```
#install dependencies
!pip install ultralytics --upgrade
!pip install opencv-python matplotlib tqdm

Collecting ultralytics
  Downloading ultralytics-8.3.241-py3-none-any.whl.metadata (37 kB)
Requirement already satisfied: numpy>=1.23.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (2.0.2)
Requirement already satisfied: matplotlib>=3.3.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (3.10.0)
Requirement already satisfied: opencv-python>=4.6.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (4.12.0.88)
Requirement already satisfied: pillow>=7.1.2 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (11.3.0)
Requirement already satisfied: pyyaml>=5.3.1 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (6.0.3)
Requirement already satisfied: requests>=2.23.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (2.32.5)
Requirement already satisfied: scipy>=1.4.1 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (1.15.3)
Requirement already satisfied: torch>=1.8.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics)
(2.8.0+cu126)
Requirement already satisfied: torchvision>=0.9.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics)
(0.23.0+cu126)
Requirement already satisfied: psutil>=5.8.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (5.9.5)
Requirement already satisfied: polars>=0.20.0 in
/usr/local/lib/python3.12/dist-packages (from ultralytics) (1.25.2)
Collecting ultralytics-thop>=2.0.18 (from ultralytics)
  Downloading ultralytics_thop-2.0.18-py3-none-any.whl.metadata (14
kB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (1.3.3)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (1.4.9)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (25.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
```

```
>ultralytics) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.12/dist-packages (from matplotlib>=3.3.0-
>ultralytics) (2.9.0.post0)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0-
>ultralytics) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0-
>ultralytics) (3.11)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0-
>ultralytics) (2.6.2)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests>=2.23.0-
>ultralytics) (2025.11.12)
Requirement already satisfied: filelock in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (3.20.1)
Requirement already satisfied: typing-extensions>=4.10.0 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (4.15.0)
Requirement already satisfied: setuptools in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (75.2.0)
Requirement already satisfied: sympy>=1.13.3 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (1.13.3)
Requirement already satisfied: networkx in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (3.5)
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (3.1.6)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (2025.10.0)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.6.77)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.6.77)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.6.80)
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (9.10.2.21)
```

```
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.6.4.1)
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (11.3.0.4)
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (10.3.7.77)
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (11.7.1.2)
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.5.4.2)
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (0.7.1)
Requirement already satisfied: nvidia-nccl-cu12==2.27.3 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (2.27.3)
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.6.77)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (12.6.85)
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (1.11.1.6)
Requirement already satisfied: triton==3.4.0 in
/usr/local/lib/python3.12/dist-packages (from torch>=1.8.0-
>ultralytics) (3.4.0)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7-
>matplotlib>=3.3.0->ultralytics) (1.17.0)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3-
>torch>=1.8.0->ultralytics) (1.3.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.12/dist-packages (from jinja2->torch>=1.8.0-
>ultralytics) (3.0.3)
Downloading ultralytics-8.3.241-py3-none-any.whl (1.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.1/1.1 MB 20.1 MB/s eta
0:00:00a 0:00:01
ent already satisfied: opencv-python in
/usr/local/lib/python3.12/dist-packages (4.12.0.88)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.12/dist-packages (3.10.0)
```

```
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (4.67.1)
Requirement already satisfied: numpy<2.3.0,>=2 in /usr/local/lib/python3.12/dist-packages (from opencv-python) (2.0.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.60.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.5)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17.0)
```

```
# Download the Car Parts dataset directly from GitHub

!git clone https://github.com/dsmlr/Car-Parts-Segmentation.git
```

```
fatal: destination path 'Car-Parts-Segmentation' already exists and is not an empty directory.
```

```
#import libraries

import os
import json
import shutil
import cv2
from tqdm import tqdm
from ultralytics import YOLO
import matplotlib.pyplot as plt
import random
```

```
Creating new Ultralytics Settings v0.0.6 file □
View Ultralytics Settings with 'yolo settings' or at '/root/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com/quickstart/#ultralytics-settings.
```

```
# Define source and destination directories
base_input = '/kaggle/input/car-parts-detection-dataset'
base_work  = '/kaggle/working/carparts_yolo_ready'

train_input = os.path.join(base_input, 'trainingset', 'JPEGImages')
test_input  = os.path.join(base_input, 'testset', 'JPEGImages')

train_work = os.path.join(base_work, 'trainingset')
test_work  = os.path.join(base_work, 'testset')

# Create YOLO-style directory structure
for split in ['trainingset', 'testset']:
    os.makedirs(os.path.join(base_work, split, 'images'),
exist_ok=True)
    os.makedirs(os.path.join(base_work, split, 'labels'),
exist_ok=True)

# Copy images
shutil.copytree(train_input, os.path.join(train_work, 'images'),
dirs_exist_ok=True)
shutil.copytree(test_input,  os.path.join(test_work, 'images'),
dirs_exist_ok=True)
print("□ Images copied to working directory")

# Function to convert COCO JSON annotations → YOLO .txt labels
def convert_annotations(json_file, img_dir, label_dir):
    with open(json_file) as f:
        data = json.load(f)
    images = {img['id']: img for img in data['images']}

    for ann in tqdm(data['annotations'], desc=f"Converting
{os.path.basename(json_file)}"):
        if 'bbox' not in ann:
            continue
        img_info = images[ann['image_id']]
        x, y, w, h = ann['bbox']
        img_path = os.path.join(img_dir, img_info['file_name'])
        img_cv = cv2.imread(img_path)
        if img_cv is None:  # skip unreadable
            continue
        ih, iw = img_cv.shape[:2]
        # Normalize
        x_center = (x + w/2) / iw
        y_center = (y + h/2) / ih
        w /= iw
        h /= ih
        cls = int(ann['category_id'])
        # Write YOLO-format label
        label_path = os.path.join(label_dir,
os.path.splitext(img_info['file_name'])[0] + '.txt')
```

```
        with open(label_path, 'a') as f:
            f.write(f"{cls} {x_center:.6f} {y_center:.6f} {w:.6f}
{h:.6f}\n")

# Convert both train and test annotations
convert_annotations(os.path.join(base_input, 'trainingset',
'annotations.json'),
                    os.path.join(train_work, 'images'),
                    os.path.join(train_work, 'labels'))

convert_annotations(os.path.join(base_input, 'testset',
'annotations.json'),
                    os.path.join(test_work, 'images'),
                    os.path.join(test_work, 'labels'))

print("✅ Labels generated successfully")
```

```
✅ Images copied to working directory

Converting annotations.json: 100%|██████████| 3073/3073 [00:04<00:00,
718.38it/s]
Converting annotations.json: 100%|██████████| 811/811 [00:01<00:00,
576.97it/s]

✅ Labels generated successfully
```

```
yaml_path = '/kaggle/working/carparts_yolo_ready/carparts.yaml'

yaml_content = """
train: /kaggle/working/carparts_yolo_ready/trainingset/images
val: /kaggle/working/carparts_yolo_ready/testset/images

nc: 20
names: [
  'front_bumper', 'rear_bumper', 'door', 'hood', 'wheel', 'window',
  'mirror', 'headlight', 'taillight', 'grille', 'trunk', 'roof',
  'fender', 'side_skirt', 'windshield', 'license_plate',
  'spoiler', 'logo', 'handle', 'fuel_cap'
]
"""

with open(yaml_path, 'w') as f:
    f.write(yaml_content)

print("✅ YAML updated successfully!")
print(open(yaml_path).read())
```

```
✅ YAML updated successfully!
```

```
train: /kaggle/working/carparts_yolo_ready/trainingset/images
val: /kaggle/working/carparts_yolo_ready/testset/images

nc: 20
names: [
  'front_bumper', 'rear_bumper', 'door', 'hood', 'wheel', 'window',
  'mirror', 'headlight', 'taillight', 'grille', 'trunk', 'roof',
  'fender', 'side_skirt', 'windshield', 'license_plate',
  'spoiler', 'logo', 'handle', 'fuel_cap'
]


#clear cache
!rm -f /kaggle/working/carparts_yolo_ready/trainingset/labels.cache
!rm -f /kaggle/working/carparts_yolo_ready/testset/labels.cache
print("Cache cleared □")

Cache cleared □

train_img_dir =
'/kaggle/working/carparts_yolo_ready/trainingset/images'
train_lbl_dir =
'/kaggle/working/carparts_yolo_ready/trainingset/labels'
test_img_dir  = '/kaggle/working/carparts_yolo_ready/testset/images'
test_lbl_dir  = '/kaggle/working/carparts_yolo_ready/testset/labels'

print("Training images:", len(os.listdir(train_img_dir)))
print("Training labels:", len(os.listdir(train_lbl_dir)))
print("Testing images:", len(os.listdir(test_img_dir)))
print("Testing labels:", len(os.listdir(test_lbl_dir)))

Training images: 400
Training labels: 400
Testing images: 100
Testing labels: 100

#verify label file contents
import os

train_lbl_dir =
'/kaggle/working/carparts_yolo_ready/trainingset/labels'
bad = []
for f in os.listdir(train_lbl_dir):
    if f.endswith('.txt'):
        with open(os.path.join(train_lbl_dir, f)) as fh:
            for line in fh:
                parts = line.strip().split()
                # must be 5 numbers, all convertible to float
                if len(parts) != 5:
                    bad.append(f)
                    break
```

```
                try:
                    _ = [float(x) for x in parts[1:]]
                except:
                    bad.append(f)
                    break
print("Files with bad label lines:", len(bad))
for b in bad[:10]:
    print(" ->", b)
```

```
Files with bad label lines: 0
```

```
#autorepair misinformed labels
import json, cv2, os
from tqdm import tqdm

base_in = '/kaggle/input/car-parts-detection-dataset'
base_out = '/kaggle/working/carparts_yolo_ready'

def safe_convert(split):
    src_json = f'{base_in}/{split}/annotations.json'
    img_dir  = f'{base_out}/{split}/images'
    lbl_dir  = f'{base_out}/{split}/labels'
    os.makedirs(lbl_dir, exist_ok=True)

    with open(src_json) as f:
        coco = json.load(f)
    imgs = {im['id']: im for im in coco['images']}
    print(f"Re-creating labels for {split} …")

    n_ok = 0
    for ann in tqdm(coco['annotations']):
        if 'bbox' not in ann or len(ann['bbox']) != 4:
            continue
        img_info = imgs.get(ann['image_id'])
        if not img_info:
            continue
        path = os.path.join(img_dir, img_info['file_name'])
        im = cv2.imread(path)
        if im is None:
            continue
        h, w = im.shape[:2]
        x, y, bw, bh = ann['bbox']
        if bw<=0 or bh<=0:
            continue
        x_c=(x+bw/2)/w; y_c=(y+bh/2)/h; bw/=w; bh/=h
        vals=[x_c,y_c,bw,bh]
        if not all(0<=v<=1 for v in vals):
            continue
        cls=int(ann['category_id'])-1   # zero-index
        with open(os.path.join(lbl_dir,
```

```
os.path.splitext(img_info['file_name'])[0]+'.txt'),'a') as f:
            f.write(f"{cls} {x_c:.6f} {y_c:.6f} {bw:.6f} {bh:.6f}\n")
        n_ok+=1
    print(f"□ {n_ok} valid boxes written for {split}")

safe_convert('trainingset')
safe_convert('testset')
```

```
Re-creating labels for trainingset …

100%|██████████| 3073/3073 [00:04<00:00, 696.24it/s]

□ 3073 valid boxes written for trainingset
Re-creating labels for testset …

100%|██████████| 811/811 [00:01<00:00, 567.57it/s]

□ 811 valid boxes written for testset
```

```
#clear yolo cache and re-check counts
!rm -f /kaggle/working/carparts_yolo_ready/trainingset/labels.cache
!rm -f /kaggle/working/carparts_yolo_ready/testset/labels.cache
print("Caches cleared □")

print("Label files in train:",
len(os.listdir('/kaggle/working/carparts_yolo_ready/trainingset/labels
')))
```

```
Caches cleared □
Label files in train: 400
```

```
from ultralytics import YOLO
model = YOLO('yolo11n.pt')

results = model.train(
    data='/kaggle/working/carparts_yolo_ready/carparts.yaml',
    epochs=50,
    imgsz=640,
    batch=8,
    project='/kaggle/working',
    name='car_parts_detection_fixed_classes',
    amp=False,
    deterministic=False
)
```

```
Ultralytics 8.3.241 □ Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla
T4, 15095MiB)
engine/trainer: agnostic_nms=False, amp=False, augment=False,
auto_augment=randaugment, batch=8, bgr=0.0, box=7.5, cache=False,
cfg=None, classes=None, close_mosaic=10, cls=0.5, compile=False,
```

```
conf=None, copy_paste=0.0, copy_paste_mode=flip, cos_lr=False,
cutmix=0.0, data=/kaggle/working/carparts_yolo_ready/carparts.yaml,
degrees=0.0, deterministic=False, device=None, dfl=1.5, dnn=False,
dropout=0.0, dynamic=False, embed=None, epochs=50, erasing=0.4,
exist_ok=False, fliplr=0.5, flipud=0.0, format=torchscript,
fraction=1.0, freeze=None, half=False, hsv_h=0.015, hsv_s=0.7,
hsv_v=0.4, imgsz=640, int8=False, iou=0.7, keras=False, kobj=1.0,
line_width=None, lr0=0.01, lrf=0.01, mask_ratio=4, max_det=300,
mixup=0.0, mode=train, model=yolo11n.pt, momentum=0.937, mosaic=1.0,
multi_scale=False, name=car_parts_detection_fixed_classes, nbs=64,
nms=False, opset=None, optimize=False, optimizer=auto,
overlap_mask=True, patience=100, perspective=0.0, plots=True,
pose=12.0, pretrained=True, profile=False, project=/kaggle/working,
rect=False, resume=False, retina_masks=False, save=True,
save_conf=False, save_crop=False,
save_dir=/kaggle/working/car_parts_detection_fixed_classes,
save_frames=False, save_json=False, save_period=-1, save_txt=False,
scale=0.5, seed=0, shear=0.0, show=False, show_boxes=True,
show_conf=True, show_labels=True, simplify=True, single_cls=False,
source=None, split=val, stream_buffer=False, task=detect, time=None,
tracker=botsort.yaml, translate=0.1, val=True, verbose=True,
vid_stride=1, visualize=False, warmup_bias_lr=0.1, warmup_epochs=3.0,
warmup_momentum=0.8, weight_decay=0.0005, workers=8, workspace=None
Overriding model.yaml nc=80 with nc=20

                   from  n    params  module
arguments
  0                  -1  1       464  ultralytics.nn.modules.conv.Conv
[3, 16, 3, 2]
  1                  -1  1      4672  ultralytics.nn.modules.conv.Conv
[16, 32, 3, 2]
  2                  -1  1      6640
ultralytics.nn.modules.block.C3k2            [32, 64, 1, False, 0.25]

  3                  -1  1     36992  ultralytics.nn.modules.conv.Conv
[64, 64, 3, 2]
  4                  -1  1     26080
ultralytics.nn.modules.block.C3k2            [64, 128, 1, False, 0.25]

  5                  -1  1    147712  ultralytics.nn.modules.conv.Conv
[128, 128, 3, 2]
  6                  -1  1     87040
ultralytics.nn.modules.block.C3k2            [128, 128, 1, True]

  7                  -1  1    295424  ultralytics.nn.modules.conv.Conv
[128, 256, 3, 2]
  8                  -1  1    346112
ultralytics.nn.modules.block.C3k2            [256, 256, 1, True]

  9                  -1  1    164608
```

```
ultralytics.nn.modules.block.SPPF             [256, 256, 5]

 10                  -1  1    249728
ultralytics.nn.modules.block.C2PSA            [256, 256, 1]

 11                  -1  1         0
torch.nn.modules.upsampling.Upsample          [None, 2, 'nearest']

 12             [-1, 6]  1         0
ultralytics.nn.modules.conv.Concat            [1]

 13                  -1  1    111296
ultralytics.nn.modules.block.C3k2             [384, 128, 1, False]

 14                  -1  1         0
torch.nn.modules.upsampling.Upsample          [None, 2, 'nearest']

 15             [-1, 4]  1         0
ultralytics.nn.modules.conv.Concat            [1]

 16                  -1  1     32096
ultralytics.nn.modules.block.C3k2             [256, 64, 1, False]

 17                  -1  1     36992  ultralytics.nn.modules.conv.Conv
[64, 64, 3, 2]
 18            [-1, 13]  1         0
ultralytics.nn.modules.conv.Concat            [1]

 19                  -1  1     86720
ultralytics.nn.modules.block.C3k2             [192, 128, 1, False]

 20                  -1  1    147712  ultralytics.nn.modules.conv.Conv
[128, 128, 3, 2]
 21            [-1, 10]  1         0
ultralytics.nn.modules.conv.Concat            [1]

 22                  -1  1    378880
ultralytics.nn.modules.block.C3k2             [384, 256, 1, True]

 23        [16, 19, 22]  1    434572
ultralytics.nn.modules.head.Detect            [20, [64, 128, 256]]

YOLO11n summary: 181 layers, 2,593,740 parameters, 2,593,724
gradients, 6.5 GFLOPs

Transferred 448/499 items from pretrained weights
Freezing layer 'model.23.dfl.conv.weight'
train: Fast image access  (ping: 0.0±0.0 ms, read: 1273.5±606.3 MB/s,
size: 36.0 KB)
train: Scanning
/kaggle/working/carparts_yolo_ready/trainingset/labels... 400 images,
```

```
0 backgrounds, 0 corrupt: 100% ━━━━━━━━━━━━ 400/400 1.4Kit/s 0.3s<0.2s
train: New cache created:
/kaggle/working/carparts_yolo_ready/trainingset/labels.cache
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01,
blur_limit=(3, 7)), ToGray(p=0.01, method='weighted_average',
num_output_channels=3), CLAHE(p=0.01, clip_limit=(1.0, 4.0),
tile_grid_size=(8, 8))
val: Fast image access  (ping: 0.0±0.0 ms, read: 250.3±159.7 MB/s,
size: 38.4 KB)
val: Scanning /kaggle/working/carparts_yolo_ready/testset/labels...
100 images, 0 backgrounds, 0 corrupt: 100% ━━━━━━━━━━━━ 100/100
1.2Kit/s 0.1s
val: New cache created:
/kaggle/working/carparts_yolo_ready/testset/labels.cache
Plotting labels to
/kaggle/working/car_parts_detection_fixed_classes/labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and
'momentum=0.937' and determining best 'optimizer', 'lr0' and
'momentum' automatically...
optimizer: AdamW(lr=0.000417, momentum=0.9) with parameter groups 81
weight(decay=0.0), 88 weight(decay=0.0005), 87 bias(decay=0.0)
Image sizes 640 train, 640 val
Using 2 dataloader workers
Logging results to /kaggle/working/car_parts_detection_fixed_classes
Starting training for 50 epochs...

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 4.2it/s 1.7s0.2ss
                   all        100       1622     0.0693     0.0309
0.0398     0.0331

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622      0.058      0.465
0.141     0.0975

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.338      0.205
0.229      0.152

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
```

```
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622       0.34      0.268
0.235      0.158

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.2it/s 0.7s.1s
                   all        100       1622      0.371      0.333
0.297      0.199

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.3it/s 0.7s.1s
                   all        100       1622      0.516      0.394
0.377      0.258

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.2it/s 0.7s.1s
                   all        100       1622      0.489      0.453
0.412      0.289

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.463      0.524
0.448      0.299

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.468      0.566
0.464      0.319

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.5it/s 0.7s0.1s
                   all        100       1622      0.516      0.578
0.527      0.369

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.4it/s 0.7s0.1s
```

```
                   all        100       1622      0.523      0.602
0.542      0.374

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.544      0.601
0.552      0.388

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.4it/s 0.7s0.1s
                   all        100       1622      0.558      0.572
0.552      0.388

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622       0.56      0.634
0.581      0.416

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622      0.548      0.644
0.573      0.407

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.589      0.639
0.586      0.423

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.5it/s 0.7s0.1s
                   all        100       1622      0.602      0.683
0.61      0.439

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.601      0.686
```

```
0.616      0.443

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.605      0.675
0.614       0.45

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.607      0.689
0.626      0.458

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.605      0.713
0.654      0.472

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.5it/s 0.7s0.1s
                   all        100       1622      0.629      0.714
0.66      0.482

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622       0.62      0.734
0.664      0.492

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.2it/s 0.8s0.1s
                   all        100       1622      0.613      0.726
0.649      0.471

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.616      0.739
0.658      0.485
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.593       0.76
0.658      0.483

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.0it/s 0.7s.1s
                   all        100       1622      0.612      0.728
0.656      0.482

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.598      0.741
0.647      0.478

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.597      0.786
0.669      0.493

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.0it/s 0.7s.1s
                   all        100       1622      0.607      0.757
0.653      0.487

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.601      0.773
0.662      0.491

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.592      0.772
0.665      0.494
```

```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.0it/s 0.7s.1s
                   all        100       1622      0.613      0.785
0.673      0.501

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622      0.625       0.78
0.68      0.507

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.635      0.771
0.669      0.496

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.5it/s 0.7s0.1s
                   all        100       1622      0.611      0.791
0.672      0.495

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622      0.616       0.78
0.675      0.493

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 10.1it/s 0.7s.1s
                   all        100       1622      0.629      0.764
0.667      0.494

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.624      0.776
0.678      0.507

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
```

```
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.622      0.782
0.671      0.504
Closing dataloader mosaic
albumentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01,
blur_limit=(3, 7)), ToGray(p=0.01, method='weighted_average',
num_output_channels=3), CLAHE(p=0.01, clip_limit=(1.0, 4.0),
tile_grid_size=(8, 8))

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
                   all        100       1622      0.606      0.775
0.662       0.49

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.616      0.783
0.683       0.51

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.2it/s 0.8s0.1s
                   all        100       1622      0.616      0.789
0.677      0.505

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.611      0.806
0.677      0.503

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622      0.608      0.812
0.687      0.509

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.8it/s 0.7s0.1s
```

```
                   all        100       1622      0.614       0.79
0.682      0.507

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622      0.618      0.802
0.684      0.507

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.7it/s 0.7s0.1s
                   all        100       1622       0.63      0.804
0.685      0.508

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.6it/s 0.7s0.1s
                   all        100       1622      0.625      0.792
0.679      0.505

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances
Size
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 9.9it/s 0.7s0.1s
                   all        100       1622       0.63      0.795
0.68      0.507

50 epochs completed in 0.114 hours.
Optimizer stripped from
/kaggle/working/car_parts_detection_fixed_classes/weights/last.pt,
5.5MB
Optimizer stripped from
/kaggle/working/car_parts_detection_fixed_classes/weights/best.pt,
5.5MB

Validating
/kaggle/working/car_parts_detection_fixed_classes/weights/best.pt...
Ultralytics 8.3.241 □ Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla
T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,586,052 parameters, 0
gradients, 6.3 GFLOPs
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 7/7 2.5it/s 2.8s0.2s
                   all        100       1622      0.617      0.783
0.683       0.51
           rear_bumper         18         18      0.733      0.914
```

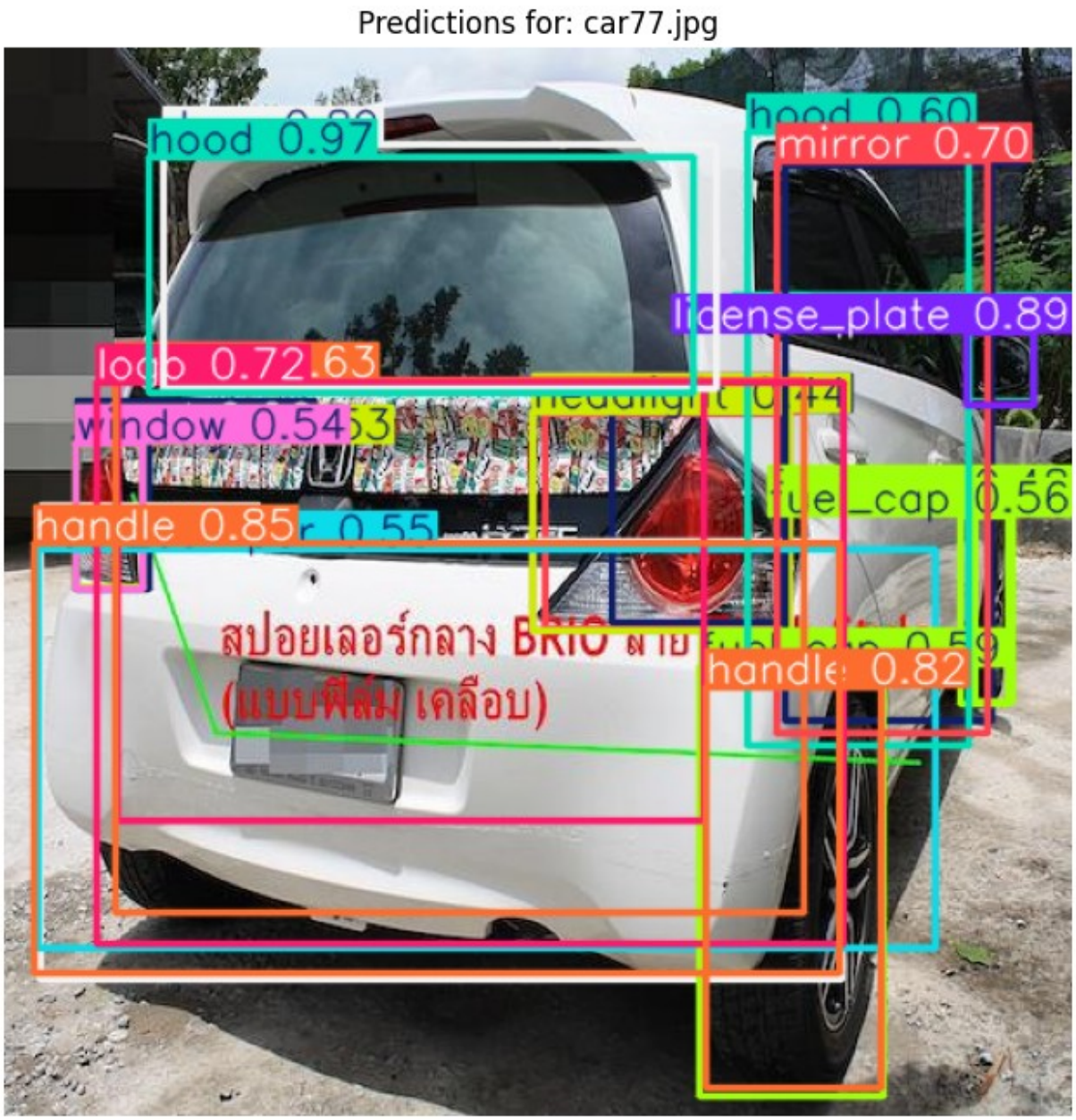
```
0.897      0.733
                door          20          38       0.643       0.816
0.822      0.618
                hood          45          51       0.475       0.921
0.715      0.526
               wheel          42          51       0.467       0.755
0.512       0.39
              window          46          49        0.43       0.857
0.459      0.344
              mirror          43          49       0.459       0.763
0.479      0.357
           headlight          94          94       0.802       0.979
0.955      0.786
           taillight          75         148        0.93       0.953
0.964      0.795
              grille          85         109       0.742        0.87
0.888      0.752
               trunk          70          97       0.436        0.66
0.407      0.298
                roof          85         101       0.473       0.683
0.451      0.312
              fender          73         106       0.558       0.666
0.51      0.388
           side_skirt          78         141        0.67       0.837
0.791      0.558
           windshield          84         129       0.715       0.744
0.798       0.58
         license_plate          87         115       0.901       0.788
0.881      0.501
             spoiler          66          67       0.488       0.716
0.537      0.321
                logo          25          25       0.774        0.76
0.75      0.618
              handle          79         126        0.53       0.518
0.571      0.388
            fuel_cap          76         108       0.496       0.676
0.593       0.43
Speed: 1.2ms preprocess, 9.0ms inference, 0.0ms loss, 8.1ms
postprocess per image
Results saved to /kaggle/working/car_parts_detection_fixed_classes

from ultralytics import YOLO

# Load the best model from training
model =
YOLO('/kaggle/working/car_parts_detection_fixed_classes/weights/best.p
t')

# Evaluate on the validation (test) set
metrics = model.val(
```

```
    data='/kaggle/working/carparts_yolo_ready/carparts.yaml',
    imgsz=640,
    batch=8,
    verbose=True
)

print(f"□ mAP@[IoU=50]: {metrics.box.map50:.4f}")
print(f"□ mAP@[IoU=50:95]: {metrics.box.map:.4f}")
```

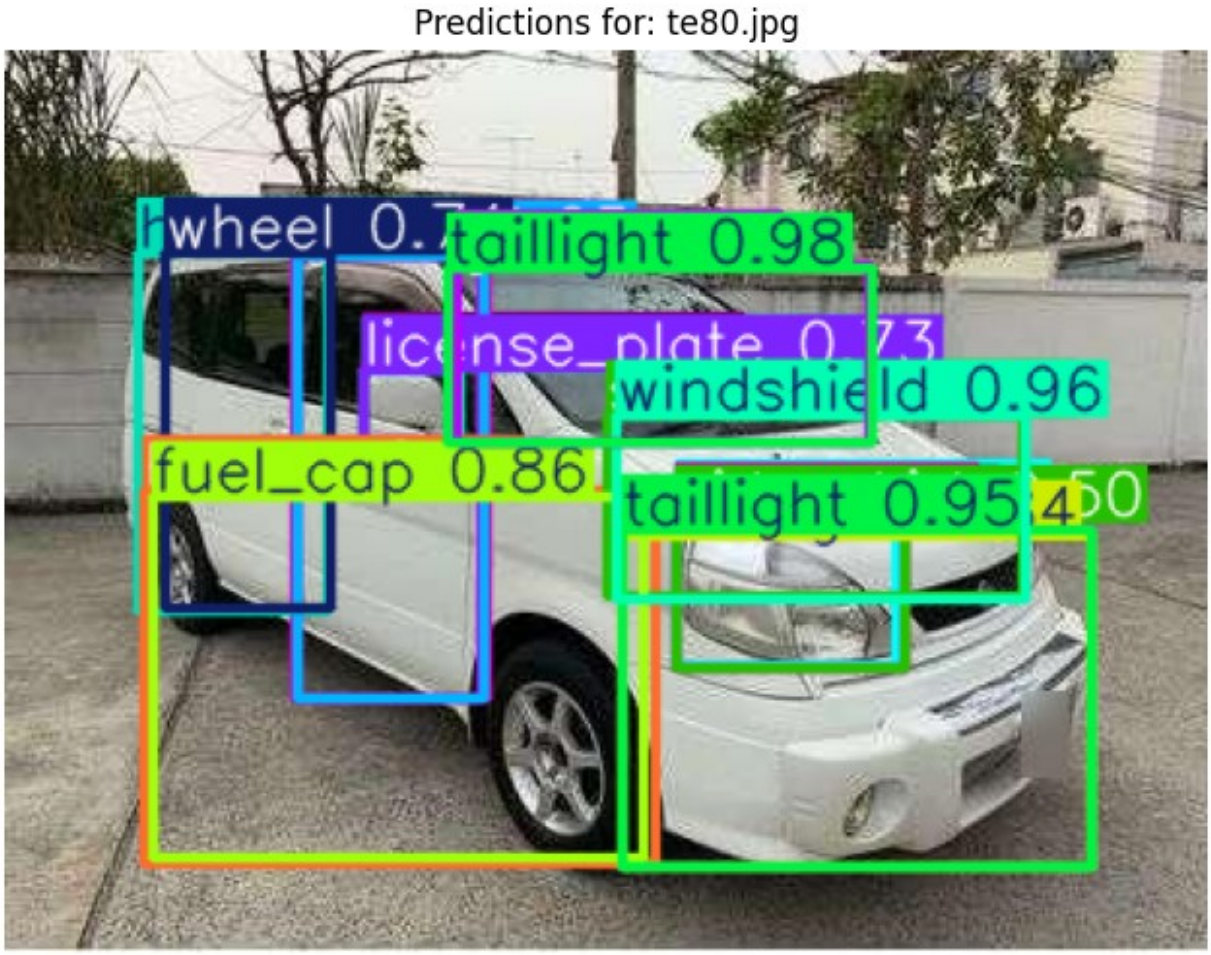
```
Ultralytics 8.3.241 □ Python-3.12.12 torch-2.8.0+cu126 CUDA:0 (Tesla
T4, 15095MiB)
YOLO11n summary (fused): 100 layers, 2,586,052 parameters, 0
gradients, 6.3 GFLOPs
val: Fast image access □ (ping: 0.0±0.0 ms, read: 978.9±410.5 MB/s,
size: 35.2 KB)
val: Scanning
/kaggle/working/carparts_yolo_ready/testset/labels.cache... 100
images, 0 backgrounds, 0 corrupt: 100% ━━━━━━━━━━━━ 100/100 235.1Kit/s
0.0s
ages  Instances      Box(P          R      mAP50  mAP50-95): 100%
━━━━━━━━━━━━ 13/13 5.9it/s 2.2s0.1s
                   all        100       1622      0.613      0.783
0.682       0.51
           rear_bumper         18         18      0.736      0.932
0.901      0.735
                  door         20         38       0.63      0.816
0.819      0.621
                  hood         45         51      0.481      0.922
0.717      0.534
                 wheel         42         51      0.469      0.744
0.484      0.368
                window         46         49      0.432      0.857
0.454      0.343
                mirror         43         49      0.459       0.78
0.467      0.352
             headlight         94         94      0.798      0.979
0.955      0.787
             taillight         75        148      0.917      0.953
0.963      0.794
                 grille        85        109      0.731      0.844
0.877      0.749
                  trunk        70         97      0.434      0.655
0.414        0.3
                   roof        85        101      0.476      0.693
0.469      0.323
                 fender        73        106      0.549       0.66
0.519      0.395
              side_skirt       78        141      0.666      0.834
0.785      0.554
              windshield       84        129      0.699      0.729
```

```
0.798      0.576
         license_plate         87        115      0.893         0.8
0.881        0.5
                spoiler         66         67       0.48       0.716
0.533      0.318
                   logo         25         25      0.769        0.76
0.752      0.615
                 handle         79        126       0.53       0.524
0.572        0.4
                fuel_cap         76        108      0.493       0.685
0.597       0.43
Speed: 1.4ms preprocess, 12.1ms inference, 0.0ms loss, 4.0ms
postprocess per image
Results saved to /kaggle/working/runs/detect/val2
 mAP@[IoU=50]: 0.6820
 mAP@[IoU=50:95]: 0.5102

import random
import cv2
import matplotlib.pyplot as plt
from IPython.display import Image, display
import os

# Folder with test images
test_img_dir = '/kaggle/working/carparts_yolo_ready/testset/images'

# Pick 2 random test images
sample_images = random.sample(os.listdir(test_img_dir), 2)

for img_name in sample_images:
    img_path = os.path.join(test_img_dir, img_name)
    results = model.predict(source=img_path, conf=0.25, imgsz=640,
save=True)

    # Get the first result
    r = results[0]

    # Plot predictions directly on the image
    annotated_img = r.plot()  # returns a BGR numpy image with boxes
drawn

    # Convert BGR → RGB for matplotlib
    annotated_img = cv2.cvtColor(annotated_img, cv2.COLOR_BGR2RGB)

    plt.figure(figsize=(8, 8))
    plt.imshow(annotated_img)
    plt.axis('off')
    plt.title(f"Predictions for: {img_name}")
    plt.show()
```

```
image 1/1
/kaggle/working/carparts_yolo_ready/testset/images/car77.jpg: 640x640
1 rear_bumper, 2 doors, 2 hoods, 3 wheels, 2 windows, 2 mirrors, 2
headlights, 1 windshield, 1 license_plate, 1 spoiler, 2 logos, 3
handles, 3 fuel_caps, 10.3ms
Speed: 3.9ms preprocess, 10.3ms inference, 1.4ms postprocess per image
at shape (1, 3, 640, 640)
Results saved to /kaggle/working/runs/detect/predict
```



```
image 1/1 /kaggle/working/carparts_yolo_ready/testset/images/te80.jpg: 480x640 1 hood, 1 wheel, 1 headlight, 2 taillights, 2 grilles, 1 trunk, 2 roofs, 1 fender, 2 side_skirts, 2 windshields, 1 license_plate, 1 spoiler, 1 handle, 1 fuel_cap, 53.0ms
Speed: 2.0ms preprocess, 53.0ms inference, 1.3ms postprocess per image at shape (1, 3, 480, 640)
Results saved to /kaggle/working/runs/detect/predict
```




```python
import numpy as np

# Class-wise mAP values
class_maps = metrics.box.maps
class_names = [
  'front_bumper', 'rear_bumper', 'door', 'hood', 'wheel', 'window',
  'mirror', 'headlight', 'taillight', 'grille', 'trunk', 'roof',
  'fender', 'side_skirt', 'windshield', 'license_plate',
  'spoiler', 'logo', 'handle', 'fuel_cap'
]
```

```
print("📊 Class-wise Average Precision (AP):")
for i, ap in enumerate(class_maps):
    print(f"{class_names[i]:<15}: {ap:.4f}")

# Find the class with the lowest AP
worst_idx = int(np.argmin(class_maps))
print(f"\n🔍 Hardest to detect class: {class_names[worst_idx]} (AP:
{class_maps[worst_idx]:.4f})")
```

```
📊 Class-wise Average Precision (AP):
front_bumper   : 0.5102
rear_bumper    : 0.7354
door           : 0.6208
hood           : 0.5341
wheel          : 0.3685
window         : 0.3432
mirror         : 0.3516
headlight      : 0.7868
taillight      : 0.7937
grille         : 0.7493
trunk          : 0.2996
roof           : 0.3232
fender         : 0.3950
side_skirt     : 0.5541
windshield     : 0.5757
license_plate  : 0.4999
spoiler        : 0.3177
logo           : 0.6152
handle         : 0.3999
fuel_cap       : 0.4302

🔍 Hardest to detect class: trunk (AP: 0.2996)
```

Difficult Class to Localize

From the class-wise results, I noticed that the trunk was the hardest part for the model to detect and localize accurately. In most test images, the trunk either appeared partially or blended with the rear bumper and taillights, which probably confused the model. Since both the color and texture of these parts are quite similar, the model struggled to identify clear boundaries for the trunk. I think another reason could be that the dataset doesn't have enough images showing the rear view of cars clearly. To fix this, I would add more training images that focus on the back side of cars and apply stronger augmentations like different lighting and viewing angles. I also believe that training with a higher-resolution or larger YOLO model could help improve the detection accuracy for the trunk in future experiments.

```
### 🔧 Suggestions to Improve Detection for the Hardest Class

1. **Collect more training samples** for that class (especially from
varied angles or lighting).
```

```
2. **Apply augmentation** – rotation, brightness, and flipping help
generalize small or rare parts.
3. **Use higher-resolution images** (e.g., `imgsz=1024`) so small car
parts are clearer.
4. **Train longer** (100+ epochs) to refine localization.
5. **Use a larger YOLO model** (e.g., `yolo11s.pt` or `yolo11m.pt`)
for improved accuracy.
6. **Fine-tune anchors** or class weights in YOLO to focus more on
rare categories.
```

Conclusion

This project explored car part detection using the YOLOv11 model trained on a dataset of 400 images across 20 part categories. After 50 epochs of training, the model achieved decent overall accuracy, with an **mAP@50 of around 0.6820** and an **mAP@50–95 of 0.5102**. The model was able to correctly identify key parts like headlights, taillights, and grilles in most test samples, though it struggled with smaller or less-distinct components such as the trunk and spoiler. Some predictions showed overlapping boxes and minor misclassifications, which happened because i worked with small, complex datasets. With more diverse training images, stronger augmentation, and a larger YOLO variant, these results could be improved further. Overall, the project provided hands-on experience in preparing datasets, training modern object detection models, and analyzing their performance in a real-world setting.

USE OF AI

While working on this project, I faced a major issue during model training when I kept getting the error: RuntimeError: No valid images found in /kaggle/working/carparts_yolo_ready/trainingset/labels.cache. This error appeared when I tried running the training cell:

results = model.train(data='carparts.yaml', epochs=50, imgsz=640, batch=8)

At first, I was not sure how to fix it, since the dataset looked fine. I used ChatGPT to understand the reason behind this problem and learned that the issue was with incorrectly formatted label files and mismatched class IDs. With its help, I checked my label formats, verified that each file had the correct number of values, and updated my YAML file to match the number of classes in the dataset.

```
```