

```

# vid = cv2.VideoCapture(0) # detect from webcam
vid = cv2.VideoCapture('https://192.168.253.163:8080/video') # detect from
mobile feed

# by default VideoCapture returns float instead of int
width = int(vid.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(vid.get(cv2.CAP_PROP_FRAME_HEIGHT))
fps = int(vid.get(cv2.CAP_PROP_FPS))
# codec = cv2.VideoWriter_fourcc(*'XVID')
# # output_path must be .mp4
# out = cv2.VideoWriter(output_path, codec, fps, (width, height))

NUM_CLASS = read_class_names(CLASSES)
key_list = list(NUM_CLASS.keys())
val_list = list(NUM_CLASS.values())
while True:
    _, frame = vid.read()

    try:
        original_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        original_frame = cv2.cvtColor(original_frame,
cv2.COLOR_BGR2RGB)
    except:
        break

    image_data = image_preprocess(np.copy(original_frame), [
        input_size, input_size])
    #image_data = tf.expand_dims(image_data, 0)
    image_data = image_data[np.newaxis, ...].astype(np.float32)

    t1 = time.time()
    if YOLO_FRAMEWORK == "tf":
        pred_bbox = Yolo.predict(image_data)
    elif YOLO_FRAMEWORK == "trt":
        batched_input = tf.constant(image_data)
        result = Yolo(batched_input)
        pred_bbox = []
        for key, value in result.items():
            value = value.numpy()
            pred_bbox.append(value)

    #t1 = time.time()
    #pred_bbox = Yolo.predict(image_data)

```