In [ ]:
```
#This code uses the ImageAI detection module and the OpenCV library to create
an object detection pipeline that uses the YOLOv3 model to make predictions on
real-world images captured by a webcam.

# set the path to the YOLO model file on your computer
modelpath = r"C:\Users\reonq\yolo.h5"


# import the ImageAI detection module and create a YOLO object detection insta
nce
from imageai import Detection
yolo = Detection.ObjectDetection()
yolo.setModelTypeAsYOLOv3()

# load the YOLO model from the specified path
yolo.setModelPath(modelpath)
yolo.loadModel()

# import the OpenCV library for accessing camera and displaying images
import cv2

# create a video capture object for accessing the camera feed
cam = cv2.VideoCapture(0) # 0=front-cam, 1=back-cam

# set the frame dimensions for the video capture object
cam.set(cv2.CAP_PROP_FRAME_WIDTH, 1300)
cam.set(cv2.CAP_PROP_FRAME_HEIGHT, 1500)

# enter the loop for making predictions on the video stream
while True:
    # read frames from the video capture object
    ret, img = cam.read()

    # use the YOLO object detection model to make predictions on the captured
image
    img, preds = yolo.detectCustomObjectsFromImage(input_image=img,
                      custom_objects=None, input_type="array",
                      output_type="array",
                      minimum_percentage_probability=70,
                      display_percentage_probability=False,
                      display_object_name=True)

    # display the predictions on the captured image
    cv2.imshow("", img)

    # wait for the user to press "q" or "Esc" to quit
    if (cv2.waitKey(1) & 0xFF == ord("q")) or (cv2.waitKey(1)==27):
        break

# release the camera and close all windows
cam.release()
cv2.destroyAllWindows()
```

In [ ]:
```

```