

문서 스캐너 Android 개발자 가이드

Ver. 8.0.1

2019.03.20

(주) 셀바스 AI

목차

1.	개요	3
2.	실행 환경	3
3.	기능 소개	3
4.	API Reference	4
5.	SDK 구성	8
6.	예제에 대한 설명을 통한 개발 가이드	9

1. 개요

본 문서는 문서 스캐너 SDK 에 대한 기술적인 정보를 제공하고 SDK 를 사용하여 기능을 빠르게 구현할 수 있는 방법을 설명한다.

2. 실행 환경

SDK 는 다음과 같은 실행 조건에서 동작하도록 개발되었다.

- 운영 체제: Android 4.0 이상의 ARM CPU 기반 장치
- 인식 대상: 일반 문서 (A4 사이즈)
- 런타임 메모리(RAM): 25MB ~ 185MB
- 예제 개발 환경: Android Studio

3. 기능 소개

SDK 는 다음과 같은 기능을 어플리케이션에 제공한다.

■ 촬영

어플리케이션이 촬영을 요청하면, SDK 의 촬영 모듈은 단말기의 카메라를 구동하며 이후 preview 또는 촬영된 이미지에서 문서를 자동으로 탐지한다.

■ 문서 보정 처리

SDK 의 영상 처리 모듈은 촬영한 이미지의 원근 왜곡, 기울어짐을 파악하여 보정한 이미지를 생성한다.

■ 이미지 보정

SDK 의 이미지 보정 모듈은 촬영한 이미지의 밝기, 대비, 선명도를 조절하고, 흑/백의 이진화 이미지로 변환한다.

4. API Reference

다음과 같은 클래스의 API 를 이용하여 기능을 구현할 수 있다.

■ com.selvasai.selvyocr.mobilesan.ImageProcessor (이미지 영역 관련 처리 및 기타 기능)

- Bitmap **rotateImage**(Bitmap bitmap, int degrees, boolean recycleInputImage)

기능	입력된 비트맵을 회전하여 출력
입력 값	bitmap: 원본 비트맵 degrees: 회전 각도 (90 도 단위) recycleInputImage: inputImage 를 recycle 할지 여부
출력 값	회전된 비트맵

- Bitmap **getPerspectiveImage**(Bitmap inputImage, int[] area, int outWidth, int outHeight, boolean recycleInputImage)

기능	입력된 비트맵의 특정 영역을 잘라 왜곡 보정
입력 값	inputImage: 원본 비트맵 area: 보정하려는 영역 int outWidth: 결과 이미지 너비 int outHeight: 결과 이미지 높이 recycleInputImage: inputImage 를 recycle 할지 여부
출력 값	area 영역으로 잘라 보정한 비트맵

- double **getBlurValue**(byte[] previewData, int previewWidth, int previewHeight)

기능	프리뷰 영상에서 초점이 흐린 정도를 측정
입력 값	previewData: 프리뷰 영상 previewWidth: 프리뷰 너비 previewHeight: 프리뷰 높이
출력 값	0~1 사이의 값 또는 NaN 1 이면 매우 흐림

- double **getBlurValue**(byte[] previewData, int previewWidth, int previewHeight, Rect area)

기능	프리뷰 영상에서 초점이 흐린 정도를 측정
입력 값	previewData: 프리뷰 영상 previewWidth: 프리뷰 너비 previewHeight: 프리뷰 높이 area: 흐린 정도를 측정할 대상 영역

출력 값	0~1 사이의 값 또는 NaN 1 이면 매우 흐림
------	--------------------------------

- double **getBlurValue**(Bitmap bitmap)

기능	이미지에서 초점이 흐린 정도를 측정
입력 값	bitmap: 비트맵 이미지
출력 값	0~1 사이의 값 또는 NaN 1 이면 매우 흐림

- double **getBlurValue**(Bitmap bitmap, Rect area)

기능	이미지에서 초점이 흐린 정도를 측정
입력 값	bitmap: 비트맵 이미지 area: 흐린 정도를 측정할 대상 영역
출력 값	0~1 사이의 값 또는 NaN 1 이면 매우 흐림

- int **isValidArea**(int[] detectedAreaPoints, Rect pivotAreaRect)

기능	getCardArea 를 통해 검출된 영역을 카메라 촬영 가이드 력트와 비교하여 올바른지 판단
입력 값	detectedAreaPoints: getCardArea 를 통해 검출한 문서 영역 pivotAreaRect: 촬영 가이드 력트
출력 값	판단 결과 Utils.AREA_VALID: 유효한 영역 Utils.AREA_NOT_FOUND: 입력 값에 오류가 있음 Utils.AREA_TOO_SMALL: 영역이 너무 작음

- int[] **getArea**(Bitmap inputImage, Rect guideRect, PointF[] quadPoints)

기능	입력된 비트맵에서 지정된 영역내의 영역검출 수행 quadPoints 값이 있으면 int[] 로 변경만 해주고 종료
입력 값	image: 영역 검출 대상 이미지 guideRect: 영역 검출을 수행할 영역 quadPoints: Crop 화면에서 사용자가 편집한 영역
출력 값	검출된 영역 좌표

- Bitmap **cropImage**(Bitmap input, Rect cropRect, int resizeWidth, int resizeHeight, boolean recycleInputImage)

기능	입력 이미지를 Crop 후 Resize
입력 값	input: 입력 비트맵 이미지

	cropRect: 자르려는 영역 resizeWidth: 자른 영역을 크기 변경할 너비 값 resizeHeight: 자른 영역을 크기 변경할 높이 값 recycleInputImage: 입력 이미지를 recycle 할지 여부
출력 값	Crop 후 Resize 된 이미지

- int **checkAreaSimilar**(byte[] data, int width, int height, int[] docArea, Rect guideRect, int cameraOrientation)

기능	입력 이미지에서 문서 영역을 검출하고 이전에 검출한 영역과 비교하여 유사도 반환
입력 값	data: 입력 비트맵 이미지 width: 자르려는 영역 height: 자른 영역을 크기 변경할 너비 값 docArea: 자른 영역을 크기 변경할 높이 값 (새로 검출된 영역이 출력됨) guideRect: 촬영 가이드 영역 cameraOrientation: 카메라 회전값
출력 값	영역검출 후 비교 결과 Utils.FIND_AREA_FAIL: 영역검출 실패 Utils.FIND_AREA_SIMILAR: 영역검출 결과 비슷함 Utils.FIND_AREA_DIFFERENT: 영역검출 결과가 다름 입력 값인 int[]docArea 로 새로 검출된 영역이 전달됨

- void **contrastBrightness**(Bitmap input, Bitmap output, int contrast, int brightness)

기능	이미지의 밝기, 대비 값 조절
입력 값	input: 입력 이미지 output: 보정된 이미지 (gray 이미지) contrast: 보정하려는 대비 값 (값 범위 -100 ~ 100) brightness: 보정하려는 밝기 값 (값 범위 -100 ~ 100)
출력 값	-

- void **sharpen**(Bitmap input, Bitmap output, int value)

기능	이미지를 선명하게 보정
입력 값	input: 입력 이미지 output: 보정된 이미지 (gray 이미지) value: 보정하려는 선명도 값 (값 범위 0 ~ 10)
출력 값	-

- void **soften**(Bitmap input, Bitmap output, int value)

기능	이미지를 흐리게 보정
입력 값	input: 입력 이미지 ouput: 보정된 이미지 (gray 이미지) value: 보정하려는 흐림 값 (값 범위 0 ~ 10)
출력 값	-

- void **auto**(Bitmap input, Bitmap output)

기능	이미지 자동 보정
입력 값	input: 입력 이미지 ouput: 보정된 이미지 (gray 이미지)
출력 값	-

- void **binarizationTheashold**(Bitmap input, Bitmap output, int threshold)

기능	이미지 이진화
입력 값	input: 입력 이미지 ouput: 보정된 이미지 (흑백 이미지) threshold: 이진화 값 (값 범위 0 ~ 255)
출력 값	-

- void **binarizationAdaptiveTheashold**(Bitmap input, Bitmap output, int blockSize, int c)

기능	이미지 이진화
입력 값	input: 입력 이미지 ouput: 보정된 이미지 (흑백 이미지) blockSize: threshold 계산에 사용되는 이웃하는 픽셀의 수(3, 5, 7, ...) c: threshold 계산에 사용되는 상수 값
출력 값	-

- void **saveBitmapToTiff**(Bitmap input, String path, boolean useJPEG2000, int dpi)

기능	TIFF 파일로 이미지 저장
입력 값	input: 입력 이미지 path: 저장 경로 useJPEG2000: TIFF 파일 안에 JPEG2000 형식으로 저장할 지 여부 <ul style="list-style-type: none"> - True: JPEG2000(칼라) 이미지로 저장 - False: 이진화(흑백) 이미지로 저장 dpi: DPI 값 설정
출력 값	-

■ **com.selvasai.selvyocr.mobilesan.util.LicenseChecker (라이선스 확인)**

- boolean **isValidLicense**(Context context)

기능	라이선스가 유효한지 여부 판단 SDK 사용 초기에 호출하지 않으면 제한된 날짜까지만 사용제한
입력 값	context : Application context
출력 값	라이선스 유효 여부 (true – 유효, false – 만료)

■ **com.selvasai.selvyocr.mobilesan.util.BuildInfo (버전 확인)**

- String **getVersion**()

기능	SDK 버전 확인
입력 값	-
출력 값	SDK 의 버전

5. SDK 구성

Android 용 SDK 는 본 개발자 가이드와 아래와 같은 라이브러리와 예제 앱으로 구성된다

■ **문서 스캐너 라이브러리**

- app\libs\
 - libSelvyMobileScan.jar: OCR 모듈
- app\src\main\jniLibs\armeabi-v7a\
 - libgnustl_shared.so: GNU STL 라이브러리
 - libSelvyImageProcessing.so: 이미지 처리 JNI 라이브러리
- app\src\main\jniLibs\arm64-v8a\
 - libgnustl_shared.so: GNU STL 라이브러리
 - libSelvyImageProcessing.so: 이미지 처리 JNI 라이브러리

■ **안드로이드 스튜디오 예제**

- app\src\main\java\com\selvasai\selvyocr\mobilesan\sample
 - SelvyMobileScanActivity.java
: 문서 스캐너에서 촬영과 영역 검출의 과정을 처리하는 Activity
 - ImageProcessActivity.java
: 문서 스캐너에서 이미지를 보정 하는 Activity

Copyright 2016. Selvas AI Inc. all rights reserved.

- MainActivity.java
: 문서 스캐너 SDK 의 기능을 사용하는 클라이언트 Activity
- ResultActivity.java
: 문서 스캔 결과 보정된 이미지를 표시하는 Activity
- app\src\main\java\com\selvasai\selvyocr\mobilescan\sample\util
 - BackgroundThread.java
: Background 에서 동작하기 위한 Thread 설정 클래스
 - Utils.java
: 반복 사용되는 코드 모음
- app\src\main\java\com\selvasai\selvyocr\camera
 - CameraView.java
: CameraPreviewView 와 CameraOverlayView 를 포함하고 제어하는 카메라
 - CameraPreviewView.java
: 카메라 화면이 보여질 서피스 뷰
 - CameraOverlayView.java
: 카메라 뷰 위에서 가이드 라인을 그리는 뷰

6. 예제에 대한 설명을 통한 개발 가이드

SDK 의 촬영 및 보정 기능을 이용하는 어플리케이션은 다음과 같은 절차로 SDK 를 어플리케이션에 통합한다.

(1) 필요 권한 및 요소 추가

- A. 카메라 사용을 위한 권한 및 요소 추가 (app\src\main\AndroidManifest.xml 참고)

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<uses-feature android:name="android.hardware.camera" />
```

```
<uses-feature android:name="android.hardware.camera.autofocus" />
```

- B. Camera 화면 실행 전에 Runtime permission 추가

(app\src\main\java\com\selvasai\selvyocr\mobilescan\sample\MainActivity.java
참고)

(2) Activity 추가

- A. app\src\main\java\com\selvasai\selvyocr\mobilescan\sample\
 - SelvyMobileScanActivity.java

(3) 모듈 추가

- A. app\src\main\java\com\selvasai\selvyocr\camera

Copyright 2016. Selvas AI Inc. all rights reserved.

- CameraView.java
 - CameraPreviewView.java
 - CameraOverlayView.java
- B. app\src\main\java\com\selvasai\selvyocr\widcard\sample\util
- BackgroundThread.java
 - Utils.java

(4) 라이브러리 추가

- A. Jar 파일 추가
- app\libs\libSelvyMobileScan.jar
- B. so 파일 추가
- app\src\main\jniLibs\armeabi-v7a\
- app\src\main\jniLibs\arm64-v8a\
- libgnustl_shared.so
 - libSelvyImageProcessing.so

(5) 리소스 추가 (추후 UI/GUI 변경시 변경 예정)

- A. 레이아웃 리소스 추가
- app\src\main\res\layout\
- activity_camera.xml
 - layout_noti_focus.xml
- B. 이미지 리소스 추가
- i. app\src\main\res\drawable\btn_camera_shutter_holo.png
 - ii. app\src\main\res\drawable\recog_target.png
- C. 스트링 리소스 추가
- i. app\src\main\res\values\strings.xml

(6) 기타 Activity 작성

별도의 기능을 사용하기 위한 Activity 작성

- A. 이미지 보정 처리
- app\src\main\java\com\selvasai\selvyocr\mobilescan\sample\ImageProcessActivity.java
- 참고

(7) 호출 및 결과 처리

- A. 호출
- app\src\main\java\com\selvasai\selvyocr\mobilescan\sample\MainActivity.java
- 참고
- B. 결과

- app\src\main\java\com\selvasai\selyocr\mobilescan\sample\ResultActivity.java
참고

- 이상 끝 -.