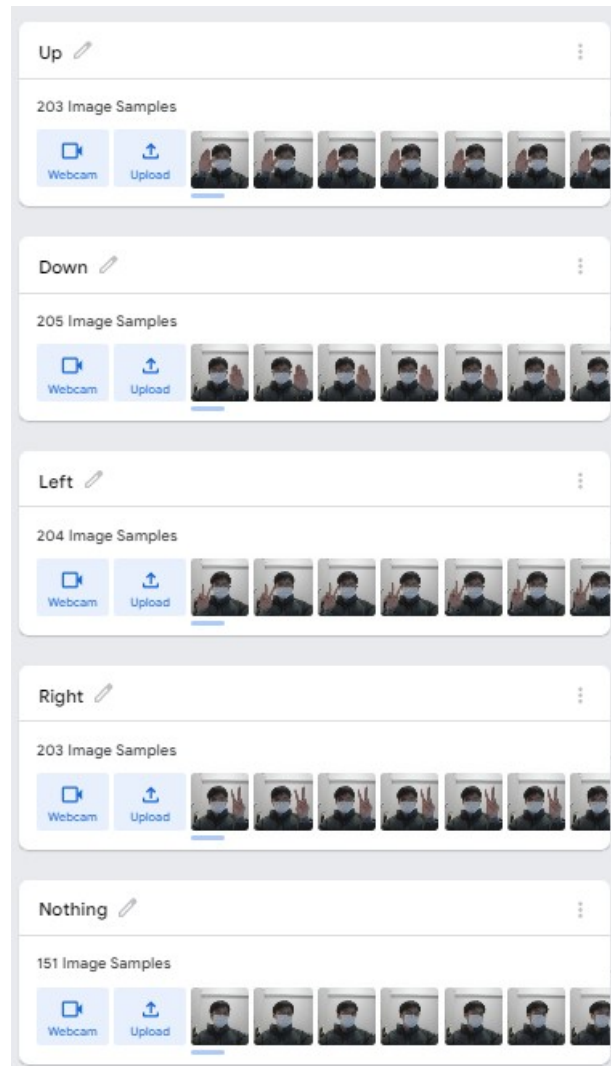


基于TeachableMachine的分类识别模型控制鼠标实验

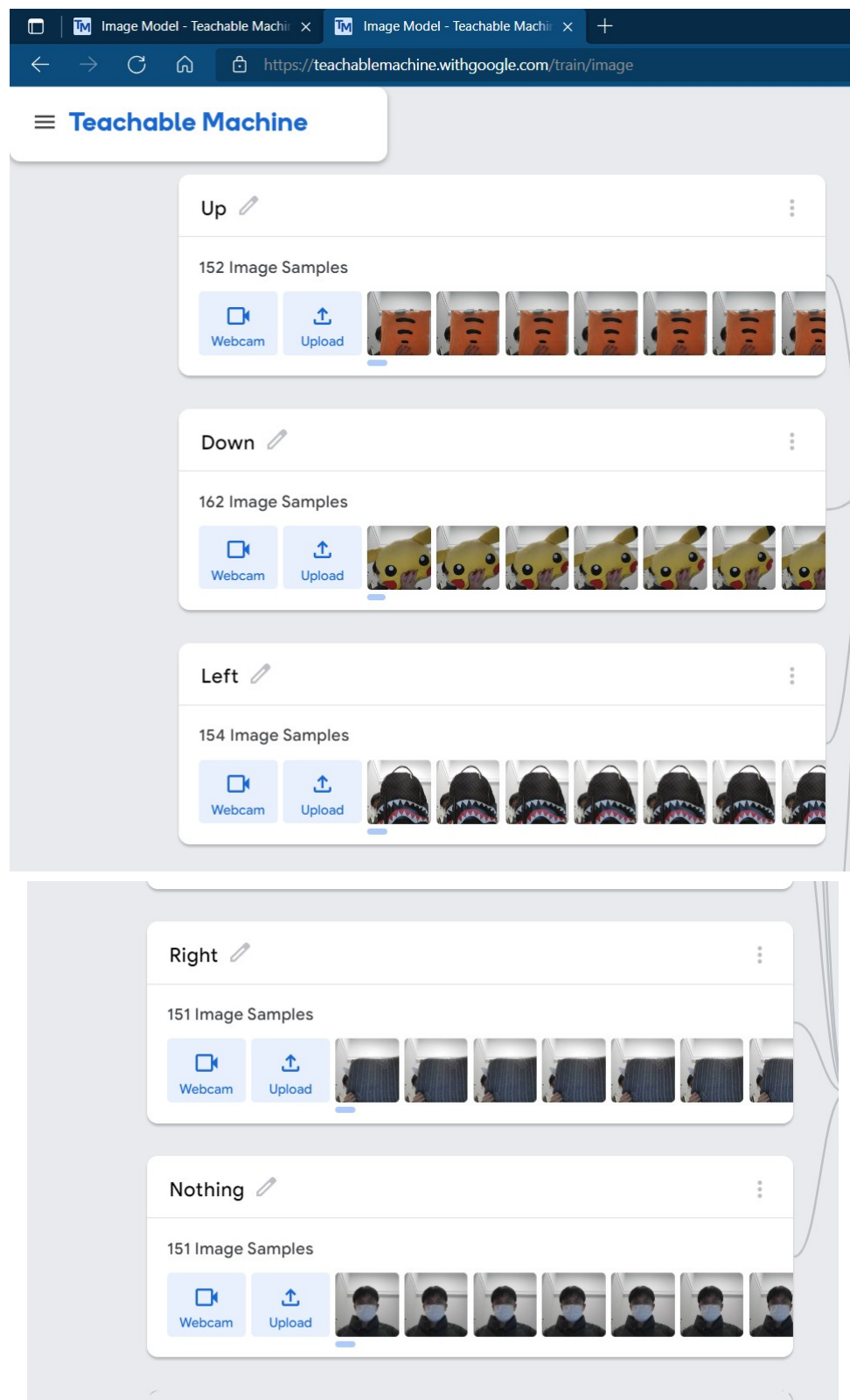
(一) 实现说明

1. 首先在google的teachable machine网站上训练模型
 - 起初打算使用识别人的"手势", 代表控制鼠标的上下左右

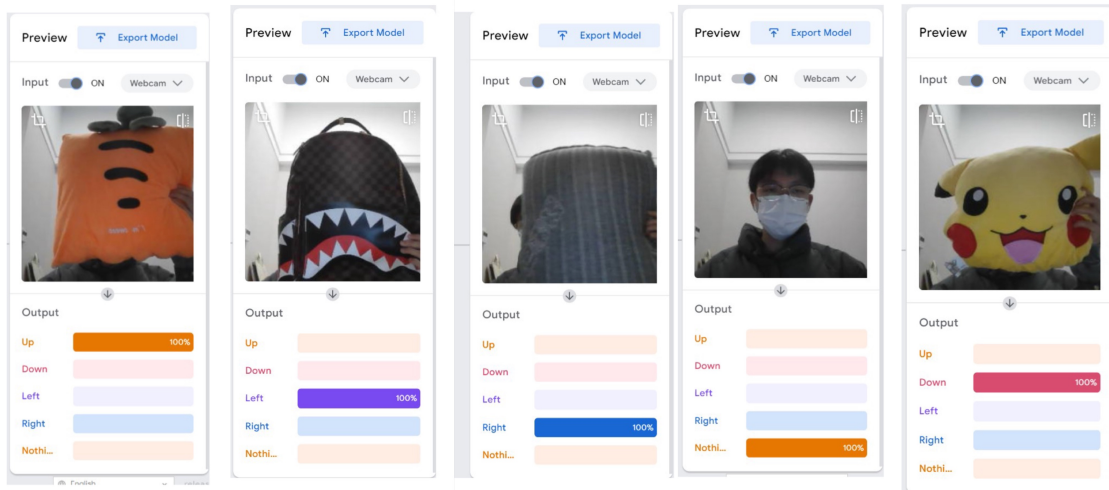


但是当h5模型下载迁移至本地运行后，识别效果不理想，故放弃该识别分类想法。猜测可能由于teachable machine采用的模型效果不佳，导致只能做简单的分类识别，区分度较低的识别存在较大障碍；也可能由于数据集过少，导致识别效果不佳等。

- 经过几次调试后，选择采用识别区分度较大的物品来进行上下左右的判别



2. 模型训练完成后，使用在线工具preview判断模型识别效果



结果较好，因此导出该模型，在本地运行

3. 导出基于tensorflow.keras的模型.h5

Export your model to use it in projects. ✕

Tensorflow.js i **Tensorflow** i Tensorflow Lite i

Model conversion type:

☒ Keras ☐ Savedmodel [Download my model](#)

Converts your model to a keras .h5 model. Note the conversion happens in the cloud, but your training data is not being uploaded, only your trained model.

Code snippets to use your model:

[Keras](#) Contribute on Github

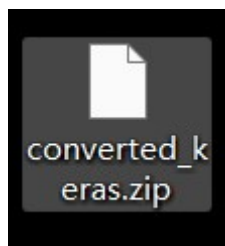
```
from keras.models import load_model
from PIL import Image, ImageOps
import numpy as np

# Load the model
model = load_model('keras_model.h5')

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1.
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)
# Replace this with the path to your image
image = Image.open('<IMAGE_PATH>')
#resize the image to a 224x224 with the same strategy as in TM2:
#resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.ANTIALIAS)
```

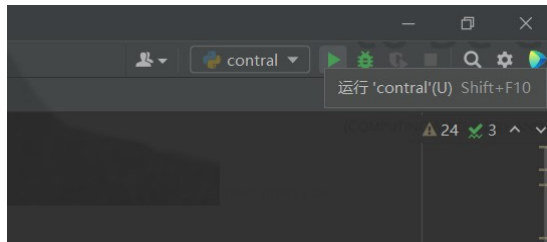
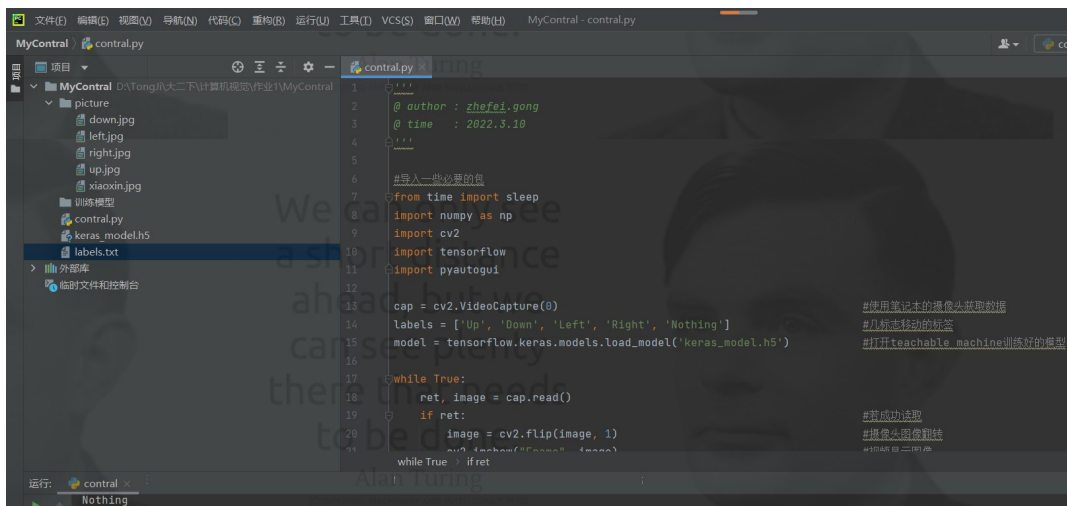
Copy

- 得到含模型的压缩包

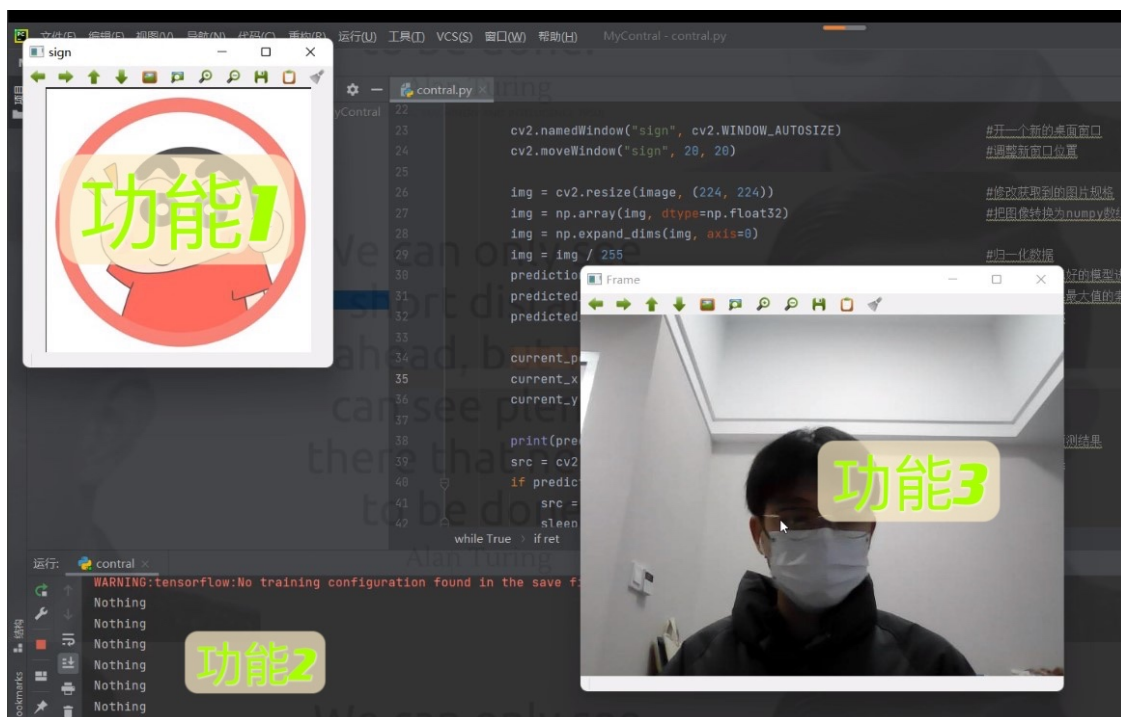


(二) 运行说明

- 在本地python环境中运行(具体环境信息见README.md文件)



- 运行界面如下：

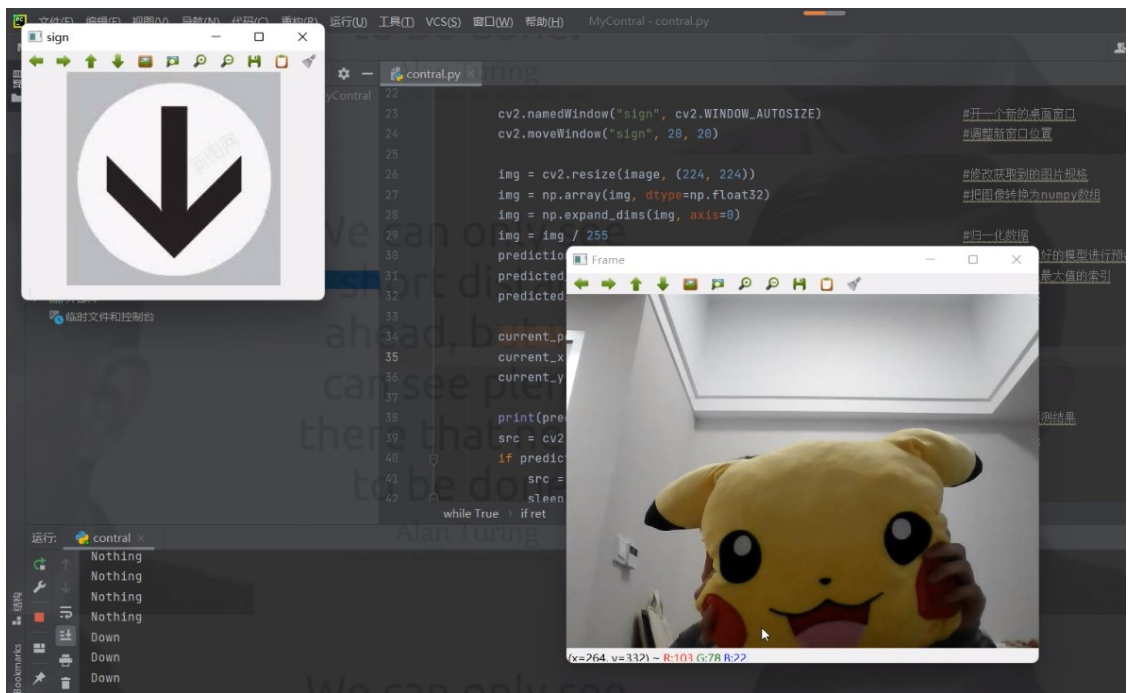


- 功能1：根据识别结果更换图像信息
- 功能2：在终端窗口打印结果信息(Up、Down、Left、Right、Nothing)
- 功能3：显示摄像头采集的图像信息，并且鼠标随着识别结果而移动

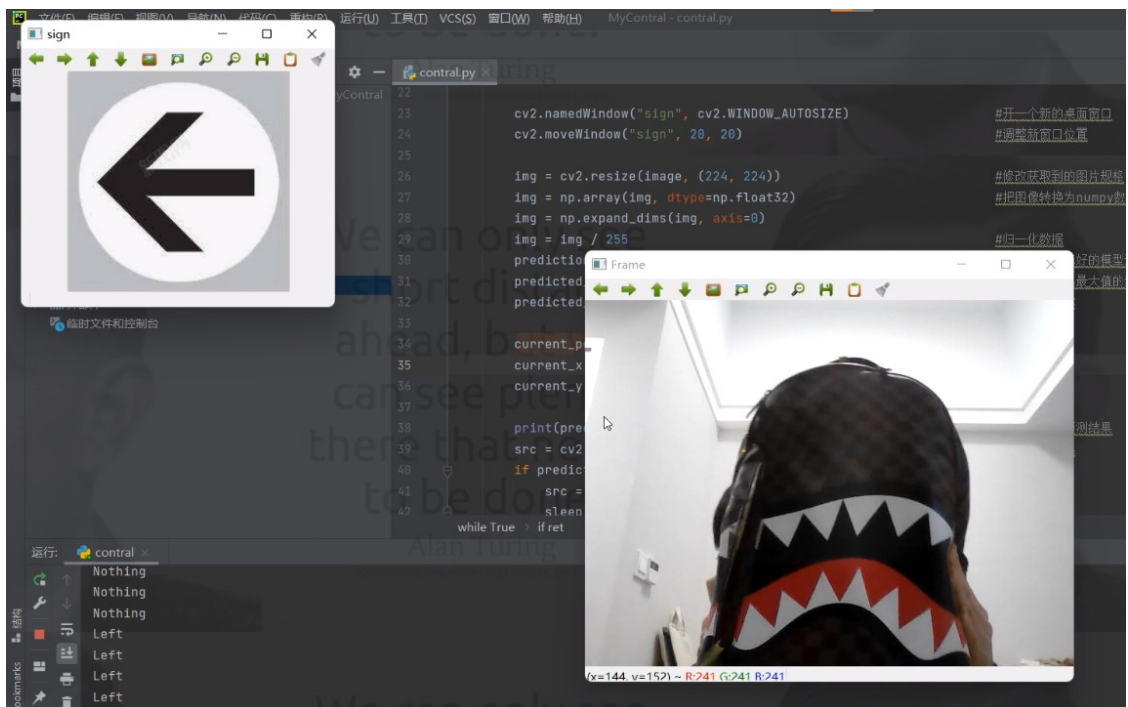
- 按下"q"键，结束识别过程

(三) 运行结果

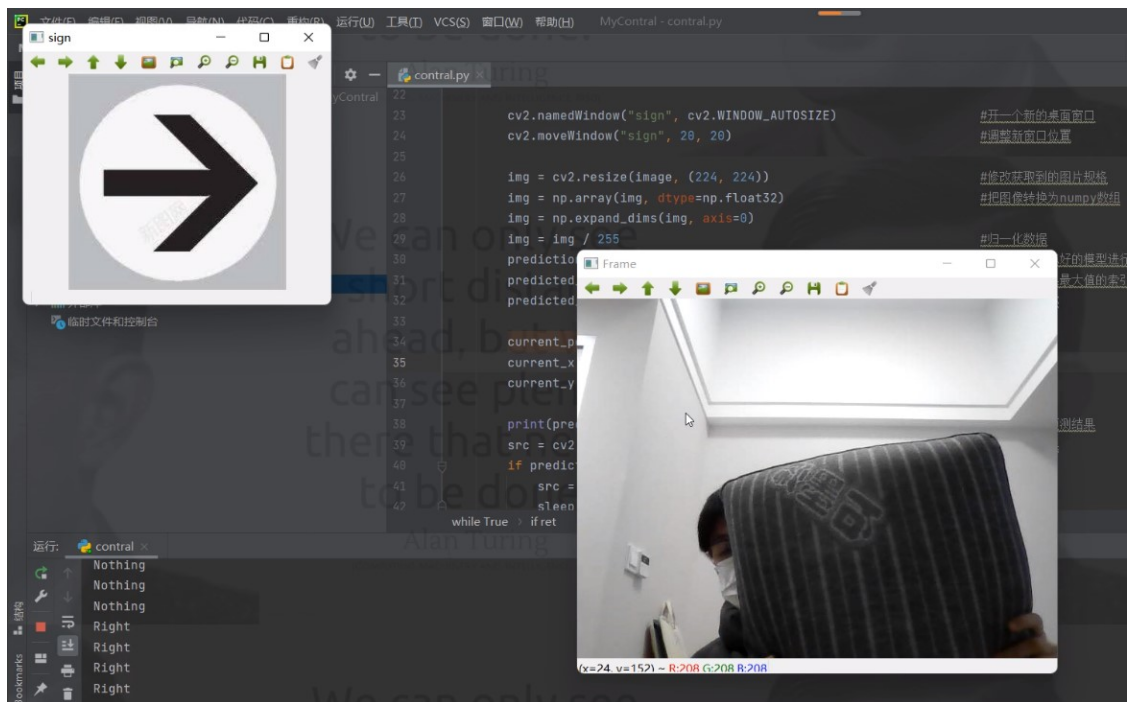
- 无指令(鼠标不动)--Nothing:



- 鼠标左移--Left:



- 鼠标右移--Right:



(四) 实验代码(python)

```
'''
@ author : zhefei.gong
@ time   : 2022.3.10
'''

#导入一些必要的包
from time import sleep
import numpy as np
import cv2
import tensorflow
import pyautogui

cap = cv2.VideoCapture(0)                                #使用笔记本
                                                         #的摄像头获取数据
labels = ['Up', 'Down', 'Left', 'Right', 'Nothing']      #几标志移动
                                                         #的标签
model = tensorflow.keras.models.load_model('keras_model.h5') #打开
teachable machine训练好的模型

while True:
    ret, image = cap.read()
    if ret:                                                #若成功读取
                                                            #摄像头图像
        image = cv2.flip(image, 1)
        cv2.imshow("Frame", image)                        #视频显示图
                                                           #像
        cv2.namedWindow("sign", cv2.WINDOW_AUTOSIZE)     #开一个新的
桌面窗口
        cv2.moveWindow("sign", 20, 20)                   #调整新窗口
位置
        img = cv2.resize(image, (224, 224))              #修改获取到
的图片规格
```

```

img = np.array(img, dtype=np.float32) #把图像转换为numpy数组
img = np.expand_dims(img, axis=0)
img = img / 255 #归一化数据
prediction = model.predict(img) #调用已经训练好的模型进行预测
predicted_class = np.argmax(prediction[0], axis=-1) #返回预测结果最大值的索引
predicted_class_name = labels[predicted_class] #获取结果名称

current_pos = pyautogui.position() #调用鼠标
current_x = current_pos.x
current_y = current_pos.y

print(predicted_class_name) #命令行打印预测结果
src = cv2.imread("picture/xiaoxin.jpg") #图片识别结果

if predicted_class_name == 'Nothing':
    src = cv2.imread("picture/xiaoxin.jpg")
    sleep(1)
elif predicted_class_name == 'Left':
    src = cv2.imread("picture/left.jpg")
    pyautogui.moveTo(current_x - 60, current_y, duration=1)
    sleep(1)
elif predicted_class_name == 'Right':
    src = cv2.imread("picture/right.jpg")
    pyautogui.moveTo(current_x + 60, current_y, duration=1)
    sleep(1)
elif predicted_class_name == 'Down':
    src = cv2.imread("picture/down.jpg")
    pyautogui.moveTo(current_x, current_y + 60, duration=1)
    sleep(1)
elif predicted_class_name == 'Up':
    src = cv2.imread("picture/up.jpg")
    pyautogui.moveTo(current_x, current_y - 60, duration=1)
    sleep(1)
cv2.imshow("sign", src) #显示对应图片

if cv2.waitKey(1) & 0xFF == ord('q'): #当下‘q’时停止显示q
    break

cap.release() #关闭连接
cv2.destroyAllWindows()

```