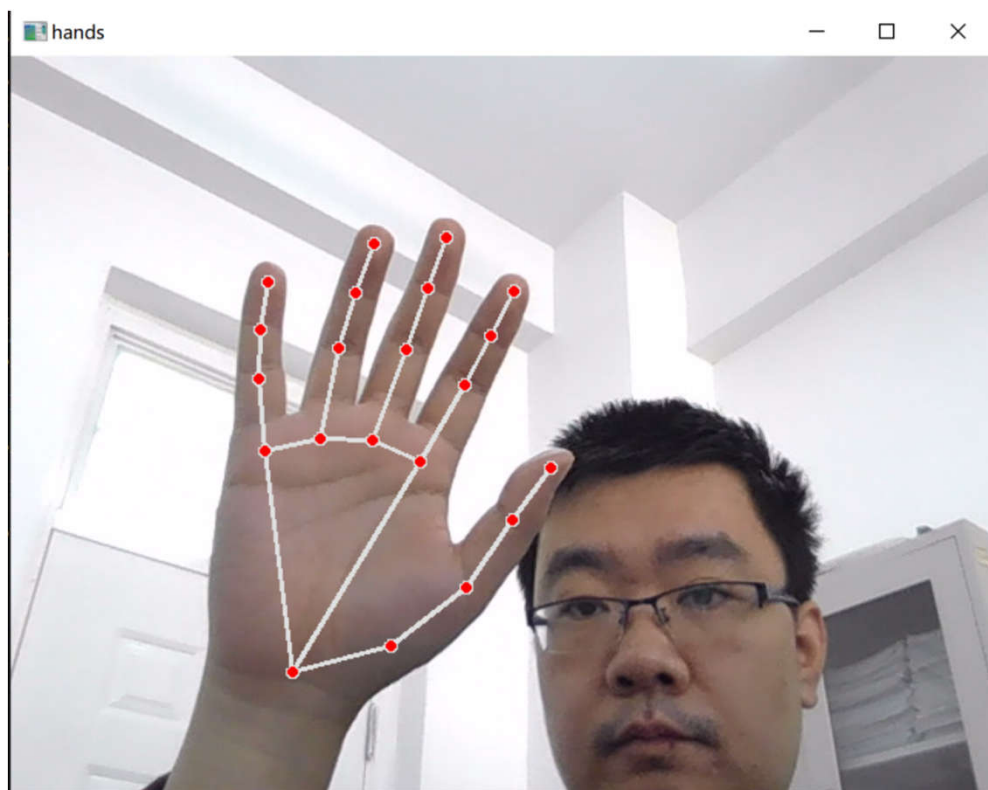


# Python编程与人工智能实践

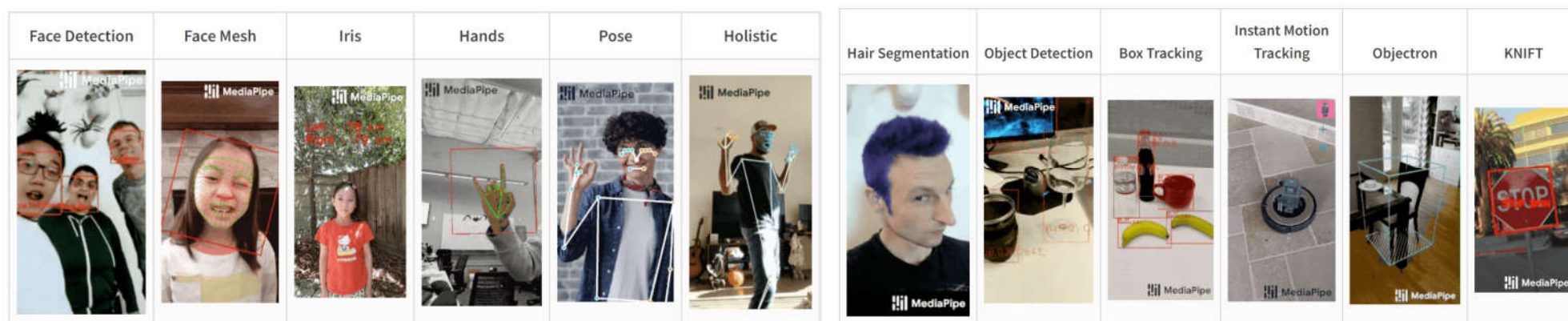


应用篇：  
基于mediapipe的手势识别

于泓  
鲁东大学  
信息与电气工程学院  
2021.11.13

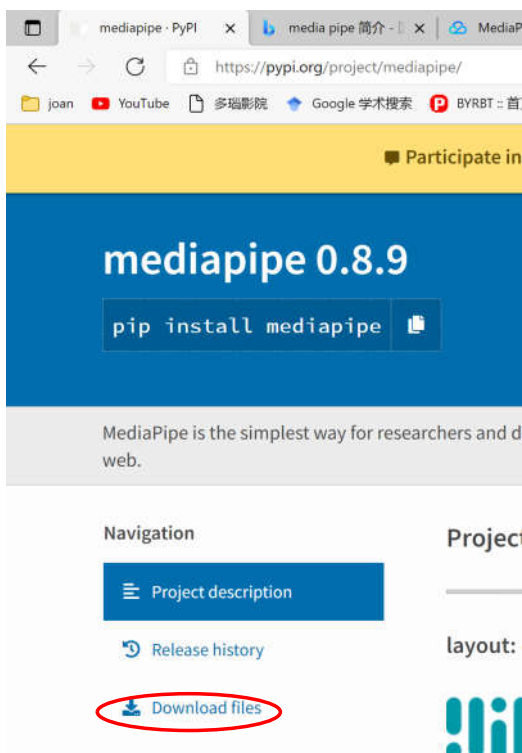
# MediaPipe: Google Research 开源的跨平台多媒体机器学习模型应用框架

作为一款跨平台框架，MediaPipe 不仅可以被部署在服务器端，更可以在多个移动端（安卓和苹果 iOS）和嵌入式平台（Google Coral 和树莓派）中作为设备端机器学习推理（On-device Machine Learning Inference）框架。



MediaPipe的Python安装: (1) 安装 opencv `pip install opencv-python`

(2) `pip install mediapipe` —————→ 直接安装由于网络问题  
有些包下载不了,  
可以到  
<https://pypi.org/project/> 手动下载  
相关的whl文件。



<a href="#">mediapipe-0.8.9-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl</a> (32.8 MB)	Wheel	cp37	Nov 4, 2021	<a href="#">View</a>
<a href="#">mediapipe-0.8.9-cp37-cp37m-win_amd64.whl</a> (48.6 MB)	Wheel	cp37	Nov 4, 2021	<a href="#">View</a>
<a href="#">mediapipe-0.8.9-cp38-cp38-macosx_10_15_x86_64.whl</a> (33.5 MB)	Wheel	cp38	Nov 4, 2021	<a href="#">View</a>
<a href="#">mediapipe-0.8.9-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl</a> (32.8 MB)	Wheel	cp38	Nov 4, 2021	<a href="#">View</a>
<a href="#">mediapipe-0.8.9-cp38-cp38-win_amd64.whl</a> (48.6 MB)	Wheel	cp38	Nov 4, 2021	<a href="#">View</a>
<a href="#">mediapipe-0.8.9-cp39-cp39-macosx_10_15_x86_64.whl</a> (33.5 MB)	Wheel	cp39	Nov 4, 2021	<a href="#">View</a>

## 手动安装

此电脑 > Data (D:) > 工作相关 > 我设计的课程 > python与人工智能课程设计 > mediapipe				
名称	修改日期	类型	大小	
hand_feature	2021/11/13 14:47	文件夹		
mediapipe-0.8.9-cp38-cp38-win_amd64.whl	2021/11/13 9:33	WHL 文件	47,423 KB	
opencv_contrib_python-4.5.4.58-cp38-cp38-win_amd64.whl	2021/11/13 9:39	WHL 文件	40,976 KB	
protobuf-3.19.1-cp38-cp38-win_amd64.whl	2021/11/13 9:44	WHL 文件	875 KB	
wheel-0.37.0-py2.py3-none-any.whl	2021/11/13 9:48	WHL 文件	35 KB	

```
pip install xxxx.whl
```

```
import mediapipe as mp
import cv2
import numpy as np

if __name__ == "__main__":

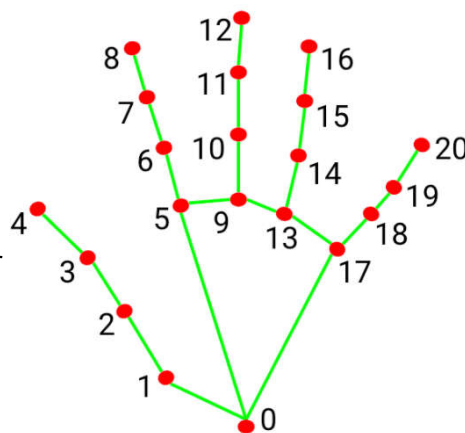
    # 打开摄像头
    cap = cv2.VideoCapture(0)

    # 定义手 检测对象
    mpHands = mp.solutions.hands
    hands = mpHands.Hands()
    mpDraw = mp.solutions.drawing_utils
```

```
mp_hands.Hands(
    model_complexity=0,
    min_detection_confidence=0.5,
    min_tracking_confidence=0.5)
```

引入跟踪  
增加速度

检测21个关键点



- |                       |                       |
|-----------------------|-----------------------|
| 0. WRIST              | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC          | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP          | 13. RING_FINGER_MCP   |
| 3. THUMB_IP           | 14. RING_FINGER_PIP   |
| 4. THUMB_TIP          | 15. RING_FINGER_DIP   |
| 5. INDEX_FINGER_MCP   | 16. RING_FINGER_TIP   |
| 6. INDEX_FINGER_PIP   | 17. PINKY_MCP         |
| 7. INDEX_FINGER_DIP   | 18. PINKY_PIP         |
| 8. INDEX_FINGER_TIP   | 19. PINKY_DIP         |
| 9. MIDDLE_FINGER_MCP  | 20. PINKY_TIP         |
| 10. MIDDLE_FINGER_PIP |                       |

Fig 2. 21 hand landmarks.

```
while True:
```

```
    # 读取一帧图像
```

```
    success, img = cap.read()
```

```
    image_height, image_width, _ = img.shape
```

```
    # 转换为RGB
```

```
    imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

```
    # 得到检测结果
```

```
    results = hands.process(imgRGB)
```

```
    if results.multi_hand_landmarks:
```

```
        for hand in results.multi_hand_landmarks:
```

```
            print("\r%.2f %.2f %.2f %.2f %.2f %.2f" % (hand.landmark[0].z, hand.landmark[4].z, hand.landmark[8].z, hand.landmark[12].z, hand.landmark[16].z, hand.landmark[20].z))
```

```
            mpDraw.draw_landmarks(img, hand, mpHands.HAND_CONNECTIONS)
```

```
            # for i in range(21):
```

```
                # pos_x = hand.landmark[i].x*image_width
```

```
                # pos_y = hand.landmark[i].y*image_height
```

```
                # # 画点
```

```
                # cv2.circle(img, (int(pos_x),int(pos_y)), 2, (0,0,255),-1)
```

```
    cv2.imshow("hands",img)
```

```
    key = cv2.waitKey(1) & 0xFF
```

```
    # 按键 "q" 退出
```

```
    if key == ord('q'):
```

```
        break
```

```
cap.release()
```

hand.landmark[i].x

hand.landmark[i].y

hand.landmark[i].z

归一化

横纵坐标

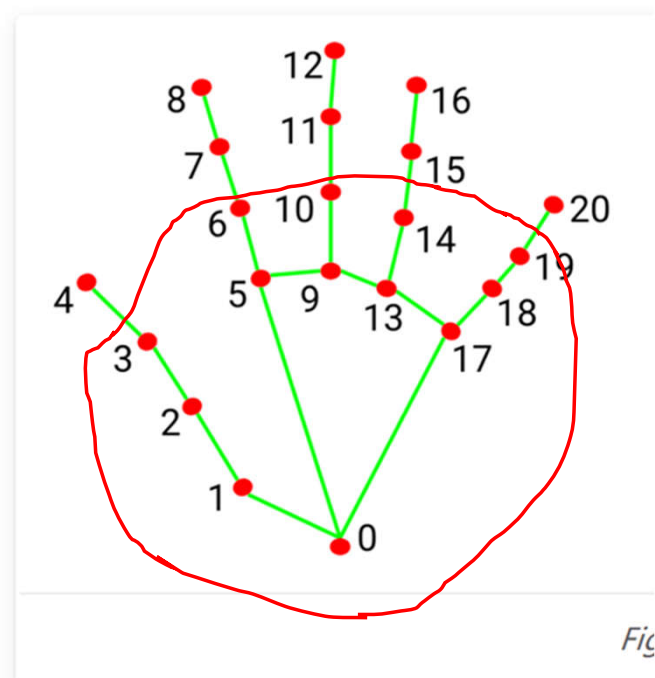
对手腕的相对位置

绘图

x,y 实际坐标的计算



手势识别:



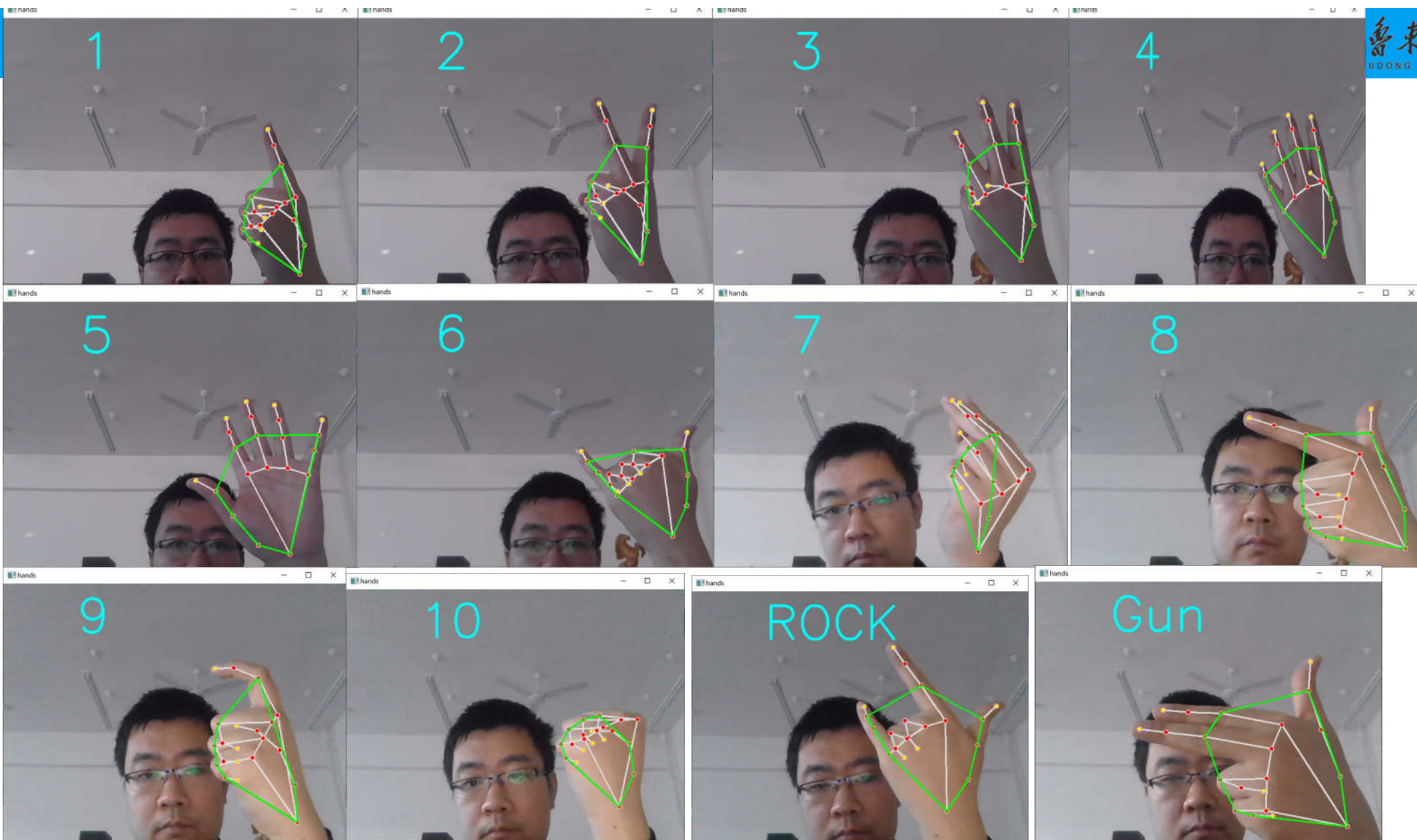
关键点 [0,1,2,3,6,10,14,19,18,17,10]

构造一个**凸包**，通过计算

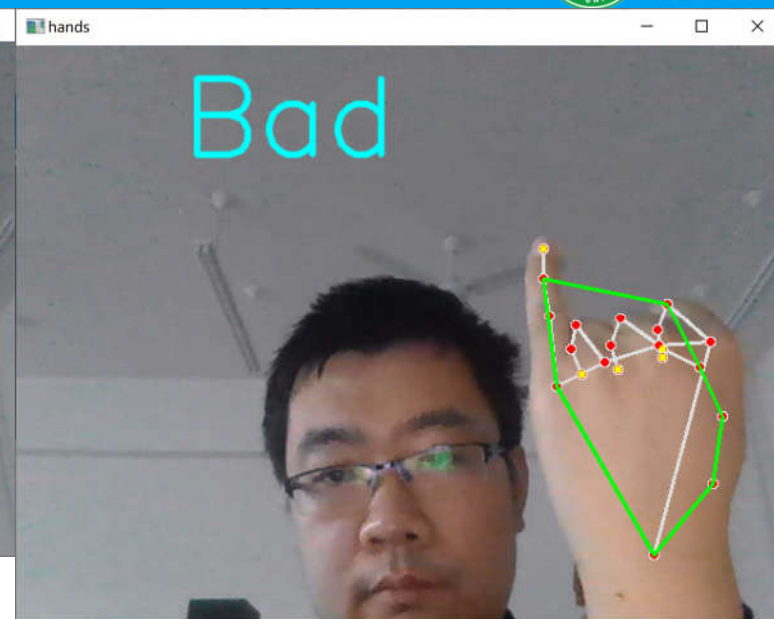
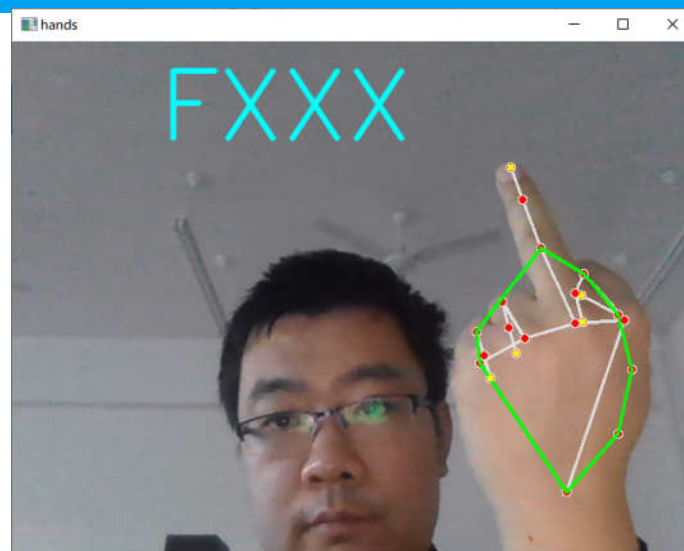
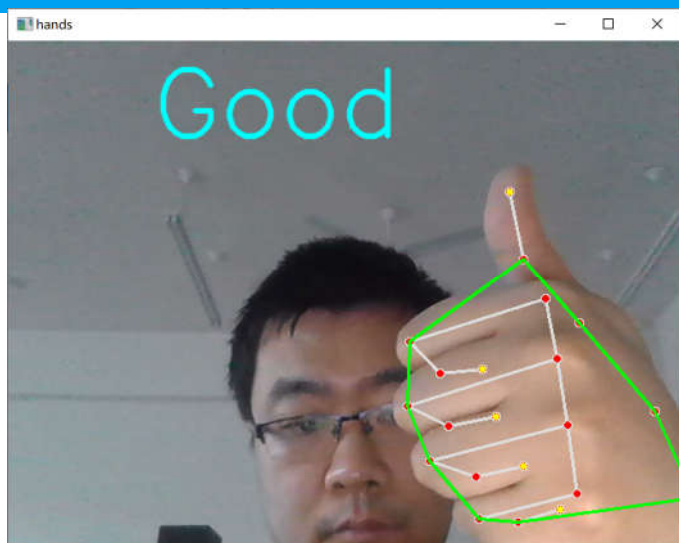
**[4,8,12,16,20] 谁在凸包外**

来进行简单的手势识别









```
import mediapipe as mp
import cv2
import numpy as np
```

```
if __name__ == "__main__":
```

```
    cap = cv2.VideoCapture(0)
    # 定义手 检测对象
    mpHands = mp.solutions.hands
    hands = mpHands.Hands()
    mpDraw = mp.solutions.drawing_utils
```

```
    while True:
```

```
        # 读取一帧图像
```

```
        success, img = cap.read()
```

```
        if not success:
            continue
```

```
        image_height, image_width, _ = np.shape(img)
```

```
        # 转换为RGB
```

```
        imgRGB = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
```

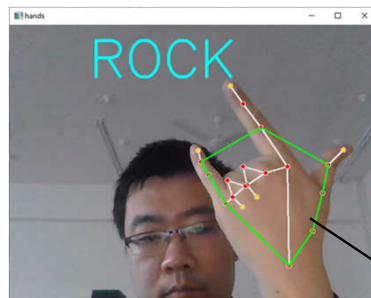
```
        # 得到检测结果
```

```
        results = hands.process(imgRGB)
```

```
        if results.multi_hand_landmarks:
```

```
            hand = results.multi_hand_landmarks[0]
```

```
            mpDraw.draw_landmarks(img, hand, mpHands.HAND_CONNECTIONS)
```



```
# 采集所有关键点的坐标
```

```
list_lms = []
```

```
for i in range(21):
```

```
    pos_x = hand.landmark[i].x*image_width
```

```
    pos_y = hand.landmark[i].y*image_height
```

```
    list_lms.append([int(pos_x), int(pos_y)])
```

```
# 构造凸包点
```

```
list_lms = np.array(list_lms, dtype=np.int32)
```

```
hull_index = [0, 1, 2, 3, 6, 10, 14, 19, 18, 17, 10]
```

```
hull = cv2.convexHull(list_lms[hull_index, :])
```

```
# 绘制凸包
```

```
cv2.polylines(img, [hull], True, (0, 255, 0), 2)
```

```
# 查找外部的点数
```

```
n_fig = -1
```

```
ll = [4, 8, 12, 16, 20]
```

```
up_fingers = []
```

```
for i in ll:
```

```
    pt = (int(list_lms[i][0]), int(list_lms[i][1]))
```

```
    dist = cv2.pointPolygonTest(hull, pt, True)
```

```
    if dist < 0:
```

```
        up_fingers.append(i)
```

```
str_guenter = get_str_guenter(up_fingers,list_lms)

cv2.putText(img,' %s'%(str_guenter),(90,90),cv2.FONT_HERSHEY_SIMPLEX,3,(255,255,0),4,cv2.LINE_AA)

for i in ll:
    pos_x = hand.landmark[i].x*image_width
    pos_y = hand.landmark[i].y*image_height
    # 画点
    cv2.circle(img, (int(pos_x),int(pos_y)), 3, (0,255,255),-1)

cv2.imshow("hands",img)

key = cv2.waitKey(1) & 0xFF

# 按键 "q" 退出
if key == ord('q'):
    break
cap.release()
```

```
def get_str_guester(up_fingers,list_lms):

    if len(up_fingers)==1 and up_fingers[0]==8:
```

```
        v1 = list_lms[6]-list_lms[7]
        v2 = list_lms[8]-list_lms[7]
```

```
        angle = get_angle(v1,v2)
```

```
        if angle<160:
            str_guester = "9"
        else:
            str_guester = "1"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==4:
        str_guester = "Good"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==20:
        str_guester = "Bad"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==12:
        str_guester = "FXXX"
```

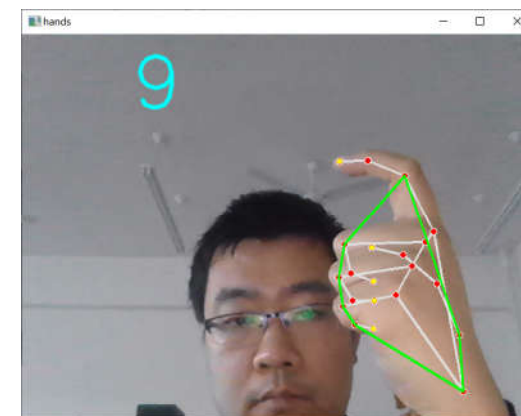
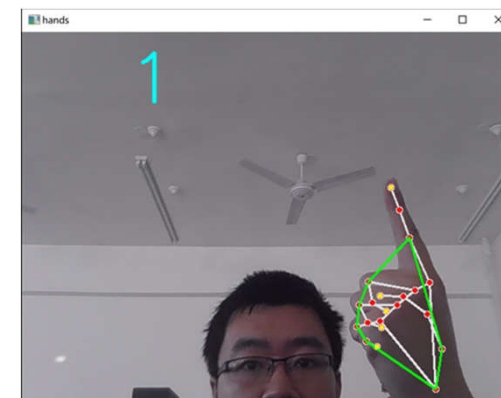
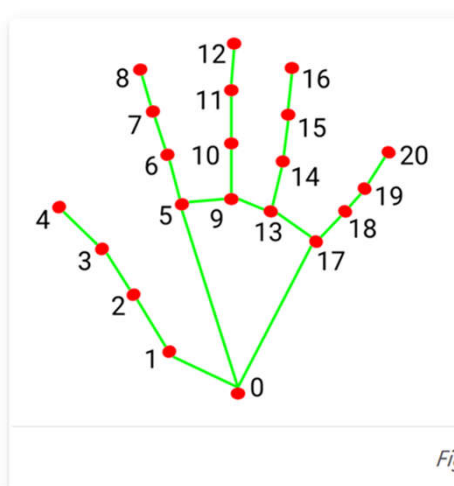
```
    elif len(up_fingers)==2 and up_fingers[0]==8 and up_fingers[1]==12:
        str_guester = "2"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==4 and up_fingers[1]==20:
        str_guester = "6"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==4 and up_fingers[1]==8:
        str_guester = "8"
```

```
def get_angle(v1,v2):
    angle = np.dot(v1,v2)/(np.sqrt(np.sum(v1*v1))*np.sqrt(np.sum(v2*v2)))
    angle = np.arccos(angle)/3.14*180

    return angle
```



```
def get_str_guesser(up_fingers, list_lms):
```

```
    if len(up_fingers)==1 and up_fingers[0]==8:
```

```
        v1 = list_lms[6]-list_lms[7]
```

```
        v2 = list_lms[8]-list_lms[7]
```

```
        angle = get_angle(v1,v2)
```

```
        if angle<160:
```

```
            str_guesser = "9"
```

```
        else:
```

```
            str_guesser = "1"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==4:
```

```
        str_guesser = "Good"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==20:
```

```
        str_guesser = "Bad"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==12:
```

```
        str_guesser = "FXXX"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==8 and up_fingers[1]==12:
```

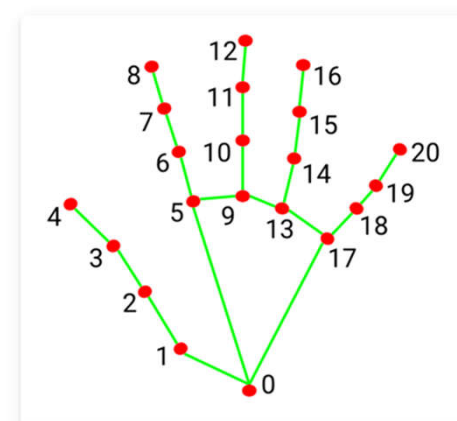
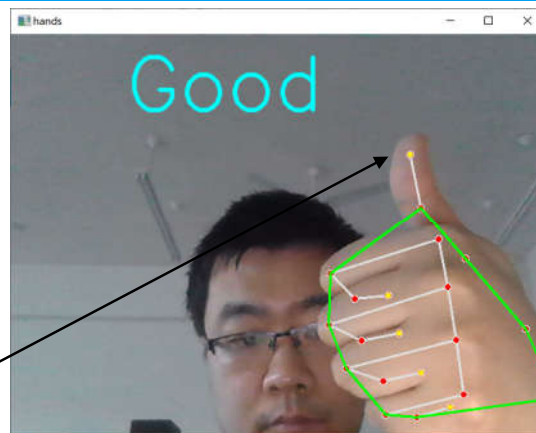
```
        str_guesser = "2"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==4 and up_fingers[1]==20:
```

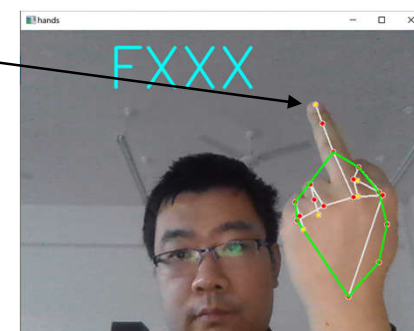
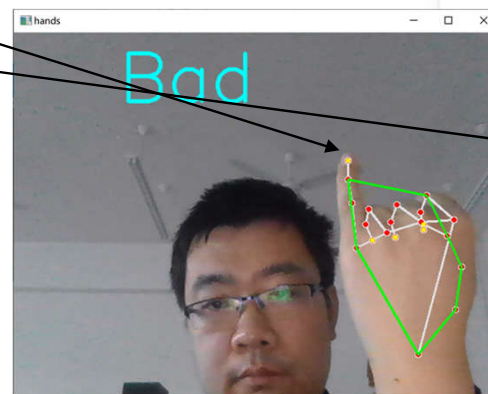
```
        str_guesser = "6"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==4 and up_fingers[1]==8:
```

```
        str_guesser = "8"
```



Fig



```
def get_str_guester(up_fingers,list_lms):
```

```
    if len(up_fingers)==1 and up_fingers[0]==8:
```

```
        v1 = list_lms[6]-list_lms[7]
```

```
        v2 = list_lms[8]-list_lms[7]
```

```
        angle = get_angle(v1,v2)
```

```
        if angle<160:
```

```
            str_guester = "9"
```

```
        else:
```

```
            str_guester = "1"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==4:
```

```
        str_guester = "Good"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==20:
```

```
        str_guester = "Bad"
```

```
    elif len(up_fingers)==1 and up_fingers[0]==12:
```

```
        str_guester = "FXXX"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==8 and up_fingers[1]==12:
```

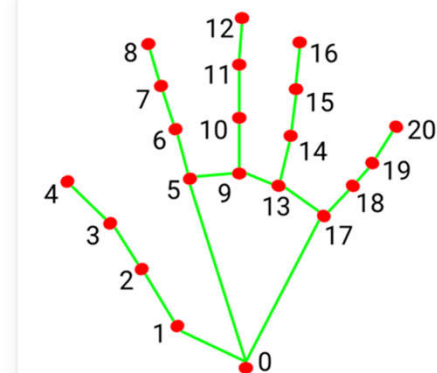
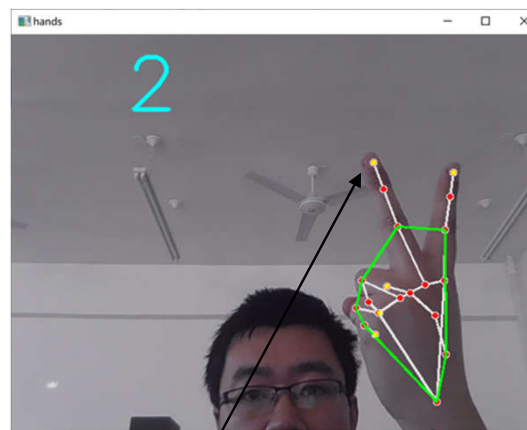
```
        str_guester = "2"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==4 and up_fingers[1]==20:
```

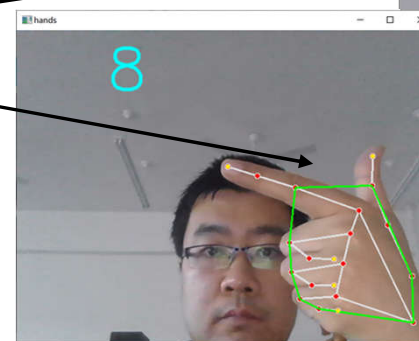
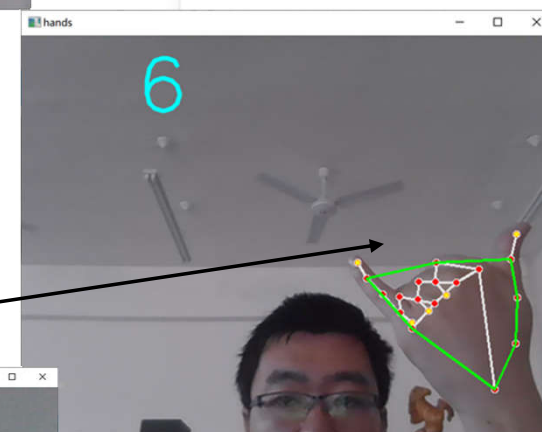
```
        str_guester = "6"
```

```
    elif len(up_fingers)==2 and up_fingers[0]==4 and up_fingers[1]==8:
```

```
        str_guester = "8"
```



Fig





```
elif len(up_fingers)==3 and up_fingers[0]==8 and up_fingers[1]==12 and up_fingers[2]==16:
    str_guesser = "3"

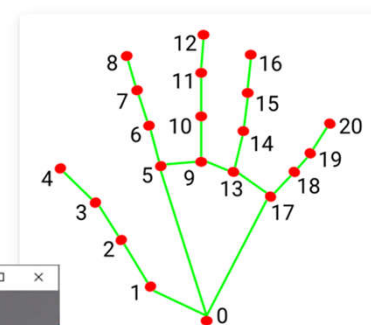
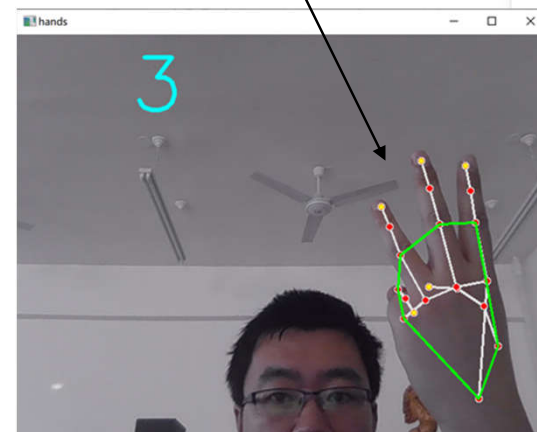
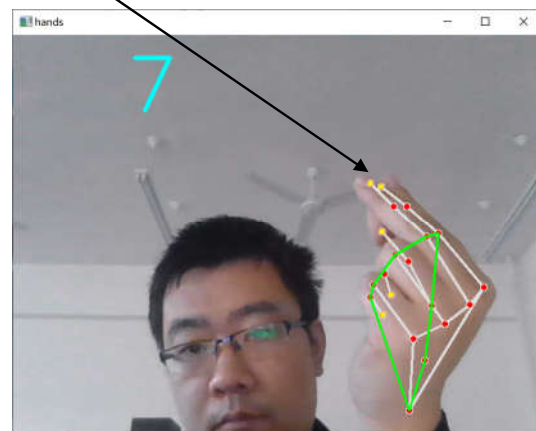
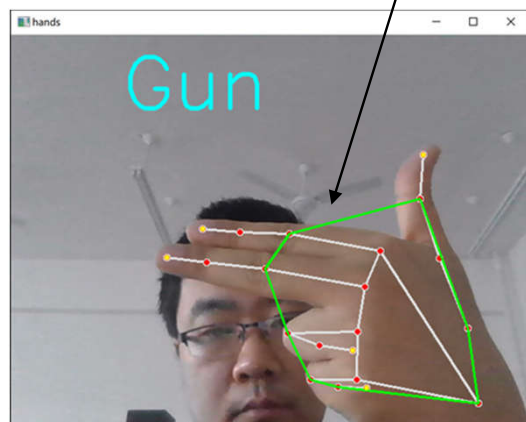
elif len(up_fingers)==3 and up_fingers[0]==4 and up_fingers[1]==8 and up_fingers[2]==12:

    dis_8_12 = list_lms[8,:] - list_lms[12,:]
    dis_8_12 = np.sqrt(np.dot(dis_8_12,dis_8_12))

    dis_4_12 = list_lms[4,:] - list_lms[12,:]
    dis_4_12 = np.sqrt(np.dot(dis_4_12,dis_4_12))

    if dis_4_12/(dis_8_12+1) <3:
        str_guesser = "7"

    elif dis_4_12/(dis_8_12+1) >5:
        str_guesser = "Gun"
    else:
        str_guester = "7"
```



Fig

2021/11/29

```

elif len(up_fingers)==3 and up_fingers[0]==4 and up_fingers[1]==8 and up_fingers[2]==20:
    str_guesser = "ROCK"

elif len(up_fingers)==4 and up_fingers[0]==8 and up_fingers[1]==12 and up_fingers[2]==16 and up_fingers[3]==20:
    str_guester = "4"

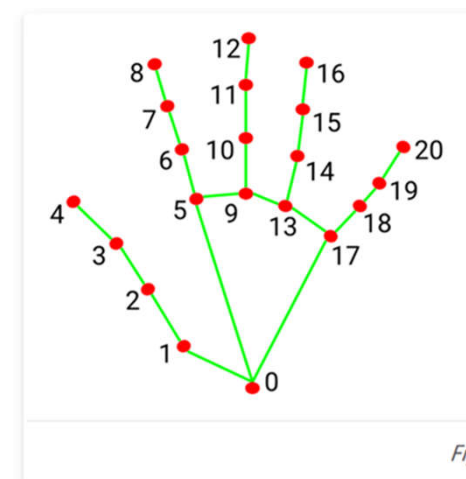
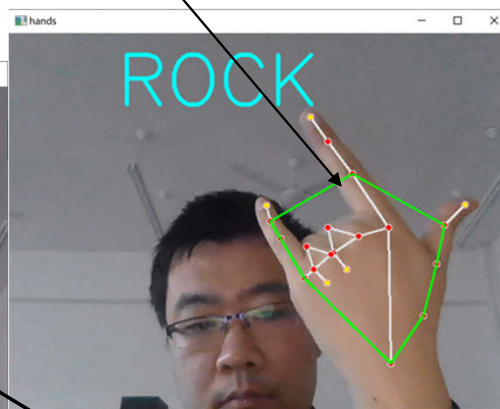
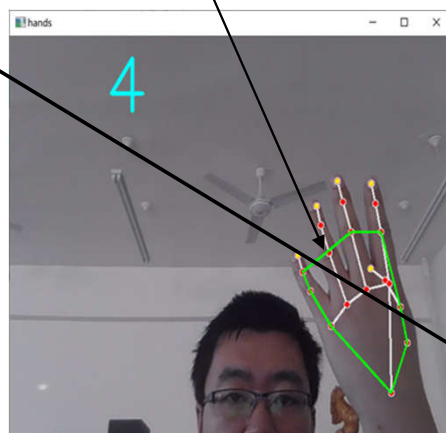
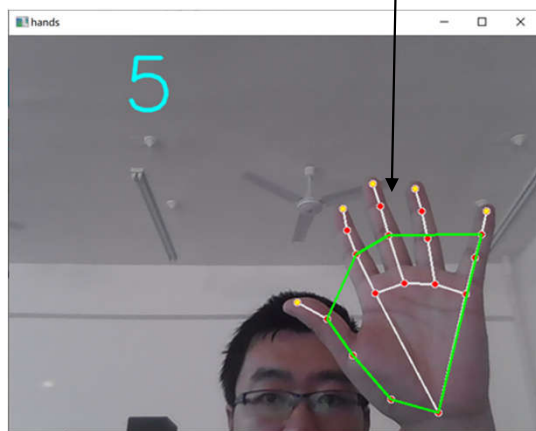
elif len(up_fingers)==5:
    str_guester = "5"

elif len(up_fingers)==0:
    str_guester = "10"

else:
    str_guester = ""

return str_guesser

```



Fig



2021/11/29

16