0)
        x1, y1, x2, y2 = box
        template_df.at[idx, 'cls'] = class_map.get(cid, f'class_{cid}')
        template_df.at[idx, 'x1']  = max(0, min(1, x1 / w))
        template_df.at[idx, 'y1']  = max(0, min(1, y1 / h))
        template_df.at[idx, 'x2']  = max(0, min(1, x2 / w))
        template_df.at[idx, 'y2']  = max(0, min(1, y2 / h))

assert list(template_df.columns) == ['ID', 'img_name', 'cls', 'x1', 'y1', 'x2', 'y2'], "Sütun sırası bozuldu!"
assert len(template_df) == 27934, "Satır sayısı şablonla uyuşmuyor!"

template_df.to_csv(OUT_CSV, index=False)
print(f'Submission hazır → {OUT_CSV}')
```

Thanks For Listening