

Interface Description of Static Portrait Segmentation SDK for Android

1.Reference library files

1.1 Add library files

Copy libs/humanseg-release.aar to the libs directory of your project.

带格式的: 缩进: 首行缩进: 1 字符

1.2 Configure build.gradle

Configure app/build.gradle and add the following content to the dependencies node:

带格式的: 缩进: 首行缩进: 1 字符

```
implementation(name: 'humanseg-release', ext: 'aar')
```

带格式的: 缩进: 首行缩进: 0 字符

Note:

For the specific configuration of build.gradle, please refer to build.gradle in the demo.

带格式的: 缩进: 首行缩进: 1 字符

2.Configure AndroidManifest.xml

Add network and storage permissions for license verification. If you only need to perform local license verification, you can remove network permissions:

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

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

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

Note:

For the specific configuration of AndroidManifest.xml, please refer to AndroidManifest.xml in the demo.

带格式的：缩进：首行缩进： 1 字符

3.Interface Description

3.1 com.wonxing.humanseg.WXLicense

删除的内容：class

This class contains the authorization interfaces. All methods in the class are static functions. To use SDK, you must call authorization interfaces first. This interface also includes two kinds of functions: fetching license (need to connect to the network) and offline authorization.

带格式的：缩进：首行缩进： 1 字符

删除的内容：online authorization

删除的内容：

删除的内容：function

3.1.1 getDeviceID()

static String getDeviceID(Context appCtxt)

带格式的：字体：倾斜

带格式的：缩进：首行缩进： 1 字符

This static method is used to get the device ID.

删除的内容：Instruction： .

带格式的：缩进：首行缩进： 1 字符

Parameters:

appCtxt - the context object of APP which cannot be null.

Return value:

String - Device local ID.

删除的内容：s

删除的内容： value

Example:

```
Context appCtx = getApplicationContext();
String deviceId = WXLicense.getDeviceID(appCtx);
```

3.1.2 fetchLicense()

删除的内容：function

static void fetchLicense(final Context appCtx, final Handler handler, final String appKey, final String appSecret)

带格式的：字体：倾斜

带格式的：缩进：首行缩进： 1 字符

This static method is the asynchronous request for the current device authorization.

删除的内容：Instruction： .

Parameters:

appCtxt - the context object of APP which cannot be null.

已下移 [3]: There is a limit on the number of times that the network can get license authorization. After the request is successful, the license should be saved locally. During the term of validity of the license, the local license can be directly used to initialize the SDK. .

handler - the handler object which receives and processes the returned license data.

appKey - the public key which is provided by Wonxing company.

appSecret - the private key which is provided by Wonxing company.

Return value:

No return value.

Note:

There is a limit on the number of times that the network can get license authorization. After the request is successful, the license should be saved locally. During the term of validity of the license, the local license can be directly used to initialize the SDK.

Example:

```
@SuppressWarnings("HandlerLeak")
private Handler mHandler = new Handler() {
    @Override
    public void handleMessage(Message msg) {
        switch (msg.what) {
            case WXLICENSE.CODE_LIC_FETCH_FAILED:
                Toast.makeText(appCtx, (CharSequence) msg.obj,
                    Toast.LENGTH_LONG).show();
                break;
            case WXLICENSE.CODE_LIC_FETCH_SUCCESSFUL:
                Toast.makeText(appCtx, "fetch license successful.",
                    Toast.LENGTH_LONG).show();
                byte[] licData = (byte[]) msg.obj;
                saveLicToSdcard(licData);
                initSDK(licData);
                break;
        }
    }
};
WXLICENSE.fetchLicense(appCtx, mHandler, AppKey, AppSecret);
```

3.1.3 fetchLicenseWithDeviceID()

static void fetchLicenseWithDeviceID(Context appCtx, Handler handler, final String deviceID, final String appKey, final String appSecret)

This static method is used to get the license authorization of the specified device.

删除的内容:

删除的内容: s

删除的内容: value

已移动(插入) [3]

删除的内容: function fetchLicenseViaDeviceID

删除的内容: Via

带格式的: 字体: 倾斜

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: Instruction: .

Parameters:

appCxt - the context object of APP which cannot be null.
handler - the handler object which receives and processes the returned license data.
deviceId - the device ID. Please refer to 3.1.1
appKey - the public key which is provided by Wonxing company.
appSecret - the private key which is provided by Wonxing company.

Return value:

No return value.

Example:

```
WXLICENSE.fetchLicenseWithDeviceID(appCxt, mHandler, deviceId, AppKey, AppSecret);
```

3.1.4 setLicense()

```
static int setLicense(Context appCxt, byte[] licData)
```

This static method is to initialize the SDK with the license data.

Parameters:

appCxt - the context object of APP which cannot be null.
licData - license data.

Return value:

Int - 0 means the setting is successful, non-0 means the setting is failed.

Example:

```
int res = WXLICENSE.setLicense(appCxt, licData);
```

3.1.5 getLicenseValidTime()

```
static long[] getLicenseValidTime()
```

This static method is used to get the valid start and end timestamp for the license. It is called after the successful initialization of 3.1.4.

Parameters:

No parameters.

删除的内容: value

删除的内容: s

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: function initViaLicenseData

删除的内容: Data

带格式的: 字体: 倾斜

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: initVia

删除的内容: Instruction: .

删除的内容:

删除的内容: s

删除的内容: value

删除的内容: initialization

删除的内容: zero

删除的内容: initialization

删除的内容: .

删除的内容: .

带格式的: 字体: (默认) + 西文正文 (Calibri), (中文) + 中文正文 (宋体), 字体颜色: 自动设置, 非加宽 / 紧缩量, 图案: 清除

带格式的: 两端对齐, 无孤行控制

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: .initVia

删除的内容: Data

删除的内容: function

删除的内容: Timestamp

带格式的: 字体: 倾斜

带格式的: 缩进: 首行缩进: 1 字符

带格式的: 字体: (中文) 宋体, 11.5 磅, 倾斜

带格式的: 字体: 倾斜

删除的内容: getLicenseTimestamp

删除的内容: Instruction: .

删除的内容: Return value

删除的内容: return value

Return **value**:
long[] - The valid start and end timestamp for the license.

Example:
long[] licTimes = WXLICENSE.[getLicenseValidTime](#)();

3.2 [com.wonxing.humanseg.WXHumanSeg](#)

This class contains portrait segmentation interfaces. The first function is the static function `init()`. After the initialization is successful, all other portrait segmentation interfaces can be called.

3.2.1 `init()`

`static void init(Context appCtx) throws WXException`

initialize the SDK, only after this method is called, the others methods of class `WXHumanSeg` can be invoked.

Parameters:

`appCtx` - the context object of APP which cannot be null.

Returns:

Int - 0 means the initialization is successful, non-0 means the initialization is failed.

Example:

`int res = WXHumanSeg.init(appCtx);`

3.2.2 `segToAlpha()`

`Bitmap segToAlpha(Bitmap srcBitmap) throws WXException`

This instance method is a portrait segmentation interface which returns Alpha Bitmap.

Parameters:

`srcBitmap` – the source Bitmap, and the format is 'Bitmap.Config.ARGB_8888'.

Return:

删除的内容: s

删除的内容: value

带格式的: 字体: (中文) 宋体, 11.5 磅, 倾斜

带格式的: 字体: 倾斜

删除的内容: `getLicenseTimestamp`

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: class

删除的内容: This class contains the portrait segmentation interfaces, and all methods in the class are instance methods.

带格式的: 缩进: 首行缩进: 1 字符

带格式的: 字体: (默认) Arial, 11.5 磅, 非加粗, 倾斜, 字体颜色: 自定义颜色 (RGB(52, 73, 94))

带格式的: 缩进: 首行缩进: 1 字符

带格式的: 字体: 倾斜

删除的内容: .

删除的内容: ,

带格式的: 字体: (默认) Arial, 11.5 磅, 字体颜色: 自定义颜色 (RGB(52, 73, 94))

带格式的: 字体: (默认) Arial, (中文) 宋体, 11.5 磅, 字体颜色: 自定义颜色 (RGB(52, 73, 94))

带格式的: 字体: (默认) Arial, (中文) 宋体, 11.5 磅, 字体颜色: 自定义颜色 (RGB(52, 73, 94))

带格式的

带格式的: 正文, 缩进: 首行缩进: 0.74 厘米

带格式的: 正文

带格式的

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: .

带格式的

带格式的

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: .

删除的内容: 1

删除的内容: function

带格式的: 字体: 倾斜

带格式的

带格式的: 字体: 倾斜

带格式的: 缩进: 首行缩进: 1 字符

删除的内容: Instruction: .

删除的内容:

删除的内容: s

删除的内容: value

The Alpha Bitmap which has the same size as the source Bitmap, and the format is 'Bitmap.Config.ALPHA_8'.

Example:

```
WXHumanSeg wxHumanSeg = new WXHumanSeg();  
Bitmap resBitmap = wxHumanSeg.segToAlpha(srcBitmap);
```

3.2.3. segAndComposition()

Bitmap segAndComposition (Bitmap srcBitmap, int[] bgcolor) throws WXException

This instance method is a portrait segmentation interface which returns the Bitmap of the combination of the specified background image and the segmented portrait.

Parameters:

srcBitmap - the source Bitmap, and the format is 'Bitmap.Config.ARGB_8888'.

bgBitmap - the background Bitmap, and the format is 'Bitmap.Config.ARGB_8888'.

Returns:

Bitmap - the Bitmap of the combination of the specified background image and the segmented portrait, and the combined Bitmap has the same size as the srcBitmap.

Note:

The size of bgBitmap should be the same as that of srcBitmap, otherwise bgBitmap will be scaled.

Example:

```
BitmapFactory.Options options = new BitmapFactory.Options();  
options.inScaled = false;  
Bitmap bgBitmap = BitmapFactory.decodeResource(getResources(),  
R.mipmap.demo_bg, options);  
Bitmap resBitmap = wxHumanSeg.segAndComposition(srcBitmap, bgBitmap);
```

Note: For specific interface calls, please refer to the MainActivity.java in the demo.

删除的内容: 2

删除的内容: function

删除的内容: To

删除的内容: BgColor

删除的内容: segToBgColor

带格式的: 字体: 倾斜

带格式的: 缩进: 首行缩进: 1 字符

带格式的: 字体: (中文) 宋体, 11.5 磅, 倾斜

带格式的: 字体: 倾斜

带格式的: 字体: (默认) Arial, (中文) 宋体, 11.5 磅, 非加粗, 倾斜, 字体颜色: 自定义颜色 (RGB (52, 73, 94))

带格式的: 字体: 倾斜

已下移 [2]: Instruction:

带格式的

删除的内容: .

删除的内容: .

带格式的: 两端对齐, 无孤行控制

已移动(插入) [1]

删除的内容: Note: For specific interface calls, please refer to the MainActivity.java in the demo. .

已上移 [1]: Note: For specific interface calls, please

带格式的: 缩进: 首行缩进: 0 厘米

删除的内容: color

删除的内容: -

删除的内容: color

删除的内容: the format is RGB.

删除的内容: value

删除的内容: source

已移动(插入) [2]

删除的内容: Instruction

删除的内容: .

带格式的: 缩进: 首行缩进: 1 字符

带格式的

删除的内容: WXHumanSeg wxHumanSeg = new

带格式的

带格式的

带格式的

带格式的

带格式的

带格式的