

# 基于Open Cv下的人体姿态估计 (引体向上计数)

第七组 王以恒 吴悠 袁忻泽 董子萱 陈云龙

LET'S START

# 目录

CONTENTS

01

研究背景介绍

—

02

环境搭建及原理分析

—

03

初步源码展示及效果

—

04

存在问题及下步计划

—



# 研究背景介绍

Introduction of research background

PART.01

## 研究背景介绍

人体姿态估计的目标是从给定的图像或视频中确定人的身体关键点（部位/关节）的位置或空间位置，如下图所示，因此，姿态估计使用基于图像的观察获得关节人体的姿态，关节人体由关节和刚性部分组成。



FIGURE 1. The estimated pose of each individual in a given image.

其应用场景非常丰富，在体育健身、医疗、美容、服装、自动驾驶、AR等领域有着非常广泛的应用

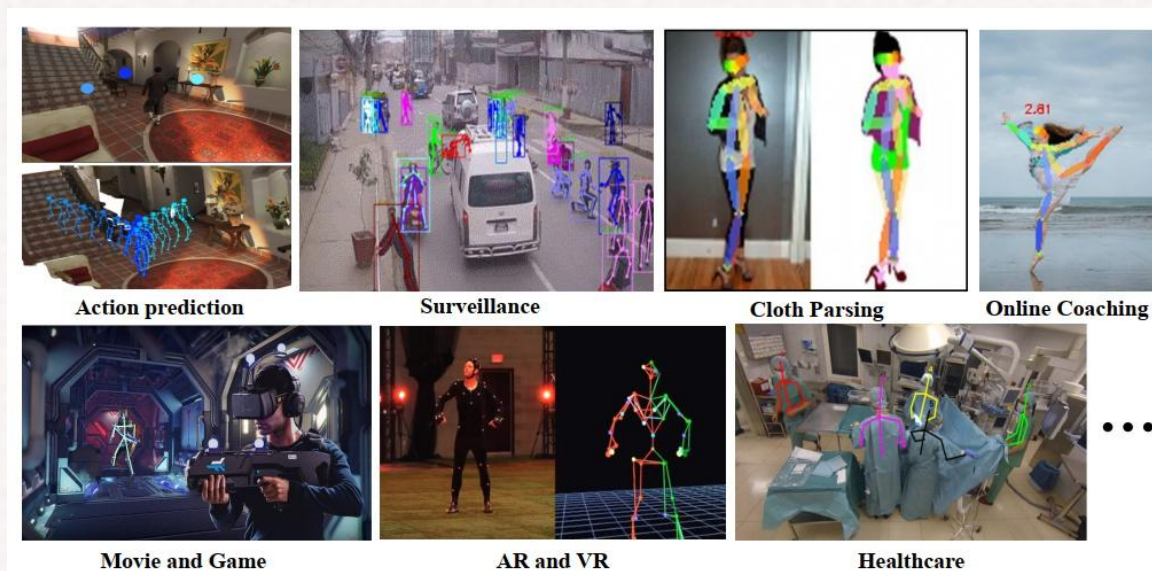


Fig. 7: Various applications of human pose estimation.



# 环境搭建及原理分析

Environment construction and principle analysis

PART.02

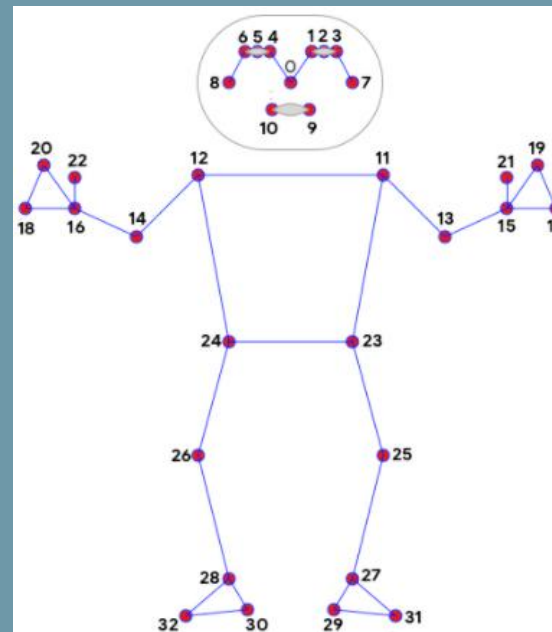
## 环境搭建及原理分析

### 环境

以Python为编程的主要工具  
应用OpenCv、mediapipe库搭建  
基本模板

### 原理

通过对人体骨架数据集的分析，  
利用Python抓取部分骨架点位，  
实现一定的功能



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

## 人体骨架点位图

通过寻找特点的点位，我们可以在设定不同的标准已达到实现不同的功能。



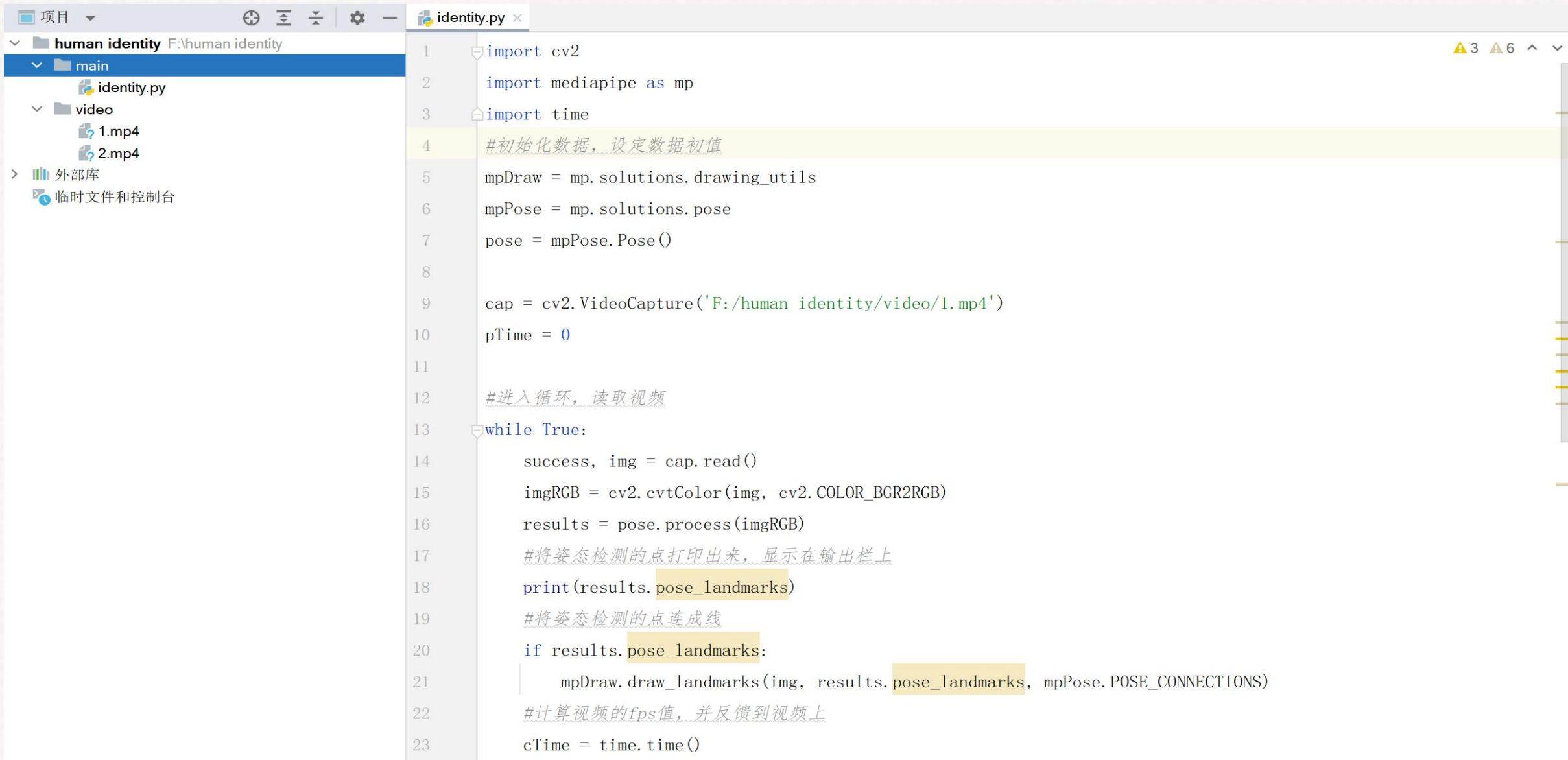
# 初步源码展示及效果

Preliminary source code display and effect

PART.03



## 初步源码展示及效果



```
1 import cv2
2 import mediapipe as mp
3 import time
4 #初始化数据, 设定数据初值
5 mpDraw = mp.solutions.drawing_utils
6 mpPose = mp.solutions.pose
7 pose = mpPose.Pose()
8
9 cap = cv2.VideoCapture('F:/human identity/video/1.mp4')
10 pTime = 0
11
12 #进入循环, 读取视频
13 while True:
14     success, img = cap.read()
15     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
16     results = pose.process(imgRGB)
17     #将姿态检测的点打印出来, 显示在输出栏上
18     print(results.pose_landmarks)
19     #将姿态检测的点连成线
20     if results.pose_landmarks:
21         mpDraw.draw_landmarks(img, results.pose_landmarks, mpPose.POSE_CONNECTIONS)
22     #计算视频的fps值, 并反馈到视频上
23     cTime = time.time()
```



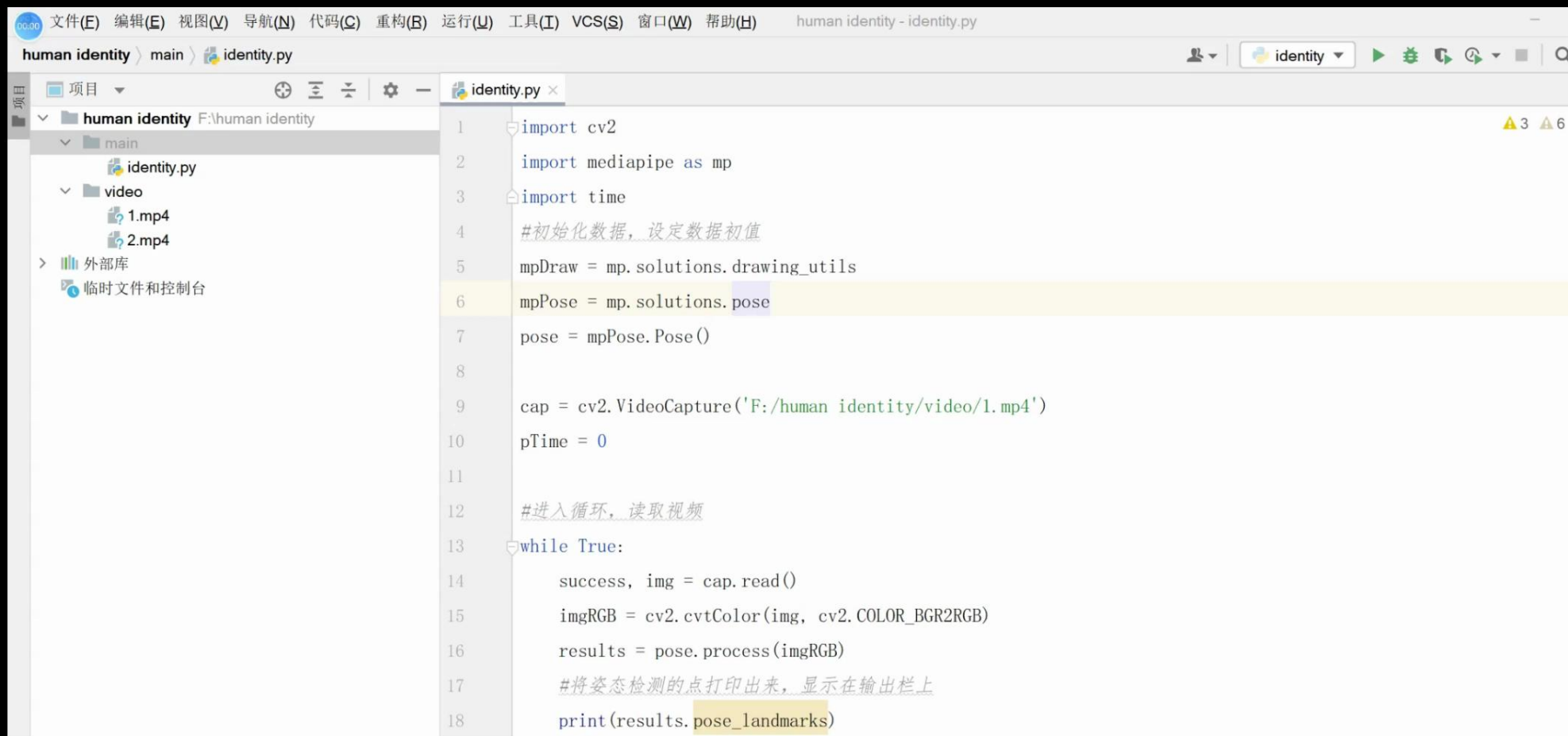


## 初步源码展示及效果

```
24     fps = 1/(cTime-pTime)
25     pTime = cTime
26     cv2.putText(img, str(int(fps)), (70, 50), cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 0), 3)
27     #设置窗口名称为Image, 调整大小为640*480
28     cv2.namedWindow('Image', cv2.WINDOW_NORMAL)
29     cv2.resizeWindow("Image", 640, 480)
30
31     cv2.imshow("Image", img)
32     cv2.waitKey(1)
33
34     # 点击窗口x即可关闭窗口, 结束播放视频
35     if cv2.getWindowProperty('Image', cv2.WND_PROP_VISIBLE) < 1:
36         break
37 cap.release()
38 cv2.destroyAllWindows()
```



## 初步源码展示及效果



```
1 import cv2
2 import mediapipe as mp
3 import time
4 #初始化数据, 设定数据初值
5 mpDraw = mp.solutions.drawing_utils
6 mpPose = mp.solutions.pose
7 pose = mpPose.Pose()
8
9 cap = cv2.VideoCapture('F:/human identity/video/1.mp4')
10 pTime = 0
11
12 #进入循环, 读取视频
13 while True:
14     success, img = cap.read()
15     imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
16     results = pose.process(imgRGB)
17     #将姿态检测的点打印出来, 显示在输出栏上
18     print(results.pose_landmarks)
```



# 存在问题及下一步计划

Existing problems and next steps

PART.04



## 存在问题及下步计划



### 程序代码

- 程序代码只有一部分，现在只完成了模板的设计，后续还需要继续补充
- 未测试是否对其他视频也能起到人体姿态检测的作用，兼容性不可知
- 计数功能尚未实现



### 功能实现

- 引体向上计数功能的实现需要：
  - 每次做完下颌是否是否过杠
  - 每次做完双手臂是否垂直再做下一个
  - 做的过程中双脚有无靠地



展示完毕  
谢谢大家的观看

LET'S START