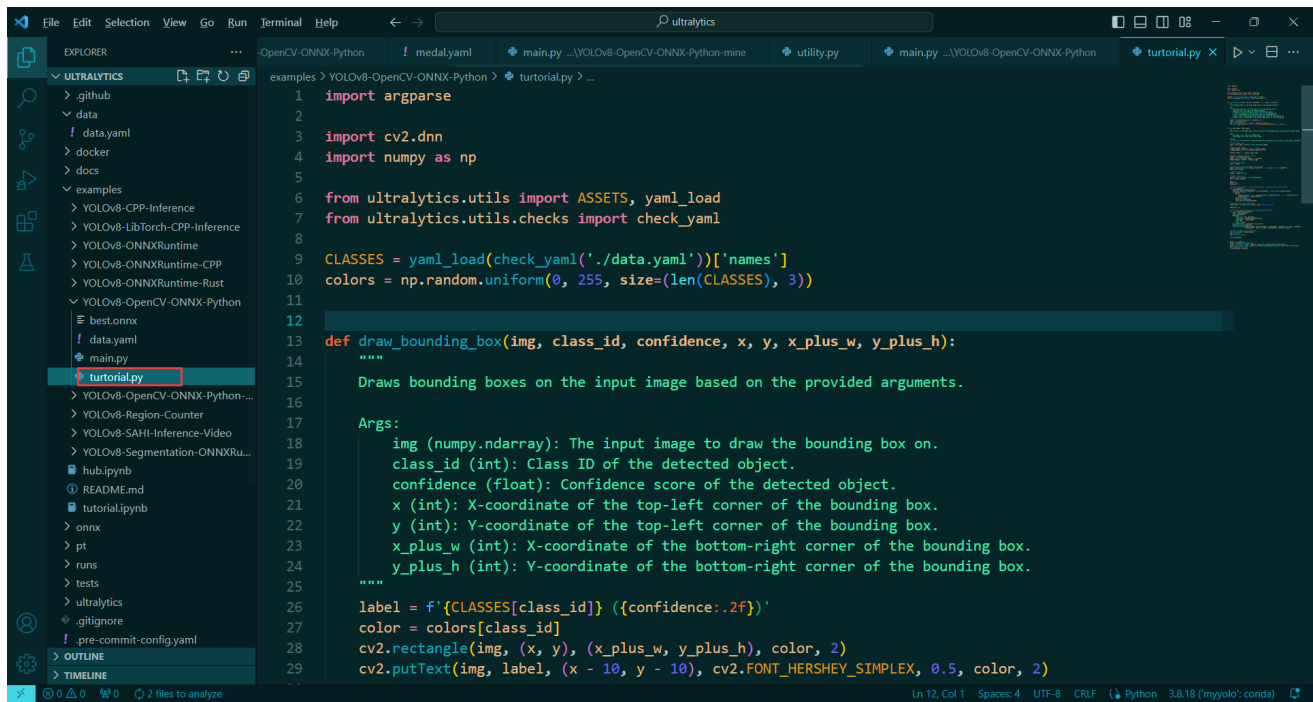
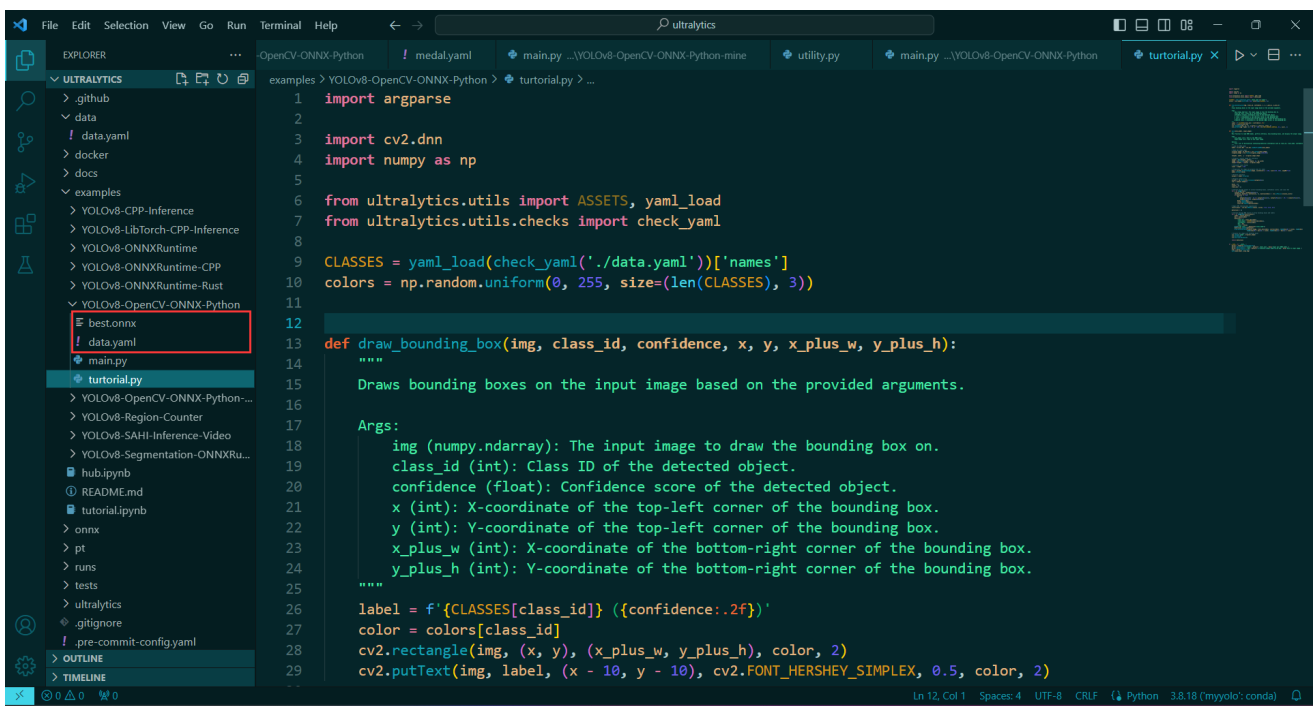


首先,在 ./examples/YOLOv8-OpenCV-ONNX-Python 窗建一个 tutorial.py 文件,将 main.py 的代码复制到 tutorial.py 中



```
1 import argparse
2
3 import cv2.dnn
4 import numpy as np
5
6 from ultralytics.utils import ASSETS, yaml_load
7 from ultralytics.utils.checks import check_yaml
8
9 CLASSES = yaml_load(check_yaml('./data.yaml'))['names']
10 colors = np.random.uniform(0, 255, size=(len(CLASSES), 3))
11
12
13 def draw_bounding_box(img, class_id, confidence, x, y, x_plus_w, y_plus_h):
14     """
15     Draws bounding boxes on the input image based on the provided arguments.
16
17     Args:
18         img (numpy.ndarray): The input image to draw the bounding box on.
19         class_id (int): Class ID of the detected object.
20         confidence (float): Confidence score of the detected object.
21         x (int): X-coordinate of the top-left corner of the bounding box.
22         y (int): Y-coordinate of the top-left corner of the bounding box.
23         x_plus_w (int): X-coordinate of the bottom-right corner of the bounding box.
24         y_plus_h (int): Y-coordinate of the bottom-right corner of the bounding box.
25     """
26     label = f'{CLASSES[class_id]} ({confidence:.2f})'
27     color = colors[class_id]
28     cv2.rectangle(img, (x, y), (x_plus_w, y_plus_h), color, 2)
29     cv2.putText(img, label, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
```

然后将你训练自己数据集得到的 data.yaml 和 best.onnx 预训练权重复制到该目录下



```
1 import argparse
2
3 import cv2.dnn
4 import numpy as np
5
6 from ultralytics.utils import ASSETS, yaml_load
7 from ultralytics.utils.checks import check_yaml
8
9 CLASSES = yaml_load(check_yaml('./data.yaml'))['names']
10 colors = np.random.uniform(0, 255, size=(len(CLASSES), 3))
11
12
13 def draw_bounding_box(img, class_id, confidence, x, y, x_plus_w, y_plus_h):
14     """
15     Draws bounding boxes on the input image based on the provided arguments.
16
17     Args:
18         img (numpy.ndarray): The input image to draw the bounding box on.
19         class_id (int): Class ID of the detected object.
20         confidence (float): Confidence score of the detected object.
21         x (int): X-coordinate of the top-left corner of the bounding box.
22         y (int): Y-coordinate of the top-left corner of the bounding box.
23         x_plus_w (int): X-coordinate of the bottom-right corner of the bounding box.
24         y_plus_h (int): Y-coordinate of the bottom-right corner of the bounding box.
25     """
26     label = f'{CLASSES[class_id]} ({confidence:.2f})'
27     color = colors[class_id]
28     cv2.rectangle(img, (x, y), (x_plus_w, y_plus_h), color, 2)
29     cv2.putText(img, label, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
```

改动以下代码:

- 第一处:

```
1 import argparse
2
3 import cv2.dnn
4 import numpy as np
5
6 from ultralytics.utils import ASSETS, yaml_load
7 from ultralytics.utils.checks import check_yaml
8
9 CLASSES = yaml_load(check_yaml('./data.yaml'))['names']
10 colors = np.random.uniform(0, 255, size=(len(CLASSES), 3))
11
12
13 def draw_bounding_box(img, class_id, confidence, x, y, x_plus_w, y_plus_h):
14     """
15     Draws bounding boxes on the input image based on the provided arguments.
16
17     Args:
18         img (numpy.ndarray): The input image to draw the bounding box on.
19         class_id (int): Class ID of the detected object.
20         confidence (float): Confidence score of the detected object.
21         x (int): X-coordinate of the top-left corner of the bounding box.
22         y (int): Y-coordinate of the top-left corner of the bounding box.
23         x_plus_w (int): X-coordinate of the bottom-right corner of the bounding box.
24         y_plus_h (int): Y-coordinate of the bottom-right corner of the bounding box.
25     """
26     label = f'{CLASSES[class_id]} ({confidence:.2f})'
27     color = colors[class_id]
28     cv2.rectangle(img, (x, y), (x_plus_w, y_plus_h), color, 2)
29     cv2.putText(img, label, (x - 10, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 2)
```

把 `check\_yaml` 内的路径换为当前目录下的 `data.yaml` 文件.

- 第二处

```
30
31
32 def main(onnx_model, input_image):
33     """
34     Main function to load ONNX model, perform inference, draw bounding boxes, and display the output image.
35
36     Args:
37         onnx_model (str): Path to the ONNX model.
38         input_image (str): Path to the input image.
39
40     Returns:
41         list: List of dictionaries containing detection information such as class_id, class_name, confidence, etc.
42     """
43     # Load the ONNX model
44     model = cv2.dnn.Net = cv2.dnn.readNetFromONNX(onnx_model)
45
46     # Read the input image
47     original_image = np.ndarray = cv2.imread(input_image)
48     [height, width, _] = original_image.shape
49
50     # Prepare a square image for inference
51     length = max((height, width))
52     image = np.zeros((length, length, 3), np.uint8)
53     image[0:height, 0:width] = original_image
54
55     # Calculate scale factor
56     scale = length / 320
57
58     # Preprocess the image and prepare blob for model
```

将 main 函数中的 scale 的分母改为训练时的 imgsiz .

- 第三处

```
> .github
  > data
    ! data.yaml
  > docker
  > docs
  > examples
    > YOLOv8-CPP-Inference
    > YOLOv8-LibTorch-CPP-Inference
    > YOLOv8-ONNXRuntime
    > YOLOv8-ONNXRuntime-CPP
    > YOLOv8-ONNXRuntime-Rust
  > YOLOv8-OpenCV-ONNX-Python
    ≡ best.onnx
    ! data.yaml
    + main.py
    + tutorial.py
  > YOLOv8-OpenCV-ONNX-Python-...
  > YOLOv8-Region-Counter
  > YOLOv8-SAHI-Inference-Video
  > YOLOv8-Segmentation-ONNXRu...
  hub.ipynb
  ④ README.md
  hub.ipynb
  > onnx
  > pt
  > runs
  > tests

46 # Read the input image
47 original_image: np.ndarray = cv2.imread(input_image)
48 [height, width, _] = original_image.shape
49
50 # Prepare a square image for inference
51 length = max((height, width))
52 image = np.zeros((length, length, 3), np.uint8)
53 image[0:height, 0:width] = original_image
54
55 # Calculate scale factor
56 scale = length / 320
57
58 # Preprocess the image and prepare blob for model
59 blob = cv2.dnn.blobFromImage(image, scalefactor=1 / 255, size=(320, 320), swapRB=True)
60 model.setInput(blob)
61
62 # Perform inference
63 outputs = model.forward()
64
65 # Prepare output array
66 outputs = np.array([cv2.transpose(outputs[0])])
67 rows = outputs.shape[1]
68
69 boxes = []
70 scores = []
```

把第二处下面的 `size` 改成训练时的 `imgsz` .

上述三处地方改完后就可以运行代码，在当前文件夹打开 `cmd` 命令行,或者打开命令行 `cmd` 然后再转移到当前目录下

```
## 其中虚拟环境
activate myyolo
```

运行:

```
python main.py --model ./best.onnx --img 图像路径
```

即可.