# 一．收集样本

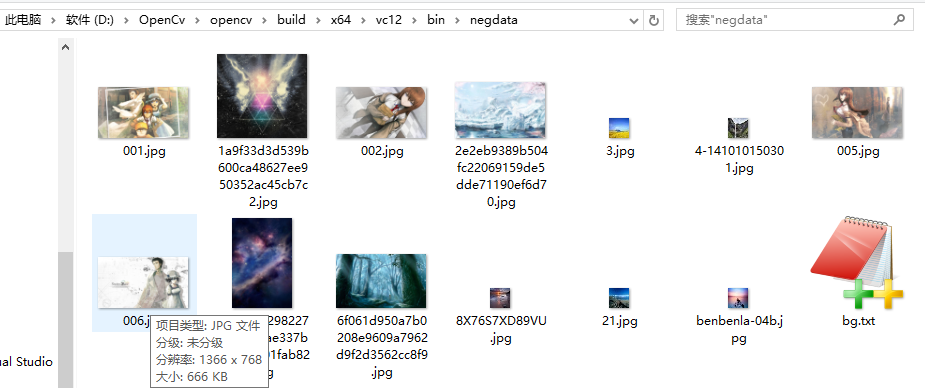
正样本：待检目标样本

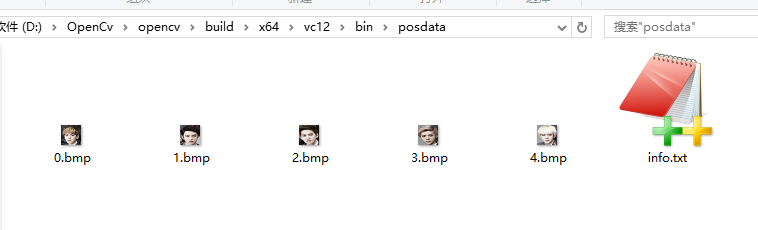
负样本：其它任意图片

PS.所有样本图片都应该有同一尺寸，如20 \* 20

ACDSee软件：对图像批量处理为20x20大小的图片

## 1.把所有正样本图片放在posdata的文件夹下，把所有负样本图片放在negdata文件夹下





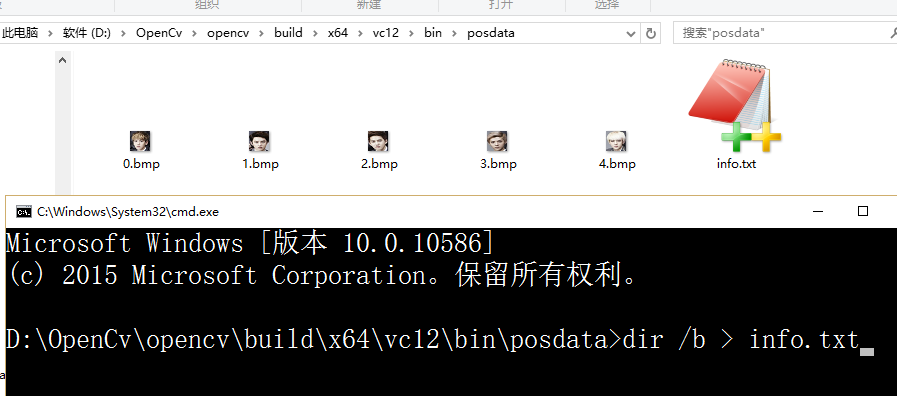
负样本数量应大于正样本数量，如(3:1)，否则在训练时会陷入死循环。。。

## 1.分别为正样本和负样本创建描述文件

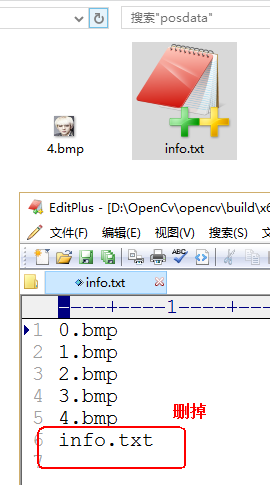
(1)为正样本创建描述文件格式文件info.txt，并且把这个文件放在与样本图片同一目录下,D:\OpenCv\opencv\build\x64\

vc12\bin\posdata

1. 在命令行下 输入以下命令： dir /b > info.txt

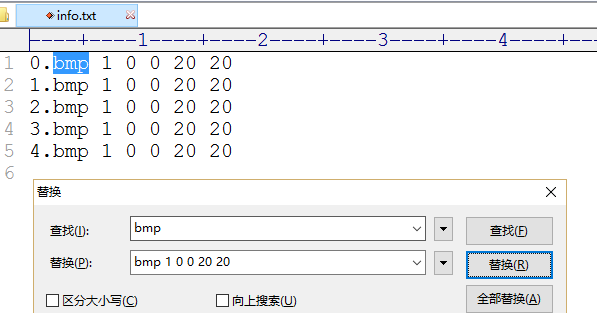


b)删除info.txt最后一行的 “info.txt”



c)打开info.txt, 按ctrl+h, 把所有的”bmp” 换成

“bmp 1 0 0 20 20”

1 代表一个文件 || 0 0 顶点坐标 || 20 20 右下顶点坐标 

 (2)为负样本创建集合文件格式文件bg.txt, 并且把这个文件放在与样本图片同一目录下

a)在命令行下 输入以下命令： dir /b > bg.txt

b)删除bg.txt最后一行的 “bg.txt”

# 创建样本

代码：

opencv\_createsamples.exe -info D:\OpenCv\opencv\build\x64\vc12\bin\posdata\info.txt -vec D:\OpenCv\opencv\build\x64\vc12\bin\a.vec -num 5 -w 20 -h 20

Ps: createsamples程序的命令行参数

|  |  |
| --- | --- |
| info<file> | 正样本描述文件 |
| vec <vec\_file\_name> | 训练好的正样本的输出文件名 |
| **num<number\_of\_samples>** | **要产生的正样本的数量，和正样本图片数目相同** |
| －w －h | 样本的宽高 |

执行脚本后，将生成正样本描述文件a.vec

-------------------------------------------------------------

其他的一些参数：

－img<image\_file\_name>

源目标图片（例如：一个公司图标）

－bg<background\_file\_name>

背景描述文件。

--maxidev<max\_intensity\_deviation>

背景色最大的偏离度。

－maxangel<max\_x\_rotation\_angle>

－maxangle<max\_y\_rotation\_angle>，

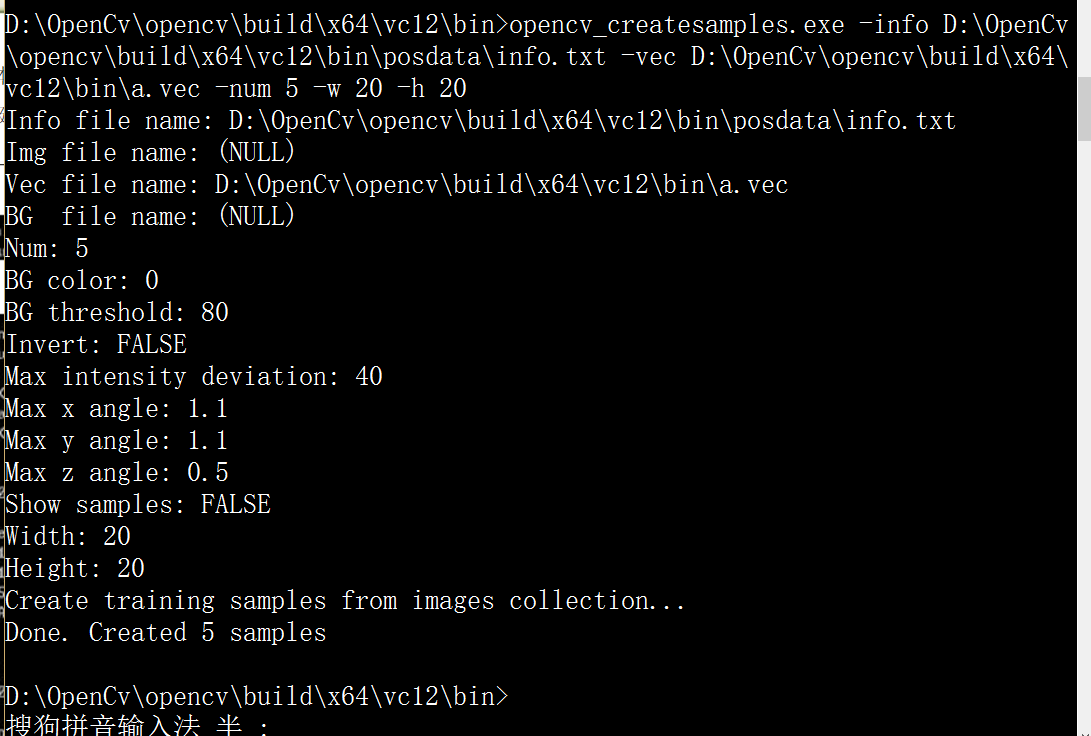
－maxzangle<max\_x\_rotation\_angle>

最大旋转角度，以弧度为单位。

－show

如果指定，每个样本会被显示出来，按下"esc"会关闭这一开关，即不显示样本图片，而创建过程继续。这是个有用的debug选项。

结果：



# 训练级联分类器

代码：

opencv\_haartraining.exe -data data -vec D:\OpenCv\opencv\build\x64\vc12\bin\a.vec -bg D:\OpenCv\opencv\build\x64\vc12\bin\negdata\bg.txt -npos 5 -nneg 13 -nstages 4 -nsplits 2 -mem 512 -nonsym -w 20 -h 20 -minpos 100

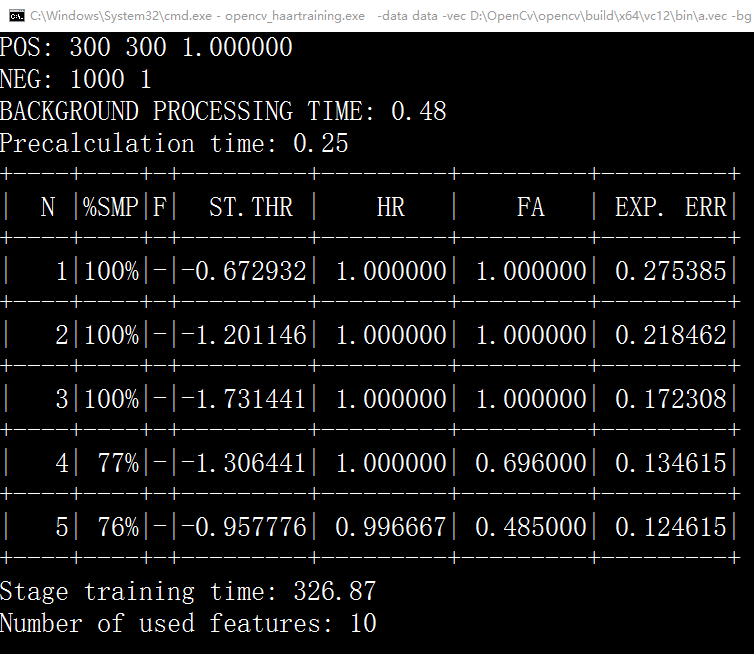
Ps: Haartraining的命令行参数如下

|  |  |
| --- | --- |
| data<dir\_name> | 存放训练好的分类器的路径名 |
| vec<vec\_file\_name> | 正样本文件名（由trainingssamples程序或者由其他的方法创建的）绝对路径 |
| bg<background\_file\_name> | 背景描述文件。绝对路径 |
| npos<number\_of\_positive\_samples> | 用来训练每一个分类器阶段的正样本数量Ps:正样本的数目(-npos)要略大于实际数 |
| nneg<number\_of\_negative\_samples> | 用来训练每一个分类器阶段的负样本数量 |
| nstages<number\_of\_stages> | 训练的阶段(层)数 |
| nsplits<number\_of\_splits> | 决定用于阶段分类器的弱分类器。如果值为1，则一个简单的stump classifier被使用。如果是2或者更多，则带有number\_of\_splits个内部节点的CART分类器被使用 |
| mem<memory\_in\_MB> | 预先计算的以MB为单位的可用内存。内存越大则训练的速度越快 |
| nonsym | 指定训练的目标对象是否垂直对称。垂直对称提高目标的训练速度。例如，正面部是垂直对称的 |
| mode all | 表示使用haar特征集的各类既有垂直的，又有45度角旋转的 |
| －w<sample\_width> －h<sample\_height> | 训练样本的尺寸，（以像素为单位）。必须和训练样本创建的尺寸相同 |

其他的一些参数：  
－sym（default）  
－minhitrate<min\_hit\_rate>  
    每个阶段分类器需要的最小的命中率。总的命中率为min\_hit\_rate的number\_of\_stages次方。  
－maxfalsealarm<max\_false\_alarm\_rate>  
    没有阶段分类器的最大错误报警率。总的错误警告率为 max\_false\_alarm\_rate的number\_of\_stages次方。每一层训练到这 个值小于0.5时训练结束，进入下一层训练  
－weighttrimming<weight\_trimming>  
    指定是否使用权修正和使用多大的权修正。一个基本的选择是0.9  
－eqw  
－mode<basic(default)|core|all>  
    选择用来训练的haar特征集的种类。basic仅仅使用垂直特征。all使用垂直和45度角旋转特征。

* 参数的配置十分重要，很多时候训练不成功多是因为参数的配置问题！！！
* -----------------------------------------------------------

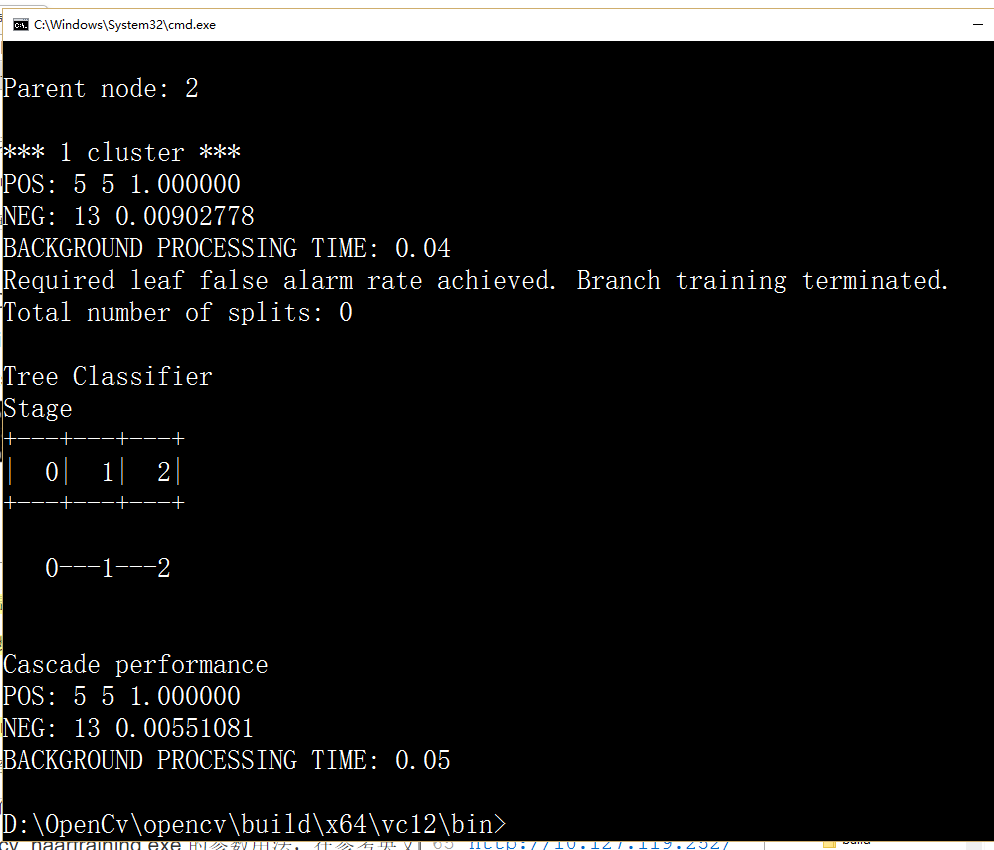
训练过程参数值解释：

* 

N：层数 %

* SMP：样本的使用率
* F : +表示通过翻转，否则是-
* ST.THR : 分类器的阈值
* HR：当前分类器 对正样本识别正确的概率
* FA：当前分类器 对负样本识别错误的概率
* EXP.ERR : 分类器的期望错误率
* -----------------------------------------------------------

结果：



# 四．测试

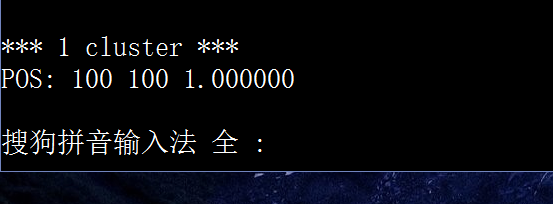
测试代码：人脸识别

因样本数量仅为 5 所以无精度可言



# 错误分析

## 死循环



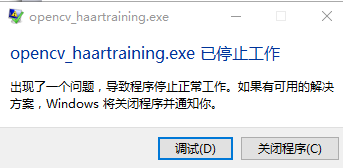
**原因：**

负样本数太少，该阶段已无可提取负样本

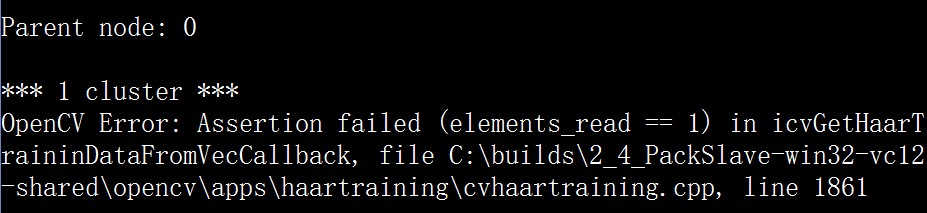
**解决方法：**

正 : 负 = 1 : 2.5~3

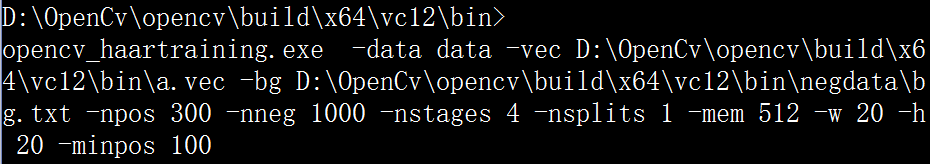
## 2.停止工作



**报错信息：**



**参数配置**：



**原因：**

在第一层训练的时候已经用完了vec文件中所有的正样本，以至于在后续的样本训练中没有新的正样本可以加入使用了，导致出现了错误的现象。

那个vec文件中的正样本数目

= npos +（nstages - 1）\*（1 - minHitRate）\* npos + s

S的意思是指在正样本中能够直接识别成背景的样本个数，如果npos=120的话，s的取值为个位数就行，不用太大。

**解决方法：**

在训练时，-npos参数值取 正样本数的 1/3

如 实际正样本数为 300 参数取值则设为 100

参考资料

[OpenCV 训练分类器生成XML文件](http://m.blog.csdn.net/article/details?id=10155813)

[训练分类器](http://wenku.baidu.com/view/9b8ea1d9d15abe23482f4d0a.html)

[训练参数设置](http://blog.csdn.net/liulina603/article/details/8197889)

[英语-训练分类器,参数设置,人脸检测](http://note.sonots.com/SciSoftware/haartraining.html?spm=5176.100239.blogcont.2.w6zgvQ" \l "e134e74e)