

# Android 人脸检测

## Android 人脸检测方法，使用 android.media, play service 和 Face++ 识别人脸

Posted on August 06, 2016 by Ein Verne | View revision history ([https://github.com/einverne/einverne.github.io/commits/master/\\_posts/2016-08-06-android-face-detect.md](https://github.com/einverne/einverne.github.io/commits/master/_posts/2016-08-06-android-face-detect.md))

最近项目使用，总结了目前 Android 中使用到的人脸检测技术，主要分成三部分来介绍：

- Android 原生支持人脸检测，从 API 1 [🔗](#) 开始就提供原生的人脸检测，但是该方法识别率相对较低
- 在后来的发展中 Google 将人脸识别技术放到了 Google Play Services [🔗](#) 中，从 Google Play Service 7.8 之后提供的 Mobile Vision API 中我们可以使用新的人脸检测API。这个方法不仅能够识别有没有人脸，还能够识别人脸的区域，甚至在视频流中追踪相同的人脸，并且提供了一系列方法来获取眼睛、嘴巴、鼻子、脸颊的位置，利用这个 API 就能够在动态的视频中给人脸增加一些有趣的效果，加个胡子，带个帽子，等等，市面上有一些动态增加表情的 App 利用的似乎就是这个技术。
- Face++，国内一家做人脸检测的提供商，不过其免费版服务需经过网络，也就意味着需要联网将照片上传到其服务器，然后他返回识别的结果回来。
- 其他服务，经过搜索 OpenCV 也能提供类似的服务 [🔗](#)，但是目前我还没有详细了解。OpenCV 做计算机视觉有一定积累，相信识别准确率应该有保障。

接下来就依次介绍前三种人脸检测的方法。

## Android 原生人脸检测API

android.media 包中的人脸检测API 有如下两个限制：

- Bitmap 必须以 Config.RGB\_565 解码
- 输入的 Bitmap 宽度需要为偶数

只要注意这两个限制，另图片眼睛的距离不要太小，其他的代码核心的没几句话。

```

/**
 * There are some limitation in this 用android.media 包中识别人脸package.
 * 使用使用使Face Detection API's input Bitmap must :
 * <p/>
 * 1. config with Config.RGB_565<br/>
 * 2. Bitmap width must be even<br/>
 * <p/>
 * more details can be checked
 * http://stackoverflow.com/q/17640206/1820217
 *
 * @param bitmap Bitmap
 */
private void detectUsingNative(final Bitmap bitmap) {
    if (null == bitmap || isRunning) {
        if (listener != null) {
            listener.onFail();
        }
        return;
    }
    facesCount = 0;
    final android.media.FaceDetector faceDetector = new android.media.FaceDetector(bitmap
        .getWidth(), bitmap.getHeight(), MEDIA_MAX_DETECT_FACE_NUMBER);
    androidNativeFacesResults = new android.media.FaceDetector.Face[MEDIA_MAX_DETECT_FACE
        _NUMBER];
    final Handler handler = new Handler();
    thread = new Thread() {
        @Override
        public void run() {
            facesCount = faceDetector.findFaces(bitmap, androidNativeFacesResults);

            handler.post(new Runnable() {
                @Override
                public void run() {
                    if (listener != null) {
                        listener.onSuccess();
                    }
                }
            });

            isRunning = false;
        }
    };
    thread.start();
    isRunning = true;
}

```

## Play Service 中人脸检测

Play Service 中的人脸检测是随着 Mobile Vision API 一同出现  的，这个库中还有一些其他的API，比

如识别二维码，识别文字等等，并且 Play Service 中的人脸识别更准确的说应该叫人脸追踪，在官方实现的 Demo [🔗](#) 中，直接调用手机摄像，能够一直追踪镜头中的同一人头像。

以下是部分实现，详细代码可参考文末给出的 GitHub 代码。

```
/**
 * 使用 Play Service 中人脸检测
 *
 * @param bitmap Bitmap
 */
private void detectUsingGms(Bitmap bitmap) {
    if (null == bitmap) {
        if (listener != null) {
            listener.onFail();
        }
        return;
    }
    facesCount = 0;

    detector = new FaceDetector.Builder(context)
        .setTrackingEnabled(false)
        .setLandmarkType(FaceDetector.ALL_LANDMARKS)
        .setClassificationType(FaceDetector.ALL_CLASSIFICATIONS)
        .build();

    // This is a temporary workaround for a bug in the face detector with respect to oper
    ating
    // on very small images. This will be fixed in a future release. But in the near te
    rm, use
    // of the SafeFaceDetector class will patch the issue.
    Detector<Face> safeDetector = new SafeFaceDetector(detector);

    // Create a frame from the bitmap and run face detection on the frame.
    Frame frame = new Frame.Builder().setBitmap(bitmap).build();

    faces = safeDetector.detect(frame);

    if (!safeDetector.isOperational()) {
        // Note: The first time that an app using face API is installed on a device, GMS
        will
        // download a native library to the device in order to do detection. Usually thi
        s
        // completes before the app is run for the first time. But if that download has
        not yet
        // completed, then the above call will not detect any faces.
        //
        // isOperational() can be used to check if the required native library is current
        ly
        // available. The detector will automatically become operational once the librar
        y
        // download completes on device.

        if (listener != null) {
            listener.onFail();
        }
        return;
    }
}
```

```
    if (listener != null) {  
        listener.onSuccess();  
    }  
}
```

## Face++ 人脸检测服务

Face++ 提供了联网的人脸检测服务，需要到其网站上注册开发者账号获取API使用权限。其大概检测代码如下：

```

/**
 * 使用 Face++ 人脸检测
 *
 * @param file File
 */
private void detectUsingFacePlus(File file) {
    if (!file.exists() || isRunning) {
        if (listener != null) {
            listener.onFail();
        }
        return;
    }
    final PostParameters parameters = new PostParameters();
    parameters.setImg(file);
    final Handler handler = new Handler();
    facesCount = 0;
    thread = new Thread() {
        @Override
        public void run() {
            boolean hasFace = false;
            boolean detectSucceed = false;
            Log.d("FacePlusDetect", "Detect Request :" + parameters.toString());
            HttpRequests httpRequests = new HttpRequests(FACEPLUSPLUS_APIKEY, FACEPLUSPLUS_APISECRET, false, true);
            JSONObject result;
            try {
                result = httpRequests.detectionDetect(parameters);
                if (result != null) {
                    detectSucceed = true;
                    JSONArray faces = result.getJSONArray("face");
                    double imgWidth = result.getDouble("img_width");
                    double imgHeight = result.getDouble("img_height");
                    if (faces != null && faces.length() > 0 && null != listener) {
                        // Has face!!
                        facesCount = faces.length();
                        facePlusResults = new RectF[facesCount];
                        hasFace = true;
                        for (int i = 0; i < facesCount; i++) {
                            float x, y, w, h;
                            facePlusResults[i] = new RectF();
                            // 需注意返回结果的center,width,height 都为0~100, 百分比
                            x = (float) faces.getJSONObject(i).getJSONObject("position").getJSONObject("center").getDouble("x");
                            y = (float) faces.getJSONObject(i).getJSONObject("position").getJSONObject("center").getDouble("y");
                            w = (float) faces.getJSONObject(i).getJSONObject("position").getDouble("width");
                            h = (float) faces.getJSONObject(i).getJSONObject("position").getDouble("height");

                            float realx = (float) (x * imgWidth / 100);
                            float realy = (float) (y * imgHeight / 100);
                            float realw = (float) (w * imgWidth / 100);

```

```

        float realh = (float) (h * imgHeight / 100);
        facePlusResults[i].set(realx - realw / 2,
                                realy - realh / 2,
                                realx + realw / 2,
                                realy + realh / 2);
    }
    //      String genderStr = playServiceFaces.getJSONObject(0).getJSO
NObject("attribute").getJSONObject("gender").getString("value");
    //      gender = Gender.valueOf(genderStr);
    } else {
        hasFace = false;
    //      detectSucceed = true;
    //      gender = Gender.OTHER;
    }
    //      Log.d("FacePlusDetect", "Detect Result : hasFace = " + hasFace
+ "; gender = " + gender.toString());
    }
    } catch (FaceppParseException e) {
        detectSucceed = false;
        Log.d(TAG, "Detect FaceppParseException !");
        e.printStackTrace();
    } catch (JSONException e) {
    //      if (hasFace) {
    //          gender = Gender.OTHER;
    //      }
        Log.d(TAG, "Detect JSONException !");
        e.printStackTrace();
    }

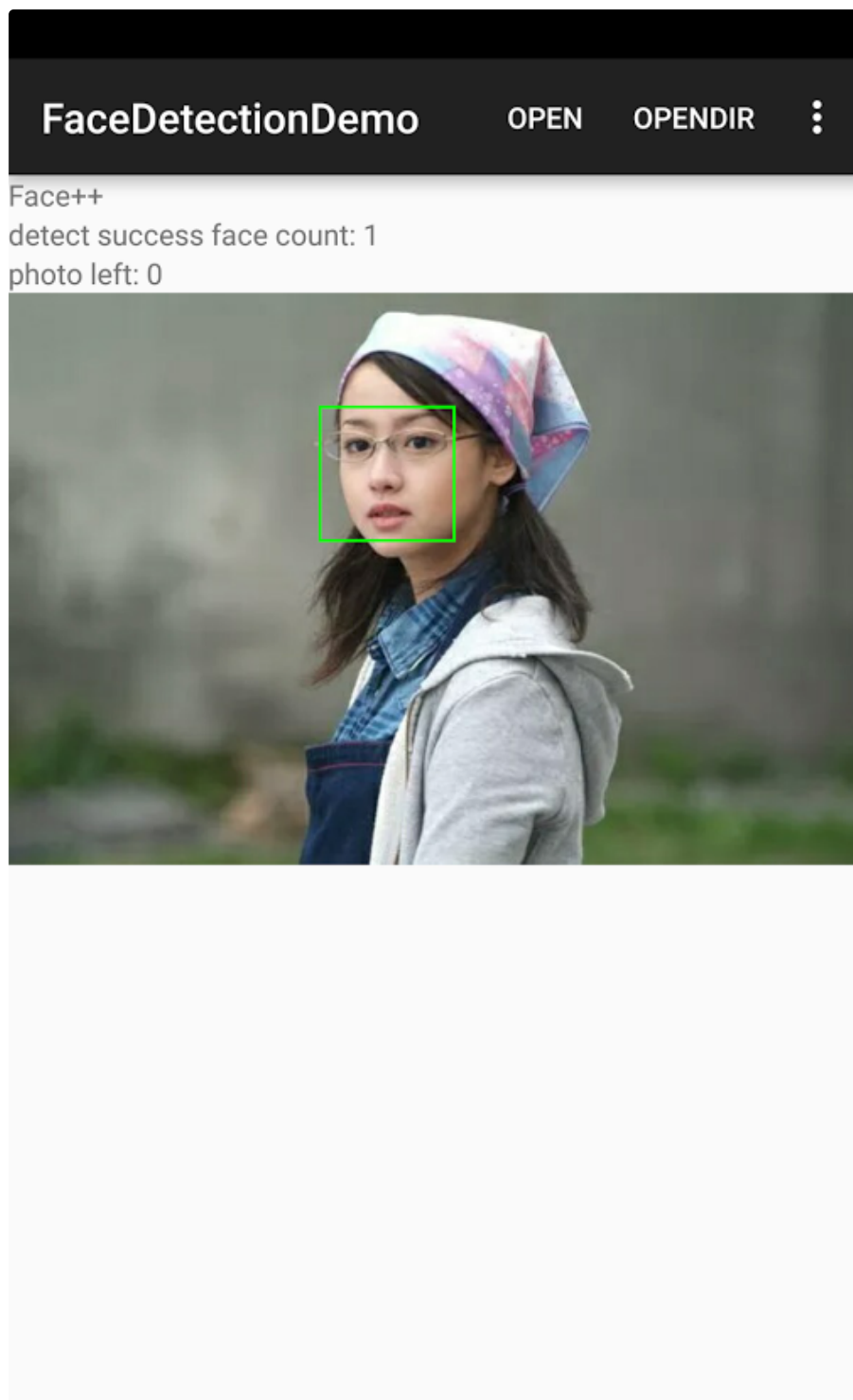
    if (detectSucceed) {
        handler.post(new Runnable() {
            @Override
            public void run() {
                if (listener != null) {
                    listener.onSuccess();
                }
            }
        });
    } else {
        handler.post(new Runnable() {
            @Override
            public void run() {
                if (listener != null) {
                    listener.onFail();
                }
            }
        });
    }

    isRunning = false;
}
};

```

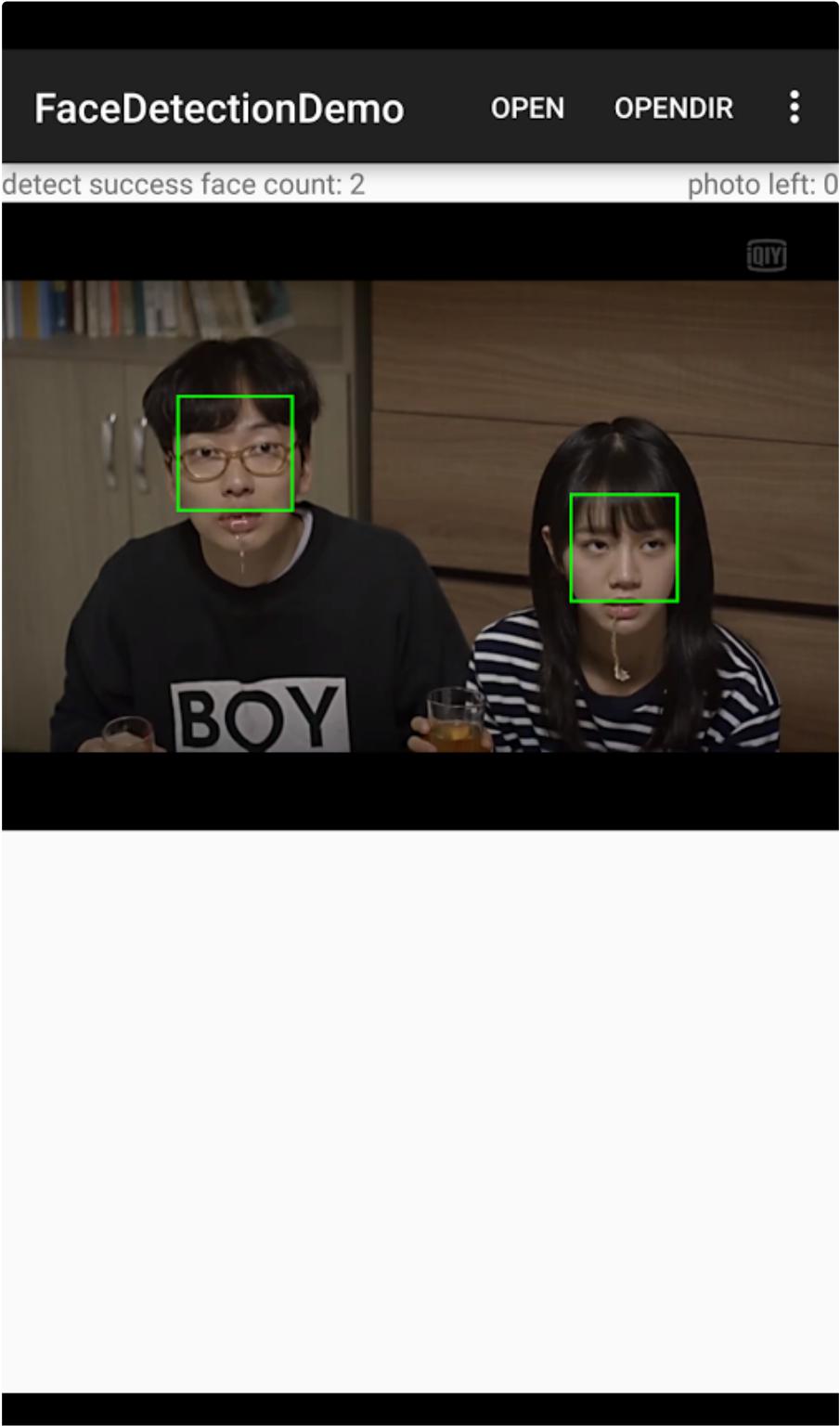
```
thread.start();  
isRunning = true;  
}
```


## Detect Result






([https://lh4.googleusercontent.com/-zNsRgSMnO10/V6899VHy\\_BI/AAAAAABA94/nejE8EBBYn0Q8Zx\\_Xbx\\_tlZFjT1SSY5YwCL0B/w506-h900-no/Screenshot\\_20160813-233236.png](https://lh4.googleusercontent.com/-zNsRgSMnO10/V6899VHy_BI/AAAAAABA94/nejE8EBBYn0Q8Zx_Xbx_tlZFjT1SSY5YwCL0B/w506-h900-no/Screenshot_20160813-233236.png))











([https://lh4.googleusercontent.com/-b2xwMmkue54/V6899QoTCzI/AAAAAABA94/qs11ZhIW2E0PUjY6AoHRS92QpFmFNOqrQCL0B/w506-h900-no/Screenshot\\_20160807-214743.png](https://lh4.googleusercontent.com/-b2xwMmkue54/V6899QoTCzI/AAAAAABA94/qs11ZhIW2E0PUjY6AoHRS92QpFmFNOqrQCL0B/w506-h900-no/Screenshot_20160807-214743.png))

FaceDetectDemo 代码可参考：<https://github.com/einverne/AndroidFaceDetectDemo> 

## Reference

全部代码可参考：<https://github.com/einverne/Android-Face-Recognition> 

- <http://opencv.org/platforms/android.html> 
- <https://github.com/crankdaworld/OpenCV-Android-FaceDetect-GoodFeature> 
- <http://code.tutsplus.com/tutorials/an-introduction-to-face-detection-on-android--cms-25212> 
- <http://stackoverflow.com/questions/17839388/creating-a-scaled-bitmap-with-createscaledbitmap-in-android> 
- <http://stackoverflow.com/questions/37300874/why-isnt-the-new-play-services-facedetector-available-on-some-devices> 
- <http://stackoverflow.com/questions/17640206/how-to-overcome-the-face-detection-api-restrictions> 

## Related Posts

- 24 Nov 2016 - Android GPS 反作弊 (</post/2016/11/android-prevent-gps-fake.html>)
- 23 Nov 2016 - Android 减小 APK 大小 (</post/2016/11/reduce-apk-size.html>)
- 16 Nov 2016 - Android Gradle 学习笔记 (</post/2016/11/android-gradle.html>)
- 15 Nov 2016 - Android ProGuard (</post/2016/11/android-proguard.html>)
- 14 Nov 2016 - Android Http 调试及抓包 (</post/2016/11/android-http-proxy-debug.html>)
- 21 Oct 2016 - Android 过渡动画框架 (</post/2016/10/android-transition-framework.html>)
- 11 Oct 2016 - Android 提醒 (</post/2016/10/android-dialog.html>)
- 09 Oct 2016 - Android 使用自定义 keystore 调试 (</post/2016/10/android-debug-with-custom-keystore.html>)
- 29 Sep 2016 - Android 常见错误 (</post/2016/09/android-common-error.html>)
- 27 Sep 2016 - android 6 runtime permission (</post/2016/09/android-6-runtime-permission.html>)
- 26 Sep 2016 - Android Snackbar 使用 (</post/2016/09/android-snackbar.html>)
- 25 Sep 2016 - Android Notification (</post/2016/09/android-notification.html>)
- 18 Sep 2016 - Gradle 重复导入错误 (</post/2016/09/gradle-dependencies-duplicate-entry.html>)
- 08 Sep 2016 - 常用 adb command 命令 (</post/2016/09/useful-adb-command.html>)
- 06 Sep 2016 - Android Animation Interpolator (</post/2016/09/android-animation-interpolator.html>)
- 05 Sep 2016 - Android ImageView ScaleType (</post/2016/09/android-imageview-scaletype.html>)
- 04 Sep 2016 - Android Animation (</post/2016/09/android-animation.html>)
- 13 Aug 2016 - 国行 Moto 360 2代一周使用感受 (</post/2016/08/moto-360-2nd-review-after-one-week.html>)
- 06 Aug 2016 - Android 人脸检测 (</post/2016/08/android-face-detect.html>)
- 15 Jul 2016 - 我使用的 Xposed module (</post/2016/07/xposed-module-i-used.html>)
- 15 Jul 2016 - 我使用的 Xposed module (</post/2016/07/xposed-module-i-used.html>)
- 24 Jun 2016 - Android lib Timber (</post/2016/06/android-lib-timber.html>)
- 08 Feb 2016 - Genymotion 安装 (</post/2016/02/genymotion-installation.html>)



扫描二维码在移动设备查看

[← Previous \( 前一篇 \) \(/post/2016/08/clonezilla-clone-system.html\)](/post/2016/08/clonezilla-clone-system.html)[Archive \( 目录 \) \(/archive.html\)](/archive.html)[Next \( 后一篇 \) → \(/post/2016/08/moto-360-2nd-review-after-one-week.html\)](/post/2016/08/moto-360-2nd-review-after-one-week.html)

0条评论

Verne in GitHub

[1 登录 ▾](#)[♥ 推荐](#)[🔗 分享](#)[评分最高 ▾](#)

开始讨论...

通过以下方式登录

或注册一个 DISQUS 帐号 [?](#)

来做第一个留言的人吧！

在 VERNE IN GITHUB 上还有

### Android Animation | Verne in GitHub

1条评论 • 1年前

Martin Tsiu — [🔗](#)，感谢分享

### C++ 解析JSON - Verne in GitHub

1条评论 • 2年前



秋源 彭 — boost property\_tree json里面带中文，怎么弄。。

### 电影网站评分机制 | Verne in GitHub

6条评论 • 8个月前



Ein Verne — 嗯呢，Dropbox我也一直在用呢，堆了各种文档，已经都95%了，最近得清理清理。

### Pokemon Go 简介 | Verne in GitHub

2条评论 • 1年前



Ein Verne — 目前都是群中整理，来源与正确性尚未验证。具体来源可到telegram群询问。

[✉ 订阅](#) [🔒 在您的网站上使用 Disqus添加 Disqus添加](#) [🔒 隐私](#)

“

实现明天理想的唯一障碍是今天的疑虑。 --佚名



学习笔记 <sup>67</sup> (/categories.html#学习笔记-ref)



Android <sup>40</sup> (/tags.html#Android-ref)

AndroidDev <sup>24</sup> (/tags.html#AndroidDev-ref)



文章目录

- Android 原生人脸检测API
- Play Service 中人脸检测
- Face++ 人脸检测服务
- Detect Result
- Reference