**实验报告**

课程名称：边缘计算开发实践 老师：皇甫江涛

实验名称： CV算法的手势识别娱乐游戏 同组学生姓名：陈雨过，陆天，钱煜，吴东升

一、实验内容与实验原理 二、实验软硬件环境

三、实验操作步骤 四、实验中遇到的问题与解决方法

五、实验结果 六、个人感想

装 订 线

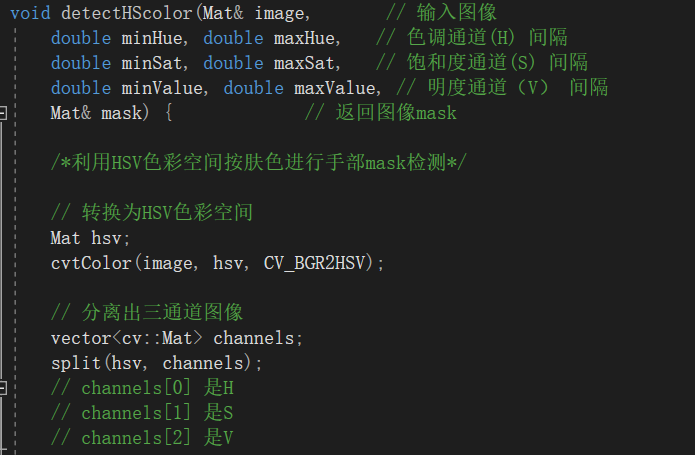
**一、实验内容与实验原理**

**实验内容：**本项目核心内容为通过计算机视觉相关算法，实现对石头，剪刀，布三种手势的识别与输赢判定，并完成了软硬件结合的边缘计算系统。

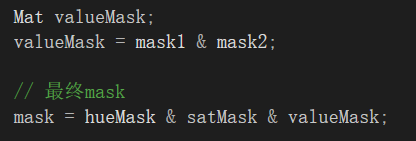
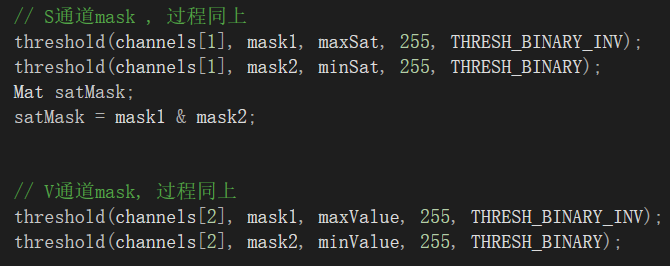
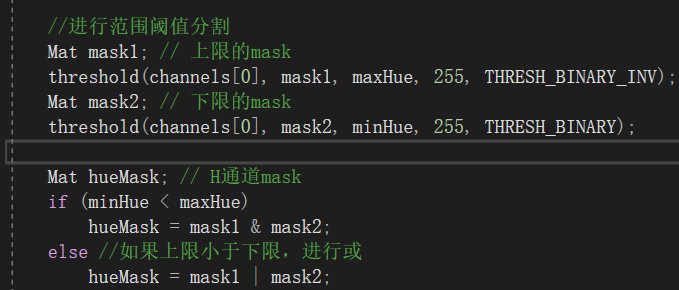
**实验内容原理：**

1.图像的转化

1）将摄像头捕捉到的RGB图像转化到HSV色彩空间；



2）对每个通道各自做阈值分割，再综合（hueMask & SatMask & ValueMask）；



2.手势的识别

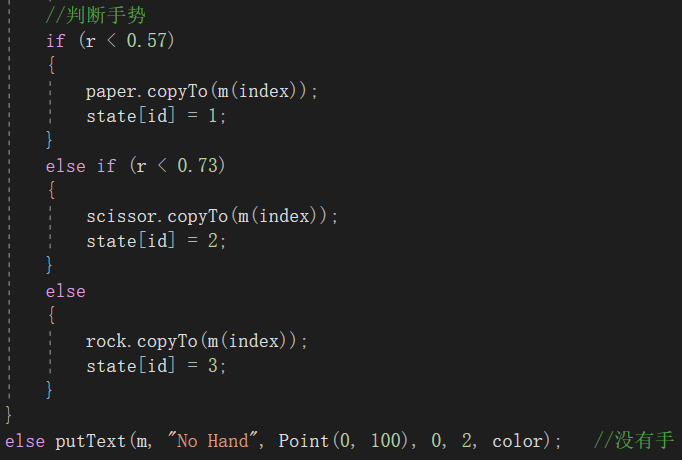
1）通过find contours函数得到手的外轮廓；



2）通过convexhull得到手的凸包；



3）计算外轮廓周长和凸包周长的比例，判断是哪个手势。



**二、实验软硬件环境**

**1.软件环境**

软件环境: 1. 依赖库Opencv

2. 编程语言C++

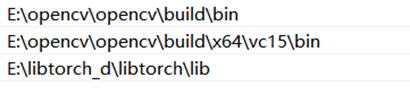
软件环境搭建过程：

1. **PC端Opencv环境搭建。**

为了方便程序的开发和调试需要在PC段搭建Opencv开发环境。

主要在VS2019上配置opencv运行环境。

**第一步**：在官网下载windows系统下编译完成opencv文件，下载完成后解压，并根据相应文件路径配置系统环境变量Path：



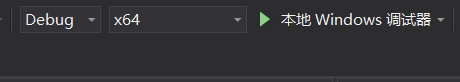
**第二步**：VS2019中opencv运行环境配置。

安装完成后，这时还无法直接调用opencv库文件，需要进一步配置。

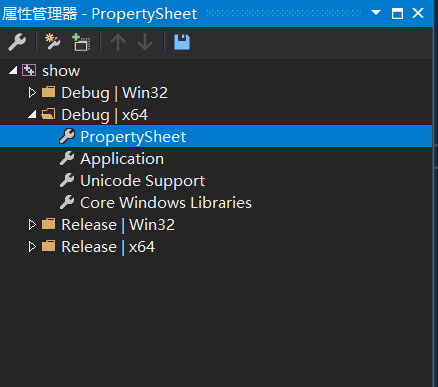
首先创建一个空的VS2019 C++项目。



进入界面后，选择调试配置为Debug模式， 64位：



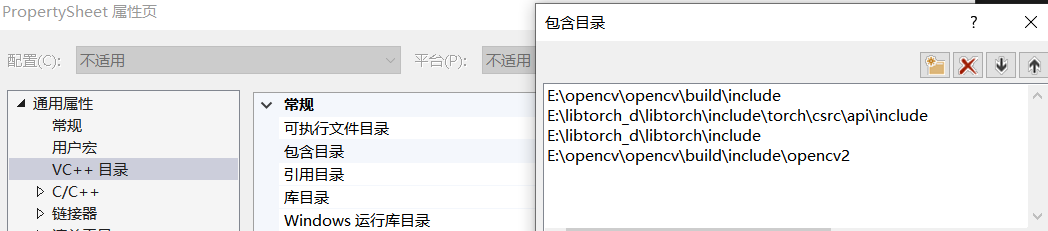
然后依次点击打开视图🡪其他窗口🡪属性管理器，显示出属性管理器界面。



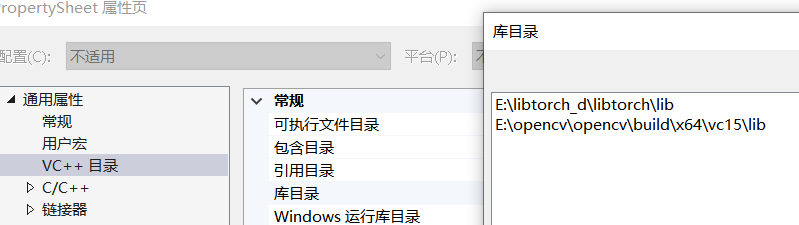
如上图所示在Debug|x64栏下新增一个属性表，进行配置，保存的属性表也可以重复使用。

双击打开新建的属性表，开始配置。

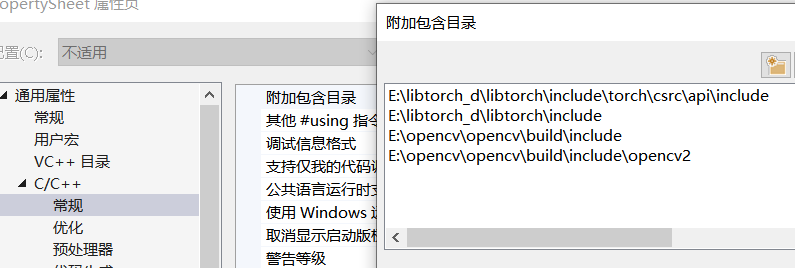
首先在VC++目录栏配置包含目录：

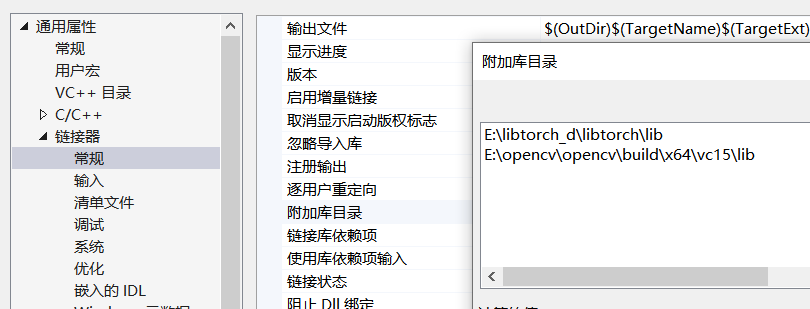


这里包含的目录主要是opencv文件中的include文件路径。

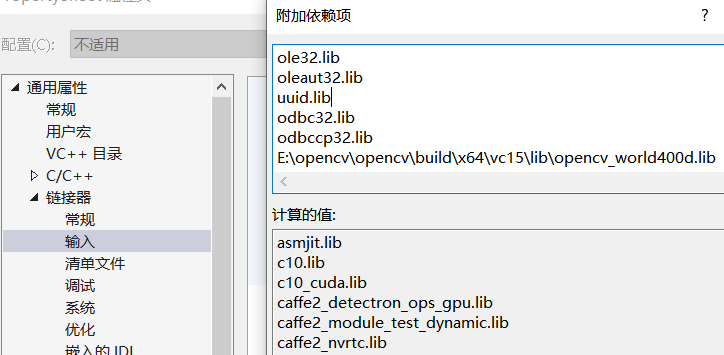
然后配置库目录：  


增加的库目录是opencv文件中的lib文件路径。Opencv中有两个lib，vc14和vc15，这里选择vc15下的lib。

在C++栏下的常规栏中配置附加包含目录：  


最后是链接器栏下的常规栏中添加附加包含目录：  


在输入栏中的附加依赖项中添加需要的.lib文件。opencv需要的是opencv\_world400d.lib文件。



完成以上步骤之后，就在VS2019中完了opencv的运行环境配置，可以进行实验相关程序的开发和调试了。

**（2）边缘计算小主机中Opencv配置。**

由于最终的功能需要在边缘计算小主机上运行，因此也需要在小主机上的Linux系统中配置opencv环境

**第一步：**安装依赖库

sudo apt-get install python-dev python-numpy

sudo apt-get install libgtk2.0-dev libavcodec-dev libavformat-dev libswscale-dev

sudo apt-get install libtbb2 libtbb-dev libjpeg-dev

sudo apt-get install libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev

sudo apt-get install libv4l-dev liblapacke-dev

sudo apt-get install libxvidcore-dev libx264-dev

sudo apt-get install libatlas-base-dev gfortran

sudo apt-get install ffmpeg

**第二步：**下载opencv源码后编译

mkdir build  
cd build

cmake -D CMAKE\_INSTALL\_PREFIX=/usr/local -D CMAKE\_BUILD\_TYPE=Release -D OPENCV\_EXTRA\_MODULES\_PATH=../opencv

make-j8

sudo make install

**第三步：**配置环境变量

sudo /bin/bash -c 'echo "/usr/local/lib" > /etc/ld.so.conf.d/opencv.conf'

sudo ldconfig

sudo gedit /etc/bash.bashrc

末尾加入PKG\_CONFIG\_PATH=$PKG\_CONFIG\_PATH:/usr/local/lib/pkgconfig export PKG\_CONFIG\_PATH后退出

source /etc/bash.bashrc

就此完成opencv编译与安装。

**2.硬件环境**

硬件环境：1.边缘计算小主机

2.摄像头支架

3.摄像头

4.黑色底板

5.显示屏

6.开发和调试用的PC

硬件环境搭建过程：

将摄像头悬挂于支架上并对准黑色底板，摄像头、小主机、显示屏按照规范连接后通电，运行程序时将手放在黑色底板上之后便可开始游戏。

硬件环境及运行效果整体：



**三、实验操作步骤**

1. 按照实际需求摆放好摄像头\底板等功能硬件。

2. 正确连接设施设备并通电。

3. 在机箱上运行已经下载好的代码程序。

4. 处理摄像头对焦,距离手部远近等问题

5. 进行使用。

**四、实验中遇到的问题与解决办法**

1手与地面之前前后景判别较为困难

解决方案：

我们小组尝试了多种前后景算法，每种方法各有优缺点。

KNN：将样本在特征空间中相近的K个点作为一类。其思路简单，无需估计参数或者是训练，但是计算量比较大，不合适在空间有限的边缘计算设备上运行。

MOG：对每个像素点指定正态分布值，对周围做拟合，符合则纳入到这一正态中，初期我们采用了这一算法，但是这一算法对于光照的较为敏感，环境适应性较差。

VIBE：主要是利用单帧视频序列初始化背景模型，然后将每一个新的像素值和样本集进行比较来判断是否属于背景点。这一算法需要一定的时间来做背景建模，同时主要适合运动物体，且容易收到手部阴影的影响。

颜色识别：这是我们小组最后采用的算法，基于纯色底板与手的颜色差值来识别，对于静态环境效果较好。

2三种手势识别存在困难

解决方案：

经过讨论，我们小组决定采用传统算法来实现手势识别而不是机器学习，神经网络等高级算法，这样获得的程序运行速度快，容量小，可以满足边缘计算开发设备的实时性要求。

在识别参数方面，我们小组考虑了手的面积，手的周长，凸包周长，凸包面积，凸包面积与手部面积之差等多种参数，但是效果都不是很好，最终选择的参数是凸包周长与手的周长之比这一无量纲参数，这样获得的识别效果最好。

**五、实验结果**

实现了可以识别三种手势的边缘计算系统，得到了较高的识别率与较好的实时性。

**六、个人感想**

**吴东升**

心得:

课程提供了学习边缘计算相关知识的契机。在导师的引导下一步步解决问题，最终实现预定的实验目标，锻炼了解决问题的能力。同时通过在公司的交流学习了更多知识在实际行业中的应用，有了更多实际感性认知，增加了学习动力。

课程建议：增进和公司的交流，学习更多实际商业中解决问题方法和应用知识。

**陆天**

心得:

1. IoT相关软件在编写时要考虑硬件的性能限制

2. 感受到了产业界的实际环境

建议:​环境搭建&设备安装调试应该更加迅速

**钱煜**

心得：

1.在建立起边缘计算的概念上，初步尝试做出了适用于此的产品，很有成就感；

2.见到了公司的工作环境，感觉对未来更加憧憬了；

3.认识到了自己所学知识的不足（至少在应用层面还很欠缺，更多的只是学到了基础知识）。

建议：

1.课程开始时更早地安排与导师对接；

**陈雨过**

心得：

通过课程，首先是对于计算机语言尤其是CV算法有了更深的认识，同时与公司导师合作也对于实际的边缘计算开发流程有了更深的认识，是一次非常有意义的体验。

建议：

希望可以在第一次课程之后尽快确定分组，第二次课即可前往公司和导师谈话