Adaboost 和 Gentle Boost 简要说明

Adaboost可以看成一个加权分类器。它核心的灵魂在于每次挑选表现最好的hiphosesis,再根据它的错误率计算它的权重。

不仅对分类器加权，对分类样本也加权，而加权的依据就是本轮分类命中与否，假如命中失败则提高这个样本的权重，否则相应降低其权值，这样做的依据是，对前一个弱分类器分类错误的样本，后来的弱分类器进行重点关注。

Gentle Boost和Adaboost的主要区别则是在于从弱分类器获得分类信息以后，权值的调整方法；以及分类是采取古典集合（discret）还是模糊集合（其他三种）。

此次实验我们训练的样本是手提电脑,共使用了81个正样本.



图1 正样本

负样本则使用了各种不相关的背景,总计467个.

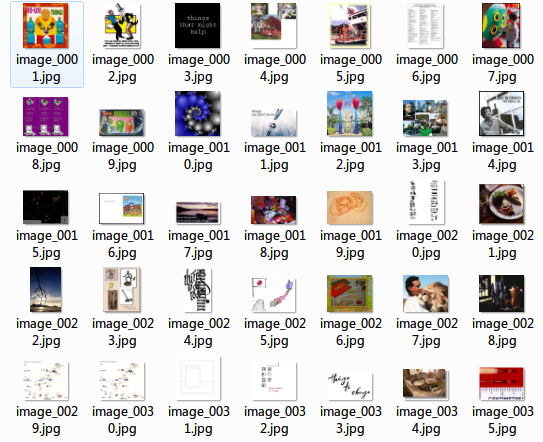


图2 负样本

Adaboost训练方法:

因为OpenCV自带了训练分类器,所以我们能很轻松的训练自己的分类器.

具体步骤为

1.建立正负样本的描述文件

2.使用CreateSamples程序建立正样本的二进制数据文件

opencv\_createsamples.exe -vec pos.vec -info pos.txt -bg neg.txt -w 40 -h 40 -num 81

3.使用haartraining训练,最后得到xml文件

opencv\_haartraining -data data -vec pos.vec -bg neg.txt -npos 81 -nneg 467 -w 40 -h 40 -mem 500 -mode ALL

4.程序使用xml文件作为分类器,识别图像.

匹配代码:

void detect\_and\_draw( IplImage\* img )

{

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}}

};

double scale = 1.3;

IplImage\* gray = cvCreateImage( cvSize(img->width,img->height), 8, 1 );

IplImage\* small\_img = cvCreateImage( cvSize( cvRound (img->width/scale),

cvRound (img->height/scale)),

8, 1 );

int i;

cvCvtColor( img, gray, CV\_BGR2GRAY );

cvResize( gray, small\_img, CV\_INTER\_LINEAR );

cvEqualizeHist( small\_img, small\_img );

cvClearMemStorage( storage );

if( cascade )

{

double t = (double)cvGetTickCount();

CvSeq\* faces = cvHaarDetectObjects( small\_img, cascade, storage,

1.1, 2, 0/\*CV\_HAAR\_DO\_CANNY\_PRUNING\*/,

cvSize(30, 30) );

t = (double)cvGetTickCount() - t;

printf( "detection time = %gms\n", t/((double)cvGetTickFrequency()\*1000.) );

for( i = 0; i < (faces ? faces->total : 0); i++ )

{

CvRect\* r = (CvRect\*)cvGetSeqElem( faces, i );

CvPoint center;

int radius;

center.x = cvRound((r->x + r->width\*0.5)\*scale);

center.y = cvRound((r->y + r->height\*0.5)\*scale);

radius = cvRound((r->width + r->height)\*0.25\*scale);

cvCircle( img, center, radius, colors[i%8], 3, 8, 0 );

}

}

cvShowImage( "result", img );

cvReleaseImage( &gray );

cvReleaseImage( &small\_img );

}

实验结果:



图3 检测结果1

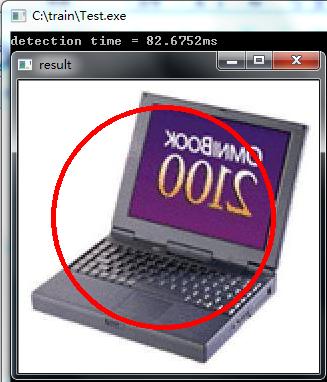


图4 检测结果2



图5 检测结果3

图片的识别速度还是可以接受的,可是识别的正确率并不高,从图3,图5可以看书,只要有明显分界而且分界上下有区别的图形区域,都会误识别为笔记本.估计和正样本中的笔记本电脑的屏幕上显示的东西不同有关.在进行过的多次实验中,也可以看出正样本的准确性和负样本的质量对最后的识别正确率影响很大.

(实验步骤参考至:

<http://hi.baidu.com/zdd007007/blog/item/b2e7f026eec9e23f8644f959.html/cmtid/c9bc5541e80cc91f9213c6f7> )

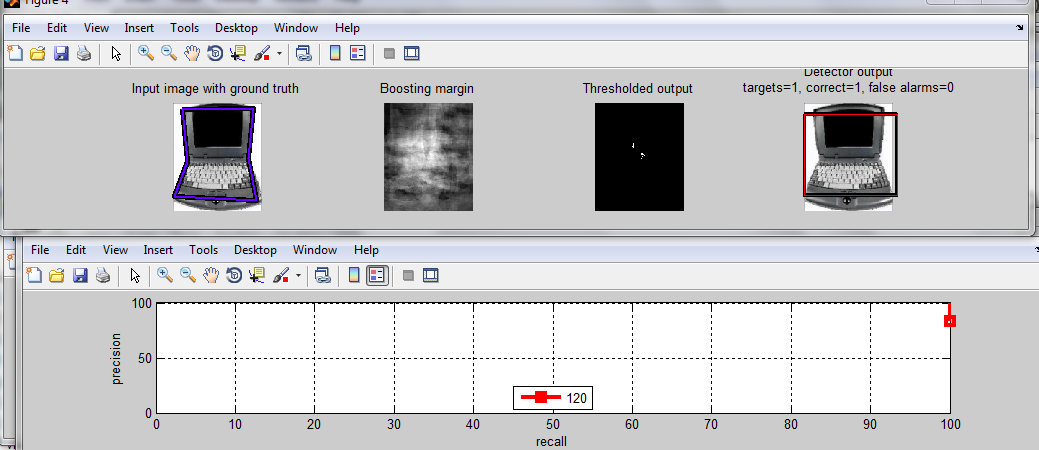
Gentleboost 训练步骤:

在<http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html> 上提供了一个基于MATLAB的教学工具，该工具提供了训练Gentleboost分类器并对其进行测试的功能，直接利用该工具进行实验。

1. 建立正样本的描述文件：

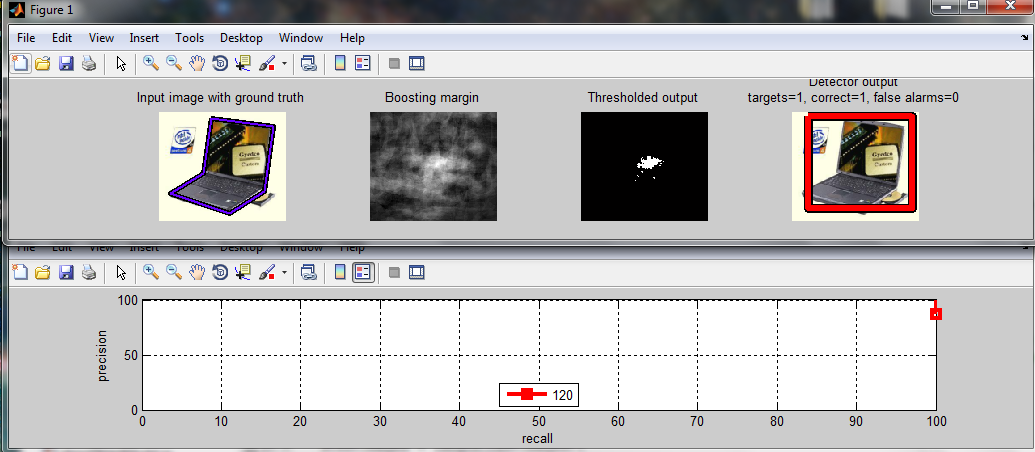
将样本图片上传到 <http://labelme.csail.mit.edu/guidelines.html> 上，使用该网站提供的工具，得到相应的xml描述文件。

1. 在 <http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html> 上下载代码和LabelMe toolbox。
2. 在initpath.m中，修改LabelMe toolbox的路径；在paremeters.m 中，修改图片和描述文件路径，将objects 改为laptop、paramfile 改为 'demoLaptop' ，并自定义sampleFromImages、numTrainImages等数据。
3. 在MatLab 上依次运行 initpath.、paremeters.m、createDatabase.m、createDictionary.m、computeFeatures.m、trainDetector.m。
4. 运行runDetector.m，程序使用分类器,识别图像。

**实验结果(与Adaboost对比)：**

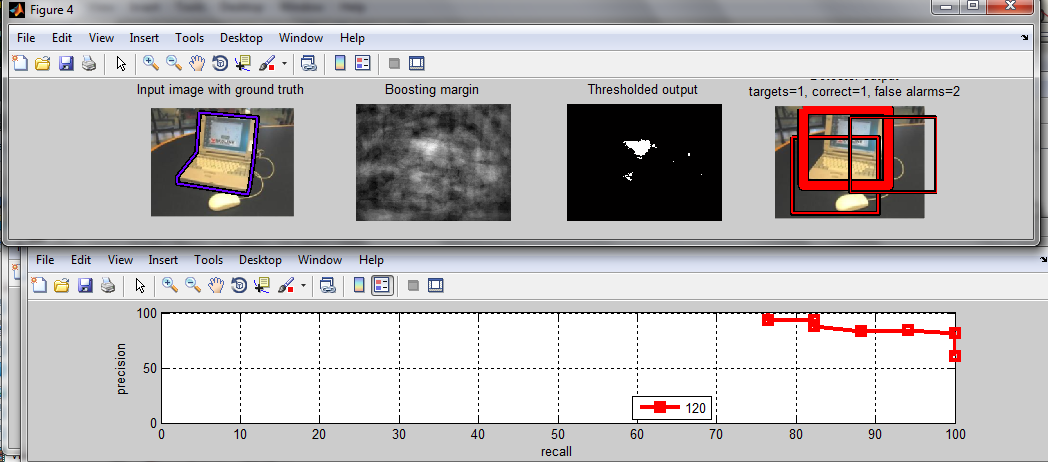


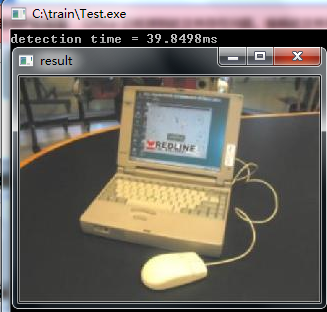
样本中的笔记本电脑多为侧向,正向的笔记本电脑Adaboost很难正确识别,而Gentle Boost则可以识别.



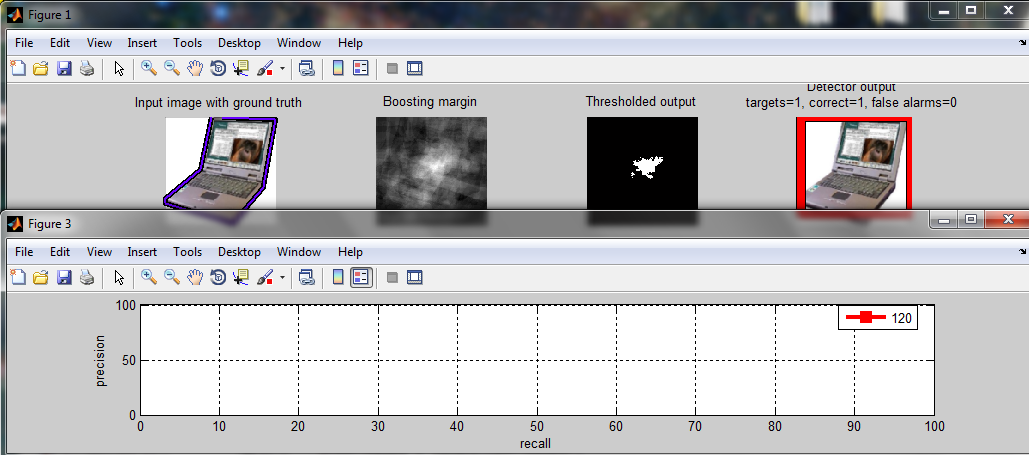


AdaBoost的识别错误率比Gentle Boost要高.





Gentle Boost虽然识别错误,但是笔记本还是能成功识别的.而Adaboost则无法识别.



两种算法都能识别,可是识别区域显然Gentle Boost要好.

**对比4组实验结果,可以看出调整了样本权值的Gentle Boost,识别率与Adaboost相比要高上不上.不过我们在进行Adaboost训练时,对样本的圈定并没有Gentle Boost那么精细,这可能也是导致Adaboost表现不佳的原因.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 姓名 | 学号 | 班级 | 手机 | 邮箱 | 备注 |
| 钟大壬 | 200830633182 | 08软件3班 | 15017527073 | 287640425@qq.com | 资料收集 |
| 邓峻 | 200830630136 | 08软件3班 | 13825402525 | 100891029@qq.com | Adaboost |
| 李存宽 | 200830634110 | 08软件3班 | 15017527072 | 307528128@qq.com | Gentle Boost |
| 邱华辉 | 200830632079 | 08软件3班 | 13580312715 | 402754944@qq.com | 资料收集 |