**课** **程 设 计 报 告**

**课程设计名称：人脸检测**

**课程设计题目**：利用opencv进行人脸检测

院（系）：计算机与信息工程学院

专 业: 13级计算机与信息科学技术

班 级：13级软件工程

学 号：20131105796

姓 名：苗超然

指导教师：朝力萌

完成日期： 2016年1月3日

**目 录**

[第1章 概要设计 2](#_Toc406624046)

[1.1题目的内容与要求 2](#_Toc406624047)

[1.2总体结构 2](#_Toc406624048)

[第2章 详细设计 2](#_Toc406624049)

[2.1主模块 2](#_Toc406624050)

[第3章 调试分析 3](#_Toc406624051)

[第4章 使用说明与执行结果 4](#_Toc406624052)

# 第1章 概要设计

## 1.1题目的内容与要求

内容：利用opencv技术实现打开计算机摄像头并检测人脸及眼睛

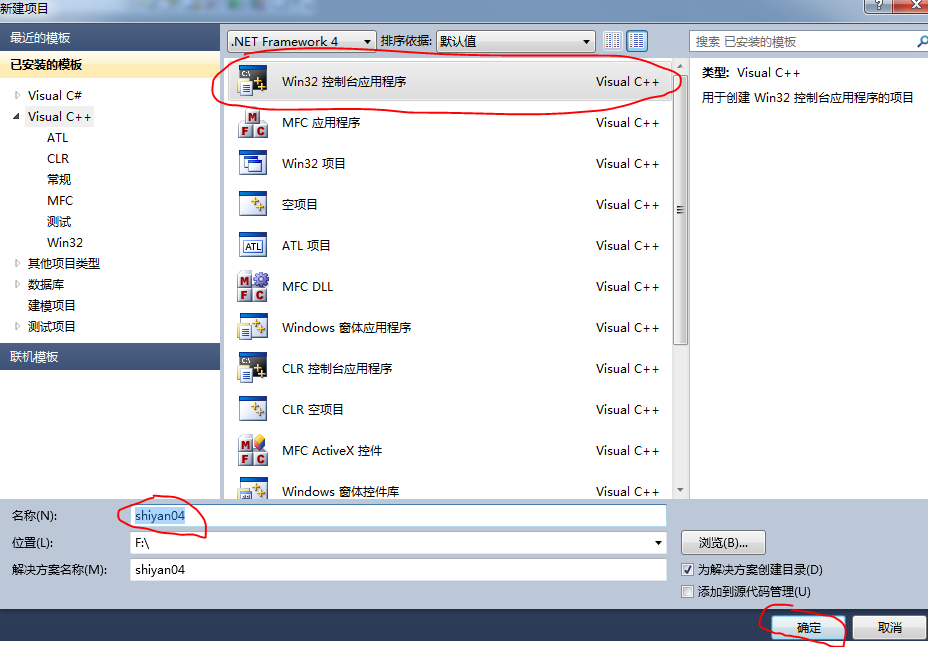
要求：1、在下载安装并调试好opencv的基础上，学会运用opencv；

2、调用摄像头并检测视频中的人脸及眼睛；

## 1.2总体结构

具体操作：

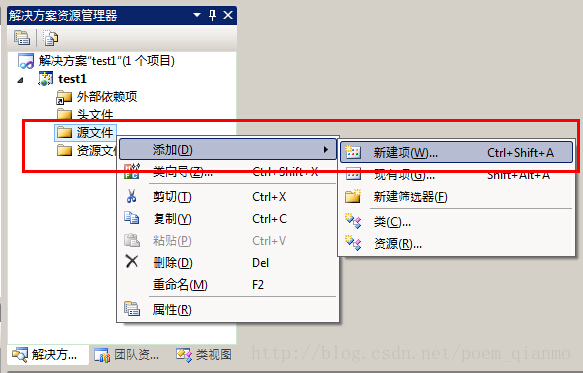
1. 首先要打开并新建一个项目（之前已经讲过如何配置opencv了，这节不再叙述），打开VS2010在菜单栏中点击文件，选择新建，项目。并根据提示完成win32应用程序向导，如下图所示：

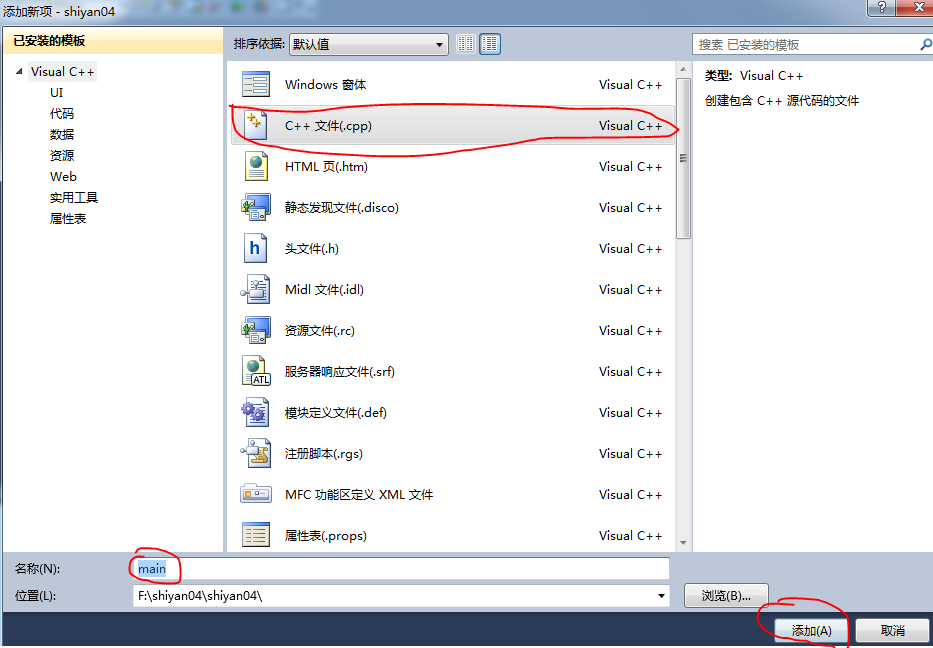






1. 在新建项目的解决方案资源管理器中数遍右键点击源文件，安装提示添加一个新建项。如图所示：





这里我们还是选择C++文(.cpp),起个名称为main.

3、准备工作到这里就算是完成了，接下来就编写程序代码，完成此次试验。

# 第2章 详细设计

## 2.1主模块

主要程序代码：

//---------------------------------【头文件、命名空间包含部分】----------------------------

// 描述：包含程序所使用的头文件和命名空间

//--------------------------------------------------------------------------------------

#include "opencv2/objdetect/objdetect.hpp"

#include "opencv2/highgui/highgui.hpp"

#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>

#include <stdio.h>

using namespace std;

using namespace cv;

//自动检测和屏幕显示

void detectAndDisplay( Mat frame );

//--------------------------------【全局变量声明】---------------------------------------

// 描述：声明全局变量

//--------------------------------------------------------------------------------------//注意，需要把"haarcascade\_frontalface\_alt.xml"和í"haarcascade\_eye\_tree\_eyeglasses.xml"这两个文件复制到工程路径下

String face\_cascade\_name = "haarcascade\_frontalface\_alt.xml";//人脸的训练数据

String eyes\_cascade\_name = "haarcascade\_eye\_tree\_eyeglasses.xml";//人眼睛的训练数据

CascadeClassifier face\_cascade;

CascadeClassifier eyes\_cascade;

string window\_name = "人脸检测";

RNG rng(12345);

//--------------------------------【help（）函数】----------------------------------

// 描述：输出帮助信息

//--------------------------------------------------------------------------------------

static void ShowHelpText()

{

//输出欢迎信息和opencv版本

cout << "\n\n\t\t\t 当前使用的opencv版本为：" << CV\_VERSION

<<"\n\n ----------------超然------------------------------------------------" ;

}

//-----------------------------------【main()函数】---------------------------------

// 描述：控制台应用程序的入口函数，我们的程序从这里开始

//--------------------------------------------------------------------------------------

int main( void )

{

VideoCapture capture;//从摄像头读入视频

Mat frame;//定义一个Mat变量，用于存储每一帧的图像

//-- 1. 加载级联（cascades） if( !face\_cascade.load( face\_cascade\_name ) ){ printf("--(!)Error loading\n"); return -1; };

if( !eyes\_cascade.load( eyes\_cascade\_name ) ){ printf("--(!)Error loading\n"); return -1; };

//-- 2. 读取视频

capture.open(0);

ShowHelpText();

if( capture.isOpened() )

{

for(;;)

{

capture >> frame;

//-- 3. 对当前帧使用分类器（Apply the classifier to the frame） if( !frame.empty() )//读取图片数据不能为空

{ detectAndDisplay( frame ); }

else

{ printf(" --(!) No captured frame -- Break!"); break; }

int c = waitKey(10);

if( (char)c == 'c' ) { break; }

}

}

return 0;

}

void detectAndDisplay( Mat frame )

{

std::vector<Rect> faces;

Mat frame\_gray;

cvtColor( frame, frame\_gray, COLOR\_BGR2GRAY );//因为用的是泪hear特征，所以都是基于灰度图像的，这里要转换成灰度图像

equalizeHist( frame\_gray, frame\_gray );//直方图均衡

//人脸检测

face\_cascade.detectMultiScale( frame\_gray, faces, 1.1, 2, 0|CASCADE\_SCALE\_IMAGE, Size(30, 30) );

for( size\_t i = 0; i < faces.size(); i++ )

{

Point center( faces[i].x + faces[i].width/2, faces[i].y + faces[i].height/2 );

ellipse( frame, center, Size( faces[i].width/2, faces[i].height/2), 0, 0, 360, Scalar( 255, 0, 255 ), 2, 8, 0 );

Mat faceROI = frame\_gray( faces[i] );

std::vector<Rect> eyes;

//-- 在人脸中检测眼睛

eyes\_cascade.detectMultiScale( faceROI, eyes, 1.1, 2, 0|CASCADE\_SCALE\_IMAGE, Size(30, 30) );

for( size\_t j = 0; j < eyes.size(); j++ )

{

Point eye\_center( faces[i].x + eyes[j].x + eyes[j].width/2, faces[i].y + eyes[j].y + eyes[j].height/2 );

int radius = cvRound( (eyes[j].width + eyes[j].height)\*0.25 );

circle( frame, eye\_center, radius, Scalar( 255, 0, 0 ), 3, 8, 0 );

}

}

//-- 显示最终效果图

imshow("人联识别",frame);//显示当前帧

}

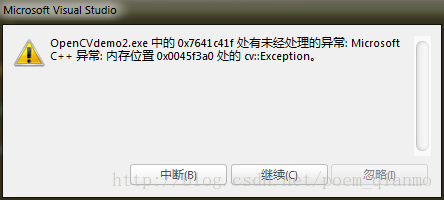
# 调试分析

出现如下错误：

1、有时会出现闪退的情况，一般是出现在第一次运行程序的时候。

解决问题：这种问题最简单的方法就是再次重新运行一下，一般情况下第二次运行都可以实现。但有时候可能就失灵了，这是调用摄像头出现的问题，运行程序后，摄像头并没有打开，所以就闪退了。重新启动计算机是不错的办发。

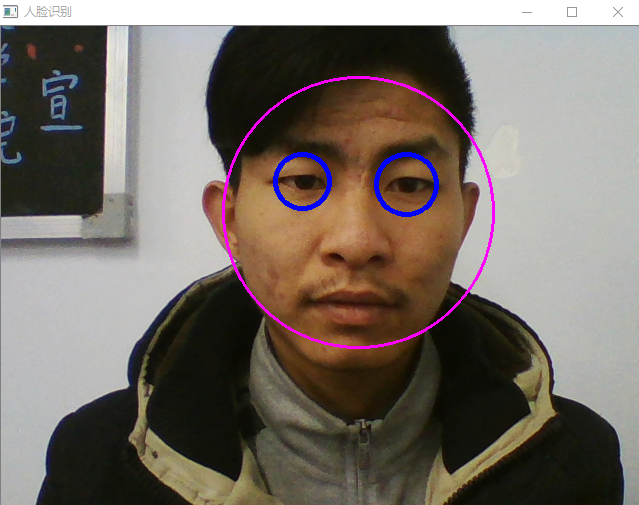
出现如下错误：



这算是opencv的一个bug，工程属性里面关于带d和不带d的lib文件的附加依赖项的问题，就算配置好了每次想debug和release下都运行还得手动在工程属性里面加。当得到这样的错误时，可以把调试方式改一改，debug和release互换；

# 第4章 使用说明与执行结果

**4.1运行操作及结果:**



**4.2课程设计小结**

**在实验中发现——分析——总结的问题有：**

对于opencv的学习也有一段时间了，但还只是学到了皮毛，从最一开始学习对于图像的对比度，亮度的调节，图像的载入，显示与输出，到缩放图片再到特征点匹配，到最后图片的人脸识别，自己一步一步去找代码，然后修改代码，积累了一定的经验，对于简单的代码修改还是可以做到的。最后利用摄像头来读入视频并检测人脸和眼睛，从静态到动态，觉得自己学到了很多。最后感谢老师的引导。

Github：https://github.com/MiaoChaoran