**NOTES**

1. **How to setup the environment (NDK, Android Eclipse, OpenCV4Android)**

Basically, you should follow these links:

<http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/O4A_SDK.html>

<http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/dev_with_OCV_on_Android.html#dev-with-ocv-on-android>

It will help you install/configure the following components/frameworks:

* Java JDK (not compatible with Open JDK).
* Android SDK (I guess you’re master of this platform – so it’s just in a blink of an eye:) for you).
* Eclipse IDE (Android Eclipse).
* ADT and CDT plug-ins for Eclipse (needed for writing C++ code – JNI part).

I would like to recommend you to use OpenCV4Android 2.4.9 – the sample project I sent you is for this version.

At the end of this phase, the OpenCV4Android SDK should have been setup and configured successfully, all the samples (tutorials) working as well. You can test these samples by building it up and running on your device (or simulator).

1. **Writing JNI part and a small sample (prototype) for testing**

Since you have setup/configured the environment successfully, now we are going to write a small sample for testing TestSearch function (FaceSearch).

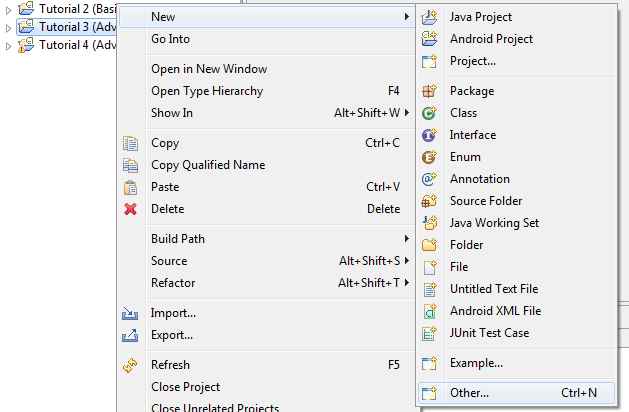
I picked up the sample Tutorial 1 – Camera Preview (under OpenCV-2.4.9-android-sdk/samples folder) which is working with OpenCV4Android to make some modification as our sample. This sample (Tutorial 1) is originally not supporting JNI, so we will need to add this support.

**2.1 Add JNI support**

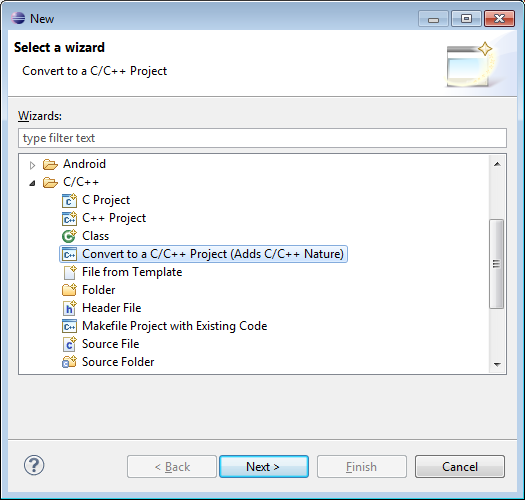
Here are two ways to add it:

* + 1. Add C/C++ Nature to the project via Eclipse menu New -> Other -> C/C++ -> Convert to a C/C++ Project. You should check the link <http://docs.opencv.org/doc/tutorials/introduction/android_binary_package/android_dev_intro.html#android-dev-intro> for more details.

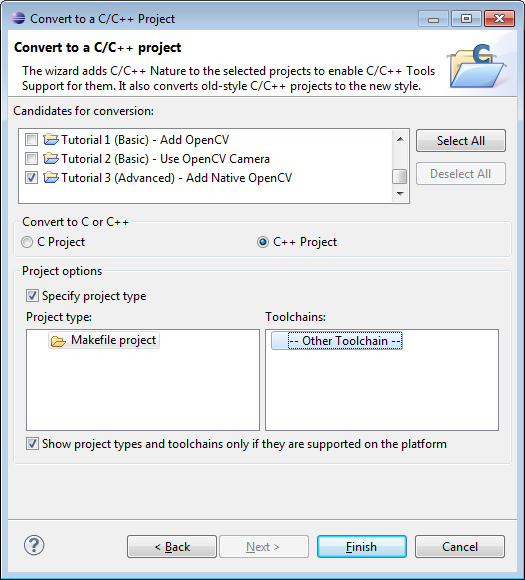
1. Step 1



1. Step 2

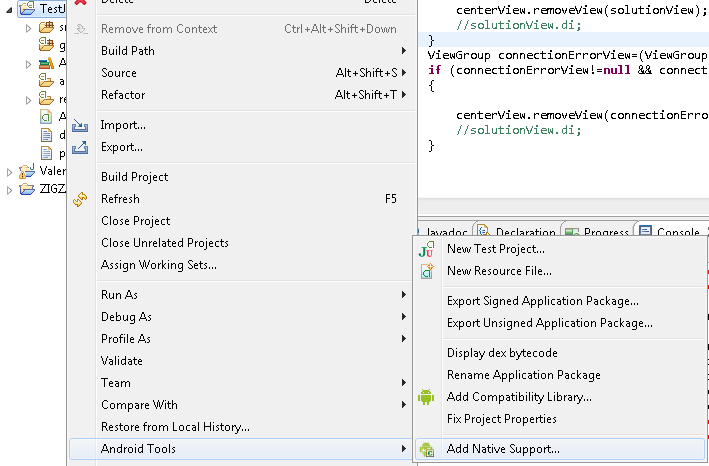


1. Step 3

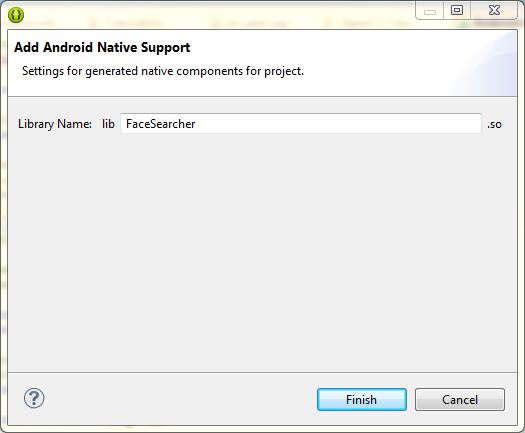


* + 1. Right click the project on Eclipse Workspace, then select Android Tools -> Add Native Support.

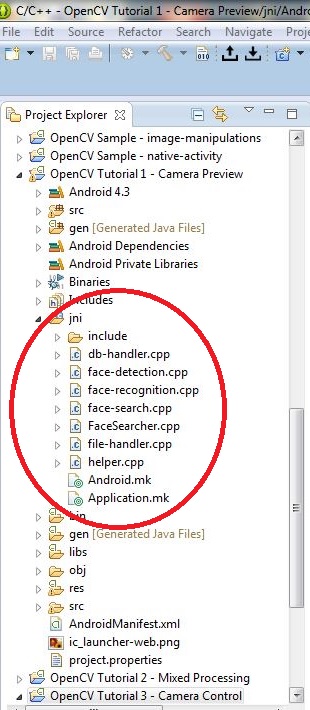
1. Step 1



1. Step 2



I prefer the second way (1.2), it’s simple and clear. So now the Tutorial 1 has JNI support, you can see jni folder under Tutorial workspace in Eclipse after adding FaceSearcher library.

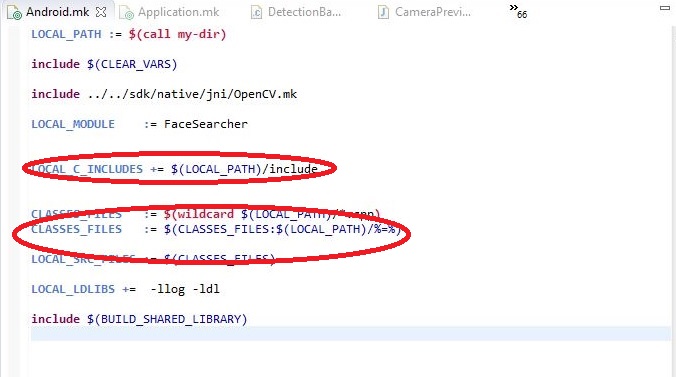


**2.2 Modify/convert the existing code for JNI part**

I copied all your header files (db-handler, face-detection, face-recognition, face-search, file-handler, helper under include folder) and added them into jni folder. After that, I also added all source files (but changed it to cpp extension for better support) into jni folder (please see above picture).

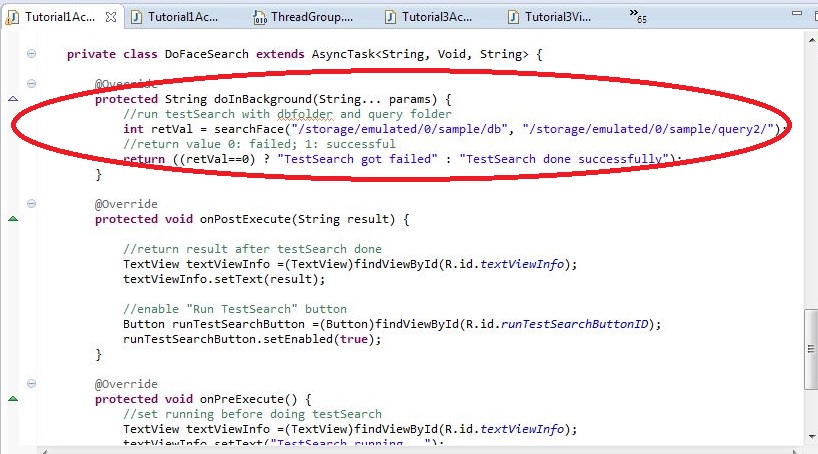
For the “FaceSearcher.cpp” file which has been generated when adding JNI support, I made it almost the same with “face-search-test.c”, but some minor modification (adding JNI function searchFace, change of TestSearch’s input parameters). Please see the source of jni part for more information.

Also, there are 2 make files (.mk) “Application.mk” and “Android.mk” generated during the creation of JNI support. I left the “Application.mk” by default, and changed the “Android.mk” file for how to compile OpenCV C++ source files, and specifying where the header files are.



**2.3 Create an Android sample to run “TestSearch” function:**

I used the original activity (Tutorial1Activity), but for our purpose (working with still images) I only needed part of that code. Besides that, I added a button “Run TestSearch” for testing. When user touches on this button, the searchFace will be executed.



Note: I was putting the “faces” and “query2” folders on my device in internal storage ("/storage/emulated/0/sample/db", "/storage/emulated/0/sample/query2/"). You can change it to whatever you want according to your settings.

Since “searchFace” is a JNI function, it should be declared with “native” reserved word. Also we have to load the library FaceSearcher, OpenCV as well before using it.

