

计算机视觉

南区计软328



群聊：计算机视觉2024—2025
第一学期



该二维码7天内(9月9日前)有效，重新进入将更新

课程助教：

李泮凯：876030122@qq.com

深圳大学
计算机与软件学院

实验内容

序号	实验主题	实验内容	实验要求	实验时数	每组人数	实验类型
1	图像处理应用实验	1. 熟悉图像的表示及基本元素、通道操作; 2. 掌握基本图像增强方法; 3. 掌握OpenCV计算机视觉库;	必做	6	1	讲授 + 实验
2	图像特征提取及综合应用实践	1. 熟悉图像处理基本操作; 2. 掌握图像边缘检测原理; 3. 掌握图像基本特征抽取方法及应用;	必做	6	1	讲授 + 实验
3	计算机视觉系统实践	1. 熟悉计算机视觉分类任务; 2. 掌握数据集的准备及模型训练过程; 3. 培养应用计算机视觉解决问题的能力;	必做	6	1	讲授 + 实验

涉及学科

- 数字图像处理
- 计算机视觉
- 模式识别
- 程序设计

实验考察形式

- 三次实验报告
- 平时编程练习及表现
- 实验汇报

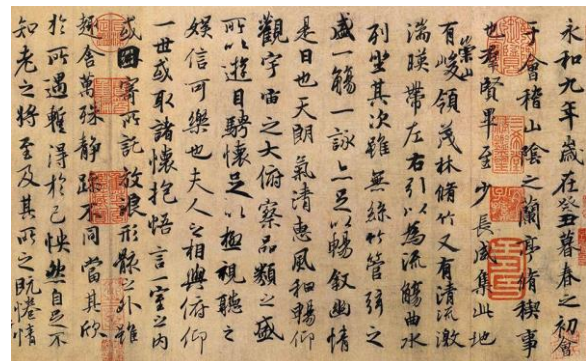
序号	实验主题	实验内容	实验要求	实验时数	每组人数	实验时间
1	图像处理应用实验	1. 熟悉图像的表示及基本元素、通道操作； 2. 掌握基本图像增强方法； 3. 掌握OpenCV计算机视觉库；	必做	6	1	2024年9月2日- 2024年10月13日
2	图像特征提取及综合应用实践	1. 熟悉图像处理基本操作； 2. 掌握图像边缘检测原理； 3. 掌握图像基本特征抽取方法及应用；	必做	6	1	2024年10月14日- 2024年11月18日
3	计算机视觉系统实践	1. 熟悉计算机视觉分类任务； 2. 掌握数据集的准备及模型训练过程； 3. 培养应用计算机视觉解决问题的能力；	必做	6	1	2024年11月19日- 2024年12月30日

实验考察内容

- 计算机视觉与模式识别基础知识掌握能力
- 算法设计能力
- 编程及系统架构能力
- 论文写作及表达能力

● 计算机视觉应用简介

图像处理应用



● 计算机视觉应用简介

视频处理应用



● 计算机视觉应用简介

视频处理应用



● 计算机视觉应用简介

视频处理应用

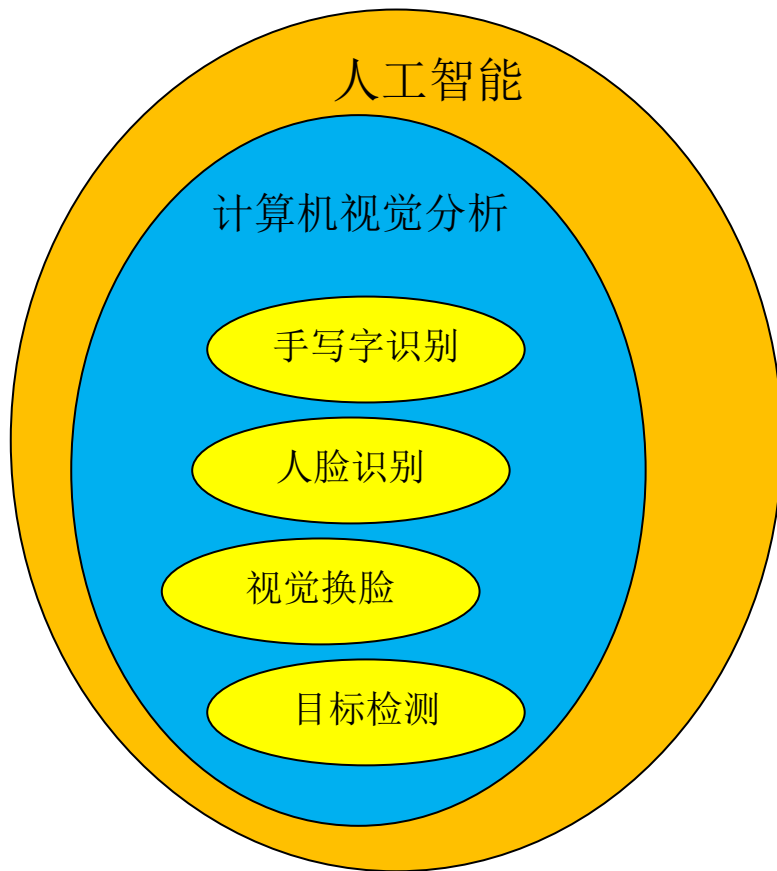


● 计算机视觉应用简介

视频处理应用



● 计算机视觉应用简介



1) 视觉分析难点

- 姿态变化;
- 光线干扰;
- 成像噪声影响（模糊、抖动、点噪声）等。

2) 视觉分析关键问题

如何在这些复杂环境下进行有效的特征提取和目标表示是视觉分析的关键问题。

3) 视觉分析研究目的

设计有效的方法对目标进行有效的特征提取，把一系列初级特征转化为具有综合语义的高级特征，实现视觉任务的模式检测、识别等问题。



●OpenCV入门

OpenCV (Open Source Computer Vision Library)是一个基于BSD许可（开源）发行的跨平台计算机视觉库，允许学术和商业的自由使用，它拥有C++、Python和Java接口并可以运行在Windows、Linux、Mac OS操作系统和Android上。OpenCV专注于各种计算机视觉问题的高效计算和实时应用，是经优化过的C/C++代码编写而成，实现了图像处理和计算机视觉方面的很多通用算法。

●OpenCV入门-参考资料

图像处理

- Rafael C. Gonzalez, Richard E. Woods著, 阮秋琦, 阮宇智译, 数字图像处理（第三版）, 电子工业出版社, 2011.6.

OpenCV3 教程

- Adrian Kaehler, Gary Bradski, Learning OpenCV 3 - Computer Vision in C++ with the OpenCV Library, O'Reilly, 2016.12
- Gary Bradski, Adrian Kaehler著, 于仕俱译, 学习OpenCV（中文版）, 清华大学出版社, 2009.10

OpenCV 官网

- <https://opencv.org/>

* OpenCV – 4.10.0

📅 2024-06-03



●OpenCV入门-参考资料

机器学习教程

- 周志华, 机器学习, 清华大学出版社, 2016.01.

深度学习教程

- Ian Goodfellow, Yoshua Bengio, Aaron Courville著, 赵申剑等译, 深度学习, 人民邮电出版社, 2017.8.

Python教程

- <https://www.liaoxuefeng.com/wiki/1016959663602400>

深度学习平台教程

- PyCharm教程 <https://www.yiibai.com/pycharm>



●OpenCV入门

- OpenCV在Python语言环境下的配置
- OpenCV在C++语言环境下的配置



●OpenCV入门

- Python+Anaconda+OpenCV
- Python+Pycharm+OpenCV （参考计算机视觉环境配置相关文档，在Blackboard下载）



●OpenCV在Python语言环境下的配置

所需安装工具:

- ① Python
- ② Anaconda
- ③ OpenCV

●OpenCV在Python语言环境下的配置

Python的安装

Python官网<https://www.python.org/downloads/>下载Windows版本的Python



The screenshot shows the Python.org website. At the top left is the Python logo. To the right is a search bar with a magnifying glass icon, a "GO" button, and a "Socialize" link. Below the search bar is a navigation menu with links: "About", "Downloads", "Documentation", "Community", "Success Stories", "News", and "Events". The main content area features a large heading "Download the latest version for Windows" in yellow. Below this heading is a yellow button labeled "Download Python 3.12.5". Underneath the button, there is text: "Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)". Below this, it says "Want to help test development versions of Python 3.13? [Prereleases](#), [Docker images](#)". On the right side of the main content area, there is an illustration of two parachutes with yellow and white stripes, each carrying a large cardboard box, set against a blue sky with white clouds.

●OpenCV在Python语言环境下的配置

Looking for a specific release?

Python releases by version number:

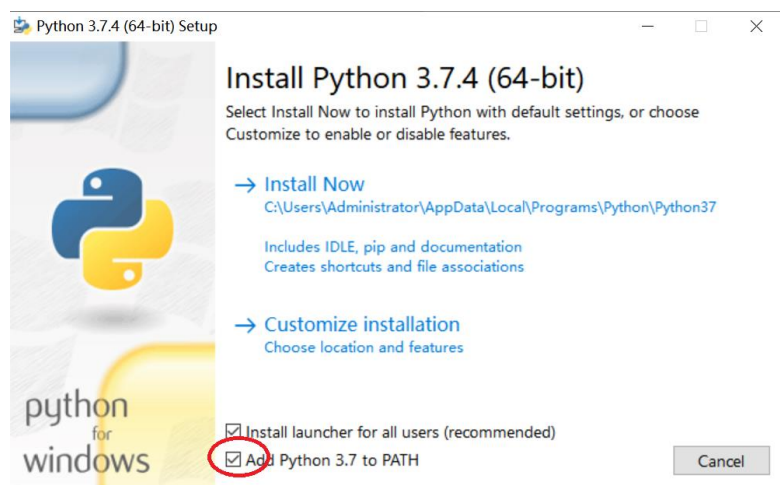
Release version	Release date		Click for more
Python 3.12.5	Aug. 6, 2024	Download	Release Notes
Python 3.12.4	June 6, 2024	Download	Release Notes
Python 3.12.3	April 9, 2024	Download	Release Notes
Python 3.11.9	April 2, 2024	Download	Release Notes
Python 3.10.14	March 19, 2024	Download	Release Notes
Python 3.9.19	March 19, 2024	Download	Release Notes
Python 3.8.19	March 19, 2024	Download	Release Notes

确定版本号

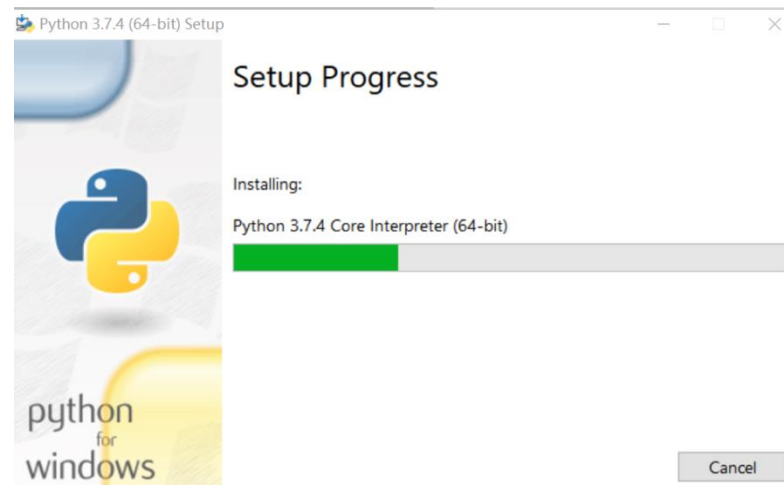
Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		68111671e5b2db4ae77b9ab01bf0f9be	23017663	SIG
XZ compressed source tarball	Source release		d33e4aae66097051c2eca45ee3604803	17131432	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7583daff1a42c8a8cee08e6	34898416	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	5dd605c38217a45773bf5e4a936b241f	28082845	SIG
Windows help file	Windows		d6399573a2c0b2ac56cade6b47cd2	8131761	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	9b00c8cf6d9ec0b9abe83184a40729a2	7504391	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	a702b4b0ad76debdb3043a583e563400	26680368	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	28cb1c608bbd73ae8e53a3bd351b4bd2	1362904	SIG
Windows x86 embeddable zip file	Windows		9fab3b81f8841879fda94133574139d8	6741626	SIG
Windows x86 executable installer	Windows		33cc602942a5444a3d6451476394789	25663848	SIG
Windows x86 web-based installer	Windows		1b670cfa5d317df82c30983ea371d87c	1324608	SIG

选择合适的安装文件



进入Python安装界面




Python安装过程

●OpenCV在Python语言环境下的配置

Anaconda的配置

在Anaconda官网<https://www.anaconda.com/download>下载Anaconda（根据Python的版本选择安装）
（例如安装目录为：C:\ProgramData\Anaconda3


Anaconda Installers



Windows

Python 3.12

64-Bit Graphical Installer (912.3M)



Mac


Python 3.12

64-Bit (Apple silicon) Graphical Installer (704.7M)

64-Bit (Apple silicon) Command Line Installer (707.3M)

64-Bit (Intel chip) Graphical Installer (734.7M)

64-Bit (Intel chip) Command Line Installer (731.2M)



Linux

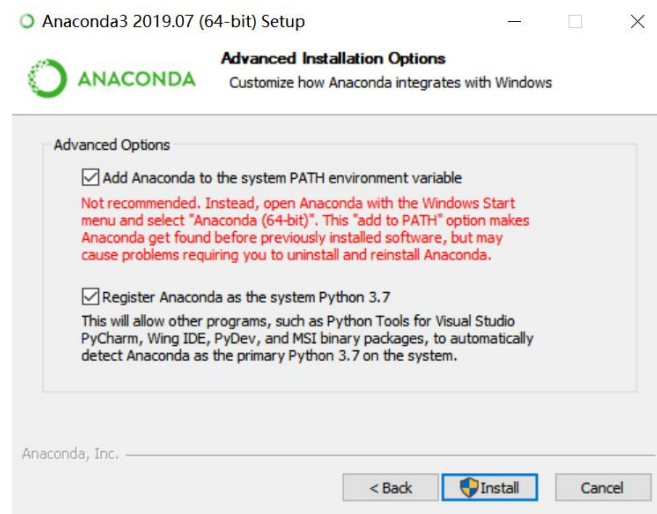
Python 3.12

64-Bit (x86) Installer (1007.9M)

64-Bit (AWS Graviton2 / ARM64) Installer (800.6M)

64-bit (Linux on IBM Z & LinuxONE) Installer (425.8M)

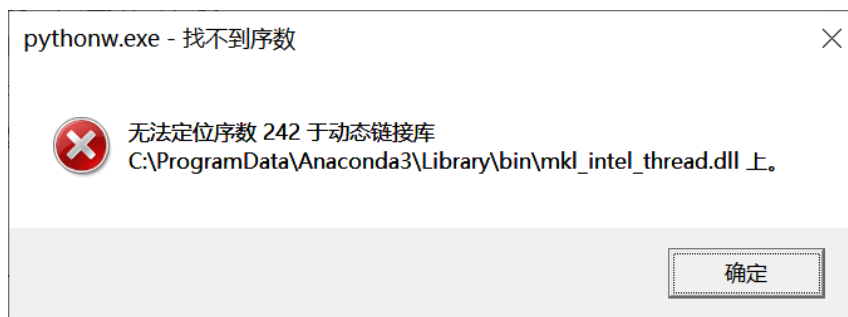
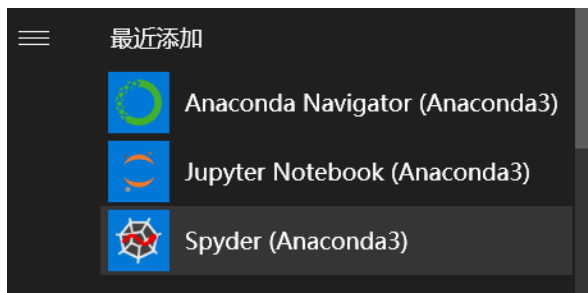
选择合适的版本



执行安装

●OpenCV在Python语言环境下的配置

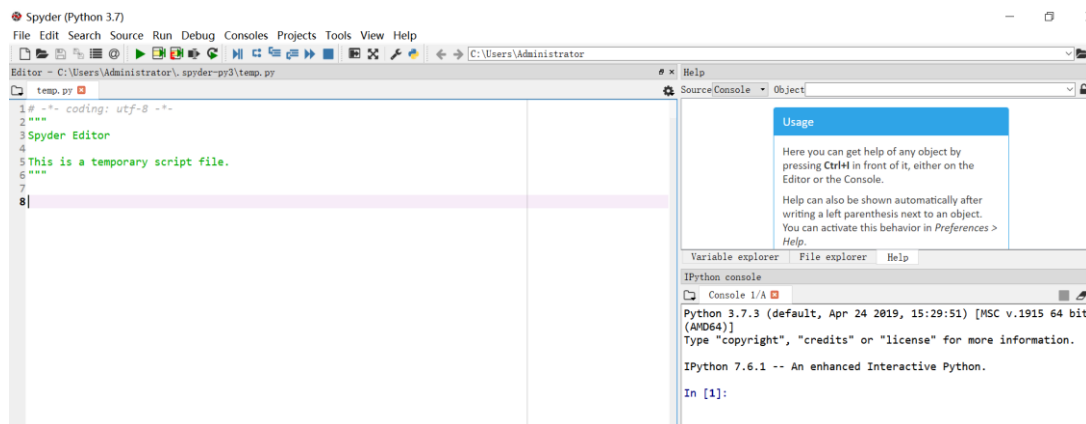
存在问题



打开Spyder失败!

解决方案

- 直接在ANACONDA Prompt命令窗口，运行`pip uninstall numpy`，卸载存在问题的numpy库；
- 然后安装`pip install numpy`。问题解决



成功打开Spyder!

●OpenCV在Python语言环境下的配置

OpenCV的安装

- ① 从网站 <https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv> 上下载（选择合适的版本）
(<https://pypi.org/project/opencv-python/#files>)
opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl
- ② 把该文件并放入安装目录C:\ProgramData\Anaconda3\Lib\site-packages下
- ③ 在Anaconda Prompt命令窗口转到目录
C:\ProgramData\Anaconda3\Lib\site-packages，并输入命令：
pip install opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl
- ④ 安装完后在Spyder窗口输入import cv2，如没有报错，表示安装成功。

← → ↻ <https://www.lfd.uci.edu/~gohlke/pythonlibs/#opencv>

OpenCV, a real time computer vision library.

opencv_python-2.4.13.7-cp27-cp27m-win32.whl
opencv_python-2.4.13.7-cp27-cp27m-win_amd64.whl
opencv_python-3.4.7+contrib-cp35-cp35m-win32.whl
opencv_python-3.4.7+contrib-cp35-cp35m-win_amd64.whl
opencv_python-3.4.7+contrib-cp36-cp36m-win32.whl
opencv_python-3.4.7+contrib-cp36-cp36m-win_amd64.whl
opencv_python-3.4.7+contrib-cp37-cp37m-win32.whl
opencv_python-3.4.7+contrib-cp37-cp37m-win_amd64.whl
opencv_python-3.4.7+contrib-cp38-cp38-win32.whl
opencv_python-3.4.7+contrib-cp38-cp38-win_amd64.whl
opencv_python-3.4.7-cp35-cp35m-win32.whl
opencv_python-3.4.7-cp35-cp35m-win_amd64.whl
opencv_python-3.4.7-cp36-cp36m-win32.whl
opencv_python-3.4.7-cp36-cp36m-win_amd64.whl
opencv_python-3.4.7-cp37-cp37m-win32.whl
opencv_python-3.4.7-cp37-cp37m-win_amd64.whl
opencv_python-3.4.7-cp38-cp38-win32.whl
opencv_python-3.4.7-cp38-cp38-win_amd64.whl
opencv_python-4.1.1+contrib-cp35-cp35m-win32.whl
opencv_python-4.1.1+contrib-cp35-cp35m-win_amd64.whl
opencv_python-4.1.1+contrib-cp36-cp36m-win32.whl
opencv_python-4.1.1+contrib-cp36-cp36m-win_amd64.whl
opencv_python-4.1.1+contrib-cp37-cp37m-win32.whl
opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl
opencv_python-4.1.1+contrib-cp38-cp38-win32.whl
opencv_python-4.1.1+contrib-cp38-cp38-win_amd64.whl

选择合适的版本

文件 主页 共享 查看

← → ↻ 此电脑 本地磁盘 (C:) > ProgramData > Anaconda3 > Lib > site-packages >

名称	修改日期	类型	大小
cython.py			
decorator.py			
docutils-0.14			
easy_install.py			
entrypoints.py			
filelock.py			
greenlet.cp37			
heapdict.py			
heapdict-1.0.0			
imageio-2.5.0			
imageio.py			
invenio			

```
(base) C:\Users\Administrator>cd C:\ProgramData\Anaconda3\Lib\site-packages
(base) C:\ProgramData\Anaconda3\Lib\site-packages>pip install opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl
Processing c:\programdata\anaconda3\lib\site-packages\opencv_python-4.1.1+contrib-cp37-cp37m-win_amd64.whl
Installing collected packages: opencv-python
Successfully installed opencv-python-4.1.1+contrib
```

```
Python console
Console 1/A

In [3]:
In [3]: import pip._internal

In [4]: print(pip._internal.pep425tags.get_supported())
[('cp37', 'cp37m', 'win_amd64'), ('cp37', 'none', 'win_amd64'), ('py3', 'none', 'win_amd64'), ('cp37', 'none', 'any'), ('cp3', 'none', 'any'), ('py37', 'none', 'any'), ('py3', 'none', 'any'), ('py36', 'none', 'any'), ('py35', 'none', 'any'), ('py34', 'none', 'any'), ('py33', 'none', 'any'), ('py32', 'none', 'any'), ('py31', 'none', 'any'), ('py30', 'none', 'any')]

In [5]: import cv2
In [6]:
```

OpenCV在Python语言环境下安装成功

●OpenCV在Python语言环境下的配置

存在问题

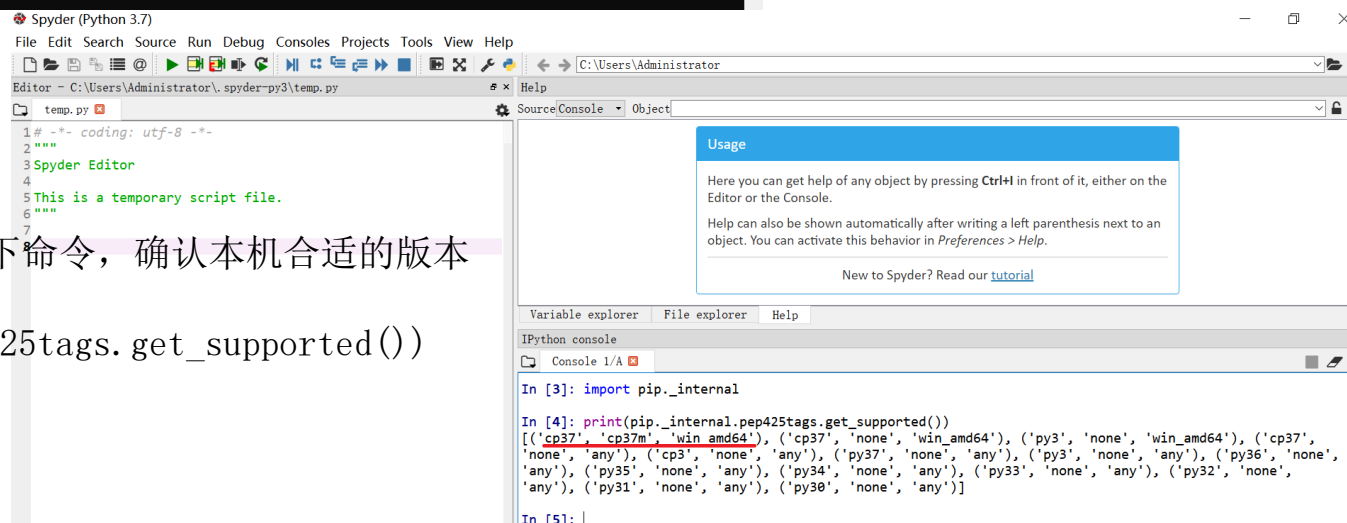
```
管理员: Anaconda Prompt (Anaconda3)

(base) C:\Users\Administrator>cd..
(base) C:\Users>cd..
(base) C:\>cd ProgramData
(base) C:\ProgramData>cd Anaconda3
(base) C:\ProgramData\Anaconda3>cd Lib
(base) C:\ProgramData\Anaconda3\Lib>cd site-packages
(base) C:\ProgramData\Anaconda3\Lib\site-packages>pip install opencv_python-4.1.1+contrib-cp38-cp38-win_amd64.whl
ERROR: opencv_python-4.1.1+contrib-cp38-cp38-win_amd64.whl is not a supported wheel on this platform.
(base) C:\ProgramData\Anaconda3\Lib\site-packages>
```

版本选择不对

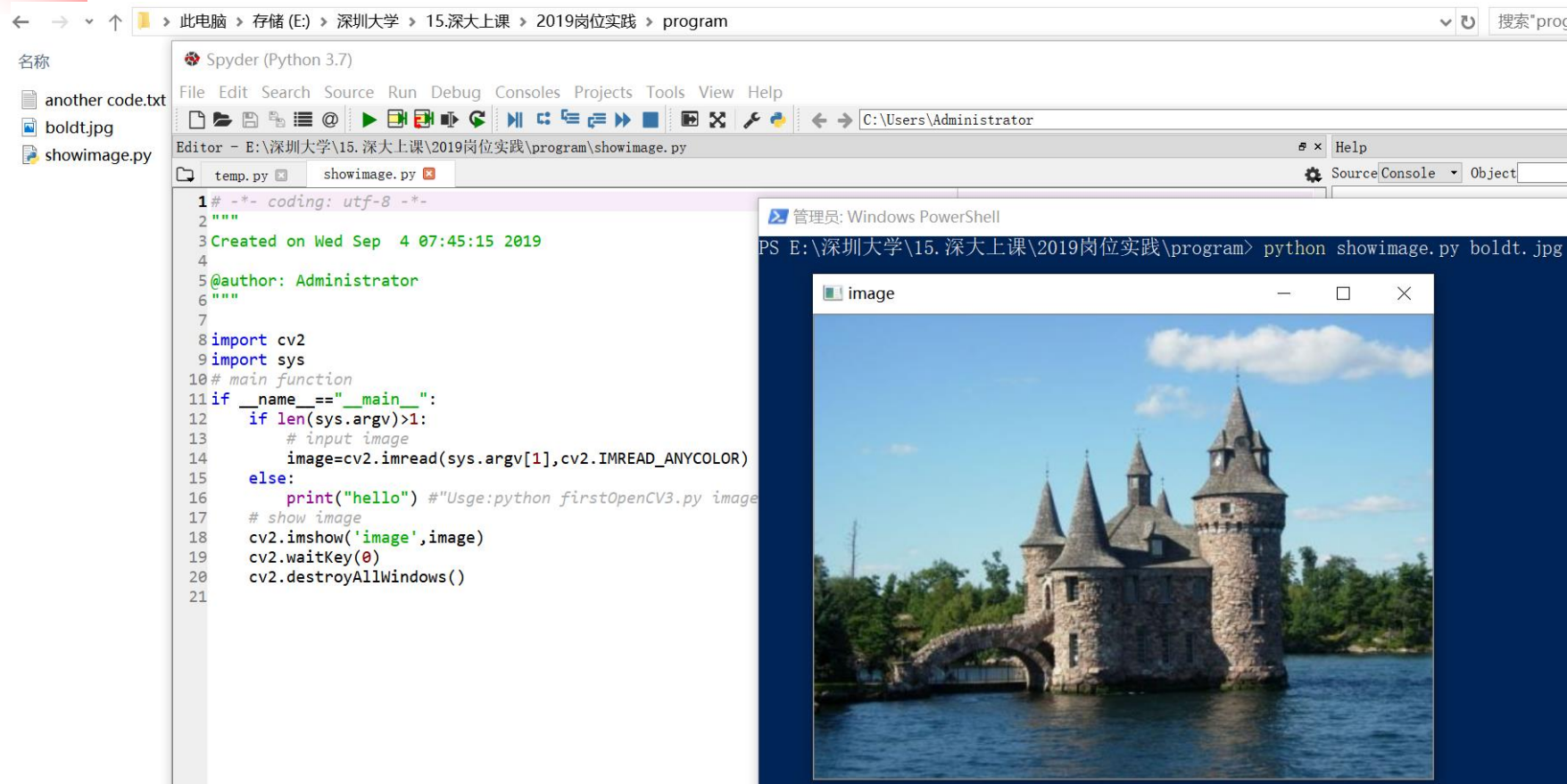
解决方案

- 在Spyder命令窗口输入以下命令，确认本机合适的版本
`import pip._internal`
`print(pip._internal.pep425tags.get_supported())`



```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\Users\Administrator\spyder-py3\temp.py
temp.py
1 # -*- coding: utf-8 -*-
2 """
3 Spyder Editor
4
5 This is a temporary script file.
6 """
7
In [3]: import pip._internal
In [4]: print(pip._internal.pep425tags.get_supported())
[('cp37', 'cp37m', 'win_amd64'), ('cp37', 'none', 'win_amd64'), ('py3', 'none', 'win_amd64'), ('cp37',
'none', 'any'), ('cp3', 'none', 'any'), ('py37', 'none', 'any'), ('py3', 'none', 'any'), ('py36', 'none',
'any'), ('py35', 'none', 'any'), ('py34', 'none', 'any'), ('py33', 'none', 'any'), ('py32', 'none',
'any'), ('py31', 'none', 'any'), ('py30', 'none', 'any')]
In [5]: |
```

●OpenCV在Python语言环境下的配置



执行一个Python程序



●OpenCV在C++语言环境下的配置

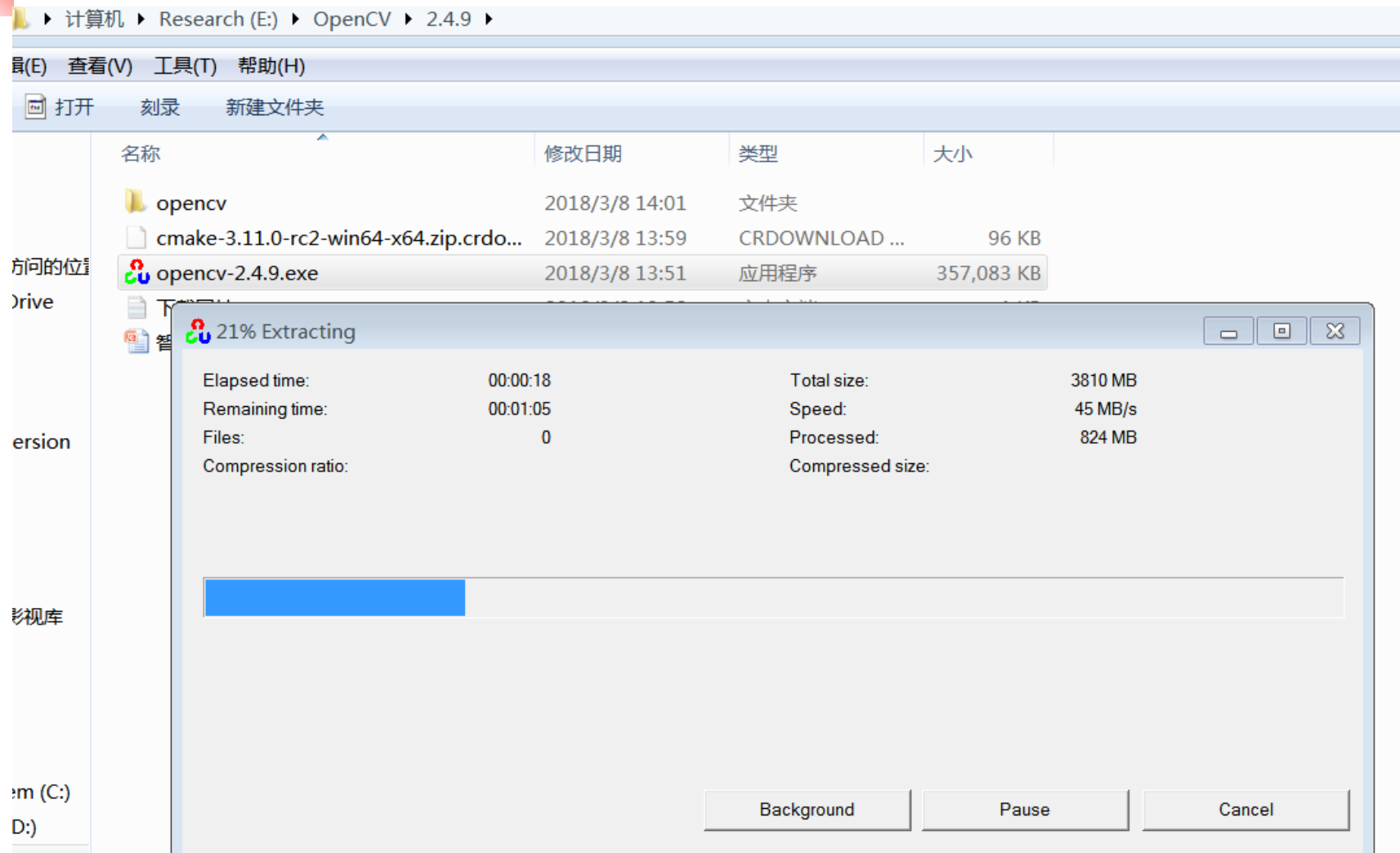
OpenCV的安装

- 准备Win7操作系统、VS2015、OpenCV 3.4.2函数库、Cmake编译器；
- 用Cmake编译对OpenCV 3.4.2进行编译；
- 用VS2015对编译出来的OpenCV.sln进行编译，生成库文件和包含文件；
- 根据库文件和包含文件的对系统进行配置。

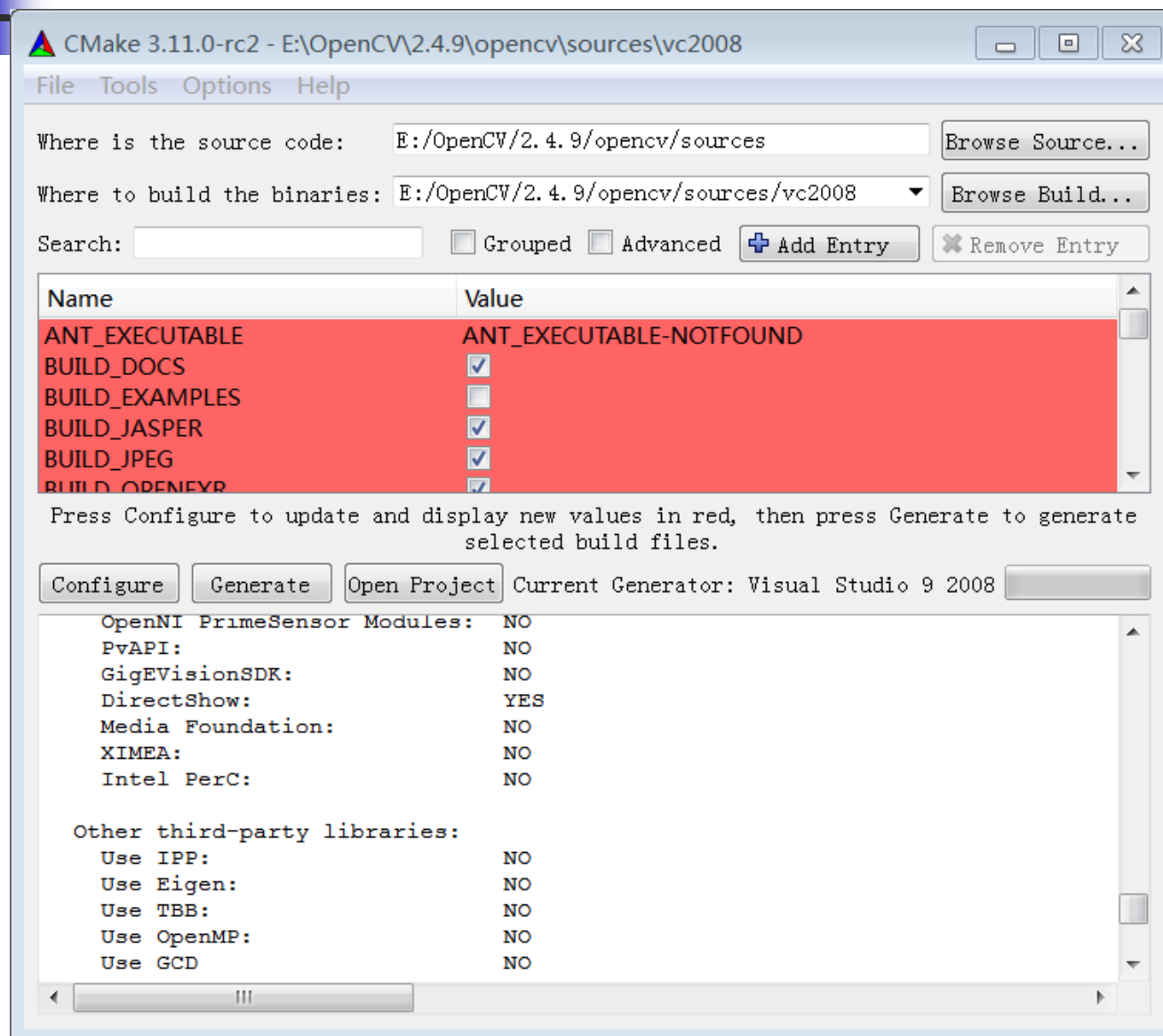
详细的安装步骤参考以下文档

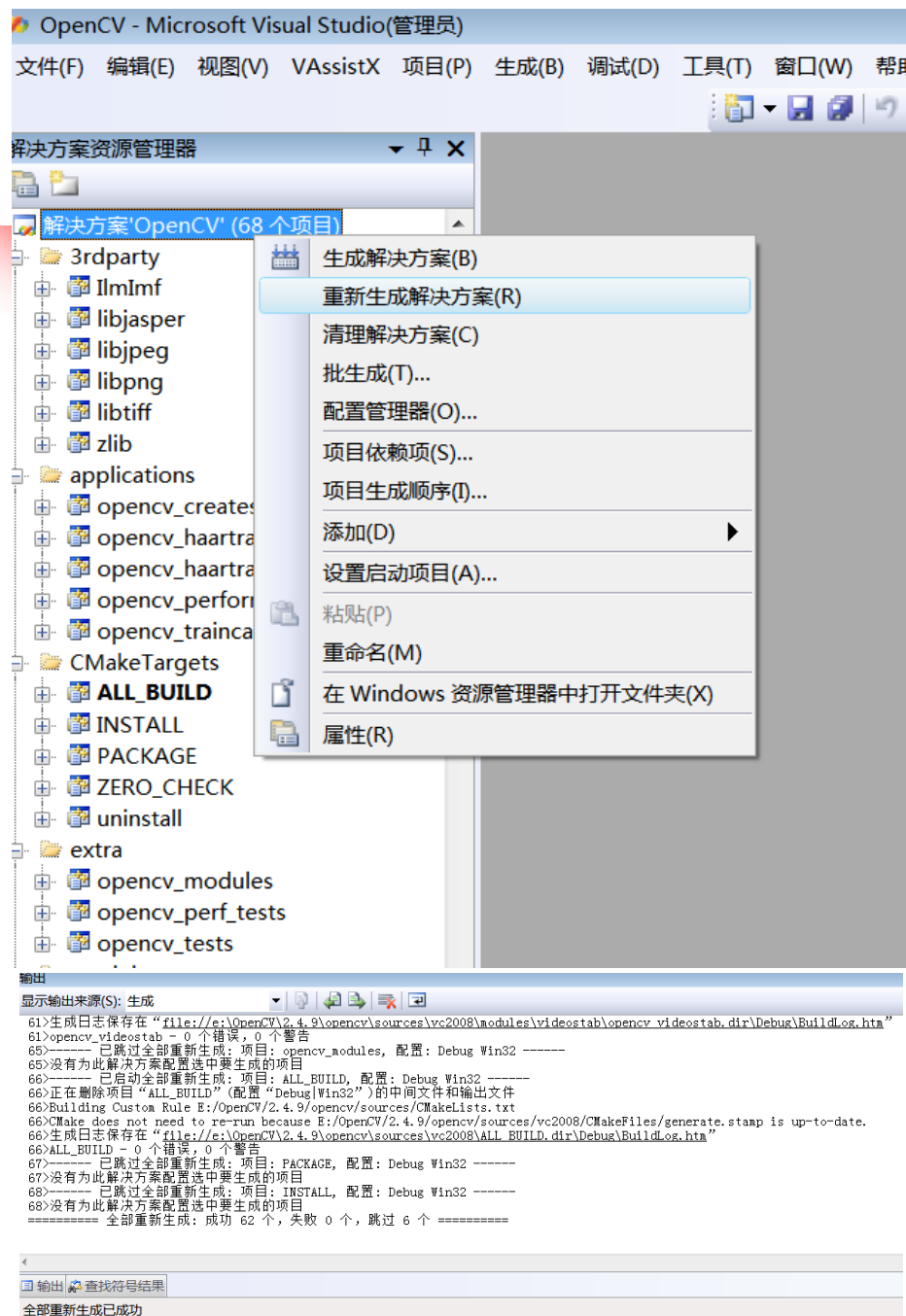
Installation method of OpenCV2.4.9+VS2012.docx

●OpenCV在C++语言环境下的配置

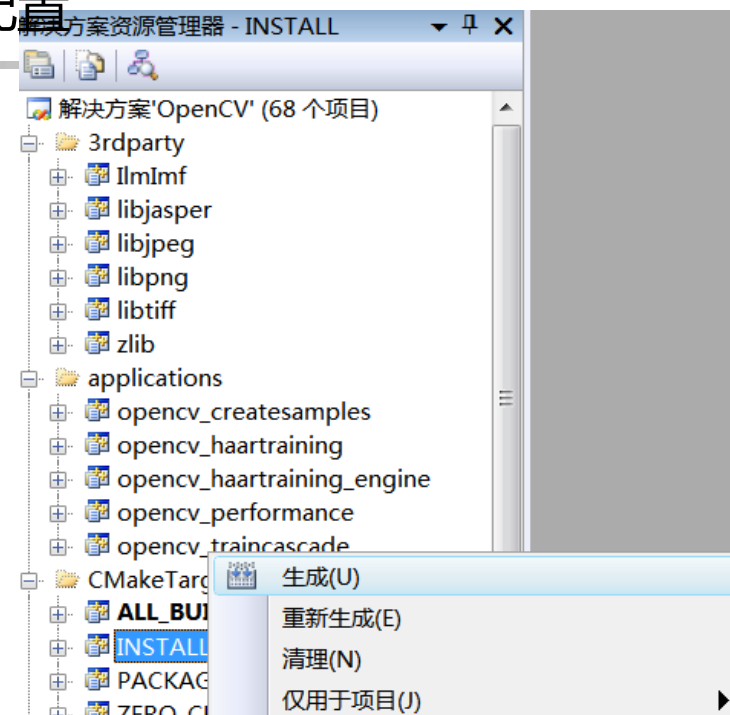


●OpenCV在C++语言环境下的配置

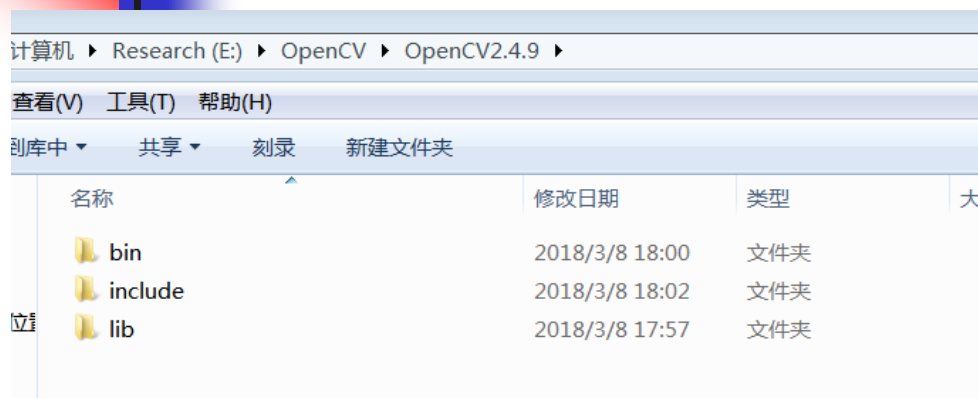




配置



●OpenCV在C++语言环境下的配置



64位系统+VC2008

include:

C:\OpenCV\OpenCV2.4.7\include

C:\OpenCV\OpenCV2.4.7\include\opencv

C:\OpenCV\OpenCV2.4.7\include\opencv2

lib:

C:\OpenCV\OpenCV2.4.7\lib

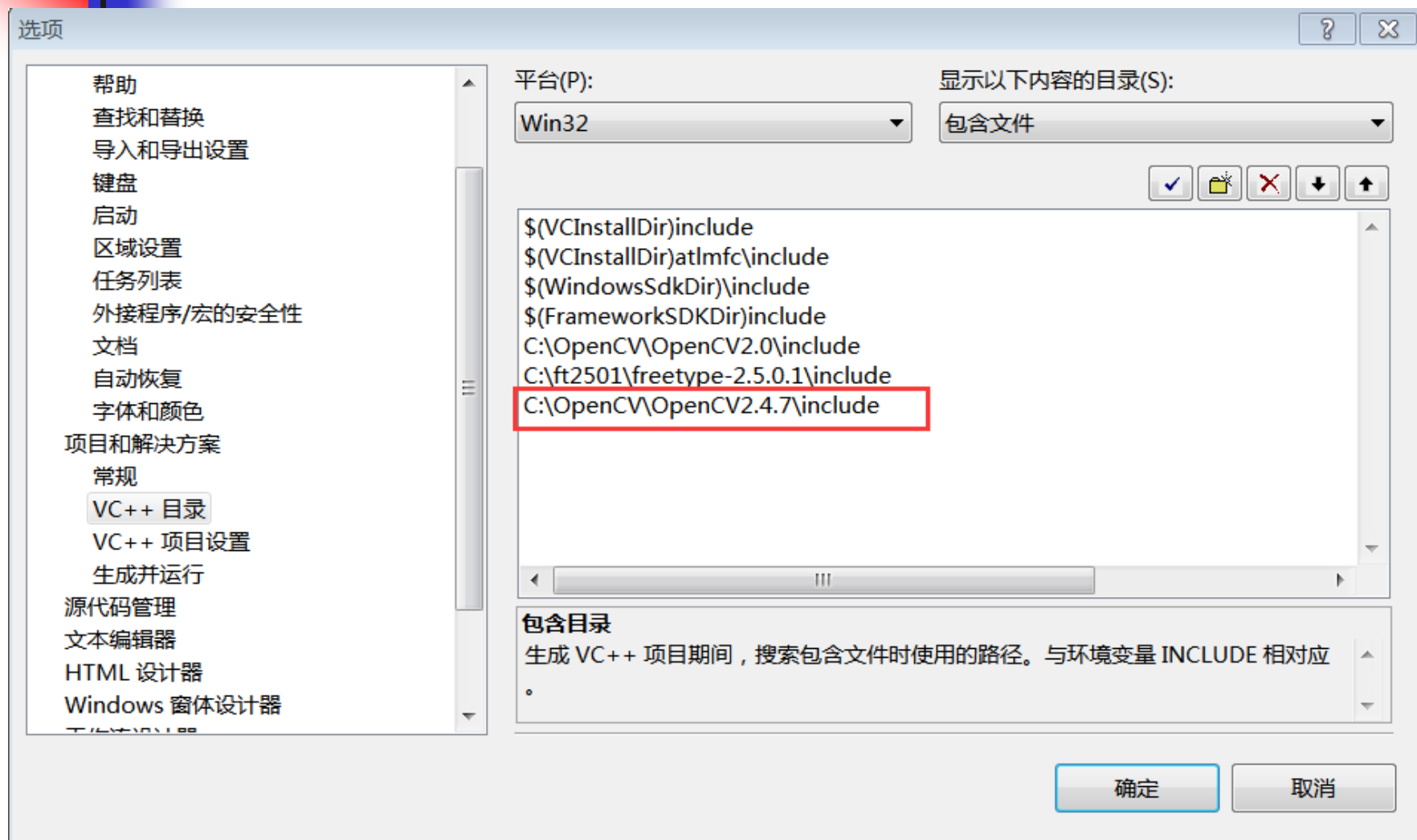
bin:

C:\OpenCV\OpenCV2.4.7\bin

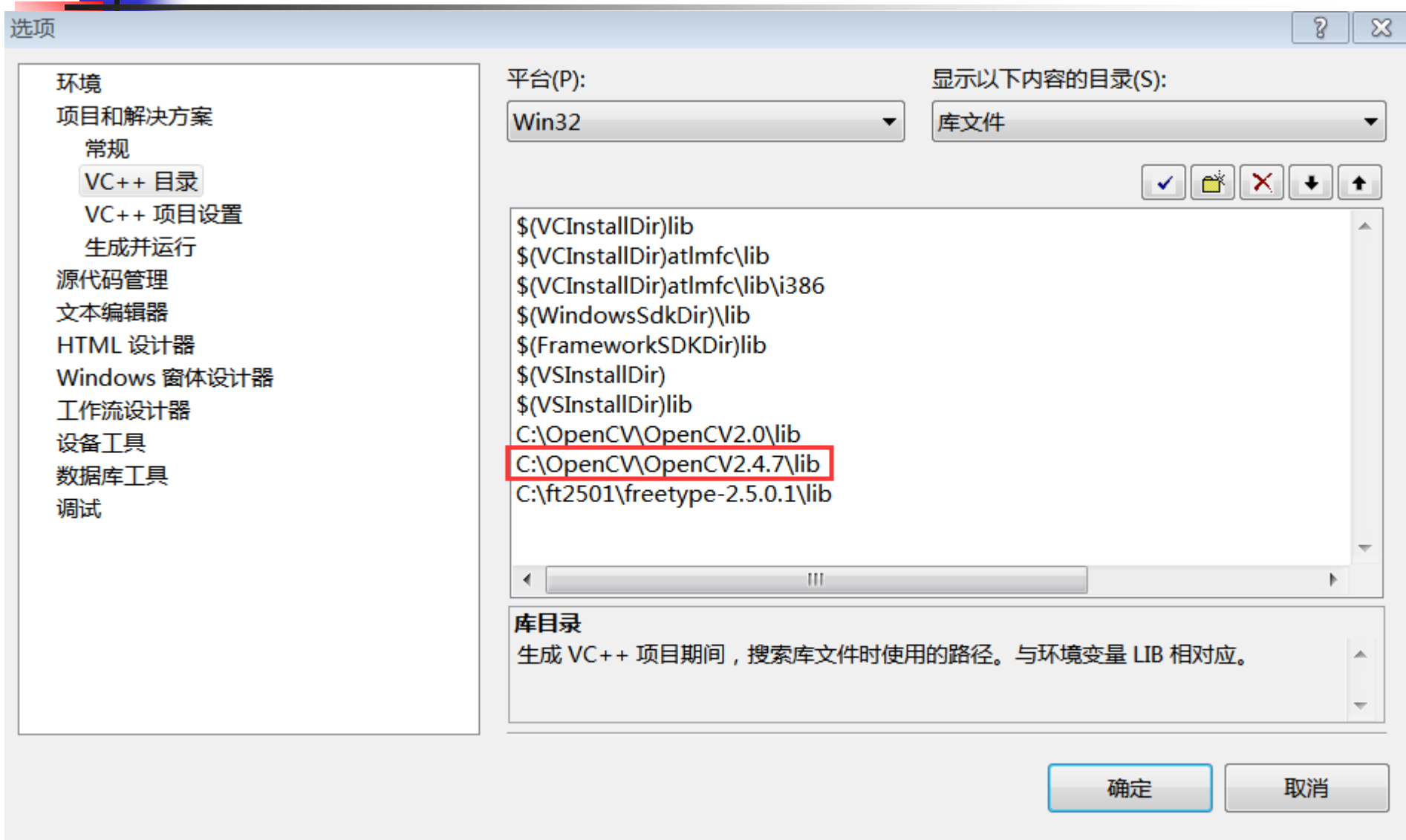
Debug配置增加 [附加依赖项 Additional Dependencies]:

opencv_highgui247d.lib;opencv_imgproc247d.lib;opencv_core247d.lib;opencv_features2d247d.lib;opencv_video247d.lib;opencv_videostab247d.lib;opencv_ml247d.lib;opencv_objdetect247d.lib;opencv_calib3d247d.lib;opencv_contrib247d.lib;opencv_flann247d.lib;opencv_gpu247d.lib;opencv_haartraining_engined.lib;opencv_legacy247d.lib;opencv_nonfree247d.lib;opencv_ocl247d.lib;opencv_photo247d.lib;opencv_stitching247d.lib;opencv_superres247d.lib;opencv_ts247d.lib

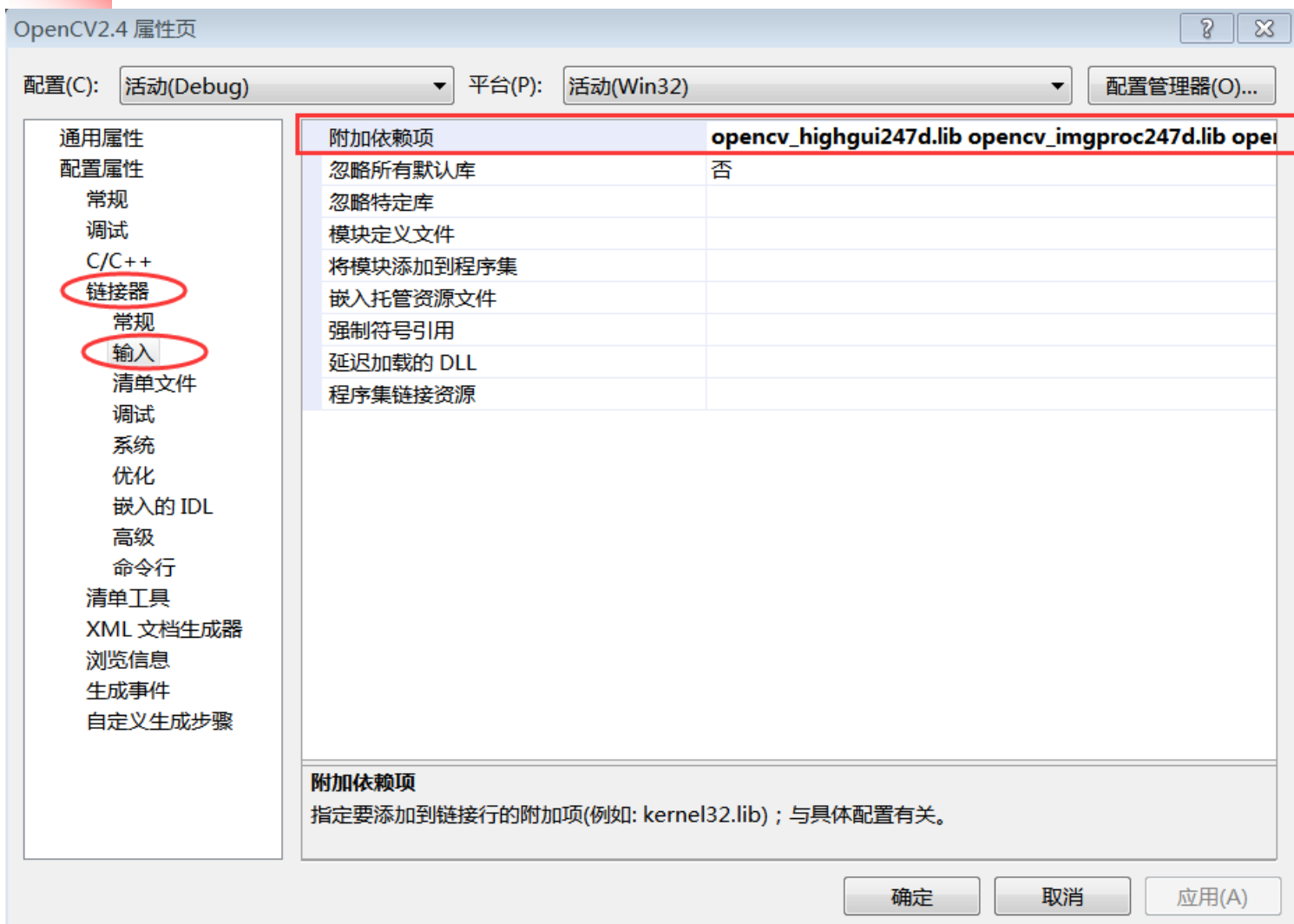
●OpenCV在C++语言环境下的配置



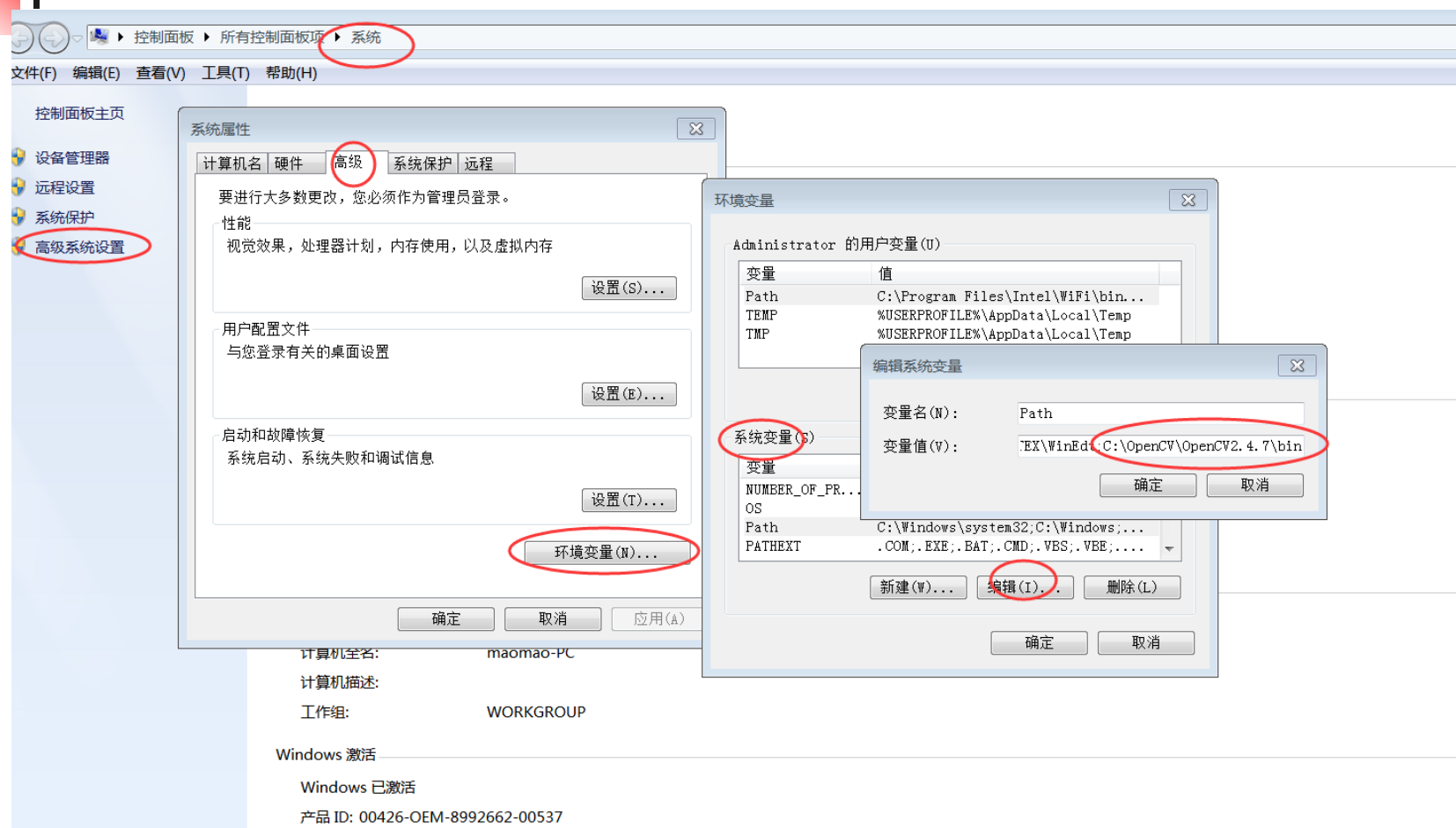
●OpenCV在C++语言环境下的配置



●OpenCV在C++语言环境下的配置



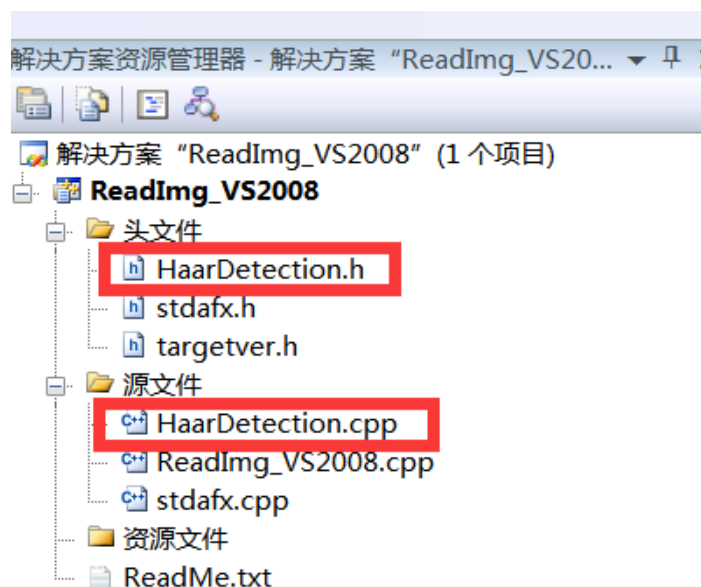
●OpenCV在C++语言环境下的配置



●OpenCV在C++语言环境下的配置

编程要求:

- ① 建立C++类;
- ② 把变量和方法写入C++类;
- ③ 通过声明类对象方式调用方法。



(1) 建立.h和.cpp文件

```
HaarDetection.h
CHaarDetection
#ifndef HAARDETECTION_H_
#define HAARDETECTION_H_
#include <opencv2/opencv.hpp>
using namespace std;
using namespace cv;
class CHaarDetection
{
public:
    CascadeClassifier cascade_faceDetector;
public:
    CHaarDetection(void);
    ~CHaarDetection(void);

    //最多返回10个人脸
    //在图像中的局部区域LocalArea进行Adaboost,
    void FaceDetection(Mat image, Rect LocalArea, vector<Rect> Faces);
    void DrawOnImg(Mat &image, vector<Rect> Faces);
    void TestOnImgDemo();
    void TestOnVideoDemo();
};
#endif
```

(2) 建立C++类

●OpenCV在C++语言环境下的配置

编程要求:

```

HaarDetection.cpp
C HaarDetection
C HaarDetection()

#include "stdafx.h"
#include "HaarDetection.h"
C HaarDetection::C HaarDetection(void)
{
    //预定义分类器路径
    string cascade_path1="haarcascade_frontalface_alt.xml";
    string cascade_path2="C:\\haarcascade_frontalface_alt.xml";

    if(cascade_faceDetector.empty()!=true)
    {
        cascade_faceDetector.load(cascade_path1);
        //cascade_faceDetector.load(cascade_path2);
    }
    if(cascade_faceDetector.empty()!=true)
    {
        printf("没有找到分类器文件\n");
    }
}

C HaarDetection::~C HaarDetection(void)
{
}

void C HaarDetection::FaceDetection(Mat image,Rect LocalArea,vector<Rect>&FaceVector,
{
    Mat ImgROI=image (Rect(LocalArea.x,LocalArea.y,LocalArea.width,LocalArea.height));//直
    int flags = CASCADE_FIND_BIGGEST_OBJECT|CASCADE_DO_ROUGH_SEARCH; //
    //int flags = CASCADE_SCALE_IMAGE; //检测多个人
    Size minFeatureSize(10, 10);
    float searchScaleFactor = 1.1f;
}
    
```

```

ReadImg_VS2008.cpp*
(全局范围)

// ReadImg_VS2008.cpp : 定义控制台应用程序的入口点
//
#include "stdafx.h"
#include <opencv2/opencv.hpp>
using namespace std;
using namespace cv;

#include "HaarDetection.h"

//void Read_Show();
//void Read_Video_FromCam();
//void Read_Video();

int _tmain(int argc, _TCHAR* argv[])
{
    //Read_Show();
    //Read_Video_FromCam();
    //Read_Video();

    C HaarDetection Obj;
    Obj.TestOnImgDemo();
    //Obj.TestOnVideoDemo();

    return 0;
}
    
```

(3) 构造函数、析构函数以及方法编写

(4) C++类的声明与调用

● OpenCV图像读取

```
void CForFreshMen::Read_Show()
{
    const char* imagename = "boldt.jpg";
    //从文件中读入图像
    Mat img = imread(imagename);
    //如果读入图像失败
    if(img.empty())
    {
        fprintf(stderr, "Can not load image %s\n", imagename);
        exit(0);
    }
    //显示图像
    namedWindow("image",1);
    cout<<"函数功能：读入并显示和保存一张图像"<<endl;
    imwrite("save.jpg",img);
    imshow("image", img);
    //此函数等待按键，按键盘任意键就返回
    waitKey(0);
}
```

```
#ifndef _FORFRESHMEN_H_
#define FORFRESHMEN_H
#include <opencv2/opencv.hpp>
#include <string>

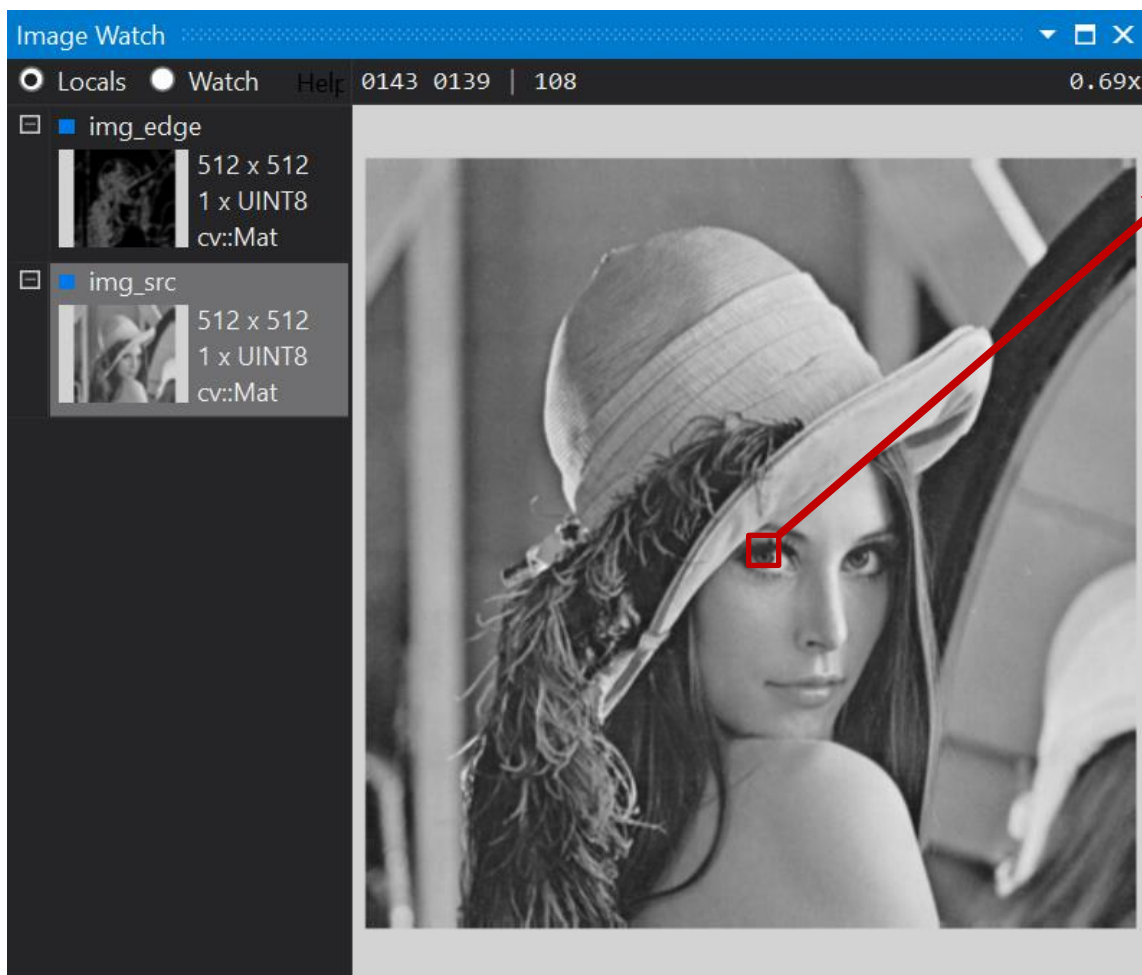
using namespace std;
using namespace cv;

class CForFreshMen
{
public:
    CForFreshMen(void);
    ~CForFreshMen(void);
}
```



● 图像基本知识

图像矩阵表示

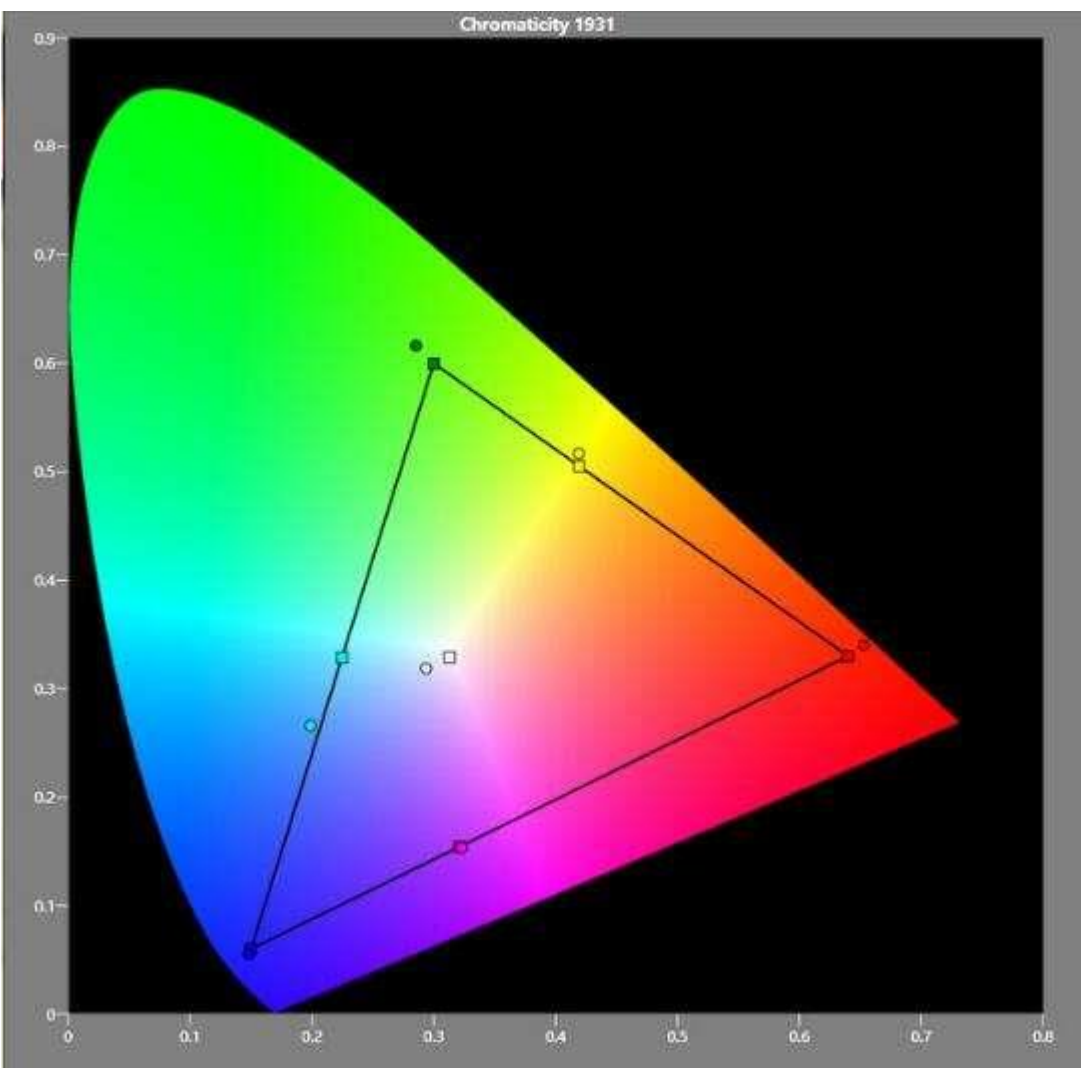


66	59	45	40	34	38	88
76	76	49	40	33	41	59
94	98	73	44	42	41	67
88	99	88	72	71	66	91
59	84	97	83	82	88	81
58	61	79	88	91	87	75
89	59	56	53	60	51	66
114	98	82	65	66	72	86

灰度图像=单矩阵 (0~255)

● 图像基本知识

图像矩阵表示 (RGB)



Leidi.jpg Properties

General Security Details Previous Versions

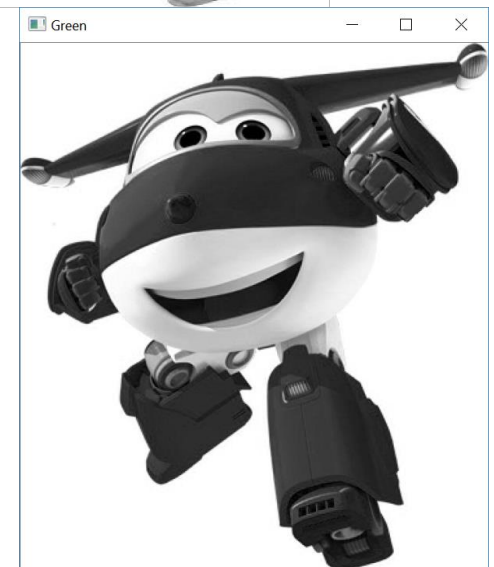
Property	Value
Image	
Image ID	
Dimensions	1200 x 1352
Width	1200 pixels
Height	1352 pixels
Horizontal resolution	72 dpi
Vertical resolution	72 dpi
Bit depth	24
Compression	
Resolution unit	
Color representation	
Compressed bits/pixel	
Camera	
Camera maker	
Camera model	
F-stop	
Exposure time	
ISO speed	
Exposure bias	

[Remove Properties and Personal Information](#)

OK Cancel Apply

● 图像基本知识

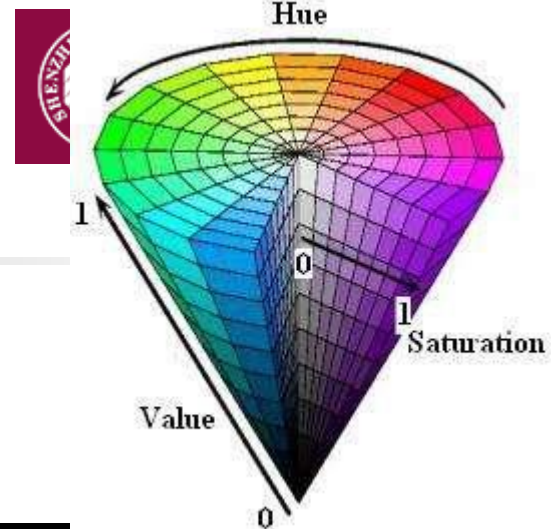
图像矩阵表示 (RGB)



`cv::split(img_work, mv);`

● 图像基本知识

图像矩阵表示 (HSV)



V: 明度



S: 饱和度

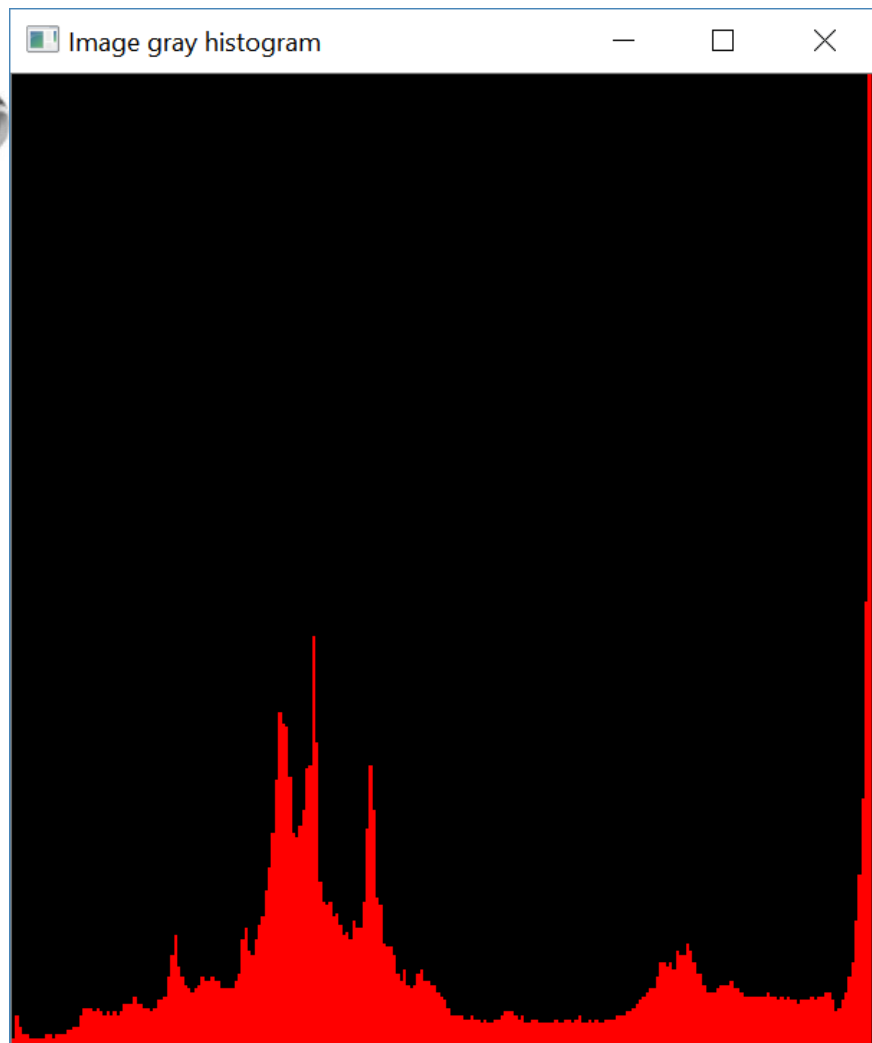


H: 色调

```
cv::cvtColor(img_work, img_hsv, CV_BGR2HSV);  
cv::split(img_hsv, mv);
```

● 图像基本知识

图像直方图表示



频率

灰度值 (0~255)

●OpenCV 边缘提取

```
void CForFreshMen::SobelDemo()
{
```

```
    cout<<"函数功能: OpenCV中的Sobel函数"<<endl;
    Mat image=imread("boldt.jpg",0);//DSC_2038
    //Mat image=imread("new_ac_4_local.bmp",0);//DSC_2038
    //Mat image=imread("fabric.jpg",0);//DSC_2038
    Mat SobelX;
    Mat SobelY;
    int ksize=3;
    double scale=0.4;//缩放值
    double delta=128;//偏移量
    /*Sobel(image,SobelX,CV_8U,1,0,ksize,1,0);
    Sobel(image,SobelY,CV_8U,0,1,ksize,1,0);
    imshow("X",SobelX);
    imshow("Y",SobelY);*/
    Sobel(image,SobelX,CV_8U,1,0,ksize,scale,delta);
    Sobel(image,SobelY,CV_8U,0,1,ksize,scale,delta);
    imshow("X_8U",SobelX);
    imshow("Y_8U",SobelY);
    Sobel(image,SobelX,CV_16S,1,0);
    Sobel(image,SobelY,CV_16S,0,1);

    Mat SobelImg;
    SobelImg=abs(SobelX)+abs(SobelY);
    double sobmin,sobmax;
    minMaxLoc(SobelImg,&sobmin,&sobmax);

    //变为位图像
    //sobelimg=-alpha*sobel+255 //反色
    SobelImg.convertTo(SobelImg,CV_8U,-255./sobmax,255);
    imshow("sobelimg",SobelImg);
```

```
}
```

```
    Mat binaryimg;
    threshold(SobelImg,binaryimg,230,255,THRESH_BINARY);
    imshow("binary",binaryimg);
```

```
    Sobel(image,SobelX,CV_32F,1,0);//必须以浮点数格式进行计算
    Sobel(image,SobelY,CV_32F,0,1);
    Mat normimg,dirimg;//计算L2范式及梯度方向
    cartToPolar(SobelX,SobelY,normimg,dirimg,true);//默认: 方向的
    //0-2pi 或0
```

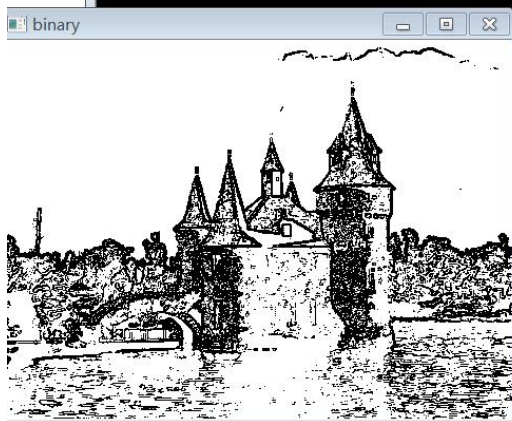
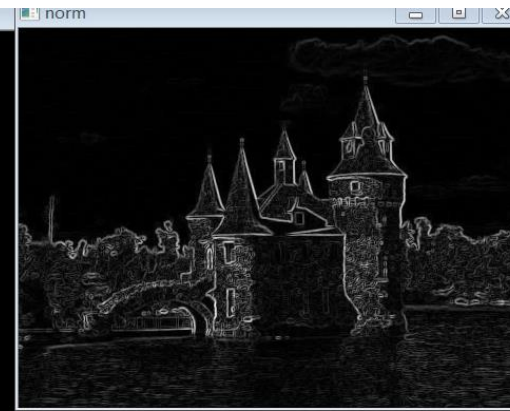
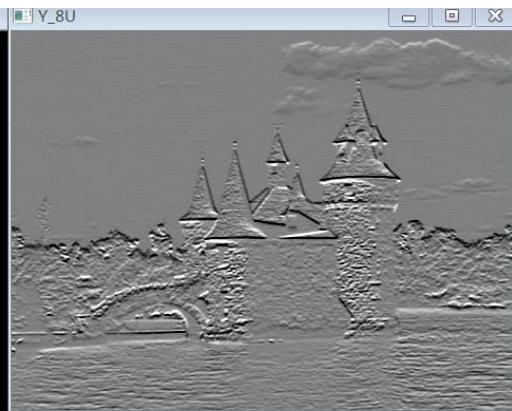
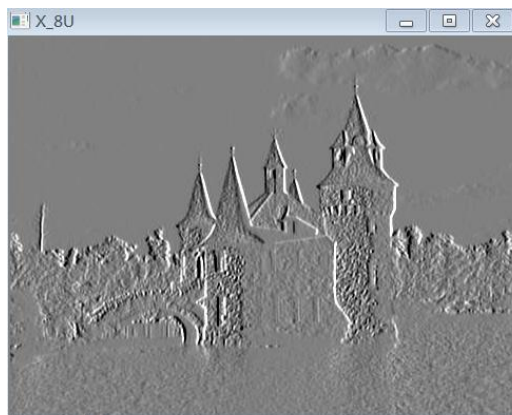
```
    cout<<"norm="<<normimg.at<float>(10,10)<<endl;
    cout<<"dirimg="<<dirimg.at<float>(10,10)<<endl;
```

```
    minMaxLoc(normimg,0,&sobmax);
```

```
    normimg.convertTo(normimg,CV_8U,255./sobmax,0);
```

```
    minMaxLoc(dirimg,0,&sobmax);
    dirimg.convertTo(dirimg,CV_8U,255./sobmax,0);
```

```
    imshow("norm",normimg);
    imshow("dir",dirimg);
    cvWaitKey(0);
```

● OpenCV 锐化图像

```
void CForFreshMen::SharpenThreePointMain()
{
    Mat img=imread("coins.png",1);
    if(img.empty())
    {
        cout<<"找不到图片"<<endl;
        exit(0);
    }
    Mat grayimg,result;
    cout<<"函数功能: 锐化图像-通过三行指针";
    cvtColor(img,grayimg,CV_RGB2GRAY,0);
    SharpenThreePoint(grayimg,result);
    namedWindow("original",1);
    imshow("original",grayimg);
    imshow("show",result);
    cvWaitKey(0);
}
```



```
void CForFreshMen::SharpenThreePoint(const Mat &img,Mat &result)
{
    //如有必要则分配图像
    cout<<"函数功能: 锐化图像-通过三行指针"<<endl;
    result.create(img.size(),img.type());
    for(int j=1;j<img.rows-1;j++)//处理除了第一行和最后一行之外的所有行
    {
        const uchar*previous=img.ptr<const uchar>(j-1);//上一行
        const uchar *current=img.ptr<const uchar>(j);//当前行
        const uchar *next=img.ptr<const uchar>(j+1);//下一行
        uchar *output=result.ptr<uchar>(j);//输出行

        for(int i=1;i<img.cols-1;i++)
        {
            *output++=saturate_cast<uchar>(5*current[i]-current[i-1]-current[i+1]-previous[i]-next[i]);
        }
    }
    //将未处理的像素设置为0
    result.row(0).setTo(Scalar(0));
    result.row(result.rows-1).setTo(Scalar(0));
    result.col(0).setTo(Scalar(0));
    result.col(result.cols-1).setTo(Scalar(0));
}
```

上机练习内容：

1. OpenCV环境配置；
2. 实现对图像的读取；
3. 把一副彩色图像的三个通道变成3个单通道图像存储到硬盘上并显示；
4. 计算一幅单通道图像的直方图；
5. 编程实现在一幅单通道图像的边缘检测。