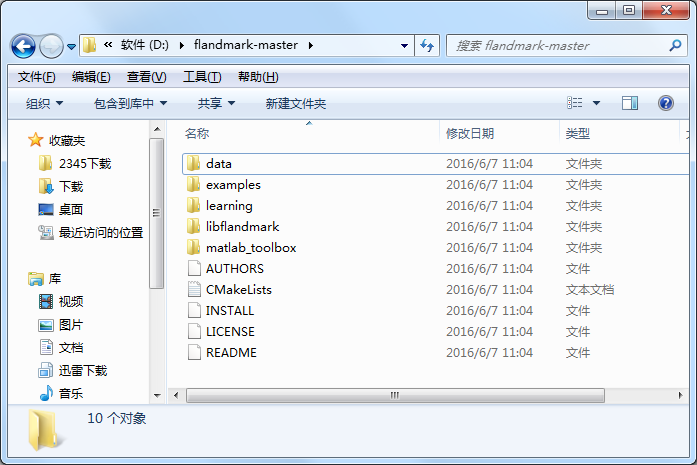
# 使用flandmark提取人脸地标点

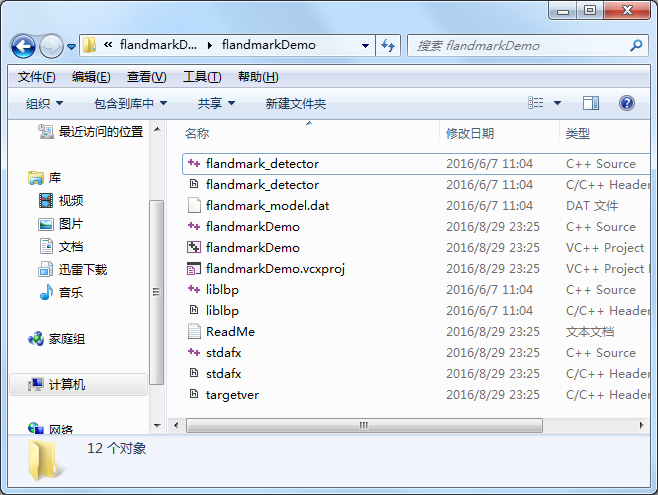
Flandmark是一个使用C语言编写的人脸地标点检测器，下载地址：<https://github.com/uricamic/flandmark>，下载完成后，解压得到以下文件：



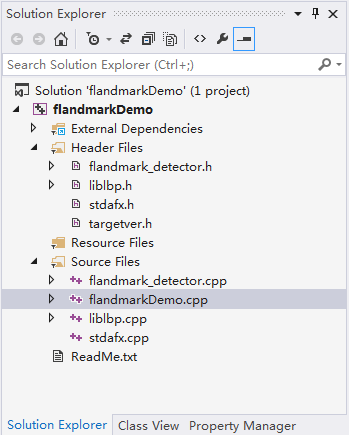
另外，该库需要用到OpenCV，因此确保OpenCV已正确配置。

现在开始我们来完成一个简单的示例程序。

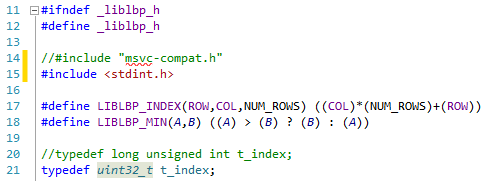
新建一个控制台工程，工程名称为flandmarkDemo，将libflandmark目录下的文件flandmark\_detect.h, flandmark\_detect.cpp, liblbp.cpp, liblbp.h文件和data目录下的flandmark\_model.dat文件拷贝到工程目录下，最后的工程目录下的文件如图所示：



将flandmark\_detect.h, flandmark\_detect.cpp, liblbp.cpp, liblbp.h文件添加到工程中：



对部分文件进行修改。打开liblbp.h文件，将#include "msvc-compat.h"修改成：#include <stdint.h>，如下图所示：



将flandmark\_detector.h中的

#include "msvc-compat.h"

#include "opencv2/core/core\_c.h"

#include "opencv2/imgproc/types\_c.h"

修改成

#include <stdint.h>

#include <opencv2/opencv.hpp>

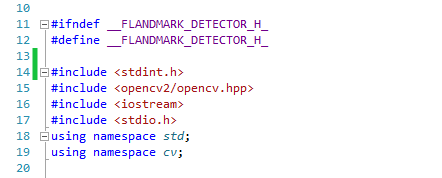
#include <iostream>

#include <stdio.h>

using namespace *std*;

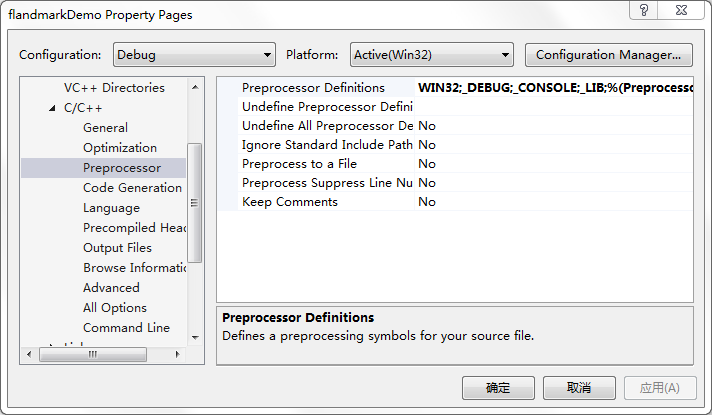
using namespace *cv*;

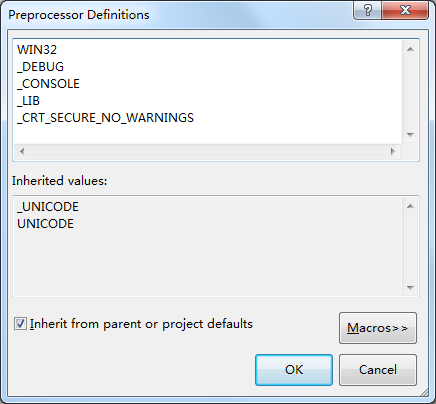
如下图所示：



在flandmark\_detector.cpp和liblbp.cpp中添加头文件：#include "stdafx.h"。

选择在工程属性中C/C++下的Preprocessor,在右侧的Preprocessor Definitions中添加\_CRT\_SECURE\_NO\_WARNINGS，如下图所示：





编写主程序：

#include "stdafx.h"

#include "flandmark\_detector.h"

#include <opencv2/opencv.hpp>

#include <iostream>

using namespace *cv*;

using namespace *std*;

#include <cmath>

#include "flandmark\_detector.h"

int *main*(int argc, char\*\* argv)

{

char flandmark\_window[] = "flandmark\_simple\_example";

int \* bbox = (int\*)*malloc*(4 \* sizeof(int));

*cvNamedWindow*(flandmark\_window, 0 );

FLANDMARK\_Model \* model = flandmark\_init("flandmark\_model.dat");

*Mat* img = *imread*("Lucy\_Liu\_0003.jpg");

if (img.*empty*())

{

*fprintf*(*stderr*, "Cannot open image %s. Exiting...\n", argv[1]);

*exit*(1);

}

*Mat* img\_grayscale;

*cvtColor*(img, img\_grayscale, *CV\_RGB2GRAY*);

bbox[0] = 71;

bbox[1] = 81;

bbox[2] = 171;

bbox[3] = 181;

double \* landmarks = (double\*)*malloc*(2 \* model->data.options.M\*sizeof(double));

if (flandmark\_detect(new *IplImage*(img\_grayscale), bbox, model, landmarks))

{

*printf*("Error during detection.\n");

}

*rectangle*(img, *Point*(bbox[0], bbox[1]), *Point*(bbox[2], bbox[3]), *Scalar*(255, 0, 0));

*rectangle*(img, *Point*(model->bb[0], model->bb[1]), *Point*(model->bb[2], model->bb[3]), *Scalar*(255, 0, 255));

*circle*(img, *Point*((int)landmarks[0], (int)landmarks[1]), 3, *Scalar*(0, 255, 0), -1);

for (int i = 2; i < 2 \* model->data.options.M; i += 2)

{

*circle*(img, *Point*(int(landmarks[i]), int(landmarks[i + 1])), 3, *Scalar*(255,0,0), -1);

}

*imshow*(flandmark\_window, img);

*waitKey*(0);

*free*(landmarks);

*free*(bbox);

flandmark\_free(model);

}

测试使用的图片来自于flandmark中的data\images目录下的图像，运行结果：

