实验八 基于 mediapipe 的深度模型应用

一、实验目的及要求

1、学习和掌握 mediapipe 的安装方法

2、掌握利用 mediapipe 进行人脸关键点检测和手部关键点检测的方法。

3、掌握利用 mediapipe 检测的脸部关键点，提取特定区域进行局部颜色变换的方法。

4、掌握利用 mediapipe 检测的手势关键点进行手势识别的方法。

二、预习要求

阅读本实验例程部，安装 mediapipe，并利用 mediapipe 进行人脸关键点和手势关键

点检测，以便能够充分利用实验时间编程调试。

三、实验设备

硬件：PC 机。

软件：Python 及相关集成开发环境。

四、实验内容

（1）实现 mediapipe 的安装

（2）利用 mediapipe 实现人脸关键点的检测并将人脸上的关键点绘制出来（face\_base.py）

import mediapipe as mp

import cv2

import numpy as np

if \_\_name\_\_ == "\_\_main\_\_":

*# 构建脸部特征提取对象*

mp\_face\_mesh = mp.solutions.face\_mesh

face\_mesh = mp\_face\_mesh.FaceMesh(static\_image\_mode=False,

max\_num\_faces=1,

refine\_landmarks=True,

min\_detection\_confidence=0.5,

min\_tracking\_confidence=0.5)

*# 构建绘图对象*

mp\_drawing = mp.solutions.drawing\_utils

mp\_drawing\_styles = mp.solutions.drawing\_styles

*# 开启摄像头*

cap = cv2.VideoCapture(0)

while True:

*# 读取一帧图像*

success, img = cap.read()

if not success:

continue

*# 获取宽度和高低*

image\_height, image\_width, \_ = np.shape(img)

*# BGR 转 RGB*

img\_RGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

*# 进行特征点提取*

results = face\_mesh.process(img\_RGB)

if results.multi\_face\_landmarks:

for face\_landmarks in results.multi\_face\_landmarks:

*# 利用 内置的mp\_drawing 进行绘图*

mp\_drawing.draw\_landmarks(image=img,

landmark\_list=face\_landmarks,

connections=mp\_face\_mesh.FACEMESH\_TESSELATION,

landmark\_drawing\_spec=None,

connection\_drawing\_spec=mp\_drawing\_styles

.get\_default\_face\_mesh\_tesselation\_style())

mp\_drawing.draw\_landmarks(image=img,

landmark\_list=face\_landmarks,

connections=mp\_face\_mesh.FACEMESH\_CONTOURS,

landmark\_drawing\_spec=None,

connection\_drawing\_spec=mp\_drawing\_styles

.get\_default\_face\_mesh\_contours\_style())

mp\_drawing.draw\_landmarks(

image=img,

landmark\_list=face\_landmarks,

connections=mp\_face\_mesh.FACEMESH\_IRISES,

landmark\_drawing\_spec=None,

connection\_drawing\_spec=mp\_drawing\_styles

.get\_default\_face\_mesh\_iris\_connections\_style())

*# # 自行计算478个关键点的坐标 并绘制*

*# if face\_landmarks:*

*# # 计算关键点坐标*

*# for i in range(478):*

*# pos\_x = int(face\_landmarks.landmark[i].x \* image\_width)*

*# pos\_y = int(face\_landmarks.landmark[i].y \* image\_height)*

*# cv2.circle(img, (pos\_x,pos\_y), 3, (0,255,0),-1)*

cv2.imshow("face-mesh",img)

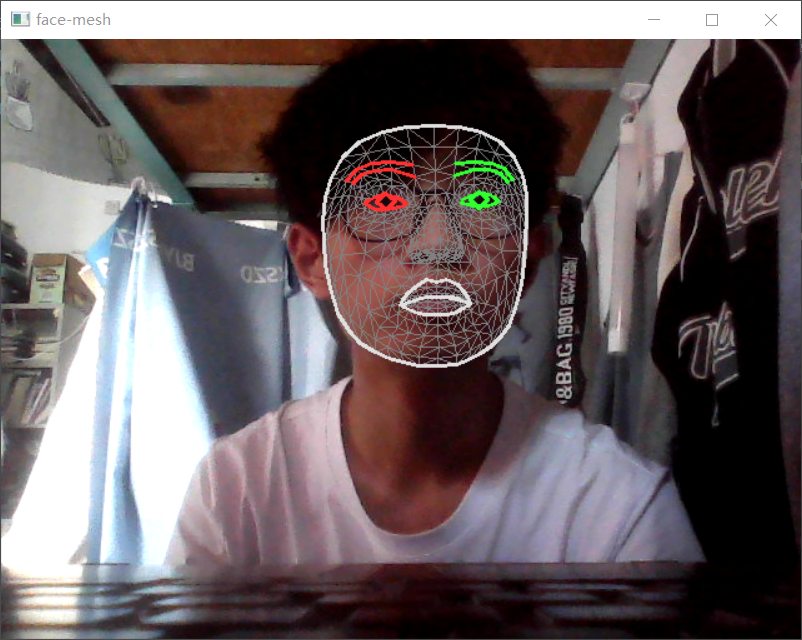
key = cv2.waitKey(1) & 0xFF

*# 按键 "q" 退出*

if key == ord('q'):

break

cap.release()



（3）利用 mediapipe 检测到的人脸关键点实现魔幻口红的特效，动态的修改唇部皮肤的

颜色（face\_LIP\_CAM.py）

import mediapipe as mp

import cv2

import numpy as np

def change\_color\_lip(img,list\_lms,index\_lip\_up,index\_lip\_down,color):

*# cv2.imshow("input",img)*

mask = np.zeros\_like(img)

points\_lip\_up = list\_lms[index\_lip\_up,:]

mask = cv2.fillPoly(mask,[points\_lip\_up],(255,255,255))

points\_lip\_down = list\_lms[index\_lip\_down,:]

mask = cv2.fillPoly(mask,[points\_lip\_down],(255,255,255))

*# cv2.imshow("mask",mask)*

img\_color\_lip = np.zeros\_like(img)

img\_color\_lip[:] = color

*# cv2.imshow("color lip",img\_color\_lip)*

img\_color\_lip = cv2.bitwise\_and(mask,img\_color\_lip)

*# cv2.imshow("color lip",img\_color\_lip)*

img\_color\_lip = cv2.GaussianBlur(img\_color\_lip,(7,7),10)

img\_color\_lip = cv2.addWeighted(img,1,img\_color\_lip,0.8,0)

return img\_color\_lip

def empty(a):

pass

if \_\_name\_\_ == "\_\_main\_\_":

*# 创建人脸关键点检测对象*

mp\_face\_mesh = mp.solutions.face\_mesh

face\_mesh = mp\_face\_mesh.FaceMesh(static\_image\_mode=False,

max\_num\_faces=1,

refine\_landmarks=True,

min\_detection\_confidence=0.5,

min\_tracking\_confidence=0.5)

*# 口红颜色调节*

cv2.namedWindow("BGR")

cv2.resizeWindow("BGR",640,240)

cv2.createTrackbar("Blue","BGR",0,255,empty)

cv2.createTrackbar("Green","BGR",0,255,empty)

cv2.createTrackbar("Red","BGR",0,255,empty)

cap = cv2.VideoCapture(0)

while True:

*# 读取一帧图像*

success, img = cap.read()

if not success:

continue

*# 获取宽度和高低*

image\_height, image\_width, \_ = np.shape(img)

*# BGR 转 RGB*

img\_RGB = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

results = face\_mesh.process(img\_RGB)

list\_lms = []

if results.multi\_face\_landmarks:

face\_landmarks = results.multi\_face\_landmarks[0]

for i in range(478):

pos\_x = int(face\_landmarks.landmark[i].x \* image\_width)

pos\_y = int(face\_landmarks.landmark[i].y \* image\_height)

list\_lms.append((pos\_x,pos\_y))

list\_lms = np.array(list\_lms,dtype=np.int32)

index\_lip\_up = [61, 185, 40, 39, 37,0, 267, 269, 270, 409, 291,308,415,310,311,312,13,82,80,191,78,61]

index\_lip\_down = [78, 95, 88, 178, 87, 14, 317, 402, 318, 324, 308,291,375,321,405,314,17,84,181,91,146,61,78]

*# 获取口红颜色*

b = cv2.getTrackbarPos("Blue","BGR")

g = cv2.getTrackbarPos("Green","BGR")

r = cv2.getTrackbarPos("Red","BGR")

color = (b,g,r)

*# img = change\_color\_lip(img,list\_lms,index\_lip\_up,index\_lip\_down,color)*

*# 灰度图像 彩色嘴唇*

img\_gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

img\_gray = cv2.cvtColor(img\_gray, cv2.COLOR\_GRAY2BGR)

img = change\_color\_lip(img\_gray,list\_lms,index\_lip\_up,index\_lip\_down,color)

cv2.imshow("BGR",img)

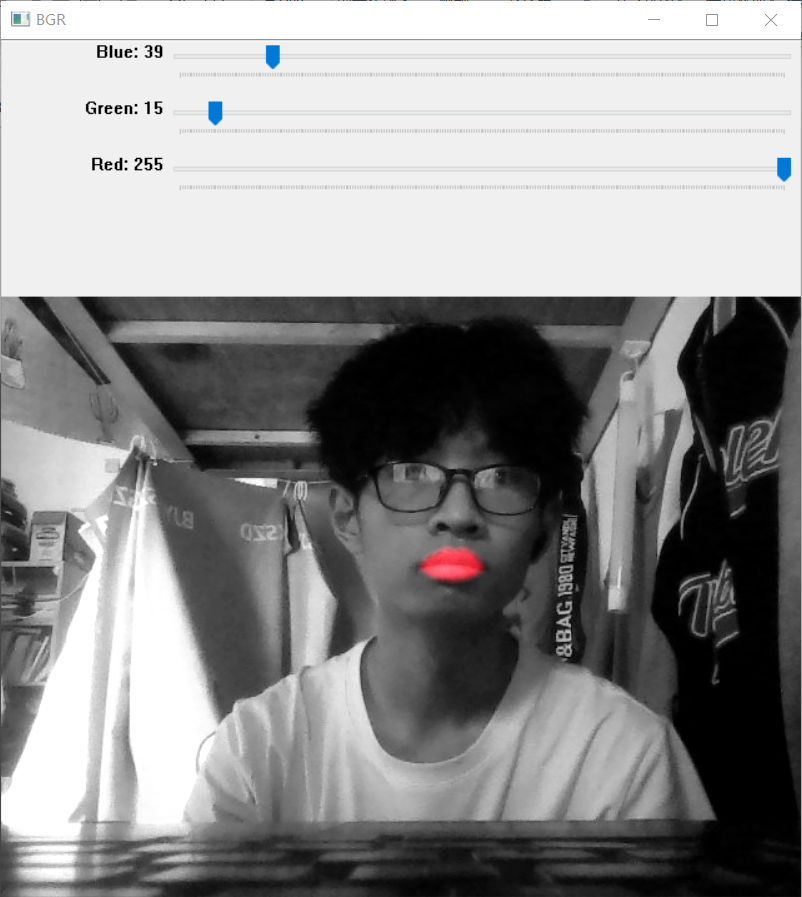
key = cv2.waitKey(1) & 0xFF

*# 按键 "q" 退出*

if key == ord('q'):

break

cap.release()



（ 4 ） 利 用 mediapipe 实 现 手 部 关 键 点 的 检 测 并 将 手 部 的 关 键 点 绘 制 出 来

（hand\_feature\_keypoints.py）

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(static\_image\_mode=False,

max\_num\_hands=2,

min\_detection\_confidence=0.5)

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:

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),end="")

*# 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)), 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()



（5）利用 mediapipe 提取的手部关键点实现简单的动作识别，能够识别 1-10 的数字手

势，以及‘good’、‘bad’等其他的自定手势。（hand\_feature.py）

import mediapipe as mp

import cv2

import numpy as np

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\_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"

elif len(up\_fingers)==3 and up\_fingers[0]==8 and up\_fingers[1]==12 and up\_fingers[2]==16:

str\_guester = "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\_guester = "7"

elif dis\_4\_12/(dis\_8\_12+1) >5:

str\_guester = "Gun"

else:

str\_guester = "7"

elif len(up\_fingers)==3 and up\_fingers[0]==4 and up\_fingers[1]==8 and up\_fingers[2]==20:

str\_guester = "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\_guester

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)

*# print(up\_fingers)*

*# print(list\_lms)*

*# print(np.shape(list\_lms))*

str\_guester = get\_str\_guester(up\_fingers,list\_lms)

cv2.putText(img,' %s'%(str\_guester),(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)

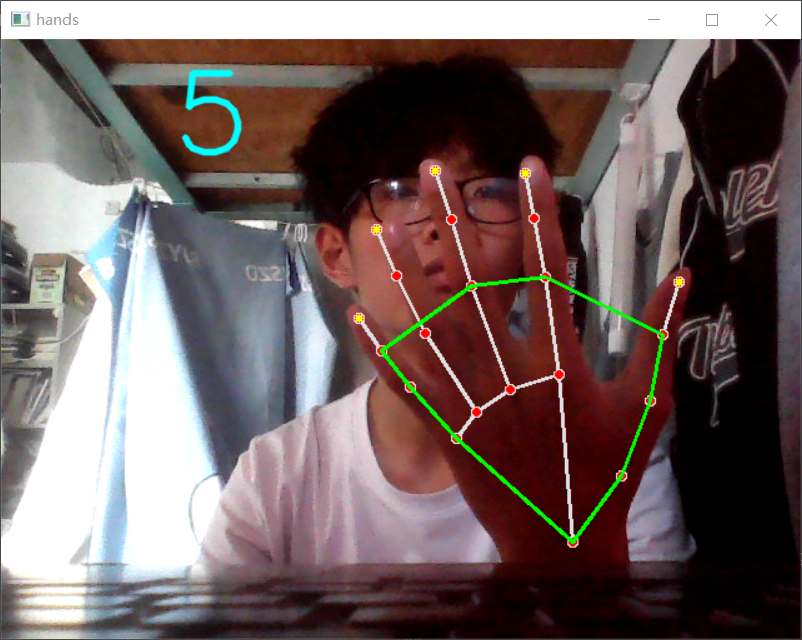
key = cv2.waitKey(1) & 0xFF

*# 按键 "q" 退出*

if key == ord('q'):

break

cap.release()



五、实验报告内容要求

1. 列出编写的 python 代码。对主要的语句进行解释。

2. 对实验结果进行截图，对结果进行必要的解释并说明实验中使用的参数。

3. 写出调试的过程，说明测试用例及调试中遇到的主要问题和解决方法。

4. 写出实验收获与不足，以及对实验的相关意见。