创建一个文件夹，命名为人脸识别，在人脸识别文件夹中新建两个文件夹pos和neg,分别存放正样本和负样本,

将opencv\_traincascade.exe和opencv\_createsamples.exe从opencv的bin文件夹中拷贝至该文件夹中。

在pos文件下新建一个txt文件，在txt文件中输入dir /b > pos.txt后保存，改后缀名为.bat生成脚本文件，双击该.bat文件运行可生成一个pos.txt。

在pos.txt文本中将所有的jpg替换成jpg 1 0 0 16 16。jpg后面的五个数字信息分别表示样本中目标个数、目标在图像中的起始位置x、y、样本的尺寸大小width、height。因为我的正样本只包含目标，所以样本的起始位置为0 0，训练之前最好将正样本规格化为同一尺寸。

pos.txt文件中不应该包含除样本实例之外的条目，刚生成之后的pos.txt后面必然会有之前.bat的条目以及pos.txt自身的条目，将这两个条目删除。

在人脸识别目录下新建一个createsamples.txt文档，输入opencv\_createsamples.exe -info pos/pos.txt -vec pos.vec -num 200 -w 16 -h 16 pause

-info表示指定的描述文件，-vec指定输出的vec文件名，-w-h表示规定的样本尺寸，-num表示创建的样本数目

修改后缀名为.bat生成脚本文件createsamples.bat文件，双击运行生成了正样本pos.vec文件

在neg文件夹下新建一个txt文件编辑内容为dir /b /s > neg.txt（加上/s指令是为了生成绝对路径），修改后缀名为.bat生成脚本文件，双击运行后生成neg.txt文件。

在记事本中编辑neg.txt，将所有”\“替换成”/“。将完成好的neg.txt放在人脸识别目录下，供后面的训练使用。

训练分类器

在人脸识别目录下新建traincascade.txt文件，输入opencv\_traincascade.exe -data data -vec pos.vec -bg neg.txt -numPos 200 -numNeg 300 -numStages 5 -w 16 -h 16 -minHitRate 0.999 maxFalseAlarmRate 0.5 -precalcValBufSize 256 -precalcIdxBufSize 256 -mode ALL pause

修改后缀名为.bat生成脚本文件traincascade.bat，双击运行

所有的stage全部训练完毕才表示训练完成，最终会生成一个总的cascade.xml文件，检测的时候只需要这一个.xml文件就可以完成检测。



测试训练效果

在OpenCV中写入代码

代码：

#include<opencv2\opencv.hpp>

#include<iostream>

using namespace cv;

using namespace std;

String filename = "C:\cascade.xml";

CascadeClassifier face\_classifier;

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

{

if (!face\_classifier.load(filename))

{

printf("could not laod face feature data..\n");

return -1;

}

Mat src = imread("C\1.jpg");

if (src.empty())

{

printf("could not laod image...\n");

return -1;

}

imshow("inupt image", src);

Mat gray;

cvtColor(src, gray, COLOR\_BGR2GRAY);

equalizeHist(gray, gray);

vector<Rect>faces;

face\_classifier.detectMultiScale(gray, faces, 1.1, 3, 0, Size(50, 50));

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

{

rectangle(src, faces[static\_cast<int>(t)], Scalar(0, 0, 255), 2, 8, 0);

}

imshow("detect faces", src);

waitKey(0);

return 0;

}

训练效果：

