

Opencv 人脸识别小结

一、实验目的

阅读并运行所给代码，观察运行结果；
更换不同验证图像，观察运行结果；
修改代码关键参数后，再次观察结果；
进行人脸检测模块训练，验证结果。

二、准备工作

首先下载 Opencv 与 VS。本次实验环境为 Opencv 3.4.0、VS 2019。

完成后，配置环境。由于所给代码版本问题，在配置过程中遇到了困难，因此下面详述配置过程，以便给后来者做参考。

2.1 配置系统变量(以 win10 为例)

桌面上找到此电脑，依次点开右键、属性、高级系统设置、环境变量。新增变量，将 opencv\build\x64\vc14\bin 路径添加到其中，如图所示。

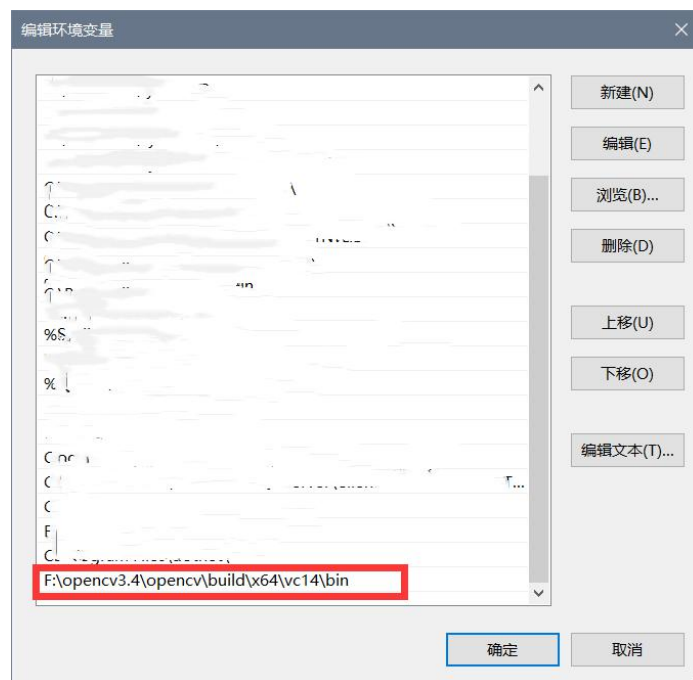


图 2.1

将 bin 目录下面的 opencv_world341.dll 和 opencv_world341d.dll 文件复制到 C:\Windows\SysWOW64 这个文件夹里，然后将 bin 目录里面的

opencv_ffmpeg341_64.dll 复制到 C:\Windows\System32 这个文件夹里面。

2.2 配置编译器(以 VS 2019 为例)

新建一个 C++ 项目，进入属性管理器，对 Debug|x64 进行配置，右键点击，再点击属性。这里我们对属性中的 VC++ 目录下包含目录和 VC++ 目录下库目录进行添加相关路径，并对链接器下输入下附加依赖项进行添加相关路径。

VC++ 目录下，包含目录中加入：

F:\opencv3.4\build\include

F:\opencv3.4\build\include\opencv

F:\opencv3.4\build\include\opencv2

VC++ 目录下，库目录中加入：F:\opencv3.4\build\x64\vc14\lib

链接器下输入下附加依赖项中加入：opencv_world340d.lib（即 2.1 中复制到 C 盘中的文件）

2.3 准备测试用例与修改错误代码

将图 2.2 中的三个文件复制到新建项目文件夹中。

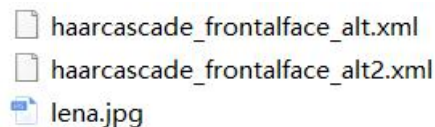


图 2.2

之后，新建 cpp 文件，命名为 face.cpp 将 facedetecttest.cpp 中的代码复制到 face.cpp 之中。

为了顺利运行，在代码中，有两处错误需要进行修改。

第一处，修改包含的 opencv 库文件，修改结果如图所示。

```
#include <opencv/cv.h>
#include <opencv/highgui.h>
```

图 2.3

第二处，修改 CvScalar 对象的初始化，修改结果如图所示。

```
static CvScalar colors[] =  
{  
    {0, 0, 255},  
    {0, 128, 255},  
    {0, 255, 255},  
    {0, 255, 0},  
    {255, 128, 0},  
    {255, 255, 0},  
    {255, 0, 0},  
    {255, 0, 255}  
};
```

图 2.4

三、人脸检测运行结果

3.1 以默认用例运行

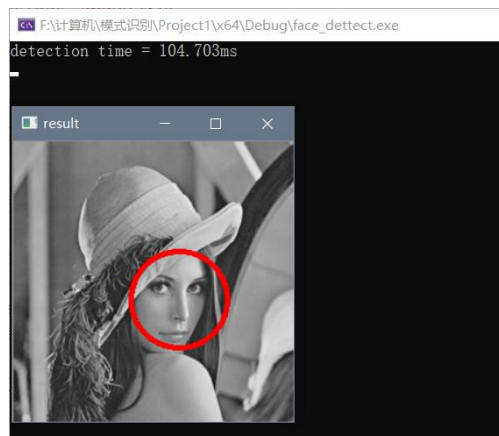


图 3.1

3.2 更换图片运行

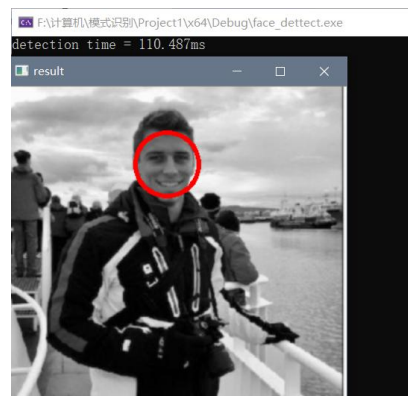


图 3.2

3.3 更改参数运行

将 `cvsize` 更改到 (1,1) 仍能识别, 改到 (60,60) 时图像无法识别。这是由于该参数表示检测窗口最小尺寸, 如果框选区域过大, 检测目标附件的相邻矩形较少, 没有符合 `min_neighbors` 的识别框。但如果将 `min_neighbors` 改为 0, 此时函数不做任何操作就返回所有的被检候选矩形框。此时重现检测框, 明显比 (30,30) 偏大。

如果将 1.1 改为 1.5, 可以发现图 3.5 中出现了两个框, 这是因为在前后两次相继的扫描中, 搜索窗口依次扩大 50%, 此时有两个框符合 `min_neighbors` 为 1 的条件, 如果将 `cvsize` 改为 (60, 60), `min_neighbors` 改为 0, 那么图像会剩下一个较大的框。

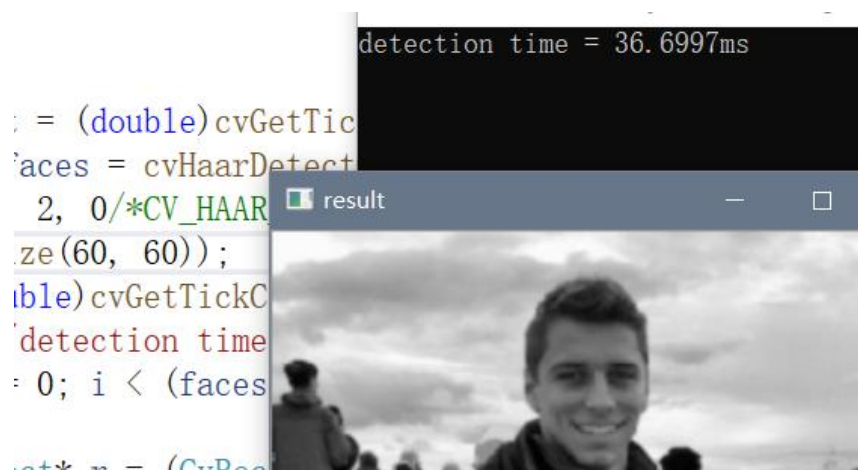


图 3.3

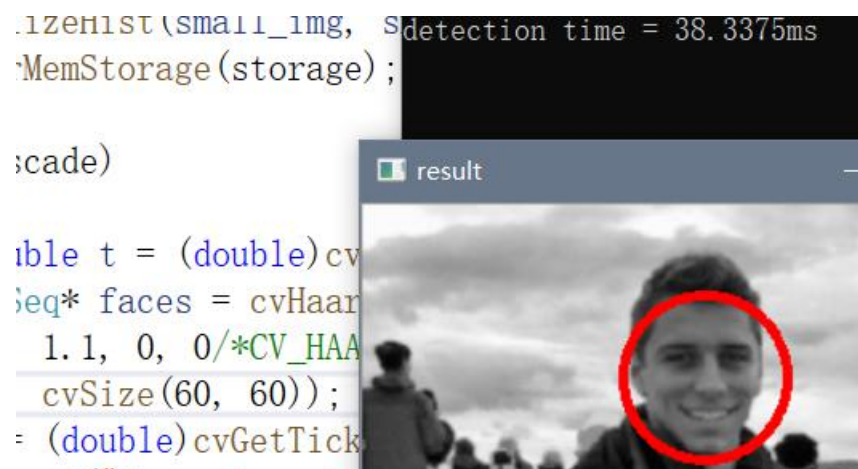


图 3.4

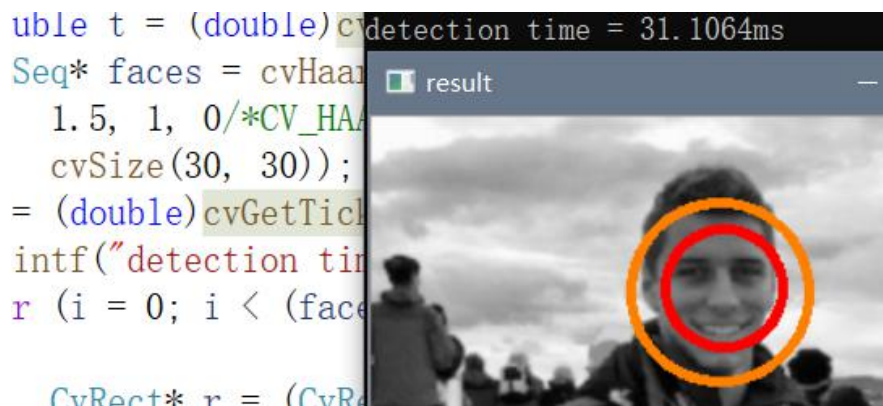


图 3.5

四、人脸检测分类器训练

由于版本更新，haartraining.exe 不再出现在 Opencv 的 release 中，改用 traincascade。

在 opencv 的安装目录中的 bin 文件夹下有两个可执行文件 opencv_createsamples.exe 和 opencv_traincascade.exe。将这两个文件拷贝到训练文件夹下，并将正、负样本的文件夹和描述文件——trainpos.txt 和 trainneg.txt 也拷贝到这个文件夹下。同时，新建两个.bat 文件——create_positive_samples.bat 和 traincascade.bat，在 negatives 中新建一个文件夹 re。这样，训练目录如下：

facedetection	2022/2/20 13:06	文件夹	
negatives	2022/2/20 15:03	文件夹	
positives	2022/2/20 14:59	文件夹	
positives1	2022/2/20 13:06	文件夹	
test	2022/2/20 13:06	文件夹	
create_positive_samples.bat	2022/2/20 14:58	Windows 批处理...	1 KB
opencv_createsamples.exe	2017/12/23 5:23	应用程序	58 KB
opencv_traincascade.exe	2017/12/23 5:23	应用程序	379 KB
pos.vec	2022/2/20 14:59	VEC 文件	313 KB
traincascade.bat	2022/2/20 14:53	Windows 批处理...	1 KB

图 4.1

然后在 create_positive_samples.bat 文件中添加如下内容：

```
opencv_createsamples.exe -info trainpos.txt -vec pos.vec -num 100 -w 40 -h 40
```

然后在 traincascade.bat 文件中添加如下内容：

```
opencv_traincascade.exe -data classifier -vec pos.vec -bg trainneg.txt -numStages
10 -minHitRate 0.999 -maxFalseAlarmRate 0.5 -numPos 1400 -numNeg 1000-w 40
-h 40 -mode ALL -precalcValBufSize 1024 -precalcIdxBufSize 1024 -featureType
LBP
```

分别运行 create_positive_samples.bat 和 traincascade.bat，经过一段时间训练之后，可以看见 re 文件夹中出现了训练好的模型。

params.xml	2022/2/20 15:08	XML 文档
stage0.xml	2022/2/20 15:08	XML 文档
stage1.xml	2022/2/20 15:08	XML 文档
stage2.xml	2022/2/20 15:08	XML 文档
stage3.xml	2022/2/20 15:08	XML 文档
stage4.xml	2022/2/20 15:08	XML 文档
stage5.xml	2022/2/20 15:08	XML 文档
stage6.xml	2022/2/20 15:08	XML 文档
stage7.xml	2022/2/20 15:08	XML 文档
stage8.xml	2022/2/20 15:08	XML 文档
stage9.xml	2022/2/20 15:08	XML 文档
stage10.xml	2022/2/20 15:08	XML 文档
stage11.xml	2022/2/20 15:08	XML 文档
stage12.xml	2022/2/20 15:08	XML 文档
stage13.xml	2022/2/20 15:08	XML 文档
stage14.xml	2022/2/20 15:08	XML 文档
stage15.xml	2022/2/20 15:08	XML 文档
stage16.xml	2022/2/20 15:09	XML 文档
stage17.xml	2022/2/20 15:09	XML 文档
stage18.xml	2022/2/20 15:11	XML 文档
stage19.xml	2022/2/20 15:14	XML 文档

图 4. 2

复制模型 params 到标题三的项目文件夹下，重命名为 mywork，并修改代码。试验模型效果。发现训练的模型也能成功实现人脸检测。

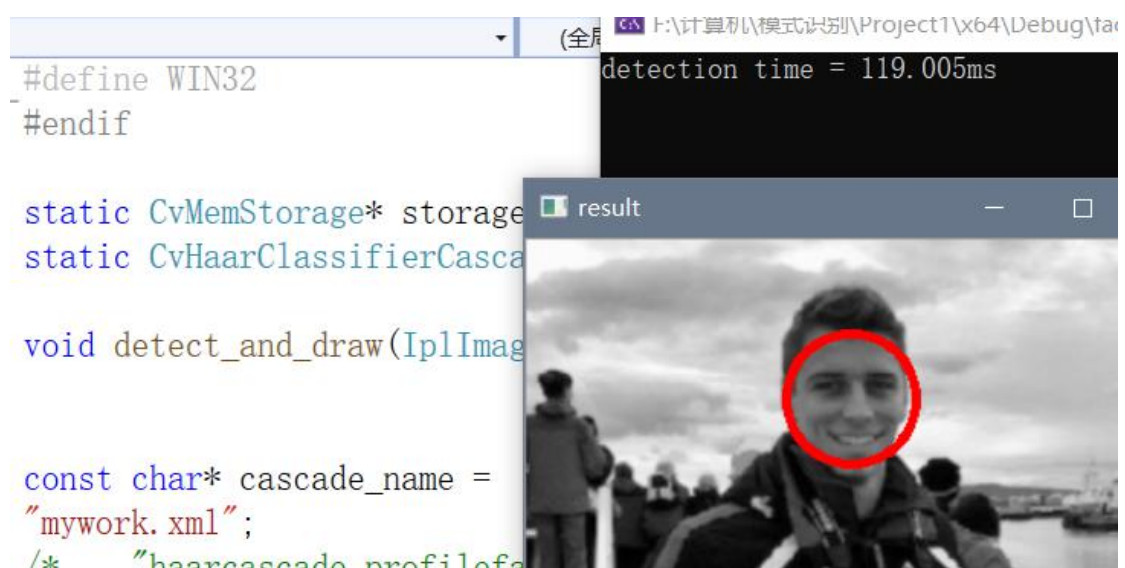


图 4. 3