首都师范大学

人体姿态估计

第七组 王以恒 吴悠 袁忻泽 董子萱 陈云龙

目 录

CONTENTS

01

研究现状

Subject Background

研究现状

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右图所示，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.

应用场景

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

我们小组选择在体育方面进行应用，具体是在人体运动姿态识别方面，如运动计数，运动数据收集，运动实时检测等等

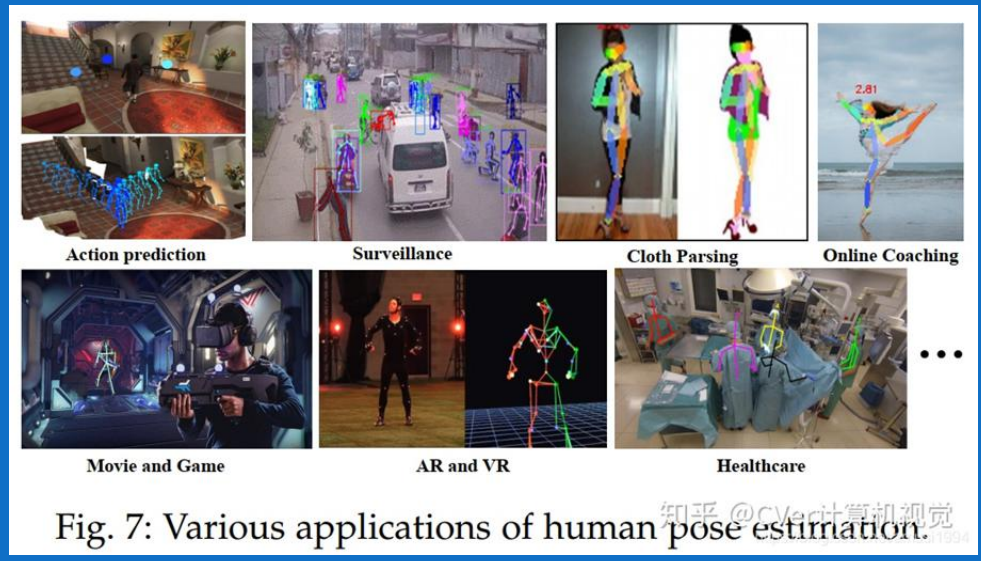
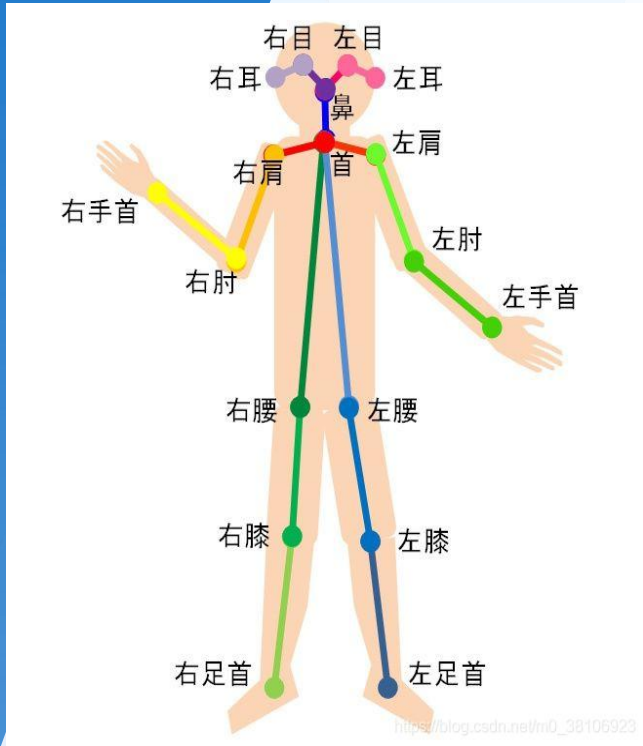


Fig. 7: Various applications of human pose estimation.

研究背景介绍

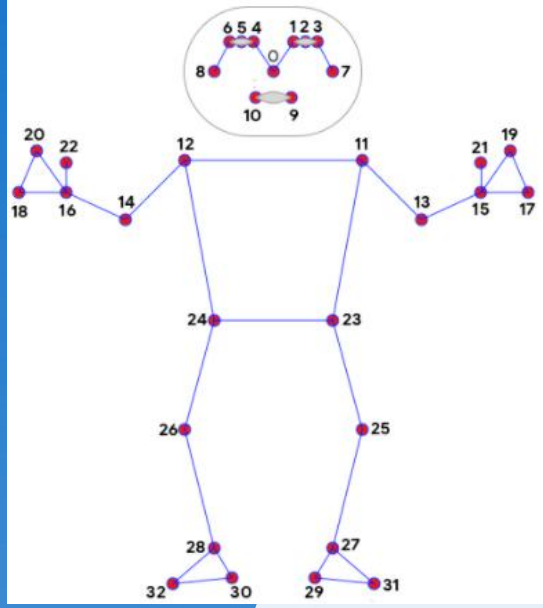
人体姿态估计（Human Posture Estimation），是通过将图片中已检测到的人体关键点正确的联系起来，从而估计人体姿态。
人体关键点通常对应人体上有一定自由度的关节，比如颈、肩、肘、腕、腰、膝、踝等，如左下图。



通过对人体关键点在三维空间相对位置的计算，来估计人体当前的姿态。
进一步，增加时间序列，看一段时间范围内人体关键点的位置变化，可以更加准确的检测姿态，估计目标未来时刻姿态，以及做更抽象的人体行为分析，例如判断一个人是否在打电话等。

人体姿态估计是计算机视觉中一个很基础的问题。从名字的角度来看，可以理解为对“人体”的姿态（关键点，比如头，左手，右脚等）的位置估计。一般我们可以这个问题再具体细分成4个任务：

单人姿态估计 (Single-Person Skeleton Estimation)
多人姿态估计 (Multi-person Pose Estimation)
人体姿态跟踪 （Video Pose Tracking)
3D人体姿态估计 （3D Skeleton Estimation)



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

Posture landmark mode

姿势世界地标

图 5. MediaPipe Pose 真实世界 3D 坐标的示例。

另一个世界坐标中的姿势地标列表。每个地标包括以下内容：

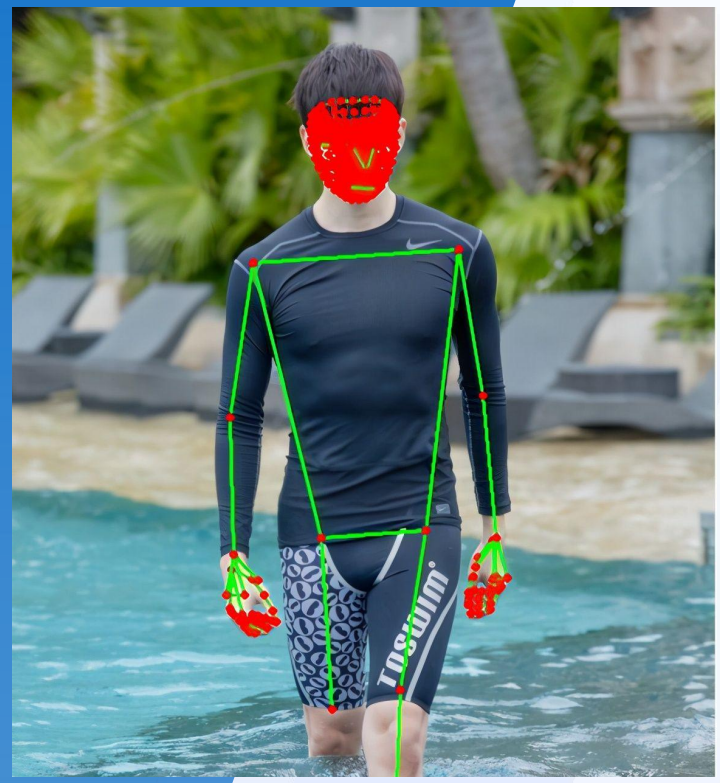
- `x`,`y`和`z`：以米为单位的真实世界 3D 坐标，原点位于臀部之间的中心。
- `visibility`：与对应的pose_landmarks中定义的相同。

分割掩码

输出分段掩码，仅在enable_segmentation设置为时预测`true`。掩码与输入图像具有相同的宽度和高度，并且包含的值分别`[0.0, 1.0]`表示“人类”和“背景”像素的高确定性。有关使用详情，请参阅下面特定于平台的使用示例。`1.0` `0.0`

Mediapipe

MediaPipe 是一款由 Google Research 开发并开源的多媒体机器学习模型应用框架。自2012年起，谷歌在内部的多个产品和服务中使用了它。它最初是为了实时分析YouTube上的视频和音频而开发的。渐渐地，它被整合到更多的产品中，如 Google Lens、ARCore、Google Home 以及 ，都已深度整合了 MediaPipe。



一款多媒体机器学习应用的成败除了依赖于模型本身的好坏，还取决于设备资源的有效调配、多个输入流之间的高效同步、跨平台部署上的便捷程度、以及应用搭建的快速与否。

基于这些需求，谷歌开发并开源了 MediaPipe 项目。除了上述的特性，MediaPipe 还支持 TensorFlow 和 TF Lite 的推理引擎（Inference Engine），任何 TensorFlow 和 TF Lite 的模型都可以在 MediaPipe 上使用。同时，在移动端和嵌入式平台，MediaPipe 也支持设备本身的 GPU 加速。

02

问题定义

Research Content

问题定义

01. 目标

实现引体向上计数的功能

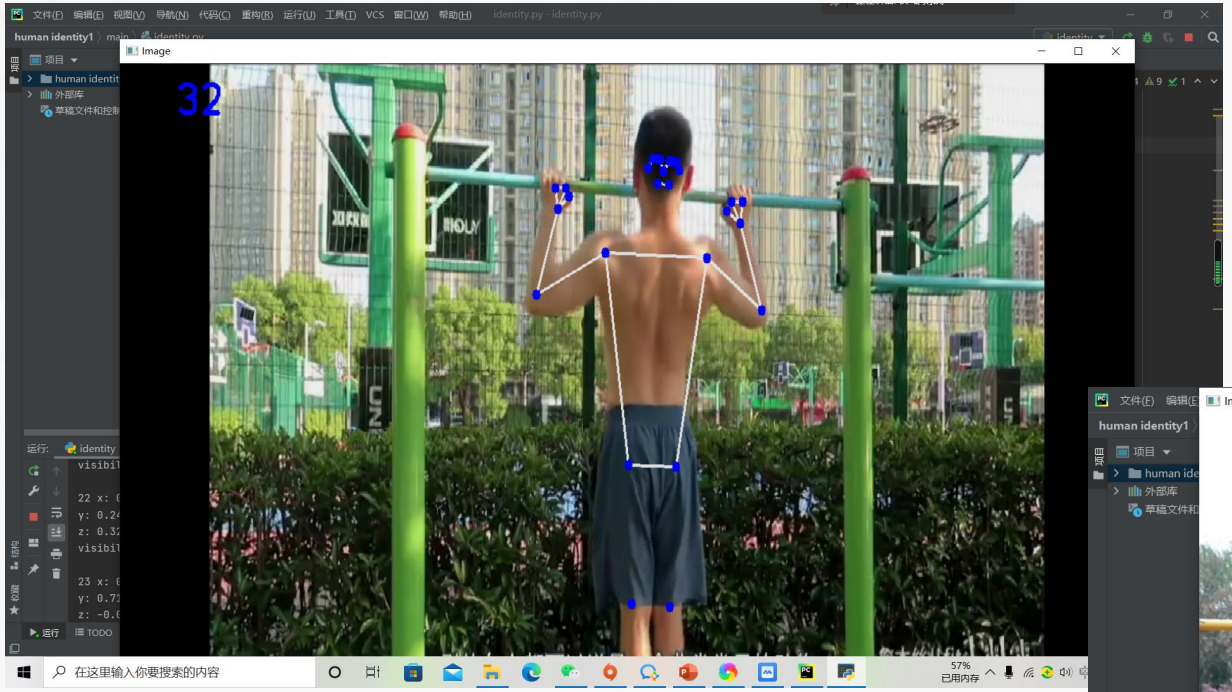
02. 基本原理

通过识别人体33个关键点，将人体坐标化

03. 计数方式

通过坐标点的位置关系进行计数

例图



03

数据采集

Results and Discuss

数据采集

Main函数代码

解释：mian函数是我们的函数主体，主体包含一下几个内容，一个是进行视频导入（导入的视频我们进行fps帧率计算，以及设置窗口大小）；另一个就是调研我们设置的类别dector里的函数，分别是函数findpose以及函数findpostion。

```python
#定义主函数main
def main():
    cap = cv2.VideoCapture('F:/human identity/video/3.mp4')
    pTime = 0
    de = Dector()
    while True:
        success, img = cap.read()
        # 进入循环，读取视频
        img = de.findpose(img)
        lmlist = de.findpostion(img)
        #打印出某一个点的x，y值坐标，例如14
        print(lmlist[14])
        #这里如果要放大14号点的位置，可以采取一下方法
        #将lmlist = de.findpostion(img)修改为lmlist = de.findpostion(img，draw= False)意为不执行画点操作
        # 然后输入cv2.circle(img, (lmlist[14][1], lmlist[14][2]), 15, (0, 0, 0), cv2.FILLED)，
        # 是把cx，cy都替换成了14号点位的【1】【2】号坐标，也就是14号位的x，y轴的实时坐标，
        # 大小更改为了15，颜色改成了黑色（颜色可以自调），在findpose里面默认用的是红色（0,0,255）



        # 计算视频的fps值，并反馈到视频上
        cTime = time.time()
        fps = 1 / (cTime - pTime)
        pTime = cTime
        cv2.putText(img, str(int(fps)), (70, 50), cv2.FONT_HERSHEY_PLAIN, 3, (255, 0, 0), 3)
        # 设置窗口名称为Image，调整大小为640*480
        cv2.namedWindow('Image', cv2.WINDOW_NORMAL)
        cv2.resizeWindow("Image", 1280, 780)
        cv2.imshow("Image", img)
        cv2.waitKey(1)
        # 点击窗口x即可关闭窗口，结束播放视频
        if cv2.getWindowProperty('Image', cv2.WND_PROP_VISIBLE) < 1:
            break
    cap.release()
    cv2.destroyAllWindows()
#定义程序入口
if __name__ == "__main__":
    main()
```

数据采集

模块代码

初始化函数__init__
根据mediapipepose.pose函数进行初始化设置

定义函数findpose：检测人体姿态，并且将检测到的每个点连起来。

定义函数findposition：具体到某一个点，将其进行放大处理，或者找到其详细坐标定位。

```
class Dector():
    #创建类别，将其命名为Dector
    def __init__(self, mode= False, complexity= 1, landmarks= True, enableSeg= False,
                 smoothSeg= True, detectionCon= 0.5, trackingCon= 0.5):
        #这些是初始化pose函数里面的指标，通过按ctrl+左键pose = mpPose.Pose()里面的pose，可以得到检测函数的初始值
        self.mode = mode
        self.complexity = complexity
        self.landmarks = landmarks
        self.enableSeg = enableSeg
        self.smoothSeg = smoothSeg
        self.detectionCon = detectionCon
        self.trackingCon = trackingCon
#这是初始化medpipe库中的检测函数
        self.mpDraw = mp.solutions.drawing_utils
        self.mpPose = mp.solutions.pose
        self.pose = self.mpPose.Pose(self.mode, self.complexity, self.landmarks, self.enableSeg,
                                     self.detectionCon, self.trackingCon)
```

```
#findpose是检测人体姿态，并且把检测到的每个点连起来
    def findpose(self, img, draw= True):
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        self.results = self.pose.process(imgRGB)
        if self.results.pose_landmarks:
            if draw:
                self.mpDraw.draw_landmarks(img, self.results.pose_landmarks, self.mpPose.POSE_CONNECTIONS)
        return img
```

```
#findpostion是我们要具体到某个点，把它进行放大处理，或者找到其详细坐标定位
    def findpostion(self, img, draw= True):
        lmlist = []
        for id, lm in enumerate(self.results.pose_landmarks.landmark):
            h, w, c = img.shape
            cx, cy = int(lm.x * w), int(lm.y * h)
            lmlist.append([id, cx, cy])
            if draw:
                cv2.circle(img, (cx, cy), 5, (255, 0, 0), cv2.FILLED)
        return lmlist
```

04

总结

Conclusion

总结

遇到的困难

1.

袁忻泽：首先是刚开始的时候下库出问题，版本啥的不对导致下库失败。然后是在学习这几个库用法的时候大致了解了但自己实施起来又比较麻烦。

2.

吴悠：刚开始接触这个课题时有些无从下手，后来看完老师发的资料，并且上网查阅过往相关资料后。跟组员确定了大致方向以及最终要实现的目标。之后在b站上和csdn等网站上学习相关知识，但是比如学习mediapipe的库就有些困难。在现有的程序基础上想改进为目标这一过程，不知道该如何修改程序。

3.

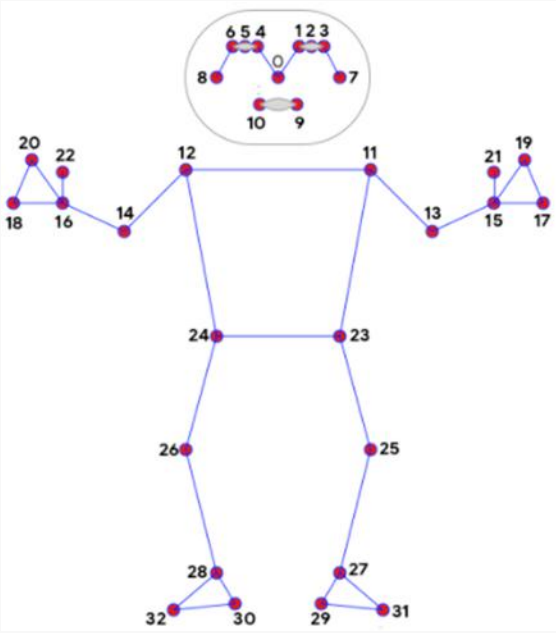
陈云龙：开始时候的一些函数库无法下载，在网上看解决方案最终解决了。但是又出现了读取视频图片失败的问题，经过一系列整改之后，可以正常运行了。可是对这些需要用到的库函数还不是很了解，学起来也相当吃力，尤其是mediapipe库。

总结

下步打算

我们小组目前对人体运动姿态识别进行了模块化处理，这样对以后的应用我们可以直接进行调用。下一步就是对做具体运动时，如引体向上，进行实时检测以及计数

- 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



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

Thank You

感 谢 聆 听

学生：第七小组

指导老师：刘铁