OCR Library Integration Guide

TABLE OF CONTENTS

1. What is OCR Library?.....1
2. Requirement.....2
3. Adding Library to your project.....2
4. Integration.....3

1. What is OCR Library?

The Optical Character Recognition (OCR) is the recognition of printed or written text characters by Mobile Camera.

This involves photo scanning of the text character-by-character, analysis of the scanned-in image, and then translation of the character image into character codes, such as ASCII.

This Library will scan the text character and return the text in string format to your project.

This source code is posted by google on Github.

https://github.com/googlesamples/android-vision/tree/master/visionSamples/ocr-reader

2. Requirement

Before you start to integrate the OCR Library, please make sure you have the following components:

1. Properly installed latest version of Android Studio. Please refer link for downloading Android Studio and related documents.
2. Basic knowledge on setting up Android Libraries. (This document will explain the specific steps for the OCR Library only).
3. Download the OCR Library onto your hard drive.
4. Size of OCR Library will be approximately 150kb

3. Adding Library to your project

Please follow the steps below to integrate OCR Library into your existing android studio project.

1. Extract the provided zip file containing OCR Library "OCRLibrary.aar".
2. OCR Library supports API level 14 and above.
3. Open your existing app in android studio.
4. Click File > New Module
5. Click Import .JAR/.AAR Package then click Next
6. Enter the location of the OCRLibrary.aar file then click Finish.
7. Make sure the library is listed at the top of your settings.gradle file, as shown below:

```
include ':app', ':OCRLibrary'
```

8. Open the app module's build.gradle file:

<your_app>\app\build.gradle

9. Add the following lines to dependencies section:

```
compile 'com.google.android.gms:play-services-vision:11.0.1'
```

10. After adding above line to build.gradle Android Studio will tell you to sync gradle project, the Library classes will be available after that.

4. Integration

4.1 Implement Interface

Implement “OCRCallback” Interface to your Activity/Fragment and get its override method, which will return you the Captured Text.

4.2 Declare Variable

Declare a variable “OCRInit“ of type “OCRInitialization” in your Activity/Fragment.

E.g. OCRInitialization OCRInit;

4.3 Library Initialization

Initialize the declare variable by passing two parameter to “OCRInitialization” method, One is Context and other is Call back Listener.

E.g. OCRInit = new OCRInitialization(this, this);

(Assuming you are using it on Activity)

4.4 Call Capture Method

On your on click event get instance of a Library and call Capture Method to start scanning.

E.g. OCRInit.getInstance(context).captureText();

Finally, you will get a scanned result in your Overrided method in String format.

Please see below snap of code.

```
public class OCRTest extends AppCompatActivity implements OCRCallback{

    OCRInitialization OCRInit;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_ocrtest);

        OCRInit = new OCRInitialization(this, this);
        Button btnSetLibrary = (Button) findViewById(com.vj.coder.ocrlibrary.R.id.btnSetLibrary);
        btnSetLibrary.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                OCRInit.getInstance(OCRTest.this).captureText();
            }
        });
    }

    @Override
    public void onTextReceive(String text) {

        Log.d("Text Capture", text);
        Toast.makeText(OCRTest.this, text, Toast.LENGTH_LONG).show();

    }
}
```

Thank you, Happy Coding ;)