



# Cpp Project

## Hand Recognition Based on OpenCV

汤星宇 11912217

王烨祥 12012529

唐麟杰 12111204

# OpenCV

- OpenCV是一个基于Apache2.0许可（开源）发行的跨平台计算机视觉和机器学习软件库，可以运行在Linux、Windows、Android和Mac OS操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类构成，同时提供了Python、Ruby、MATLAB等语言的接口，实现了图像处理和计算机视觉方面的很多通用算法。
- OpenCV用C++语言编写，它具有C++, Python, Java和 MATLAB接口，并支持Windows, Linux, Android和Mac OS, OpenCV主要倾向于实时视觉应用，并在可用时利用MMX和SSE指令，如今也提供对于C#、Ch、Ruby, GO的支持。



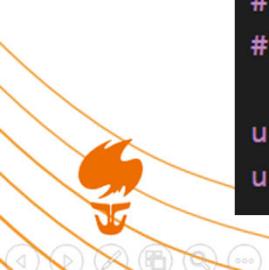
# OpenCV

- 在project中使用OpenCV的库和接口

```
#include <iostream>           // for standard I/O
#include <string>             // for strings
#include <iomanip>            // for controlling float print precision
#include <sstream>             // string to number conversion
#include <opencv2/imgproc/imgproc.hpp> // Gaussian Blur
#include <opencv2/core/core.hpp>      // Basic OpenCV structures (cv::Mat, Scalar)
#include <opencv2/highgui/highgui.hpp> // OpenCV window I/O
#include <opencv2/imgproc/imgproc_c.h>

#include <opencv2\video\background_segm.hpp>
#include <opencv2\opencv.hpp>
#include "opencv2/objdetect/objdetect.hpp"
#include <math.h>
#include <ctime>
#include <Windows.h>

using namespace cv;
using namespace std;
```



# Hand Recognition

- 使用 .xml 文件和相应的OpenCV接口函数加载已经训练的模型，进行手势识别和人脸识别

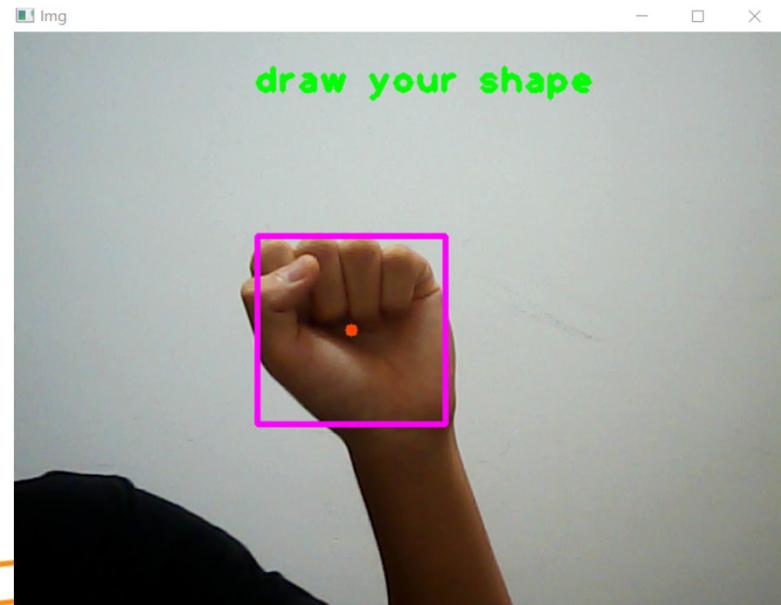
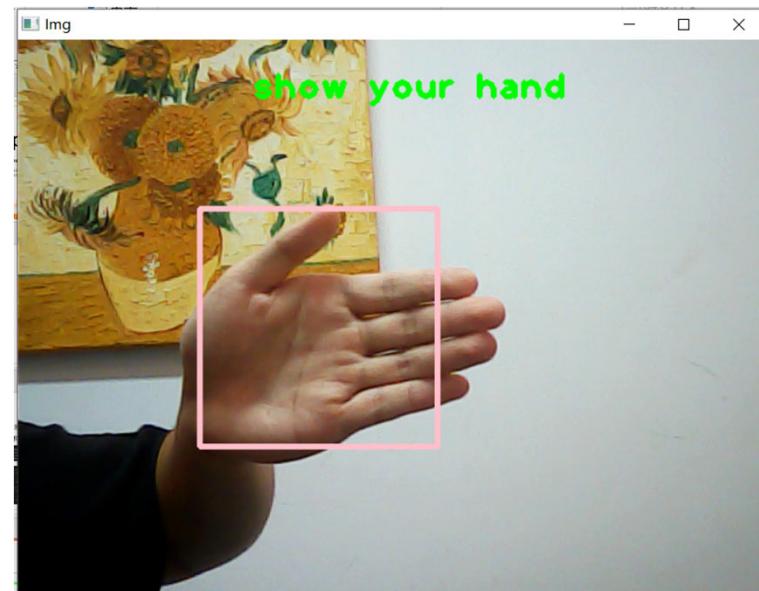
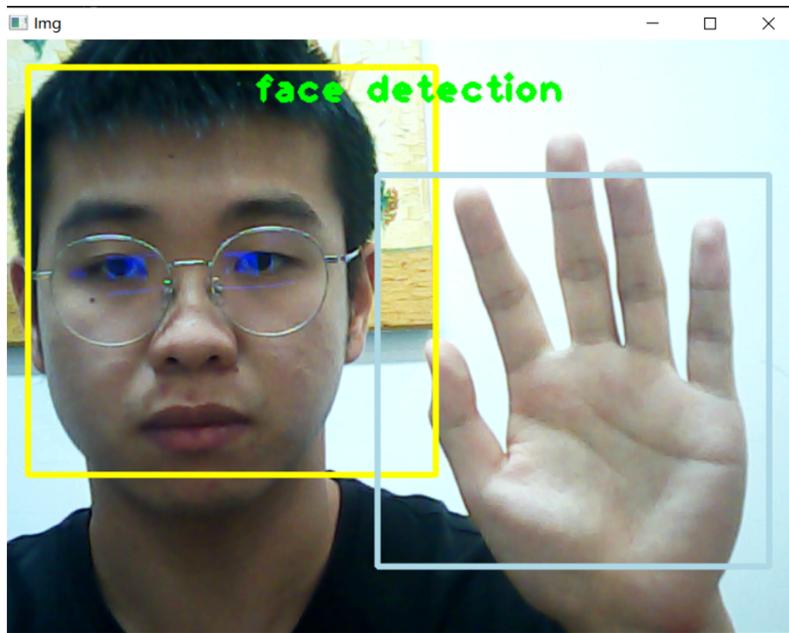
```
✓ own_project
  ✓ face_photos
    fist.xml
    haarcascade_frontalface_alt.xml
  ↗ main.cpp
  ↗ project.cpp
  ↗ recognition.cpp
    right.xml
    rpalm.xml
  ↗ test.cpp
```

```
Fist_Cascade.detectMultiScale(img, fists, 1.1, 2, 0 | CASCADE_SCALE_IMAGE, cv::Size(30, 30)); //识别拳头
Rdirection_Cascade.detectMultiScale(img, rdirections, 1.1, 2, 0 | CASCADE_SCALE_IMAGE, cv::Size(30, 30)); //识别向右
Rpalm_Cascade.detectMultiScale(img, rpalms, 1.1, 2, 0 | CASCADE_SCALE_IMAGE, cv::Size(30, 30)); //识别右掌
Face_Cascade.detectMultiScale(img, faces, 1.1, 2, 0 | CASCADE_SCALE_IMAGE, cv::Size(30, 30)); //识别人脸
```

```
//识别人脸
for (int i = 0; i < faces.size(); i++)
{
    //识别人脸 用黄色长方形框起来
    rectangle(img, Point(faces[i].tl().x - 50, faces[i].tl().y - 50), faces[i].br(), Scalar(0, 255, 255), 3);
}
//识别右手掌
for (int i = 0; i < rpalms.size(); i++)
{
    //识别右手掌 用蓝色长方形框起来
    rectangle(img, rpalms[i].tl(), rpalms[i].br(), Scalar(230, 216, 173), 3);
}
```

# Affect

- 三种手势+人脸



# Structure

- 使用全局变量 mode 来进行不同功能模块之间的跳转

多种模式

```
mode = 0 主界面  
mode = 1 画板  
mode = 2 识别图像  
mode = 3 自拍 人脸截图  
mode = 4 鼠标动作
```

mode 0 主界面  
三种模式  
向右 进入画板画图 ->mode 1  
拳头 进入人脸识别 ->mode 3  
手掌 进入鼠标动作 ->mode 4

mode 1 画板  
画轨迹  
向右 进入识别图像 ->mode 2  
拳头 画轨迹  
手掌 清空画板 重新画

mode 2 识别图像  
将画出的轨迹拟合 估计形状并输出结果  
向右 返回主界面 ->mode 0  
拳头 将结果加载入队列 用于鼠标动作  
手掌 返回mode 1 不满意重新画 / 继续添加结果载入队列

识别向右来切换模式 默认顺序为mode 0 -> mode 1 -> mode 2

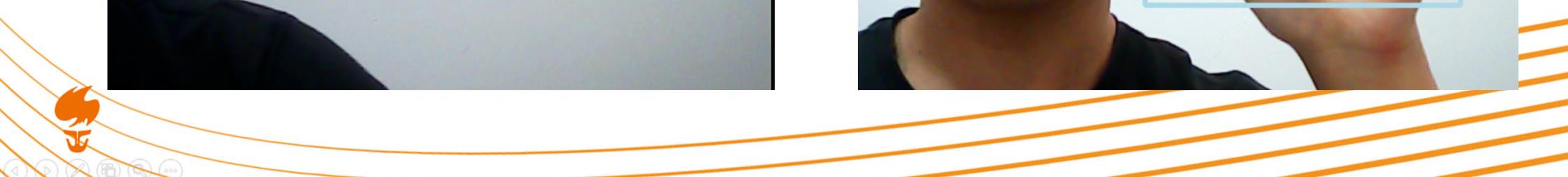
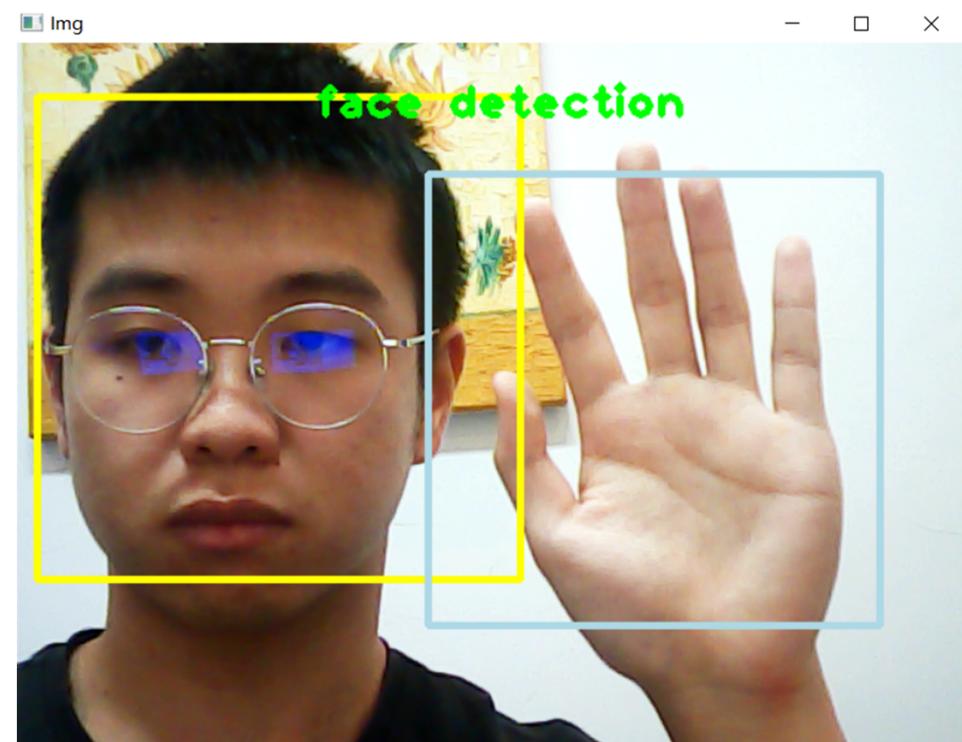
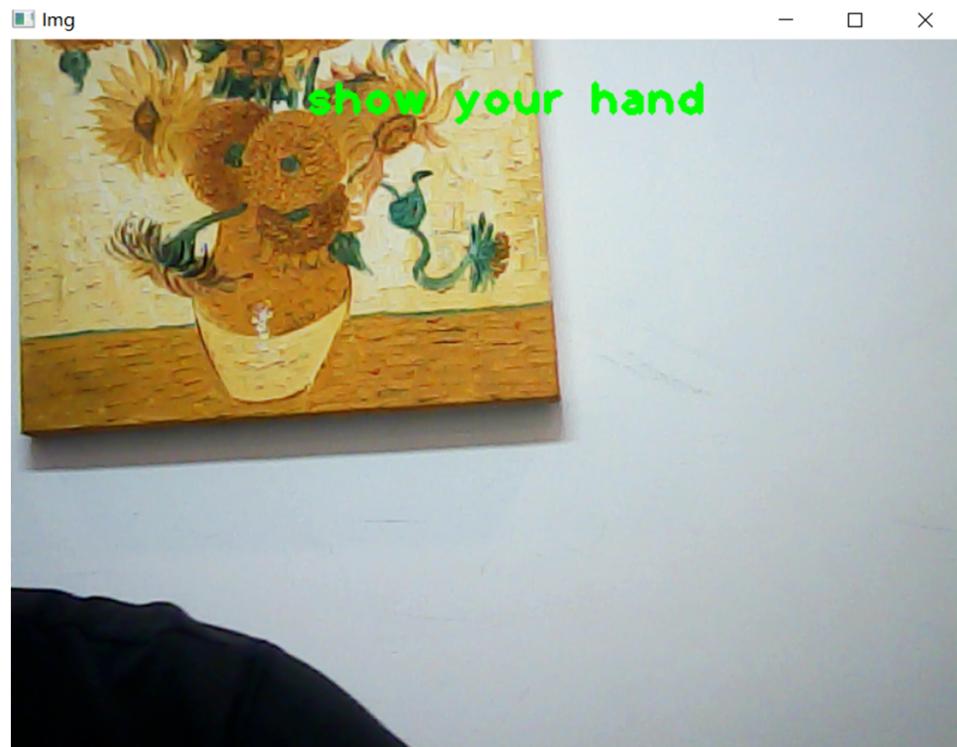
mode 3 自拍 人脸识别  
识别人脸然后可以选择截图  
向右 返回主界面 ->mode 0  
右手掌 将脸部截图保存至文件夹

mode 4 光标动作

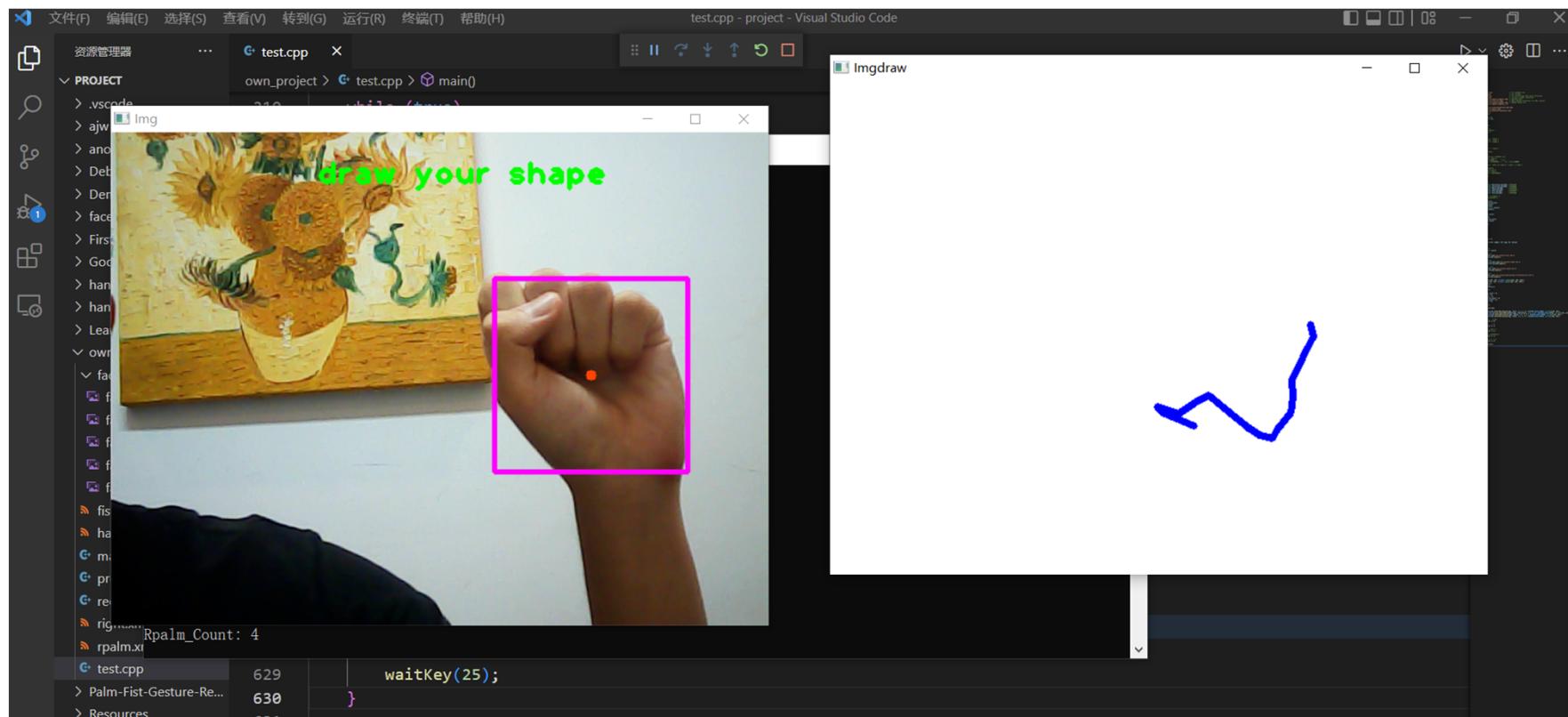
```
//模式0 开始界面  
if (mode == 0) ...  
  
//模式1 画轨迹  
if (mode == 1) ...  
  
//模式2 识别图像  
if (mode == 2) ...  
  
//模式3 人脸识别自拍  
if (mode == 3) ...  
  
//模式4 光标动作  
if (mode == 4) ...
```



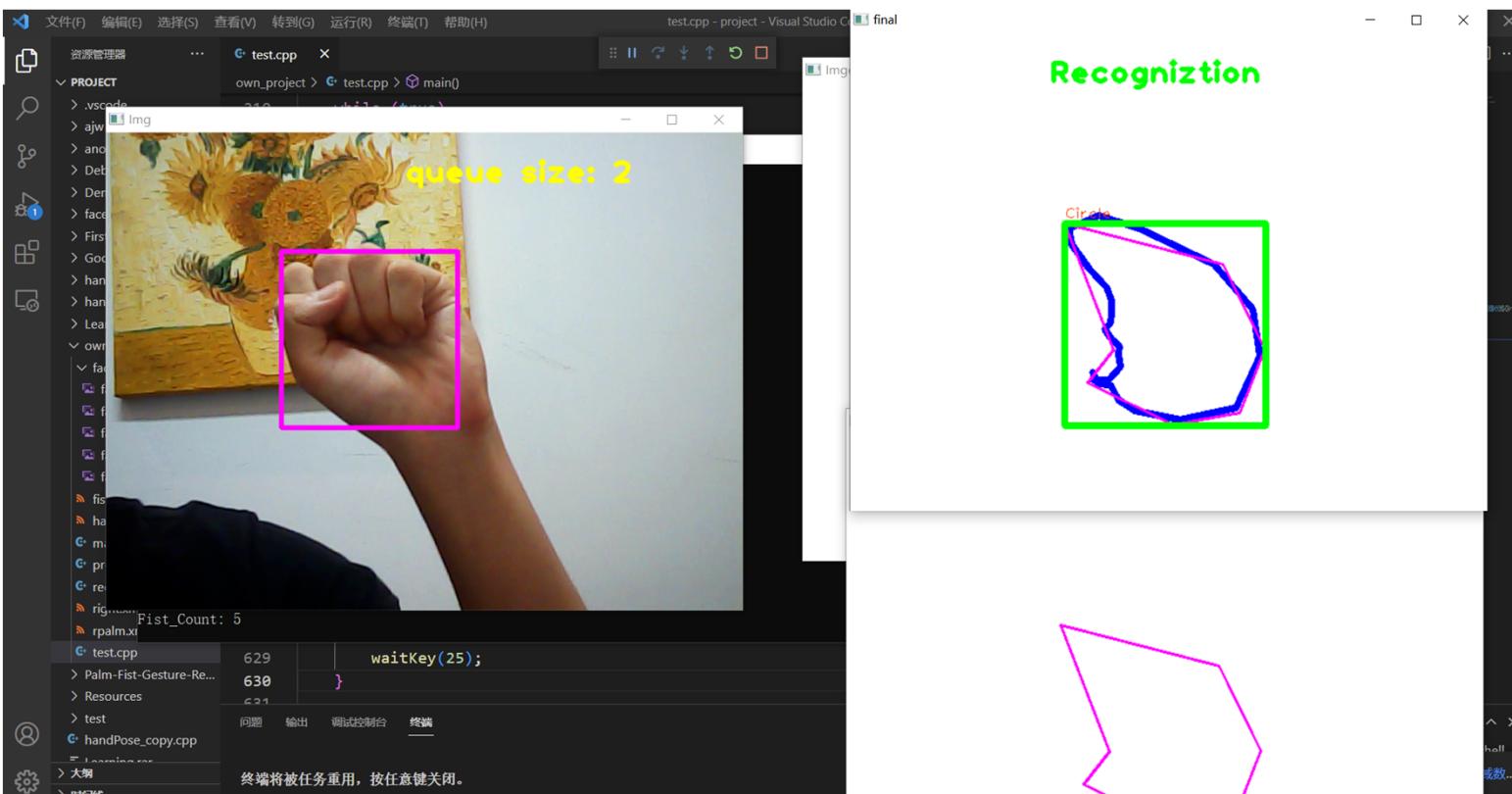
# Affect



# Affect

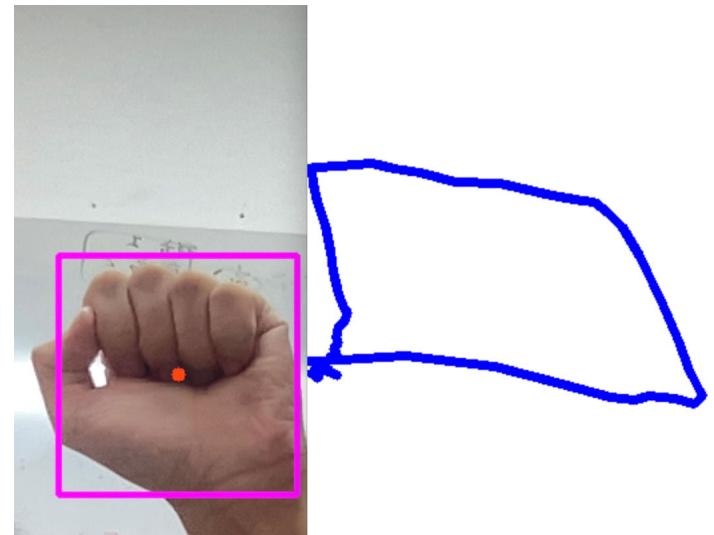


# Affect



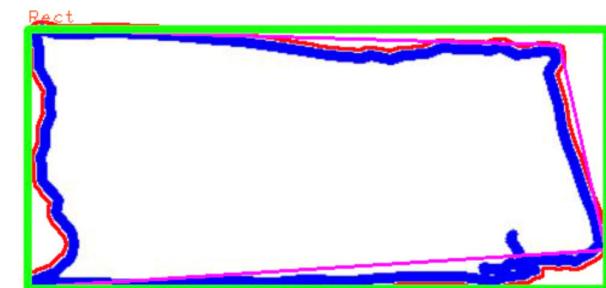
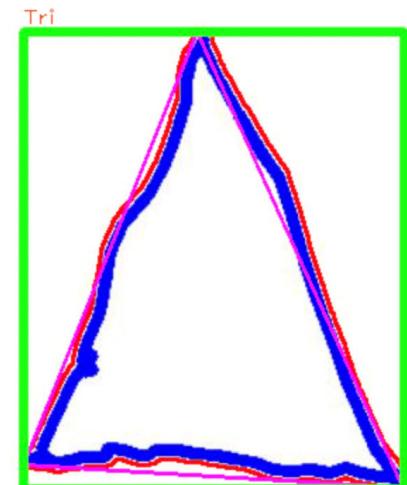
## Trajectory Detection& Hand Draw Graphics

- 拳头作为被检测目标
- 每帧计算拳头所在外接矩形的中心，并保存为Point，储存于Vector中
- 所得Vector即为轨迹，即可用于鼠标控制
- 将Vector中相邻的两点连线，画于Mat上得到图像，即可用于图形识别和相应控制



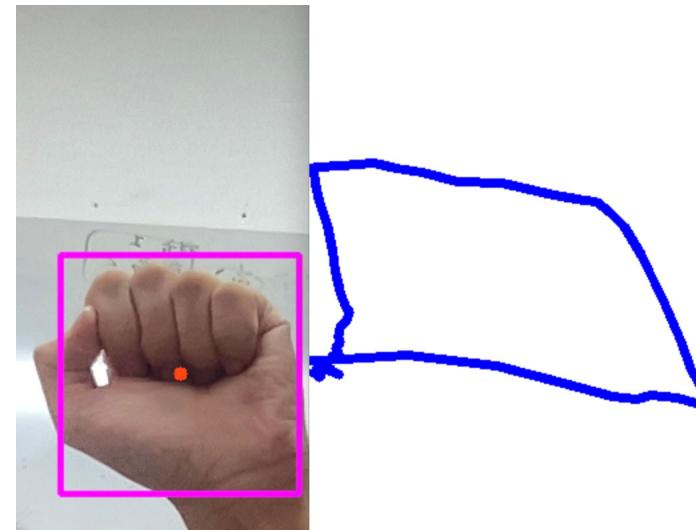
## Pattern Correction & Recognition

- 所识别物体的轨迹（即点集）被画在Mat上，并首尾相连保证闭合
- 对该存有轨迹/手绘图的Mat预处理成二值图形
- 对预处理后的图形进行边缘识别（图形的外轮廓为识别对象，排除内部干扰），得到点集
- 舍弃包含面积较小的轮廓
- 将剩余轮廓点集进行多边形拟合（Douglas-Peucker算法），得到新标准多边形点集
- 根据拟合出的角数和长宽比进行判断



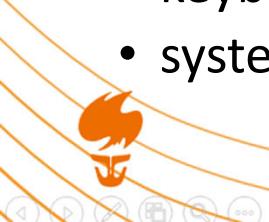
# simple controlling functions

- 根据不同的图形有相应不同的操作
- Tri--1--控制鼠标
- Squa--2--截图
- Rect--3--单击
- Pentagon--4--双击
- Circle--5--虚拟键盘
- 拳头作为被检测目标
- 原理与图像识别相同



# simple controlling functions

- 调用系统自带的函数实现控制鼠标
- SetCursorPos();//设置鼠标的位置
- GetCursorPos(&p);//更新鼠标的位置
- mouse\_event(MOUSEEVENTF\_LEFTDOWN,0,0,0,0);//模拟鼠标键按下
- mouse\_event(MOUSEEVENTF\_LEFTUP,0,0,0,0);//模拟鼠标键抬起
- keybd\_event(VK\_LWIN, 0, 0, 0);
- keybd\_event(VK\_LSHIFT, 0, 0, 0);
- keybd\_event(83, 0, 0, 0);
- keybd\_event(VK\_LWIN, 0, 2, 0);
- keybd\_event(VK\_LSHIFT, 0, 2, 0);
- keybd\_event(83, 0, 2, 0);// win+shift+S 实现快速截图
- system("C:\\Windows\\system32\\osk.exe");//打开键盘





Southern University  
of Science and  
Technology

# Thanks !