

人脸识别算法SDK测试平台使用说明文档

V 1.0

智能安全技术研究中心

**版本更新**

| 日期 | 修订版本 | 修改章节 | 修改描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2019/7/3 |  |  | 初始版本 | 刘勇 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[1. 引言 1](#_Toc13214830)

[2. 专用术语 1](#_Toc13214831)

[3. 脚本使用说明 1](#_Toc13214832)

[3.1 配置文件config.ini中的相对路径 1](#_Toc13214833)

[3.2 各测试用例有自己单独的读/存文件路径，但也有例外 1](#_Toc13214834)

[3.3 依照配置文件config.ini放置好人脸图片文件 3](#_Toc13214835)

[3.4 用新版SDK编译出测试平台的可执行文件 3](#_Toc13214836)

[3.5 Linux编译可执行文件需要安装opencv-2.4.11 3](#_Toc13214837)

[3.6 运行可执行文件 3](#_Toc13214838)

[3.7 测试用例执行结果 4](#_Toc13214839)

[3.8 执行结果打印的耗时统计方法 4](#_Toc13214840)

[3.9 执行结果中的错误率统计方法 5](#_Toc13214841)

[3.10 出现“系统找不到指定的文件”时不必理会 5](#_Toc13214842)

[3.11 测试用例执行先后顺序要求 5](#_Toc13214843)

[3.12 将前台打印输出到文件 6](#_Toc13214844)

[4. 系统出错处理设计 6](#_Toc13214845)

[5. 技术设计 6](#_Toc13214846)

[6. 数据库设计 6](#_Toc13214847)

[7. 词汇表 6](#_Toc13214848)

[8. 进度计划 6](#_Toc13214849)

# 引言

作为人脸识别算法SDK测试平台自动化脚本的使用说明。

# 专用术语

|  |  |  |
| --- | --- | --- |
| 编号 | 专用术语 | 意义 |
| 1 | 主菜单 | 脚本运行起来后的初始列表界面即为主菜单； |
| 2 | 测试套 | 主菜单列表的每一个条目为一个测试套； |
| 3 | 测试用例 | 选中某个测试套，进入的新的列表界面即为该测试套下的测试用例列表界面，列表的每个条目为一个测试用例，选择需要执行的测试用例即可运行此用例； |
| 4 | 人脸检测 | 检测出人脸矩形的过程； |
| 5 | 人脸识别 | 使用提取的特征，通过特征比对找出合乎要求人脸的过程； |

*表2-1 专用术语表*

# 脚本使用说明

本文档将脚本使用过程中可能遇到的问题逐条罗列。

## 配置文件config.ini中的相对路径

配置文件config.ini中的路径全部为相对路径，是相对于可执行文件的路径，Windows下为相对FaceSDKTest.exe的路径，Linux为相对FaceSDKTest文件的路径。

## 依照配置文件config.ini放置好人脸图片文件

*[ISCalFaceInfoPath]*

*;图片路径A*

*faceInfoImgPathA = ../../Data/ISCalFaceInfoPath/faceInfoImgA*

*;图片路径B*

*faceInfoImgPathB = ../../Data/ISCalFaceInfoPath/faceInfoImgB*

*图3-1 config.ini中配置的待测人脸图片路径*

人脸图片目录包含A/B两个目录，如图3-1所示。如果只有一个人脸图片文件夹，比如不进行人脸比对，只进行人脸检测，并且只有一个人脸图片文件夹，拿图3-1举例，那么将其命名为faceInfoImgA或者faceInfoImgB均可，但是没有人脸图片的文件夹不能删除，只能空着。

## config.ini文件中compareSamples的含义

在1:1比对的时候，如果A文件夹有10000张现场照图片，B文件夹有10000张身份证照，如果把A中的每张图片都与B文件夹中的所有图片进行比对，那么将要比对1亿次，没有必要意义也不是很大，纯属浪费时间。

引入compareSamples后，对A目录中的任一张现场照图片，首先在B文件夹中按名字找到对应的身份证照片，只与找到的身份证照片到往后的共compareSamples张身份证照片进行1:1比对，这样就不用比对1亿次，而是比对了10000\*compareSamples次。

1:N同。

## 支持首选识别率和误警率设计

首先明确此两概念只适用于1:N比对场景，然后举例说明首选识别率和误警率的概念：

比如A文件夹为现场照100张，B文件夹为身份证照片100张，A文件夹中的其中50张照片（对应50个乘客Pin）的身份证照片在B文件夹中，另外50张照片（对应另50个乘客Pout）的身份证照片不在B文件夹中。

Pin在与B文件夹进行1:N对比时，可能出现和本人身份证比对结果小于判别门限的情况，也可能出现虽然大于判别门限，但是和B文件夹非本人的身份证照比对的分值更高的情况，首选识别率为比对的分值大于判别门限且为最大比对分值的比率。

Pout在与B文件夹进行1:N对比时，可能出现比对分值大于判别门限的情况，误警率就是所有50个人中出现过比对分值高于判别门限的比率。

以上的首选识别率和误警率都不可能高于100%。

最后，在测试时，怎么区分哪些是Pin哪些是Pout呢？完全按照文件名来区分，如果A文件夹中的文件名（相当于人名）在B文件夹中，那么属于Pin，否则属于Pout。

## 各测试用例有自己单独的读/存文件路径，但也有例外

各测试用例有自己单独的读/存文件路径，如图3-2所示。

*Appliance*

*Error*

*ISCalFaceInfoPath*

*ISCompare*

*ISFaceDetectPath*

*ISFaceDetectRgb*

*ISFaceDetTrackRgb*

*ISGetFeaturePath*

*图3-2 测试用例单独使用一个文件夹*

测试用例专有的文件夹中包括了人脸图片文件夹，单线程、多线程执行结果文件夹等。图3-3为ISFaceDetectPath测试用例专有文件夹中包含的人脸图片文件夹、单/多线程特征文件夹、单/多线程pca文件夹、单/多线程人脸矩形文件夹等。

*detectImgA*

*detectImgB*

*feaMA*

*feaMB*

*feaSA*

*feaSB*

*pcaMA*

*pcaMB*

*pcaSA*

*pcaSB*

*recDetectMA*

*recDetectMB*

*recDetectSA*

*recDetectSB*

*图3-3 测试用例专有文件夹*

## 用新版SDK编译出测试平台的可执行文件

在face目录，将新的SDK lib库替换掉老的lib库，简单地，直接将face目录替换掉即可。

将face目录下的models文件夹拷贝到build下可执行文件所在的位置，覆盖掉原有的models文件夹。注意Windows和Linux分别是不同文件夹下的不同可执行文件，Windows下的可执行文件名为FaceSDKTest.exe，Linux下可执行文件名为FaceSDKTest。

## Linux编译可执行文件需要安装opencv-2.4.11

Linux编译测试平台的可执行文件需要安装opencv-2.4.11，安装的过程相当复杂，有可能是两三天，也可能更久，不属于本文档内容，不做扩展。

## 运行可执行文件

Windows下，双击可执行文件即可。

Linux下，在命令行中敲入“./test.py”或者“python test.py”或者“python3 test.py”，根据Ubuntu系统python设置不同，其中可能只有不等个可以正常启动运行脚本。

根据脚本提示，选择需求选项，即可在各个测试套之间切换、执行测试用例、退出脚本、返回主菜单等。

## 测试用例执行结果

测试用例执行结果有两类，一类是执行结果数据存文件，测试用例执行结果存文件夹目录在用例输入人脸图片同级目录。

另一类是前台打印，前台打印有些是成功率、读/存路径、内存消耗情况等，另一类则可能什么都没有，这就需要看用例名了，比如图3-2、图3-3所示，该用例的用例名就说明是在不创建通道直接调用人脸检测接口会出现错误，至于是什么错误码，在测试代码中用断言进行了校验，只要断言校验成功（无异常打印）即说明SDK版本此测试用例通过。

*[ RUN ] ftError.detectWithOutCreatingDectectChannel*

*ProcessMemory::getProcessMemory(): 22032*

*ProcessMemory::getProcessMemory(): 23048*

*[ OK ] ftError.detectWithOutCreatingDectectChannel (731 ms)*

*图3-2 断言类测试用例执行成功打印*

*[ RUN ] ftError.detectWithOutCreatingDectectChannel*

*ProcessMemory::getProcessMemory(): 25568*

*e:\projects\private\facesdktest\source\src\fterror.cpp(35): error: Value of: IMG\_DATA\_ERROR == ISFaceDetectRgb(0, (char\*)image.data, image.rows\*image.cols\*3, image.cols, image.rows, outRst, &outLength)*

*Actual: false*

*Expected: true*

*ProcessMemory::getProcessMemory(): 26536*

*[ FAILED ] ftError.detectWithOutCreatingDectectChannel (635 ms)*

*[----------] 1 test from ftError (637 ms total)*

*[----------] Global test environment tear-down*

*[==========] 1 test from 1 test case ran. (641 ms total)*

*[ PASSED ] 0 tests.*

*[ FAILED ] 1 test, listed below:*

*[ FAILED ] ftError.detectWithOutCreatingDectectChannel*

*图3-3 断言类测试用例执行失败打印*

## 执行结果打印的耗时统计方法

执行结果打印的耗时为SDK接口消耗的时间，其它读存文件等耗时不计入。

对于多线程用例，由于多个线程是并行执行，所以在每个线程下统计SDK接口净耗时，然后累加并除上线程数。

总体上，对于人脸图片文件不是很少的情况下，多线程耗时小于单线程耗时。

在统计平均耗时时，比如检测100张人脸，则是总耗时除上100。如果分别100张图片的两个文件夹进行特征比对，比对次数为10000，则平均耗时为总耗时除上10000。

## 执行结果中的错误率统计方法

在人脸检测及特征提取时，统计的是检测成功次数与图片总数比或者特征提取成功次数与图片总数比，很好理解。

对于特征比对时，比如A/B文件夹下各有100张图片，比对时，将A文件夹中的每张图片与B文件夹下的每张图片进行特征比对，实际比对次数进行了10000次，错误率的计算就是以10000作为分母，比如10000次比对累计出现20次不是本人，但是特征对比结果大于0.85的门限，那么错误率就是0.2%。

## 出现“系统找不到指定的文件”时不必理会

测试用例执行结果存文件夹时，可能之前已经执行过该用例，之前的执行结果已经存在了，所以在执行测试用例之前要先删除掉前次执行产生的执行结果。对于初次执行测试用例，没有前次执行的测试结果，所以会出现删除一个不存在的文件夹，前台告警“系统找不到指定的文件”，不必理会。

## 测试用例执行先后顺序要求

一、由于测试用例ISFaceDetectPath和ISGetFeatureWithFacePosPath测试用例共用了数据源，在使用ISGetFeatureWithFacePosPath测试用例提取人脸特征时需要使用ISFaceDetectPath测试用例生成的人脸位置数据，因此需要先运行ISFaceDetectPath测试用例，再运行ISGetFeatureWithFacePosPath测试用例。

二、在执行ISCompare测试套测试用例时，需要先执行第一条，以产生进行比对所需要的特征文件和pca文件。

其它测试用例之间没有先后顺序要求，可以根据需要执行。

【注】多线程特征提取需要的人脸位置文件为多线程人脸检测用例检测出的人脸位置文件，单线程特征提取需要的人脸位置文件为单线程人脸检测用例检测出的人脸位置文件。

## 将前台打印输出到文件

测试脚本代码行数有限，可读性很强，根据自己需求修改无难度。如果要将前台打印输出到文件，则直接使用重定向符号“>”即可，Windows和Linux均可，如图3-4。

*打印到前台*

*.\Debug\FaceSDKTest.exe --gtest\_filter=ftISCompare.ISCompare\_\**

*打印到文件*

*.\Debug\FaceSDKTest.exe --gtest\_filter=ftISCompare.ISCompare\_\* >文件名*

*图3-4 将前台打印输出到文件*

# 系统出错处理设计

略

# 技术设计

略

# 数据库设计

略

# 词汇表

略

# 进度计划

略