**Mediapipe-control部署文档**

**安装要求**

* mediapipe 0.8.1
* OpenCV 3.4.2 or Later
* Tensorflow 2.3.0 or Later  
  tf-nightly 2.5.0.dev or later (Only when creating a TFLite for an LSTM model)
* scikit-learn 0.23.2 or Later (Only if you want to display the confusion matrix)
* matplotlib 3.3.2 or Later (Only if you want to display the confusion matrix)
* PyAutoGUI 0.9.53 and PyDirectInput 1.0.4 (For Computer Control)

**安装方式**

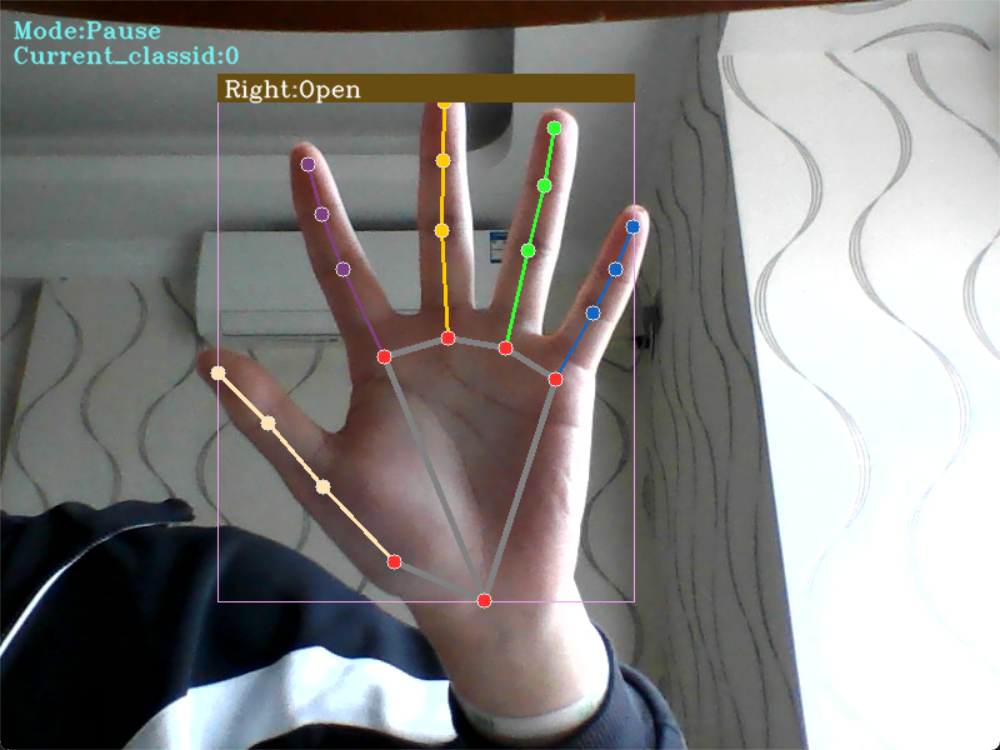
|  |
| --- |
| Bash pip install mediapipe pip install opencv-python pip install pyautogui pip install PyDirectInput pip install --upgrade tensorflow pip install -U scikit-learn pip install matplotlib |

**训练**

安装完需要的程序之后用编译器跑creat\_dataset.py或者在terminal 里跑一下这个程序

|  |
| --- |
| Bash python creat\_dataset.py |

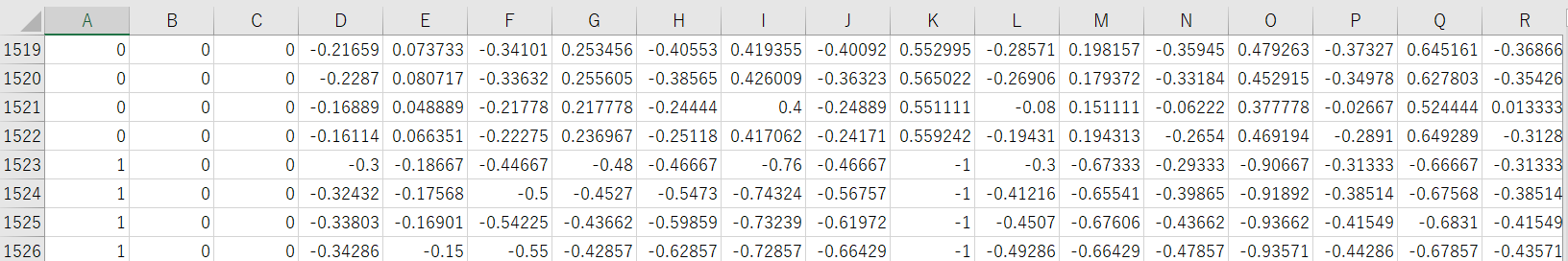
**1.输入数据**



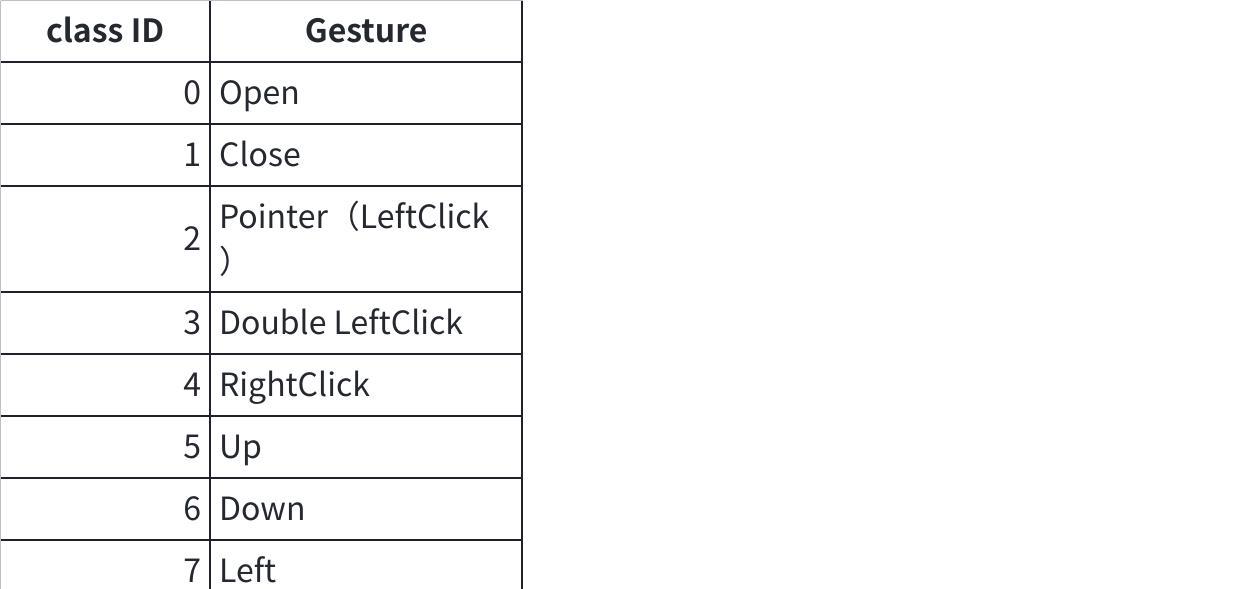
按 “0” 到 “9” 会把当前的手势节点信息(keypoints) 调整需要录入的对应的class\_id 。按c 会把当前的手势节点信息(keypoints) 录入到对应的class\_id这些信息会加入到"model/gesture\_classifier/keypoint.csv" 里面。 如果你要加个新的手势，想录入到class\_id 9这个位置。 你可以先把想录入的手势摆出来，然后按9 进入id为9的手势录入模式，然后做出想录入的手势 -> 按 c -> 新的位置 -> 做出想录入的手势 -> 按 c -> 循环。

新的手势添加完之后记得在 "model/gesture\_classifier/gesture\_classifier\_label.csv" 里面把这个手势的标签也给加上。

在"model/gesture\_classifier/keypoint.csv" 里面， 第一列对应的就是class\_id 0-9，后面的都是手势节点信息



在这个版本里面, class\_id 0-8 都被占用了。可以自行更改添加或者删除， 后面有说。如果要在这个版本上面添加的话可以从9 开始加起。



**点击图片可查看完整电子表格**

**2.模型训练**

用Jupyter Notebook打开 "train\_model.ipynb" 如果需要改一下手势标签的话请自行在 "model/gesture\_classifier/gesture\_classifier\_label.csv" 里面进行更改  
在train\_model.ipynb 文件里面根据要训练的手势数量改一下变量 "NUM\_CLASSES = 9"

或者在train\_model.py文件里改变量 "NUM\_CLASSES = 9"再运行也是同样的效果。

**3.删除手势**

所有训练用的手势节点信息都存在这里"keypoint.csv". 先进入以上的这个文件，删除掉目标的class\_id的讯息。也可以利用remove\_gesture进行删除，可以跑一下这行代码，但是class\_id 要换成你想删除的手势的class\_id。

|  |
| --- |
| Bash python .\remove\_gesture.py -i class\_id |

删除完训练用的手势节点，我们需要重新训练一个新的模型。旧模型的label 不要删 不过你可以把label 改成NULL 或者直接无视掉也行。

**4.更改或者添加新的手势指令**

首先你可以根据步骤1 - 2添加一个新的手势 手势添加成功之后，打开main\_control.py， 然后你可以更改290行的内容。 你可以把'UP' 换成 'a' 或者把hand\_sign\_id (也就是class\_id) 改成别的手势。

|  |
| --- |
| Python if is\_left\_hand: # 判定为左手  # Press UP  if hand\_sign\_id == 5 and pre\_gesture != 5:  pydirectinput.press('UP')  # Press DOWN  elif hand\_sign\_id == 6 and pre\_gesture != 6:  pydirectinput.press('DOWN')  # Press LEFT  elif hand\_sign\_id == 7 and pre\_gesture != 7:  pydirectinput.press('LEFT')  # Press RIGHT  elif hand\_sign\_id == 8 and pre\_gesture != 8:  pydirectinput.press('RIGHT') # 判定为右手 else:  # 鼠标移动  if hand\_sign\_id == 0:  pydirectinput.moveTo(x, y - 200, duration=0.1, \_pause=False)  pass  # 持续点击左键  elif hand\_sign\_id == 1:  pydirectinput.click(button='left')  # 点击左键  elif hand\_sign\_id == 2:  pydirectinput.click(button='left')  # 防止过快点击  time.sleep(1)  # 双击左键  elif hand\_sign\_id == 3:  pydirectinput.doubleClick(button='left')  # 防止过快点击  time.sleep(1)  # 点击右键  elif hand\_sign\_id == 4:  pydirectinput.click(button='right')  # 防止过快点击  time.sleep(1) |

**测试**

运行文件main\_control.py。

|  |
| --- |
| Bash python main\_control.py |

文件中附带2048的网页版游戏文件，运行main\_control.py文件会直接打开2048的网页版游戏，或者可以在编译器中运行index.html即可打开2048小游戏。

**手势控制**

右手的控制中：open控制鼠标的光标移动close控制持续点击鼠标左键，pointer单击鼠标左键，LeftClick双击鼠标左键，RightClick单击鼠标右键。

左手的控制中：Up控制键盘的“↑”方向键的输入，Down控制键盘的“↓”方向键的输入，Left控制键盘“←”方向键的输入，Right控制键盘“→”方向键的输入。