把License放置到 \bin\Licenses文件夹下面，并且把TrialFlag.txt中的TRUE修改为FALSE

打开Tutorials/Biometrics/CPP/VerifyCPP/VerifyCPP.vcxproj 打开后点击VeriFingerCPP.cpp生成解决方案

编译好的exe在bin/win32\_x86下面

Verifinger对于输入图像的要求是500dpi

verifinger官网可以下载Cross\_Match\_Sample\_DB 进行试验。这个文件夹里的图片全是符合verifinger要求的。

Verifinger储存的指纹细节点template是按照NTemplate储存的，也可以改写代码使得其按照iso 19794-2标准储存。

D:\verifinger\Neurotec\_Biometric\_12\_2\_SDK\Tutorials\Biometrics\CPP\EnrollFingerFromImageCPP，将里面的输出格式改成ISO,编译即可。或者在命令行输入的时候，末尾输入ISO。

示例: EnrollFingerFromImage.exe test.jpg test ISO

可以用github上的相应的C++代码进行解析。

<https://github.com/DongChengdongHangZhou/ISO_19794-2-2005-converter>

编译此代码非常简单，按照上面的要求进行操作即可。在Ubuntu电脑上依次输入四个命令即可：

mkdir build cd build cmake .. make

运行此代码的时候，一定要./ISO\_19794\_2\_2005\_converter <fmd.ist> <fmd.txt>。

其中， ‘./’ 不能漏！！！

示例：./ISO\_19794\_2\_2005\_converter test result

这时就会生成一个名为result的文件（还不是txt文件，不过补上txt后缀名就可以以文本文档打开）。这个文件从第8行开始读就是细节点的坐标以及方向。

经过我改写的EnrollFingerFromImage.CPP,可以把细节点信息输出到txt文档中。在cmd命令下，运行EnrollFingerFromImageCPP.exe xx.jpg xxx,就可以将xx.jpg的指纹图像细节点提取出来，以NTemplate格式存储到xxx中，同时把细节点信息输入到txt文档中。

如果想打开NTemplate的文件，那么可以找到 SDK\Bin\Win64\_x64\NTemplateSampleCS.exe, 然后就可以了。不过这个exe无法打开iso文件格式的细节点模板。

运行EnrollFingerFromImageCPP.exe, 第一行提取core的坐标，第二行是delta的坐标。

测ROC曲线：

把VerifyFingerROCtester.cpp里面的代码代替VerifyFingerCPP.cpp里面的代码，就可以用于测ROC曲线。

生成MCC格式的模板：

把EnrollFingerFromImageMCCformat.cpp里面的代码代替EnrollFingerFromImage.cpp里面的代码，就可以用于生成MCC格式的细节点文件。