教师AI基础考核试题

请将代码按照次序，分别命名为1.py-4.py，提交。

提交地址： <http://site01.openhydra.net:30012/evt/uploadcode>

## 一、基础算法类

1. 寻找数字。

“9474”是一个有趣的数字。它的每个位上的数字的 4 次幂之和等于它本身（9⁴ + 4⁴ + 7⁴ + 4⁴ = 9474）。请编写一个python程序，找出1000到9999之间所有类似“9474”的数字，并输出（注意，不包含“9474”）。

2. 没有判断的猜拳游戏。

很多人都跟计算机玩过猜拳游戏。猜拳游戏原理一般是用数字代表“石头”“剪刀”“布”，如0、1、2，计算机随机生成一个0-2之间的数字，然后和用户选择的数字比较，最后输出结果。这类代码往往会用到分支结构，即“if”。能否实现代码中不出现“if”，编写出一个猜拳游戏？

注意，出现“if”就表示挑战失败了。

## 二、数据解析类

3. 计算AI模型的推理结果

准确率是衡量一个模型效果好坏的重要指标。函数cal\_acc能够对真实标签和推理结果进行比较，计算出准确率。

有一个三分类（三个标签）问题，参数中的“infer\_y”是具有4个样本的推理结果，比如“[[0.1,0.2,0.7],[0.4,0.1,0.5],[0.3,0.4,0.3],[0.6,0.2,0.2]]”，其中“[0.1,0.2,0.7]”是第一个样本的推理结果，以此类推。参数中的“true\_y”是样本的真实标签(Ground Truth)，比如“[2,2,1,2]”。前三个预测正确（推理结果与真实标签一致，如[0.1,0.2,0.7]表示第2项的概率最大），最后一个错误（推理结果与真实标签不一致），因此准确率为3/4，即返回值为75%。

请完善代码中的cal\_acc函数，能输出75.00%，小数点保留2位。

def cal\_acc(infer\_y,true\_y):

# 请补充代码，完善这个函数

result = [[0.1,0.2,0.7],[0.4,0.1,0.5],[0.3,0.4,0.3],[0.6,0.2,0.2]]

tag = [2,2,1,2]

result = cal\_acc(result,tag)

3. 目标检测

COCO（Common Objects in Context）是一个用于目标检测和图像分割任务的广泛使用的数据集和评估基准。它是计算机视觉领域中最重要的数据集之一，在XEduHub中内置的coco模型能够检测出80类coco数据集中的物体，任务名称为“det\_coco”。

“det\_coco”返回的是图中的目标名称和坐标。进行推理的图像、推理代码和输出信息如下所示。



进行推理的图像（img.jpg）

推理代码（给出）

from XEdu.hub import Workflow as wf # 导入库

body\_det = wf(task='det\_body') # 实例化模型

img\_path = 'img.jpg' # 指定进行推理的图片路径

body\_det.inference(data=img\_path) # 进行推理

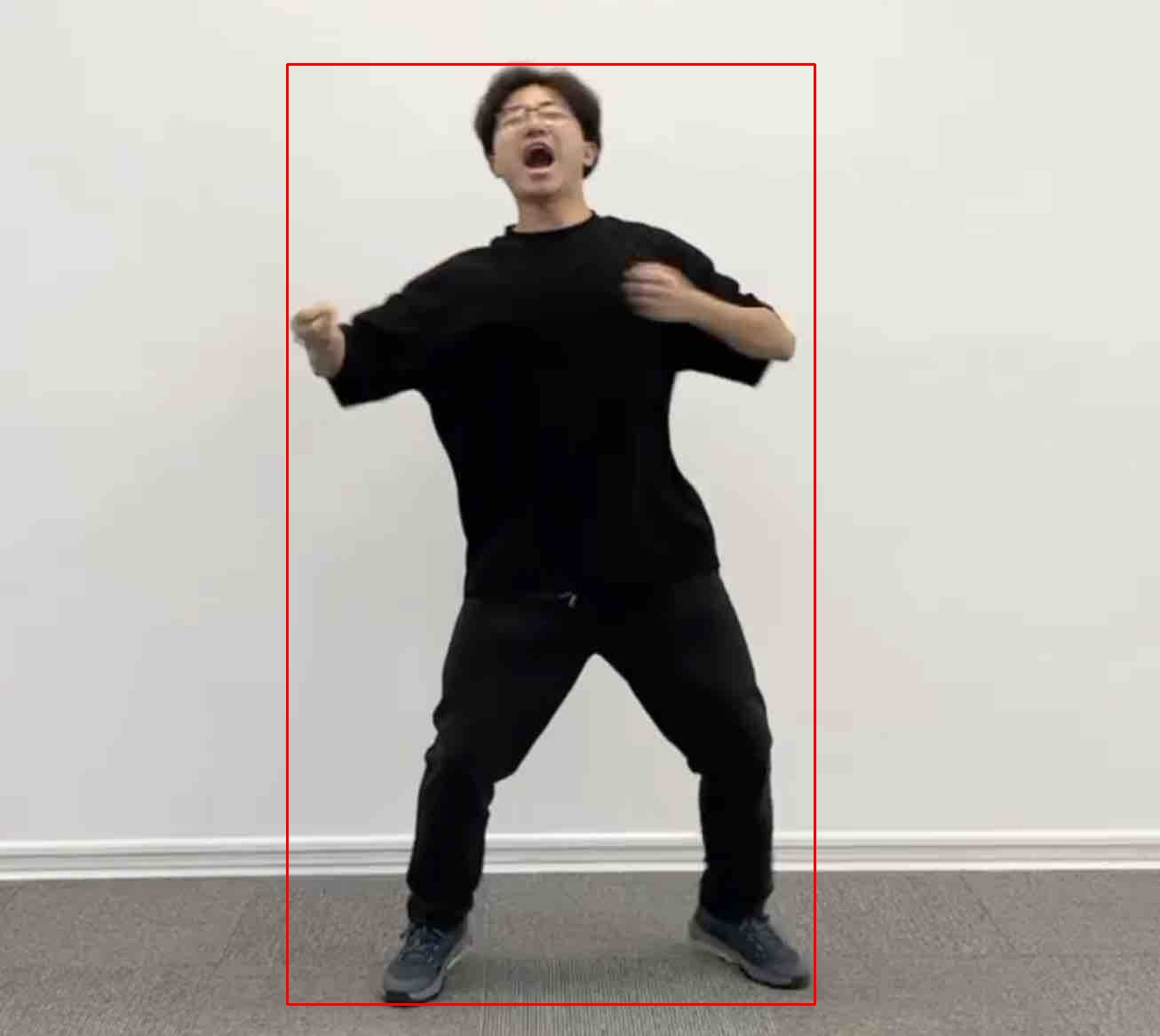
result = body\_det.format\_output(lang='zh') # 推理结果格式化

print(result)

推理结果（给出）

{'检测框': [[323.2777170453753, 72.95395088195801, 917.5945354189192, 1130.7357228142876]], '分数': [0.8851305]}

请在推理代码的基础上增加功能，实现解析模型输出的数据，提取坐标数据后用红色矩形标注，效果如图所示。注意：只能使用OpenCV库或者Pillow库。



目标图

请在这里输入完整的代码。

from XEdu.hub import Workflow as wf # 导入库

body\_det = wf(task='det\_body') # 实例化模型

img\_path = 'img.jpg' # 指定进行推理的图片路径

body\_det.inference(data=img\_path) # 进行推理

result = body\_det.format\_output(lang='zh') # 推理结果格式化

print(result)

# 请将代码补充在下方