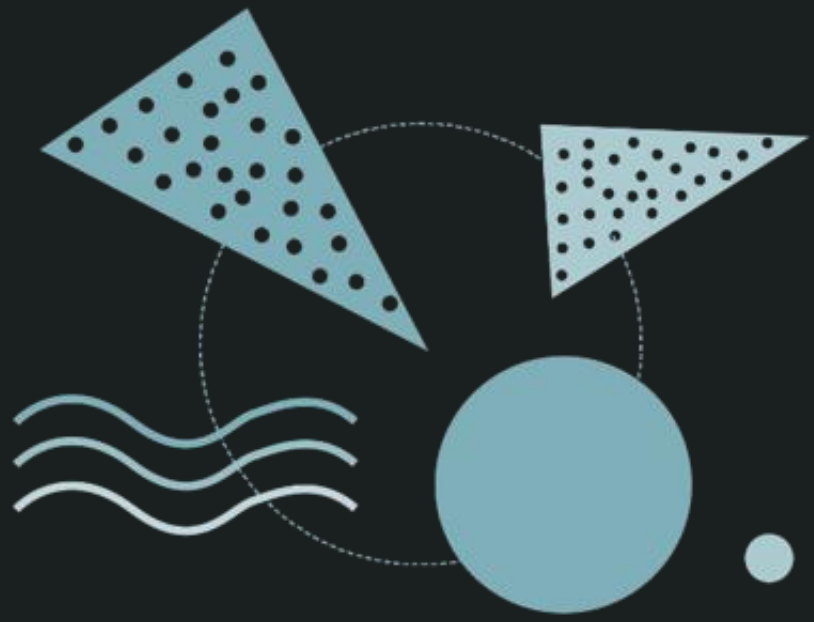
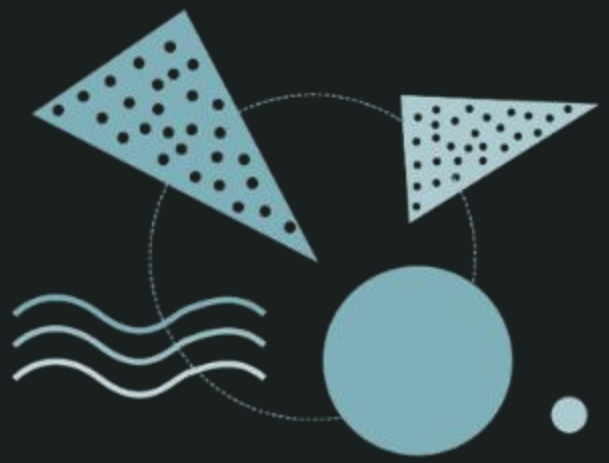
人体姿态估计

第七组组员：王以恒 袁忻泽 吴悠 陈云龙 董子萱

目录





01

Part One

视频导入模块

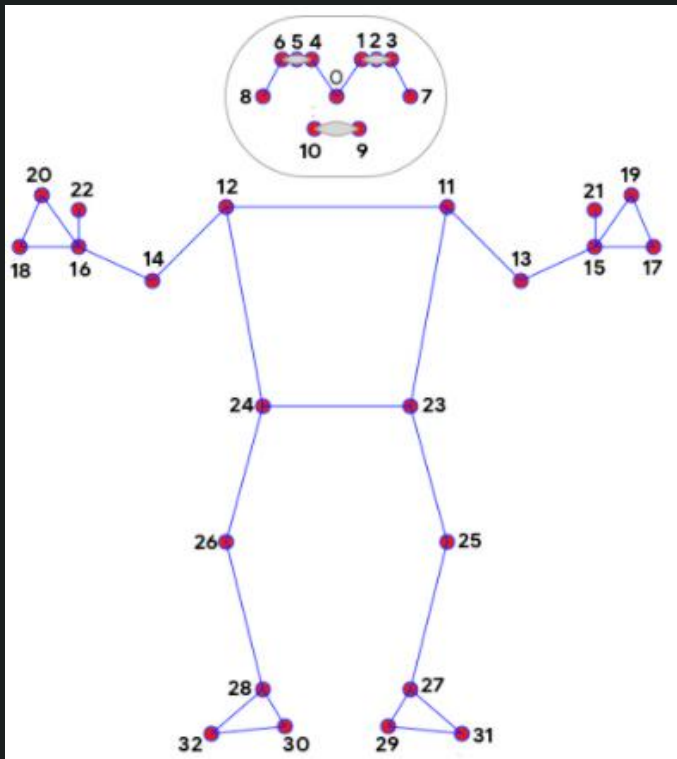
将已有的视频通过opencv导入，并用mediapipe处理

视频导入模块

这个模块中，我们选择使用opencv把找到的已有的视频进行导入。导入之后运用mediapipe对视频进行分析处理，识别出视频中的33个人体关键点（部位/关节），再将关键点连线，更直观的展现出人体姿态

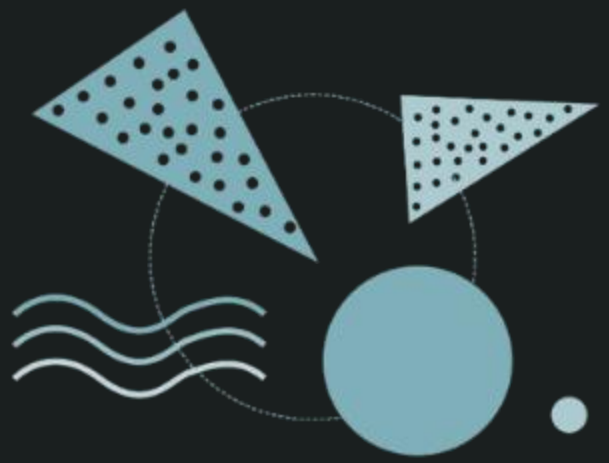
```
#进入循环，读取视频
while True:
    success, img = cap.read()
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    results = pose.process(imgRGB)
    #将姿态检测的点打印出来，显示在输出栏上
    #print(results.pose_landmarks)
    #将姿态检测的点连成线
    if results.pose_landmarks:
        lmlist = []
        mpDraw.draw_landmarks(img, results.pose_landmarks, mpPose.POSE_CONNECTIONS)
        for id, lm in enumerate(results.pose_landmarks.landmark):
            h, w, c = img.shape
            # print(id, lm)
            cx, cy = int(lm.x * w), int(lm.y * h)
            lmlist.append([id, cx, cy])
            cv2.circle(img, (cx, cy), 15, (255, 0, 0), cv2.FILLED)
    print(lmlist[9][2], lmlist[9][2]+100)
```

视频导入模块



0. nose
1. left_eye_inner
2. left_eye
3. left_eye_outer
4. right_eye_inner
5. right_eye
6. right_eye_outer
7. left_ear
8. right_ear
9. mouth_left
10. mouth_right
11. left_shoulder
12. right_shoulder
13. left_elbow
14. right_elbow
15. left_wrist
16. right_wrist
17. left_pinky
18. right_pinky
19. left_index
20. right_index
21. left_thumb
22. right_thumb
23. left_hip
24. right_hip
25. left_knee
26. right_knee
27. left_ankle
28. right_ankle
29. left_heel
30. right_heel
31. left_foot_index
32. right_foot_index

视频导入模块



02

Part One

分析计数模块

分析关键点的位置，完成计数

分析计数模块

大致思路

这个模块中，我们对已经识别出来的人体关键点进行分析，通过不同关键点之间的位置，来判断运动人做引体向上运动时是否标准，并对其进行计数

具体实现

规则一：下颌必须过杠

嘴的位置最高，其次是肩，最后是肘，代表其下颌过杠

规则二：在做完一次后，第二次做之前两个手臂保持近乎水平

下落回到初始位置时，肘最高，其次是嘴，最后是肩

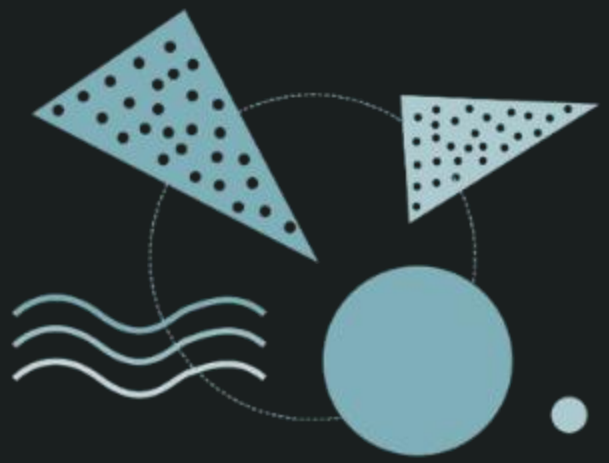
分析计数模块

```
ptime = ctime
# 带入规则函数，进行按照规定计数
log1 = re.rule1(lmlist)
log2 = re.rule2(lmlist)
count = re.Count(log1, log2)

# 将计数结果实时反馈至屏幕上
cv2.putText(img, "FPS:", (0, 50), cv2.FONT_HERSHEY_PLAIN, 2, (0, 0, 255), 3)
cv2.putText(img, str(int(fps)), (100, 50), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0, 25
# cv2.putText(img, "The number is :", (0, 100), cv2.FONT_HERSHEY_PLAIN, 2.5, (2
cv2.putText(img, str(int(count)), (450, 150), cv2.FONT_HERSHEY_PLAIN, 3, (0, 0,
#由于cv2.putText在输入中文后显示了？？？，不能输出中文，故在Chinese.py文件中封装函数，在对其
img = cv2AddChineseText(img, "引体向上数量为：", (0, 100), (255, 0, 0), 60)
cv2.imshow("Image", img)
cv2.waitKey(1)
# 点击窗口x即可关闭窗口，结束播放视频
```

```
# 规则一：下颌必须过杠
def rule1(self, lmlist):
#嘴的位置最高，其次是肩，最后是肘，代表其下颌过杠
    if lmlist[9][2] < lmlist[19][2] and lmlist[12][2] < lmlist[14][2]:
        self.tracking = 1
        self.logical1 = 1
        # time.sleep(1)
    # if lmlist[9][2] > lmlist[19][2] and lmlist[9][2] > lmlist[20][2]:
    #     self.tracking = 1
    return self.logical1
# 规则二：在做完一次后，第二次做之前两个手臂保持近乎水平
def rule2(self, lmlist, height=0):
# 下落回到初始位置时，肘最高，其次是嘴，最后是肩
# 参数height是代表下落时，第二次做引体向上的嘴、肘、肩的初始位置，默认为0
    if lmlist[14][2]+height < lmlist[9][2] and self.tracking:
        self.logical2 = 1
        # time.sleep(5)
    return self.logical2
```

分析计数模块



03

Part One

预处理模块

对视频进行处理，使其能逐帧提取

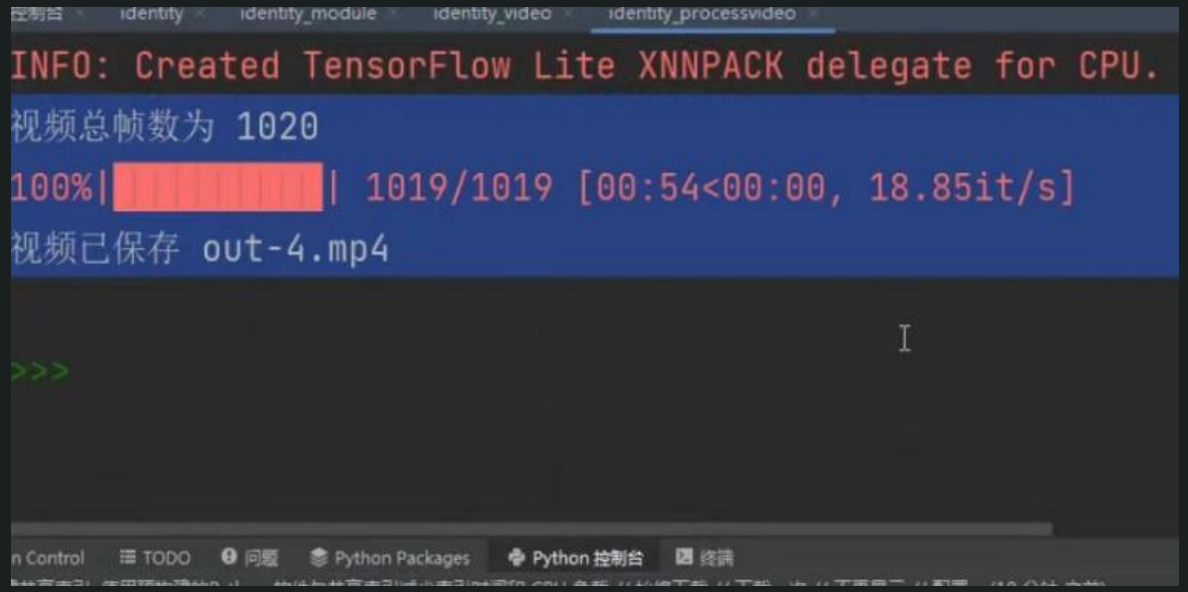
视频预处理模块

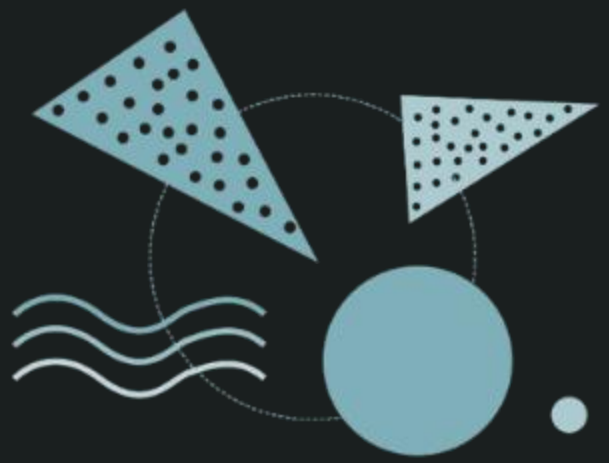
大致思路

这个模块中，我们对已有的视频进行处理，并生成一个新视频保存，在新视频中有不同颜色的关键点，也可从中逐帧提取图片

```
# VideoCaputre对象是否成功打开
while cap.isOpened():
    success, frame = cap.read()
    frame_count += 1
    if not success:
        break
cap.release()
print('视频总帧数为', frame_count)
cap = cv2.VideoCapture(input_path)
frame_size = (cap.get(cv2.CAP_PROP_FRAME_WIDTH), cap.get(
# 视频解码，解码器选成mp4格式
fourcc = cv2.VideoWriter_fourcc(*'mp4v')
fps = cap.get(cv2.CAP_PROP_FPS)
```

视频预处理模块


identity
identity_module
identity_video
identity_processvideo
INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
视频总帧数为 1020
100%| | 1019/1019 [00:54<00:00, 18.85it/s]
视频已保存 out-4.mp4
>>>
n Control
TODO
问题
Python Packages
Python 控制台
终端



04

Part One

提取图片模块

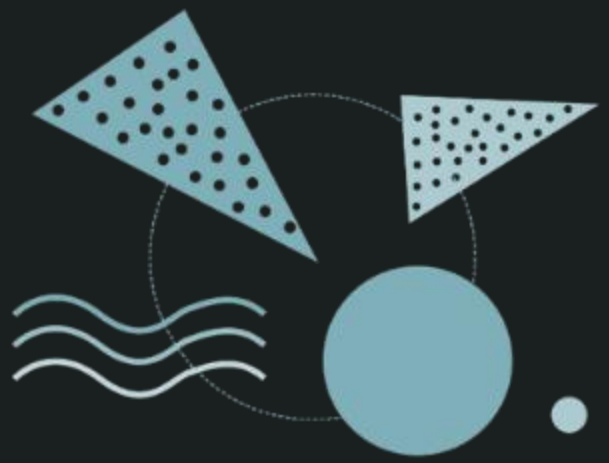
从已生成视频中提取需要的图片

提取图片模块

大致思路

这个模块中，我们从已经生成的视频的视频中提取出图片并进行储存，方便后续分析

```
file_name = glob.glob("D:/课程学习/计算机视觉/human identity/pic/
# 检测文件夹里面是否为空，如为空，则file_name = []，  进入if判断
if not file_name:
    print("File doesn't exit")
else:
    for file in file_name:
        os.remove(file)
    print("所有文件删除成功")
num = 1
video = cv2.VideoCapture(video_path)
fps = video.get(cv2.CAP_PROP_FPS)#帧率
frames = video.get(cv2.CAP_PROP_FRAME_COUNT)#总的帧数
```



05

Part One

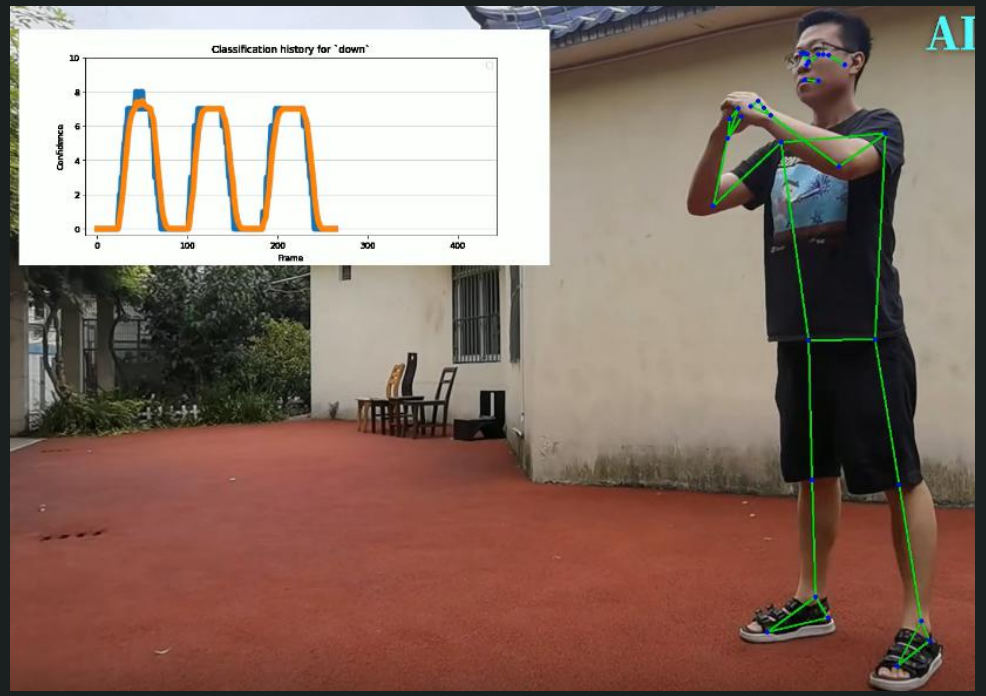
可视化模块

根据提取出的图片进行分析整合，希望能做出阈值图

图片处理模块

大致思路

这个模块是我们接下来的目标，正在努力完成。其主要希望通过对每一帧图像进行分析整合处理，最后总结出运动人的运动轨迹阈值图。



YHANKS