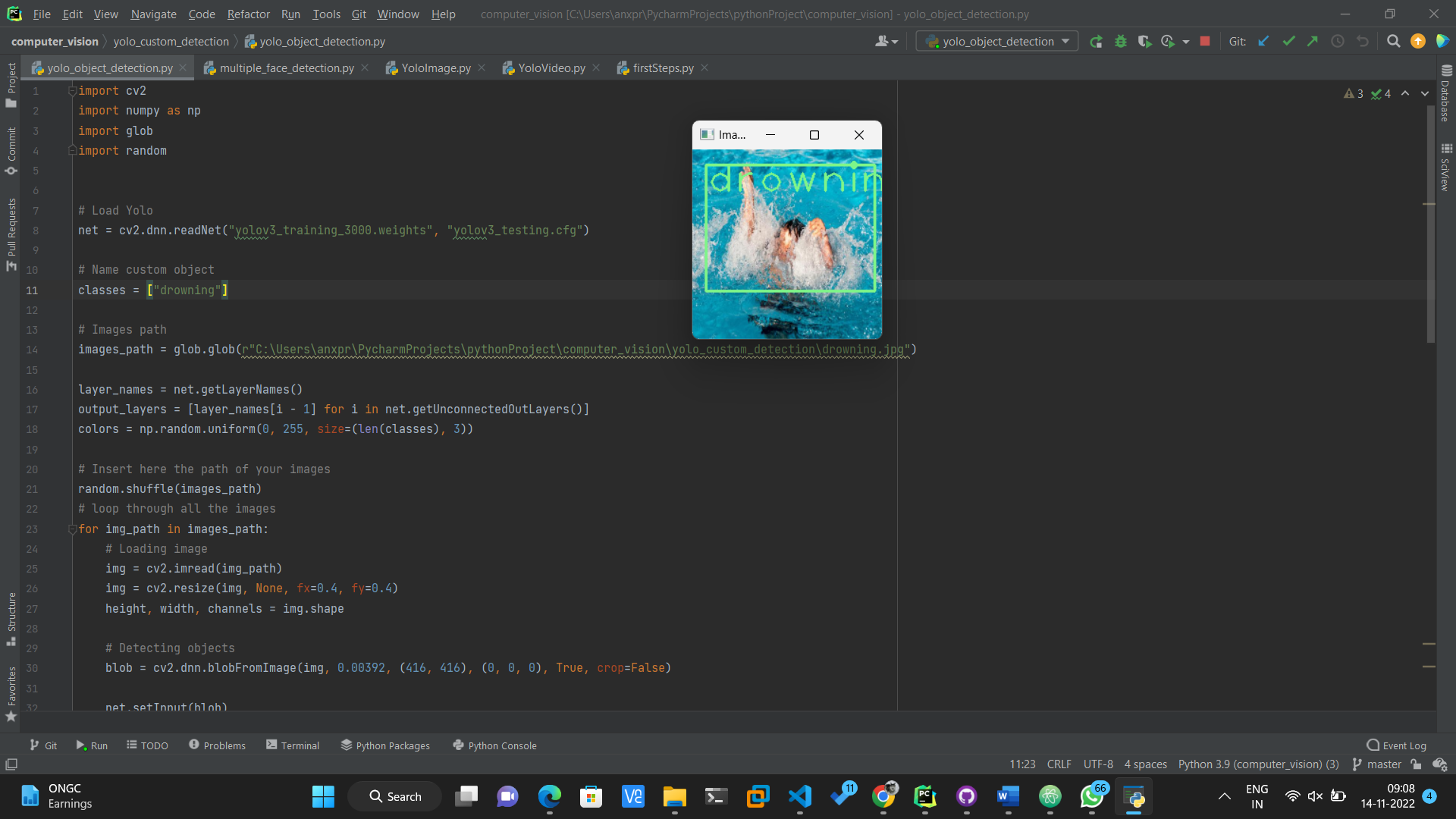
**Testing the Model**

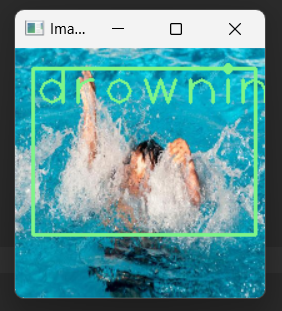
Obtained weights file “[yolov3\_training\_3000.weights](https://drive.google.com/file/d/1-ECcQYbQQvyVEwvT54T0sdTdu9R3AZkM/view)” from training is used for testing detection on a sample image of a drowning subject

*Code for Testing:*

|  |
| --- |
| import cv2  import numpy as np  import glob  import random  # Load Yolo  net = cv2.dnn.readNet("yolov3\_training\_3000.weights", "yolov3\_testing.cfg")  # Name custom object  classes = ["drowning"]  # Images path  images\_path = glob.glob(r"C:\Users\anxpr\PycharmProjects\pythonProject\computer\_vision\yolo\_custom\_detection\drowning.jpg")  layer\_names = net.getLayerNames()  output\_layers = [layer\_names[i - 1] for i in net.getUnconnectedOutLayers()]  colors = np.random.uniform(0, 255, size=(len(classes), 3))  # Insert here the path of your images  random.shuffle(images\_path)  # loop through all the images  for img\_path in images\_path:  # Loading image  img = cv2.imread(img\_path)  img = cv2.resize(img, None, fx=0.4, fy=0.4)  height, width, channels = img.shape  # Detecting objects  blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)  net.setInput(blob)  outs = net.forward(output\_layers)  # Showing informations on the screen  class\_ids = []  confidences = []  boxes = []  for out in outs:  for detection in out:  scores = detection[5:]  class\_id = np.argmax(scores)  confidence = scores[class\_id]  if confidence > 0.3:  # Object detected  print(class\_id)  center\_x = int(detection[0] \* width)  center\_y = int(detection[1] \* height)  w = int(detection[2] \* width)  h = int(detection[3] \* height)  # Rectangle coordinates  x = int(center\_x - w / 2)  y = int(center\_y - h / 2)  boxes.append([x, y, w, h])  confidences.append(float(confidence))  class\_ids.append(class\_id)  indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)  print(indexes)  font = cv2.FONT\_HERSHEY\_PLAIN  for i in range(len(boxes)):  if i in indexes:  x, y, w, h = boxes[i]  label = str(classes[class\_ids[i]])  color = colors[class\_ids[i]]  cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)  cv2.putText(img, label, (x, y + 30), font, 3, color, 2)  cv2.imshow("Image", img)  key = cv2.waitKey(0)  cv2.destroyAllWindows() |



Output:

 The model is successfully able to detect potential drowning subject from the sample image. The same could be used to detect potential subjects off of a video feed from a camera mounted over a swimming pool.