def cameraDetect(token, device, variable, sample\_time=5):

    cap = cv2.VideoCapture(0)

    init = time.time()

    # Allowed sample time for Ubidots is 1 dot/second

    if sample\_time < 1:         sample\_time = 1     while(True):         # Capture frame-by-frame         ret, frame = cap.read()         frame = imutils.resize(frame, width=min(400, frame.shape[1]))         result = detector(frame.copy())         # shows the result         for (xA, yA, xB, yB) in result:             cv2.rectangle(frame, (xA, yA), (xB, yB), (0, 255, 0), 2)         cv2.imshow('frame', frame)         # Sends results         if time.time() - init >= sample\_time:

            print("[INFO] Sending actual frame results")

            # Converts the image to base 64 and adds it to the context

            b64 = convert\_to\_base64(frame)

            context = {"image": b64}

            sendToUbidots(token, device, variable,

                          len(result), context=context)

            init = time.time()

        if cv2.waitKey(1) & 0xFF == ord('q'):

            break

    # When everything done, release the capture

    cap.release()

    cv2.destroyAllWindows()

def convert\_to\_base64(image):

    image = imutils.resize(image, width=400)

    img\_str = cv2.imencode('.png', image)[1].tostring()

    b64 = base64.b64encode(img\_str)

    return b64.decode('utf-8')