BME 363 - Lab 7

Installing OpenCV for Android

This tutorial is a guide for installing the OpenCV image processing library for Android, as well as for setting up and editing a sample project. OpenCV provides many useful image processing tools for applications such as template matching or image binarization. However, this library is written in C++, while coding for Android in Android Studio is done in Java. This tutorial goes through the steps to integrate the C++ code in the OpenCV library to an Android project, and these steps should be followed carefully to avoid the many errors that could be made in the process.

Note: Use a tablet, not an Android phone, in order to see the application menu.

1. Install the android NDK: https://developer.android.com/ndk/guides/index.html
2. Install OpenCV ***3.3.0***: https://sourceforge.net/projects/opencvlibrary/files/opencv-android/
   Extract this file in a location that is easily accessible, such as C:\ or C:\Android on PC, or /Users/yourusername/desktop on Mac.
3. Open android studio. Click File > New > Import Project
4. Go to the location that OpenCV was saved in and select OpenCV-android-sdk > samples > tutorial-2-mixedprocessing
5. Select Next > Finish
6. Click the “Project” drop-down button and ensure that the android view is selected.
7. Open the build.gradle files for each module: (openCVlibrary330 and openCVTutorial2MixedProcessing). Ensure that the compileSdkVersion corresponds to the SDK version installed. This can be checked under Tools > Android > SDK Manager under the SDK Platforms tab.
8. In each build.gradle file, note the buildTools version listed. Under the SDK Manager, select SDK Tools and click on the check box labeled “Show Package Details”. Ensure that the buildTools version listed in the build.gradle files corresponds to a version installed.
9. Sync the gradle files by clicking “sync now” on the yellow bar that appears stating that the gradle files have been changed on top of the editor.
10. Right click on the folder “openCVTutorial2MixedProcessing in the project window and select “Link C++ Project With Gradle”
11. Set the build system to ndk-build and set the project path to the Android.mk file in the openCVTutorial2MixedProcessing. If the library was installed in C:\, this pathway would be

C:\OpenCV-3.3.0-android-sdk\OpenCV-android-sdk\samples\openCVTutorial2MixedProcessing\jni\Android.mk

12. In the Android.mk file, change the line:

    “include ../../sdk/native/jni/OpenCV.mk”

    to

    “include (your filepath)/OpenCV-android-sdk/sdk/native/jni/OpenCV.mk”

    This file can be accessed by clicking on the error message that pops up when trying to build the project.

13. Ensure that the project builds successfully.

14. Install the OpenCV Manager from the Google Play store.

15. Run the project on an Android device.

Adding a Function

This should add a new menu option to the functions available. When selected, the Thresh function should create an adaptive threshold binary filter that runs in real-time using the camera on the device. The mean value of the pixels is calculated in a window of blockSize x blockSize number of pixels centered around each pixel, and the meanOffset is subtracted from the calculated mean to create a threshold. If the center pixel is above this threshold, the corresponding output pixel is set to black, and if the pixel is below this threshold, the pixel is set to white. This produces the inverted binary effect displayed on the screen.

1. Open Tutorial2Activity.java

2. At the top of the file, add:

    ```
    private static final int VIEW_MODE_THRESH = 3;
    private MenuItem mItemPreviewThresh;
    ```

3. In the method “onCreateOptionsMenu”, add:

    ```
    mItemPreviewThresh = menu.add("Thresh.");
    ```

    This code provides a new menu item to select the binary mode we’re adding.

4. In the method “onCameraFrame” add:

    ```
    case VIEW_MODE_THRESH:
    mRgba = inputFrame.rgba(); //set mRgba to the frame taken from the camera
    int maxValue = 255; //set the maximum value to be taken in as white (255)
    ```

```
int blockSize = 61; //set the window size for adaptive thresholding
int meanOffset = 15; //this is the number subtracted from the calculated mean
Imgproc.adaptiveThreshold(
        inputFrame.gray(), //set grayscale image from camera frame as source
        mIntermediateMat, //set mIntermediateMat as destination
        maxValue,
        Imgproc.ADAPTIVE_THRESH_MEAN_C, //use the mean adaptive threshold method
        Imgproc.THRESH_BINARY_INV, //set output to inverted binarization
        blockSize,
        meanOffset
);
Imgproc.cvtColor( mIntermediateMat, mRgba, Imgproc.COLOR_GRAY2RGBA, 4 );
Break;
```

5. “onOptionsItemSelected”, add:

```
else if (item == mItemPreviewThresh) {
        mViewMode = VIEW_MODE_THRESH;
}
```

Removing a Function

Next, remove the “Canny” function from the application.

Guide based on EE368/CS232 Digital Image Processing, Winter 2018, http://web.stanford.edu/class/ee368/Android/Tutorial-2.pdf