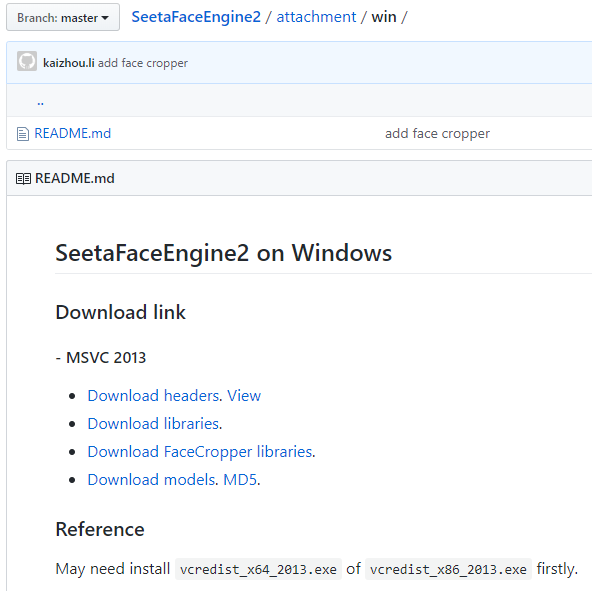
# FaceEngine项目构建记录

预先配置： openCV（本人使用opencv342）

相关SDK配置

依赖性一览

<https://github.com/seetaface/SeetaFaceEngine2/tree/master/attachment/win>



|  |  |
| --- | --- |
| 头文件 | E:\ZimengZhao\_Libs\FaceEngine\_Repo\SeetaFaceEngine2\attachment\prebuild\include |
| 库目录x2 | E:\ZimengZhao\_Libs\FaceEngine\_Repo\SeetaFaceEngine2\attachment\prebuild\lib\win  为了便于使用，将其放置于  E:\ZimengZhao\_Libs\FaceEngine\_Repo\SeetaFaceEngine2\my\_proj\SeetaFaceEngine\Source\_lib |
| 模型MD5 | 下载链接：  <https://pan.baidu.com/s/1HJj8PEnv3SOu6ZxVpAHPXg>    疑问：  关于下面这个文档说明？ |

至此开始正式项目（Cpp Doc）

<https://github.com/seetaface/SeetaFaceEngine2/blob/master/example/C++/README.md>

### Detection构建（FD对象）

|  |
| --- |
| Pipeline: 初始化FD对象，使用openCV读入图像并转化为SeeTaa图像格式，  之后将SeeTaa图喂入FD【SeetaRect \*face = FD.Detect(image, &face\_amount);】，  即可输出\*face 即 脸的bbx列表， face\_amount存储表的长度；  FD可以一次性检查出单张图像中所有的face;  缺点： 全家福、全明星级别的小、多脸图像，表现较差 |

将上述包括openCV和必要的SeeFace文件全部的包含文件、库全部纳入按照以下方式编译：

Realeasex64 Mode

需要的lib

opencv\_world342.lib

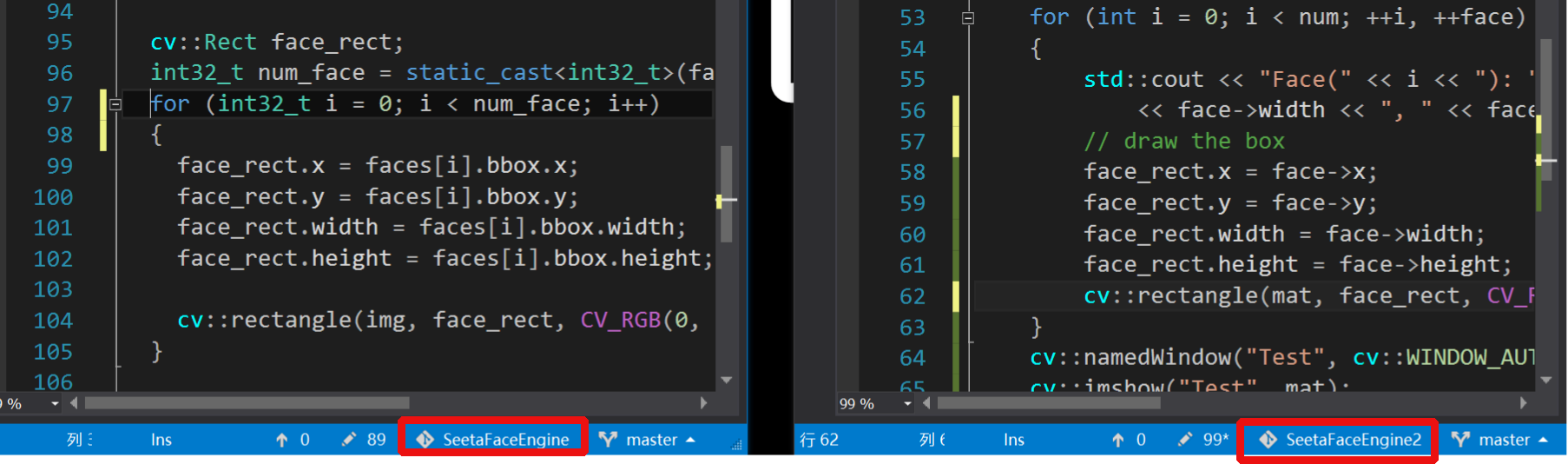
SeetaFaceDetector200.lib

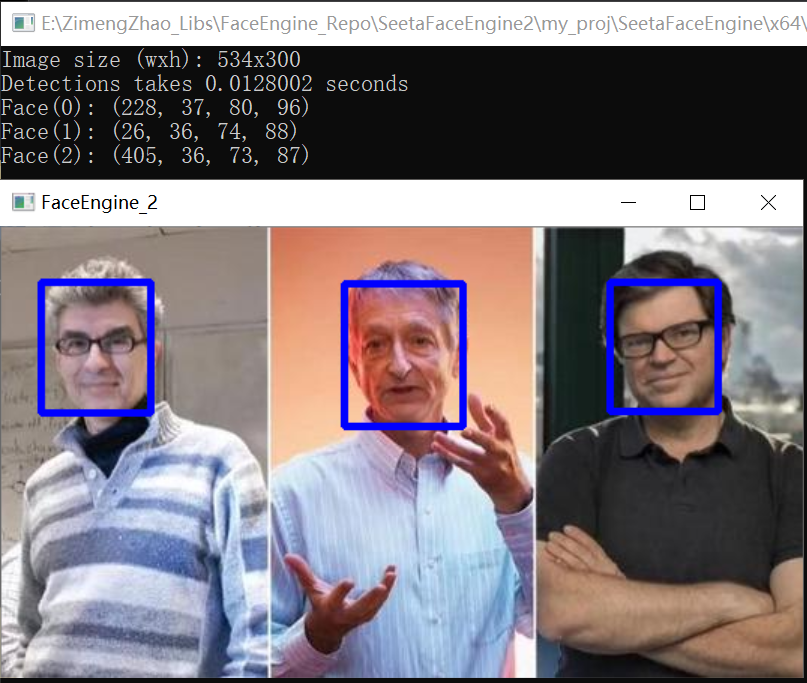
exe路径：

加入 SeetaFaceDetector200.dll



写法：参考初代代码，按照旧API的example构建；





### 2．Alignment构建（PD对象）

|  |
| --- |
| Pipeline: Alignment作用在于细化分析FD中得到的BBX内部的脸部关键点。  这一对象的输入为上一步FD的单个RoI；  输出为这一RoI中脸部的5个关键点坐标；    P1 = 蓝色，左眼；P2 = 绿色，右眼；P3 = 红色，鼻子；  P4 = 紫色，左嘴角，P5 = 青色，右嘴角； |



/\*

Realeasex64 Mode

需要的lib

opencv\_world342.lib

SeetaFaceDetector200.lib

exe路径：

加入 SeetaFaceDetector200.dll

PointDetector还需要lib：

SeetaPointDetector200.lib

exe路径还需要：

加入 SeetaPointDetector200.dll \*/

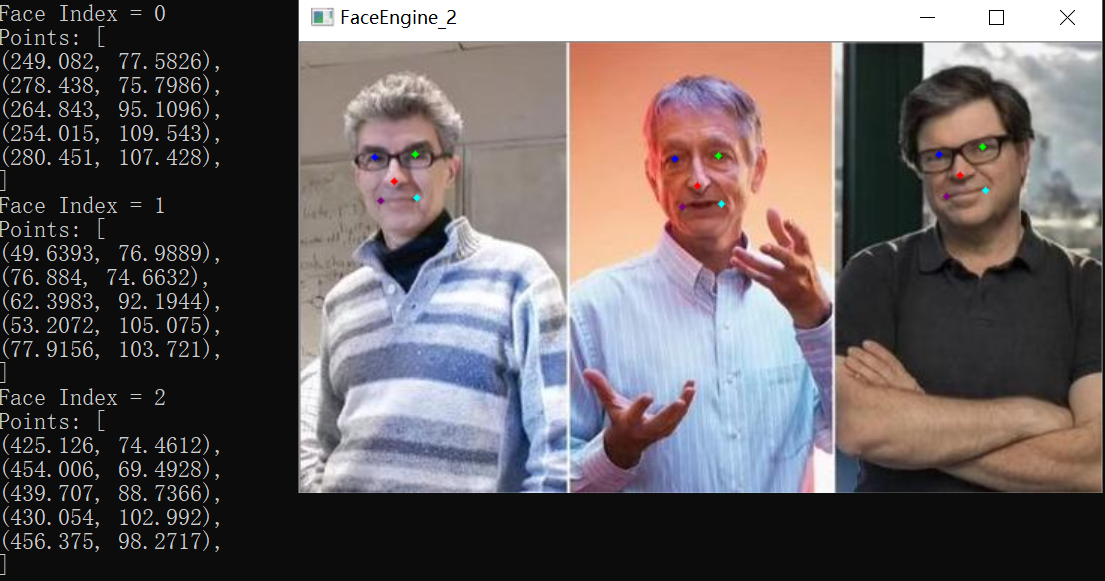
 





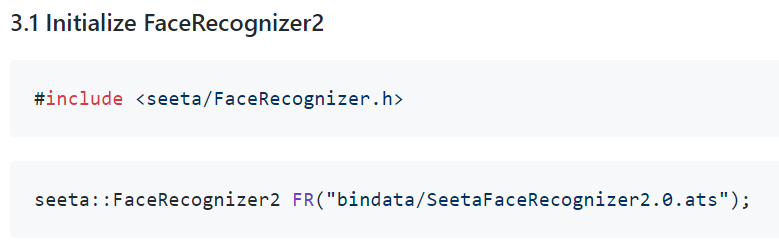
// Align 一次只能处理单张图像，要想得到多张，需要将\*face中的对象逐个送入分析

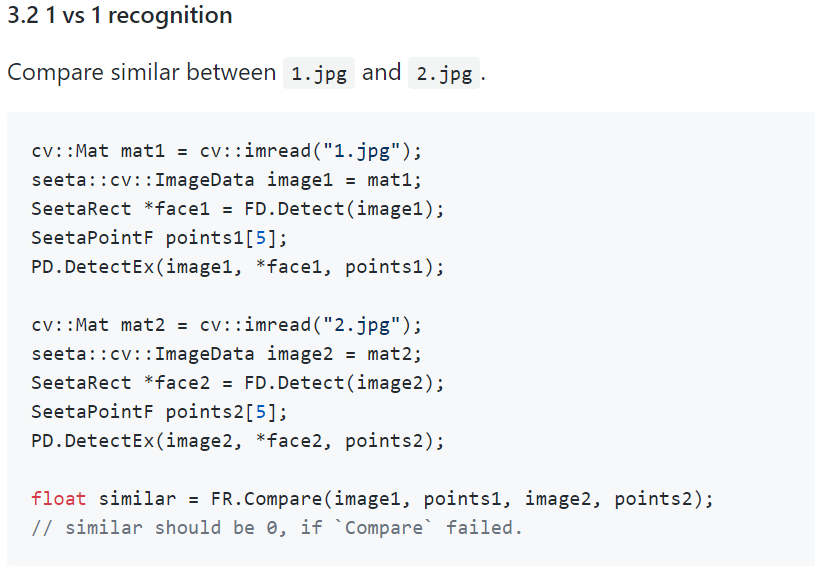
联合测试：（包括检查5点的位置）



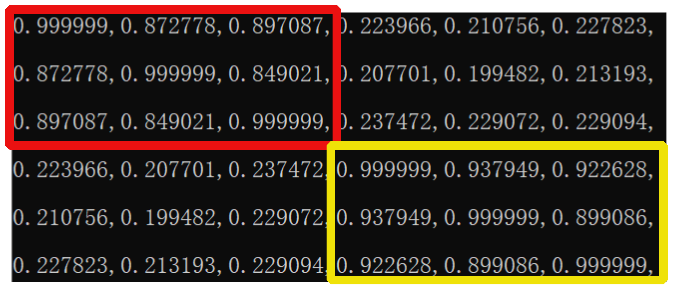
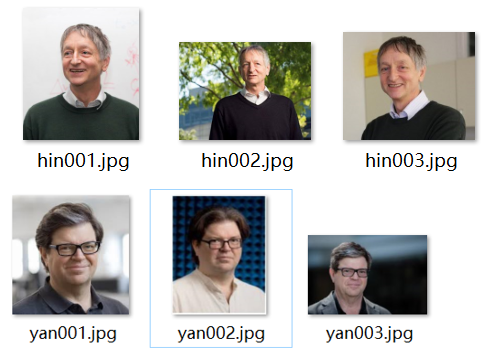
### Face Identification（FR对象）

|  |
| --- |
| pipeline：  利用FD的RoI及针对RoI的5points, 描述两组或多组数据之间的相似性（聚类任务） |

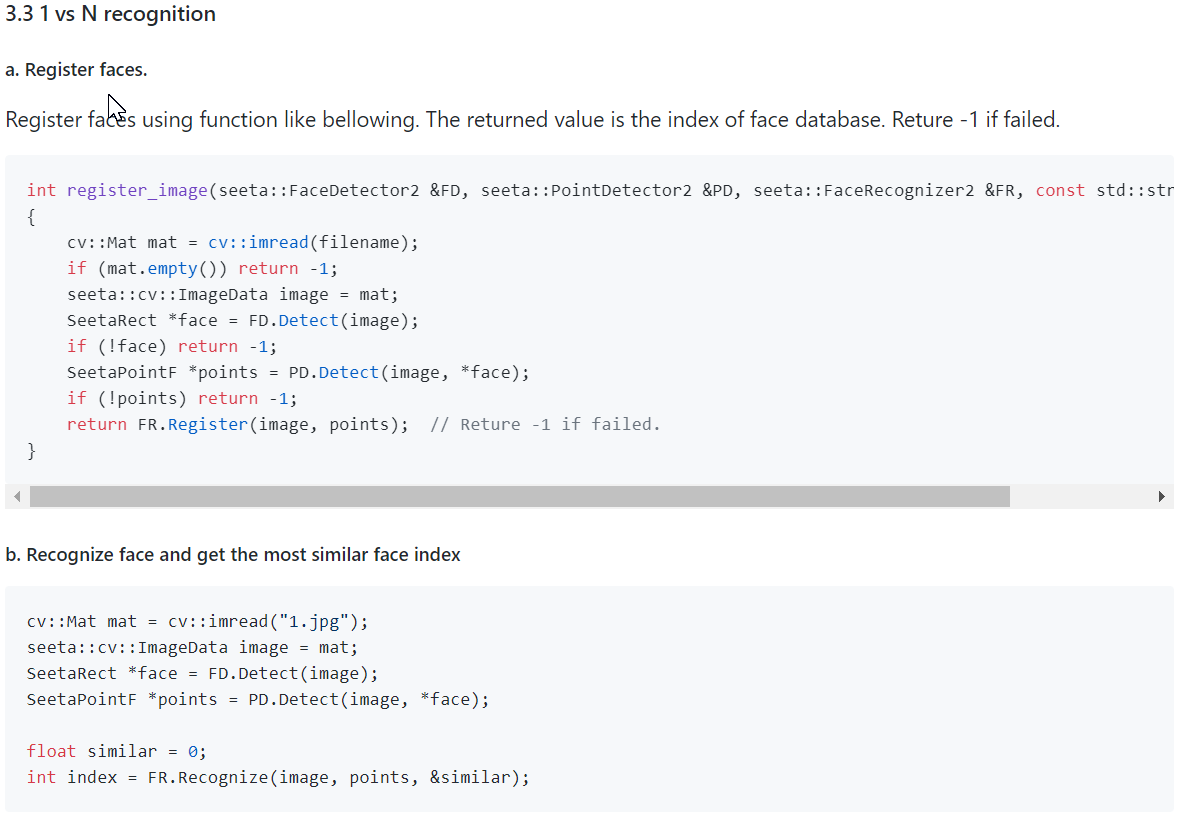


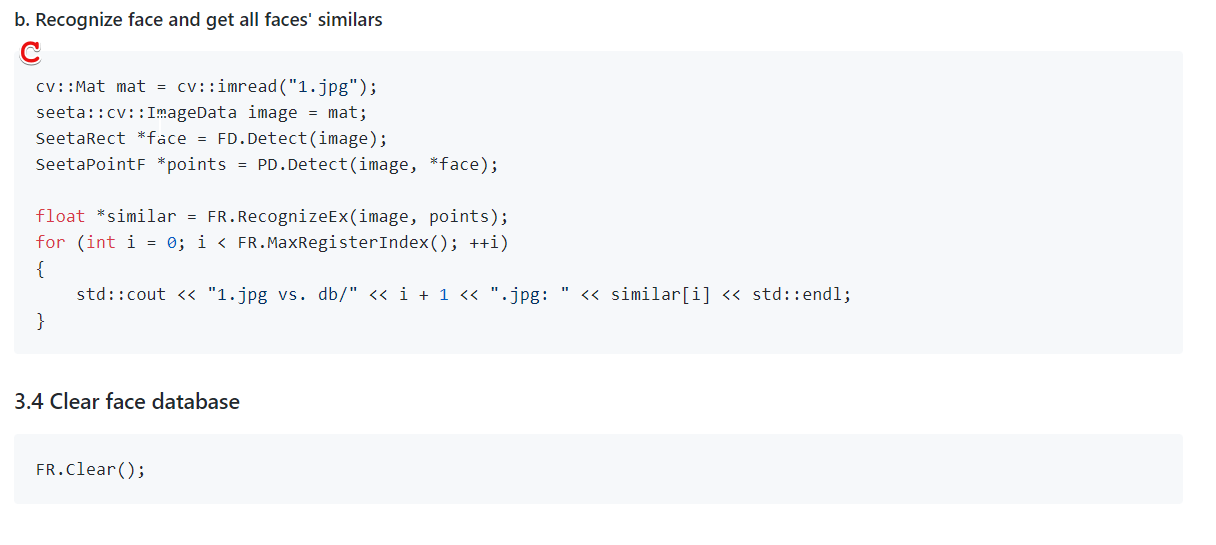


1v1 测试：

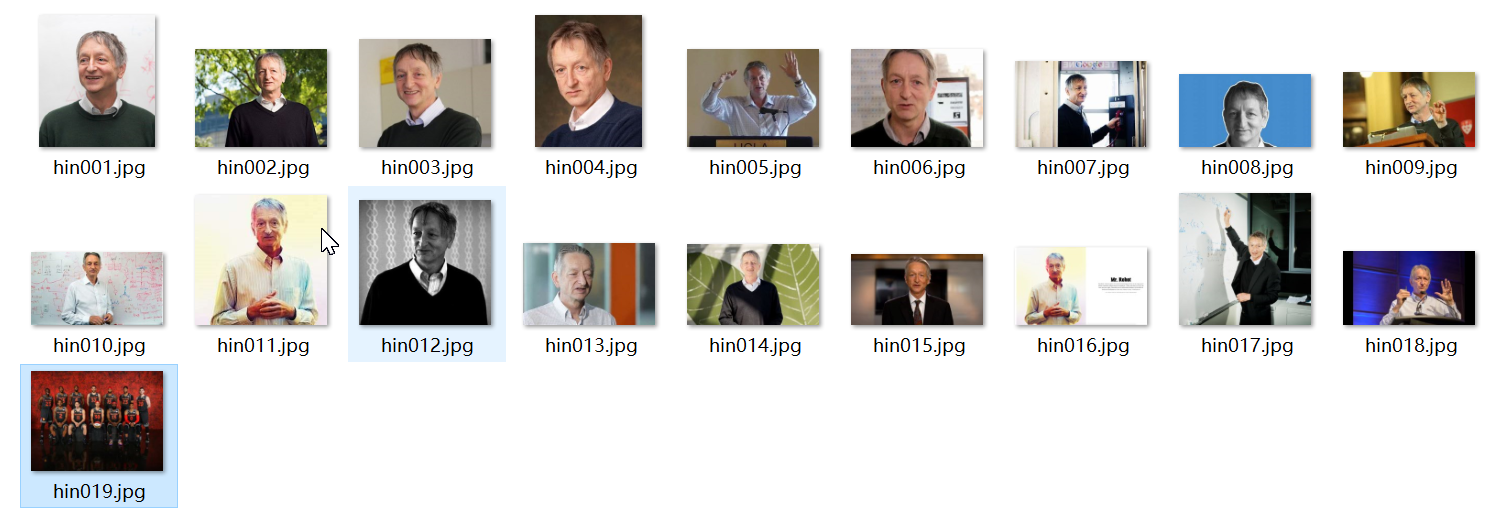


6张图两两比较结果（对角为自身匹配）↗

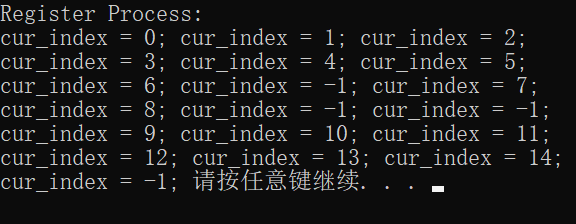




数据准备：hinton的19张图像



Register过程，建立匹配样本空间



-1代表register失败(总计19,失败= 4， 成功=15)

测试匹配过程：

