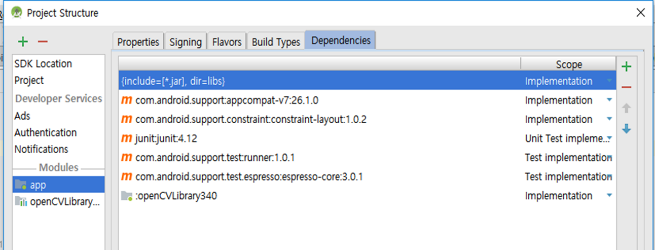
5주차 2012742027 2018-04-08 홍준혁

**안드로이드 스튜디오  
openCV 기능추가(ndk-build)  
  
개발환경  
안드로이드 3.0.1  
openCV 3.4.0**

openCV를 추가하는 방법엔 두가지가 있는데 ndk-build와 CMake이다. 둘의 차이점으론 기능상 하는 일은 똑같지만 전자는 해야할 일이 많고 후자는 간단하다.

다만 CMake가 나온지 얼마 되지 않아 기존의 소스들이 모두 ndk-build를 사용했기 때문에 소스코드를 재활용하기위해 ndk로 기능을 추가하였다.

**openCV 모듈 추가하기**

* **[file] – [new] – [import module] 클릭한뒤 opencv 디렉토리를 추가한다. 여기서 주의해야 할점은 하위 폴더 sdk/java를 선택해야 추가가 된다. 정상적으로 추가되었다면 모듈에 openCVLirary340이 추가된것을 확인 할 수 있다.**
* **[file]-[project Structure] 클릭. dependecies에서 + 를 추가한뒤 모듈을 클릭해 openCV 모듈을 추가한다.**

openCV – 안드로이드용 모듈은 다음 github에서 다운 받을 수 있다.

<https://github.com/opencv/opencv/releases>

opencv-3.4.0-android-sdk.zip을 다운받는다.

압축을 풀고 OpenCv 디렉토리를 적당 한 곳에 복사한다.

위 사진에 보이는 Dependencies 엔 여러가지 라이브러리나 모듈을 추가 할 수 있는데 간혹가다 버젼이 다른 문제로 import를 하지 못하거나 함수가 없다는 경우가 발생한다. 그렇다면 라이브러리나 버전이 다르다는 말이니 그 때 상황에 맞춰서 버전을 맞추어 보길 바람 \*\*\*

**버젼 통일하기 2018-04-11**

* openCV에 사용되는 컴파일 버전과 안드로이드 컴파일 버젼이 다를 수 있으므로 통일 시켜주는 작업을 해야한다.
* gradle 파일을 app과 openCv 하나씩 열어 통일 시킨다.
* 통일 시키지 않으면 이전 버전에서 더이상 사용하지 않는 함수나 기능들을 사용하여 오류가 날 상황들이 존재한다. 여기 까지 하고 gradle 동기화를 시켜준다.
* 이상이 있으면 더이상 넘어가지 말고 해결 할 때까지 이전 과정을 반복한다. << 꼬이면 어디서 오류가 났는지 알 수 없어 어플이 툭하면 죽는다 >>

**3. AndroidManifest 등록**

* OpenCV를 사용하기 위해선 사용자의 카메라의 권한을 얻어야 하는데 이때 권한을 얻기 위해 Manifest에 추가해야 자동적으로 권한을 허용할 것인지 묻는 창이 뜨게 된다

권한을 허용한 후,

<http://github.com/hongjunhyeok/Android> 탭에 들어와서

openCV예제 올려놓을 테니 확인 바람.

<참고>

openCV 함수 : <http://webnautes.tistroy.com/923> 참고했음 >> 설명이 잘 나와있어 따라해보면 도움이 됨.

wiringPI 함수 : <https://wiki.odroid.com/odroid-c2/application_note/gpio> 참고했음 >> 프로그램이 너무 크고 기능이 많으므로 걸러낼 필요가 있음.

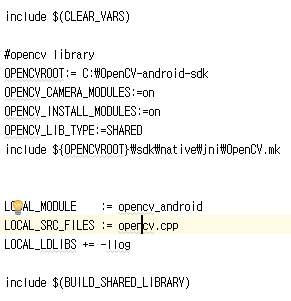
코드가 MainActivity만 하더라도 600줄이 넘기에 설명이 무리.

안드로이드에서 c++라이브러리 사용하여 opencv 작동.

6주차 2018-04-14 2012742027 홍준혁

jni 폴더안에 새로 생성할 파일 이름으로 opencv.cpp 을 적고 OK를 클릭 후 작성한다. 아직까진 mk파일로 돌리지 않았기 때문에 opencv가 인식되지않아 오류가 뜰것이다. C++로 작동하고자 하는 코드를 작성한다.

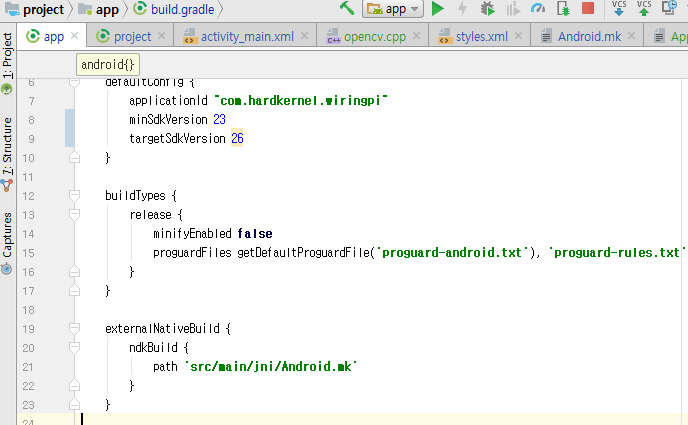
<android.mk>



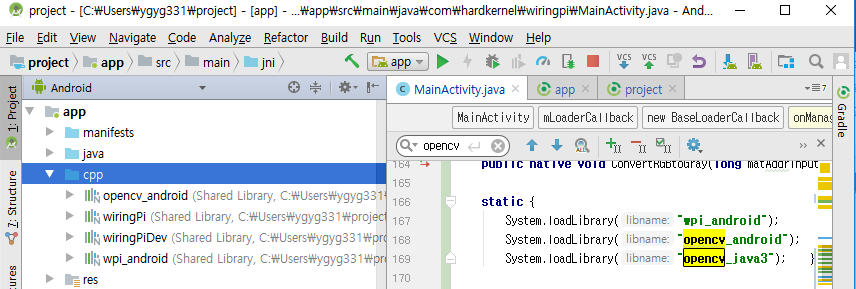
c파일과 openCV 라이브러리를 연동시키며 더불어 java code에서도 동작을 확인 시키는 역할을 한다. 주의해야 할점은

한 라이브러리 파일 안에는 무조건 1개의 Android 파일이 들어가야 하고 두개이상의 안드로이드.mk 파일이 들어가게 되면 에러가 발생한다. c나 c++ 파일의 갯수는 상관없다.

여기까지 마쳤으면, 거의 마무리단계라고 할 수 있다.



Result.



라이브러리가 잘 인식되었따면 cpp 탭에 opencv\_androi가 shared Library로 나오며 서로 연동되는 것을 확인 할 수 있다.

이를 통해 5~6주차 사이에 안드로이드로 opencv라이브러리를 완성시켜보았다.

주의해야할 점으론 단계가 복잡하여 쉽게 접근할 수 없다는 점과 구조가 이차원적이라 고려해야할 점이 많다는 점인데 기능을 다분화하여 쉽게 이해할 수 있도록 작성해보겠다

**JNIEXPORT void JNICALL** Java\_com\_hardkernel\_wiringpi\_MainActivity\_ConvertRGBtoGray  
  
(JNIEnv \*env, jobject instance, jlong matAddrInput, jlong matAddrResult){  
  
Mat &matInput = \*(Mat \*)matAddrInput;  
Mat &matResult = \*(Mat \*)matAddrResult;  
  
cvtColor(matInput, matResult,***CV\_RGB2GRAY***);

기본 소스 코드.