|  |  |
| --- | --- |
| FDDB测试步骤 | 文件类型：技术文档 |
| 文件编号： |
| 面向的部门： 研发中心 |
| 保密等级：高 |
| 作者： 刘凯 |
| 日期：2018-11-23 |
| 版本：1.0V |

FDDB测试步骤



目 录

[1 FDDB测试人脸检测 3](#_Toc530753320)

[参考文献 1](#_Toc530753321)

# FDDB测试人脸检测

1) 下载FDDB库

FDDB库的下载地址为：<http://vis-www.cs.umass.edu/fddb/index.html>，解压得到图片文件夹originalPics/，和标注文件夹FDDB-folds/。

从<http://vis-www.cs.umass.edu/fddb/results.html>下载评估程序，解压得到evaluate/

2) 准备标记文件

在FDDB-folds/里新建一个imList.txt文件，将10个FDDB-fold-x.txt(其中x=01~10)文件的内容按**顺序**复制粘贴到imList.txt文件里。

同样，新建一个ellipseList.txt文件，将10个FDDB-fold-x-ellipseList.txt里的内容按顺序复制粘贴到ellipseList.txt文件里

3) 准备测试文件

按照imList.txt里的路径依次对图片进行人脸检测，得到的检测结果保存在results.txt文件里，其格式与FDDB-fold-x-ellipseList.txt一样，即对每幅图片的检测结果，依次输入图片路径、图中人脸数和人脸检测框的左、上、宽、高、置信度五个结果。如图1所示。

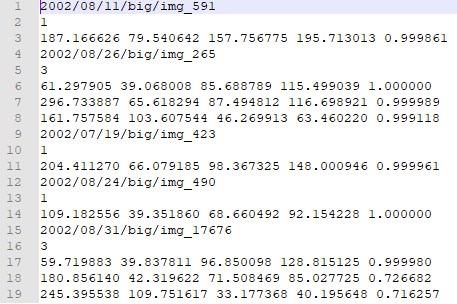


图1：输出结果格式

4) 配置OpenCv

略

5) 修改评估文件

a) 打开evaluate/Makefile文件，将

evaluate: $(OBJS)

$(CC) $(LIBS) $(OBJS) -o $@

修改为：

evaluate: $(OBJS)

$(CC) $(OBJS) -o $@ $(LIBS)

b) 打开evaluate/common.cpp，将

//#define \_\_IMAGE\_FORMAT\_\_ ".jpg"

#define \_\_IMAGE\_FORMAT\_\_ ".ppm"

修改为：

#define \_\_IMAGE\_FORMAT\_\_ ".jpg"

//#define \_\_IMAGE\_FORMAT\_\_ ".ppm"

c) 将RectanglesSingleImage.cpp，EllipsesSingleImage.cpp和evaluate.cpp里的CV\_RGB替换成Scalar，否则make的时候会报错。

6) 在evaluate/下执行make

7) 在终端执行：

./evaluate -a ../ellipseList.txt -d ../results.txt -i ../originalPics/ -l ../imList.txt

其中：

-a 标注文件

-d 检测结果

-i 图片目录

-l 图片路径文件

-f 检测结果类型(默认是矩形框，若为椭圆框则为1)

-r 输出的Roc文件名前缀，默认是temp

上述命令执行后将生成tempContROC.txt和tempDiscROC.txt两个文件，分别为continue和discrete两种指标的结果。可以用matlab等工具绘图，也可以用其自带的gunplot绘图，修改并执行runEvaluage.pl即可。

需要说明的是两个ROC文件的第一列为纵坐标，第二列为横坐标。其中离散ROC的度量只关心检测框与标注的人脸框的IoU是否大于阈值，大于阈值则认为检测正确。而连续ROC的度量则是希望IoU越大越好。

参考文献

1. V. Jain and E. G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. UMass Amherst Technical Report, 2010.