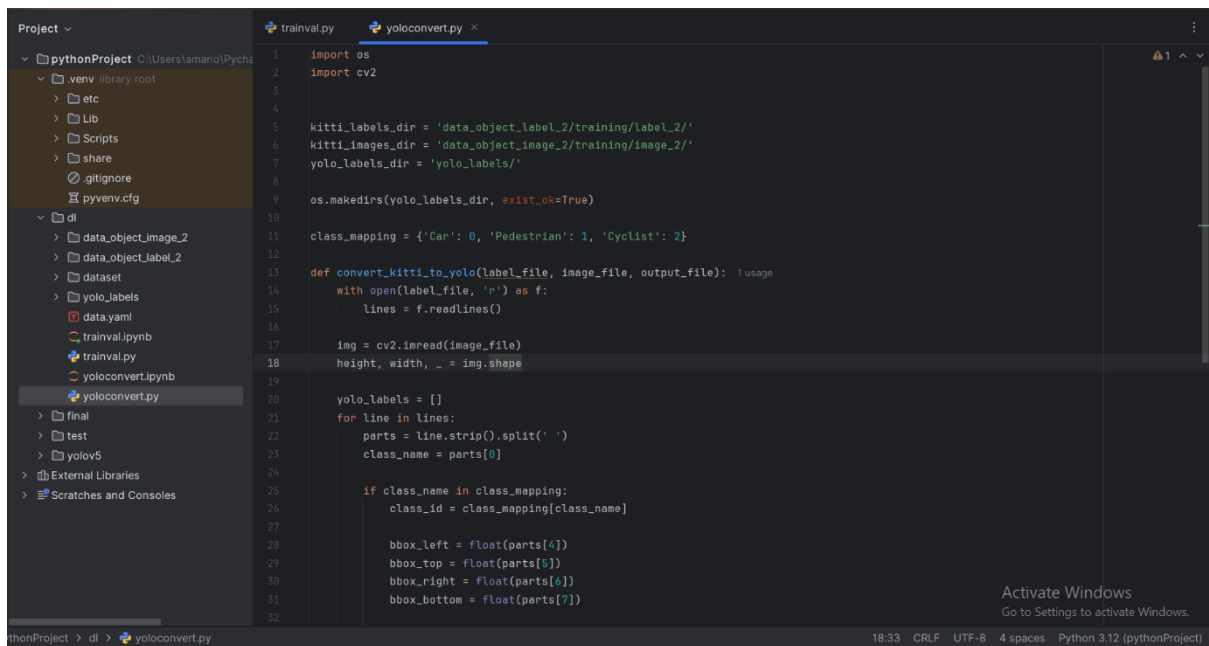
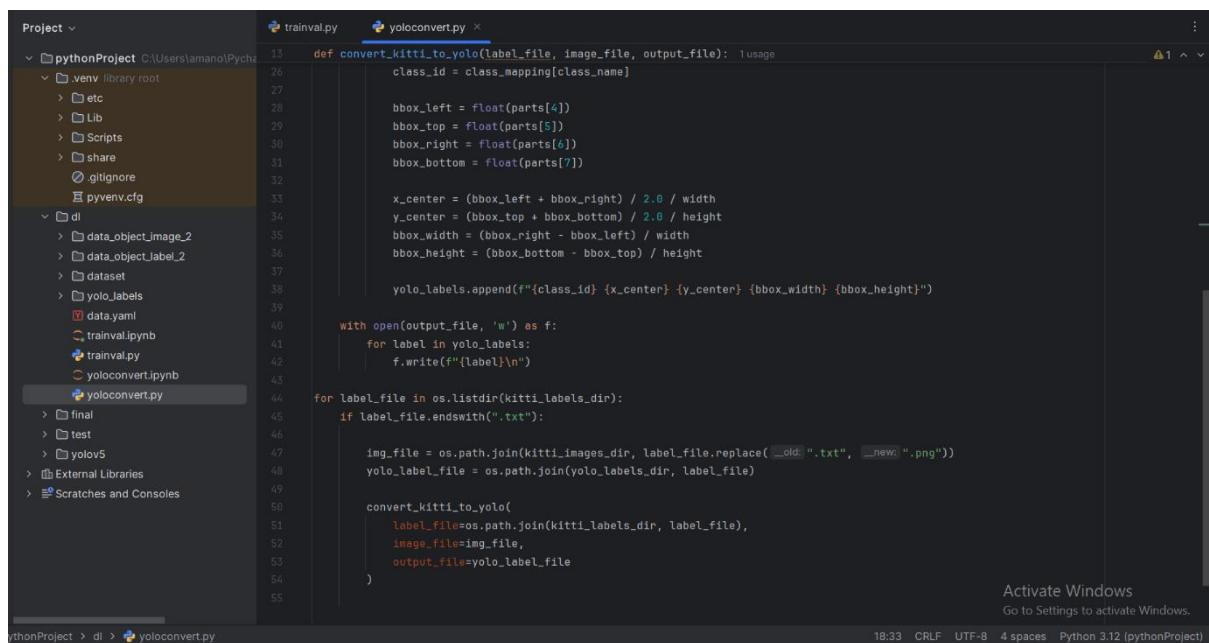


YOLOCONVERTER.PY



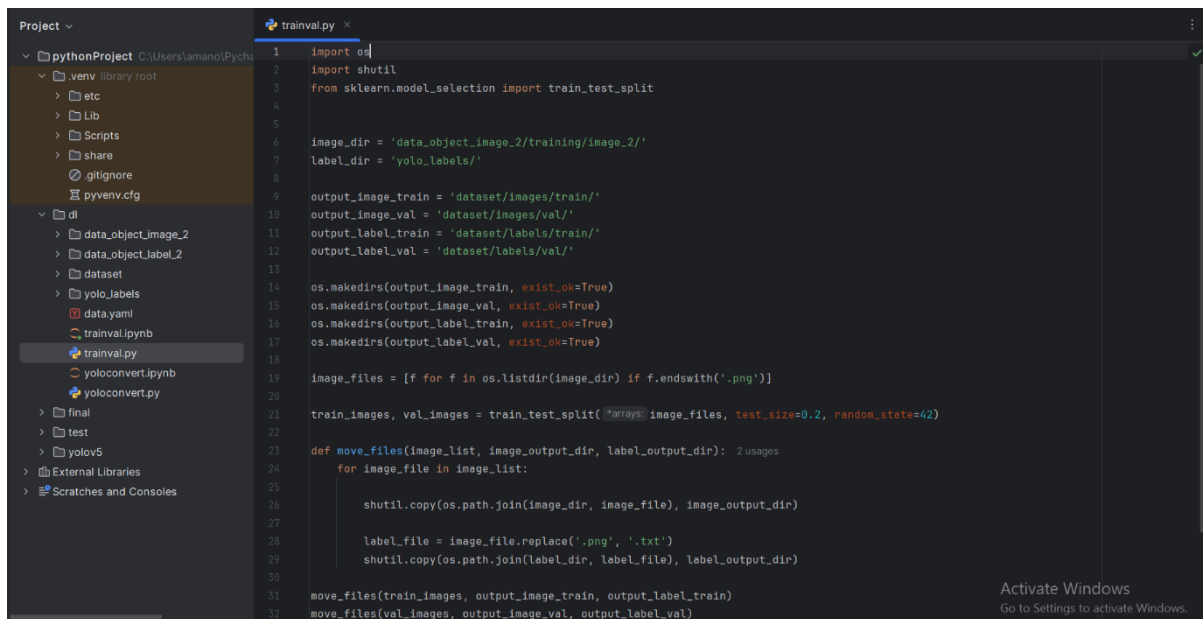
```
1 import os
2 import cv2
3
4
5 kitti_labels_dir = 'data_object_label_2/training/label_2/'
6 kitti_images_dir = 'data_object_image_2/training/image_2/'
7 yolo_labels_dir = 'yolo_labels/'
8
9 os.makedirs(yolo_labels_dir, exist_ok=True)
10
11 class_mapping = {'Car': 0, 'Pedestrian': 1, 'Cyclist': 2}
12
13 def convert_kitti_to_yolo(label_file, image_file, output_file):
14     with open(label_file, 'r') as f:
15         lines = f.readlines()
16
17     img = cv2.imread(image_file)
18     height, width, _ = img.shape
19
20     yolo_labels = []
21     for line in lines:
22         parts = line.strip().split(' ')
23         class_name = parts[0]
24
25         if class_name in class_mapping:
26             class_id = class_mapping[class_name]
27
28             bbox_left = float(parts[4])
29             bbox_top = float(parts[5])
30             bbox_right = float(parts[6])
31             bbox_bottom = float(parts[7])
```



```
32
33     x_center = (bbox_left + bbox_right) / 2.0 / width
34     y_center = (bbox_top + bbox_bottom) / 2.0 / height
35     bbox_width = (bbox_right - bbox_left) / width
36     bbox_height = (bbox_bottom - bbox_top) / height
37
38     yolo_labels.append(f"{class_id} {x_center} {y_center} {bbox_width} {bbox_height}")
39
40     with open(output_file, 'w') as f:
41         for label in yolo_labels:
42             f.write(f"{label}\n")
43
44 for label_file in os.listdir(kitti_labels_dir):
45     if label_file.endswith(".txt"):
46
47         img_file = os.path.join(kitti_images_dir, label_file.replace(_old_ ".txt", _new_ ".png"))
48         yolo_label_file = os.path.join(yolo_labels_dir, label_file)
49
50         convert_kitti_to_yolo(
51             label_file=os.path.join(kitti_labels_dir, label_file),
52             image_file=img_file,
53             output_file=yolo_label_file
54         )
55
```

This code converts KITTI into YOLO format. It processes each label file and its image file, extracting the bounding box information for classes that are mapped, which is Car, Pedestrian, and Cyclist. The bounding boxes are then transformed from KITTI format to YOLO format. The code also creates directories for the YOLO label output. The use of OpenCV ensures the image dimensions are correctly retrieved.

TRAINVAL.PY



```
1 import os
2 import shutil
3 from sklearn.model_selection import train_test_split
4
5
6 image_dir = 'data_object_image_2/training/image_2/'
7 label_dir = 'yolo_labels/'
8
9 output_image_train = 'dataset/images/train/'
10 output_image_val = 'dataset/images/val/'
11 output_label_train = 'dataset/labels/train/'
12 output_label_val = 'dataset/labels/val/'
13
14 os.makedirs(output_image_train, exist_ok=True)
15 os.makedirs(output_image_val, exist_ok=True)
16 os.makedirs(output_label_train, exist_ok=True)
17 os.makedirs(output_label_val, exist_ok=True)
18
19 image_files = [f for f in os.listdir(image_dir) if f.endswith('.png')]
20
21 train_images, val_images = train_test_split(image_files, test_size=0.2, random_state=42)
22
23 def move_files(image_list, image_output_dir, label_output_dir):
24     for image_file in image_list:
25         shutil.copy(os.path.join(image_dir, image_file), image_output_dir)
26
27         label_file = image_file.replace('.png', '.txt')
28         shutil.copy(os.path.join(label_dir, label_file), label_output_dir)
29
30
31 move_files(train_images, output_image_train, output_label_train)
32 move_files(val_images, output_image_val, output_label_val)
```

This code splits a dataset of images and their labels into training and validation sets. It uses `train_test_split` from `sklearn` to divide the dataset with 80% of the images being training one, while other 20% for validation. The `move_files` function efficiently handles copying image and label files from the source directories to the target directories.

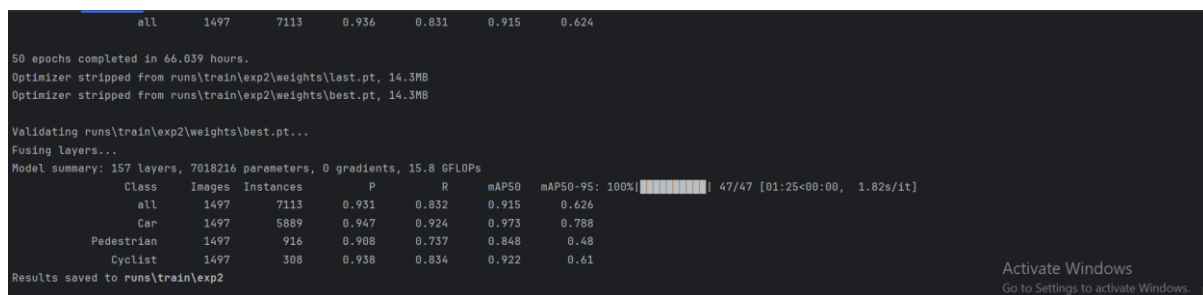
After both codes we run this command and start training:

```
!python C:\Users\amano\PycharmProjects\pythonProject\yolov5\train.py --img 640 --batch 16 --epochs 50 --data
```

```
C:\Users\amano\PycharmProjects\pythonProject\dl\data.yaml --weights
```

```
C:\Users\amano\PycharmProjects\pythonProject\yolov5\yolov5s.pt
```

RESULTS



```
all 1497 7113 0.936 0.831 0.915 0.624

50 epochs completed in 66.039 hours.
Optimizer stripped from runs\train\exp2\weights\last.pt, 14.3MB
Optimizer stripped from runs\train\exp2\weights\best.pt, 14.3MB

Validating runs\train\exp2\weights\best.pt...
Fusing layers...
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs

```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	1497	7113	0.931	0.832	0.915	0.624
Car	1497	5889	0.947	0.924	0.973	0.788
Pedestrian	1497	916	0.908	0.737	0.848	0.48
Cyclist	1497	308	0.938	0.834	0.922	0.61

```
Results saved to runs\train\exp2
```

TEST IMAGES

