

[首页](#) > [物联网](#) > 正文

Android Things 专题6 完整的栗子：运用TensorFlow解析图像

转载 谷歌开发者 2017-03-22 14:58 阅读 195 次

摘要：Android Things

现在深度学习很火，那我们就在Android Things中，利用摄像头抓拍图片，让 TensorFlow 去识别图像，最后用扬声器告诉我们结果。

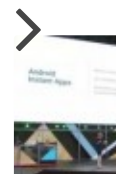
是不是很酷？说基本的功能就说了这么长一串，那垒代码得垒多久啊？

项目结构

我们就从 Android Studio 的环始境开始说起吧。

启动 Android Studio 之后，务必把 SDK Tools 的版本升级到 24 及以上。然后再把 SDK 升级到 Android 7.0 及以上。让 Android Studio 自己完成相关组件的更新，导入项目，项目的结构如下：

站长推





```
▼ app
  ► assets
  ► build
  ▼ libs
    ? Tensorflow-Android-Inference-alpha-debug.aar
  ▼ src
    ▼ main
      ▼ java
        ▼ com.example.androidthings.imageclassifier
          ▼ classifier
            ? Classifier
            ? TensorFlowImageClassifier
          ▼ env
            ? ImageUtils
            ? Logger
            ? CameraHandler
            ? ImageClassifierActivity
            ? ImagePreprocessor
        ► res
          ? AndroidManifest.xml
      ? app.iml
      ? build.gradle
```

<http://blog.csdn.net/wfing>

代码中的 imageclassifier 是用于跟 TensorFlow 做交互的，还有摄头，图片处理的相关 handler。

我们再来看看外部的引用库：

站长推



```
▶ < Android API 24 Platform > (/home/ycwang/desktop/android/  
▶ < 1.8 > (/usr/lib/jvm/java-8-openjdk-amd64)  
▶ androidthings-0.2-devpreview  
▶ driver-button-0.2  
▶ support-annotations-24.2.0  
▶ Tensorflow-Android-Inference-alpha-debug-
```

包括了 Android Things 和 TensorFlow 的相关库，当然，Android 的 API 的版本是24。gradle 的依赖和 Manifest 中的 filer 是和之前搭建开发环境的讲解一致的。

引用的 TensorFlow 的库是 aar 打包的 Tensorflow-Android-Inference-alpha-debug.aar，这就意味着，我们不需要 NDK 环境就能够编译整个项目了。

主要是留意 dependencies 这一项，包括了 TensorFlow 的库和 Android thing 的库：

站长推



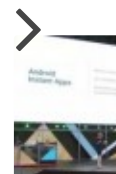


```
33 // Download model zip file into ../../assets directory
34 // unzip it to demo project's own ./assets directory
35 import de.undercouch.gradle.tasks.download.Download
36 task downloadFile(type: Download) {
37     src 'https://storage.googleapis.com/download.tensorflow.org/models/inception5h.zip'
38     dest projectDir.toString() + '../../assets/inception.zip'
39 }
40 task unzip(type: Copy) {
41     from zipTree(projectDir.toString() + '../../assets/inception.zip')
42     into file(projectDir.toString() + '/assets')
43 }
44 unzip.dependsOn downloadFile
45
46 project.afterEvaluate {
47     if (!(new File(projectDir.toString() + '/assets')).exists()) {
48         preBuild.dependsOn unzip
49     }
50 }
51
52 repositories {
53     flatDir {
54         dirs 'libs'
55     }
56 }
57
58 dependencies {
59     compile(name: 'Tensorflow-Android-Inference-alpha-debug', ext: 'aar')
60
61     provided 'com.google.android.things:androidthings:0.2-devpreview'
62     compile 'com.google.android.things.contrib:driver-button:0.2'
63 }
64
```

<http://blog.csdn.net/wfing>

再申请了摄像头相关的权限。补充一下，Android Things 是不支持动态权限的申请的。

站长推



```
16  -->
17
18  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
19    package="com.example.androidthings.imageclassifier">
20
21    <uses-permission android:name="android.permission.CAMERA" />
22    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
23
24    <uses-feature android:name="android.hardware.camera" />
25    <uses-feature android:name="android.hardware.camera.autofocus" />
26
27    <application android:allowBackup="true"
28        android:label="TensorFlowInference"
29        android:icon="@android:drawable/sym_def_app_icon"
30        android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
31
32        <activity android:name=".ImageClassifierActivity"
33            android:screenOrientation="landscape"
34            android:label="TensorFlowInference">
35            <intent-filter>
36                <action android:name="android.intent.action.MAIN" />
37                <category android:name="android.intent.category.LAUNCHER" />
38            </intent-filter>
39
40            <!-- Launch activity automatically on boot -->
41            <intent-filter>
42                <action android:name="android.intent.action.MAIN"/>
43                <category android:name="android.intent.category.IOT_LAUNCHER"/>
44                <category android:name="android.intent.category.DEFAULT"/>
45            </intent-filter>
46        </activity>
47    </application>
48
49 </manifest>
50
```

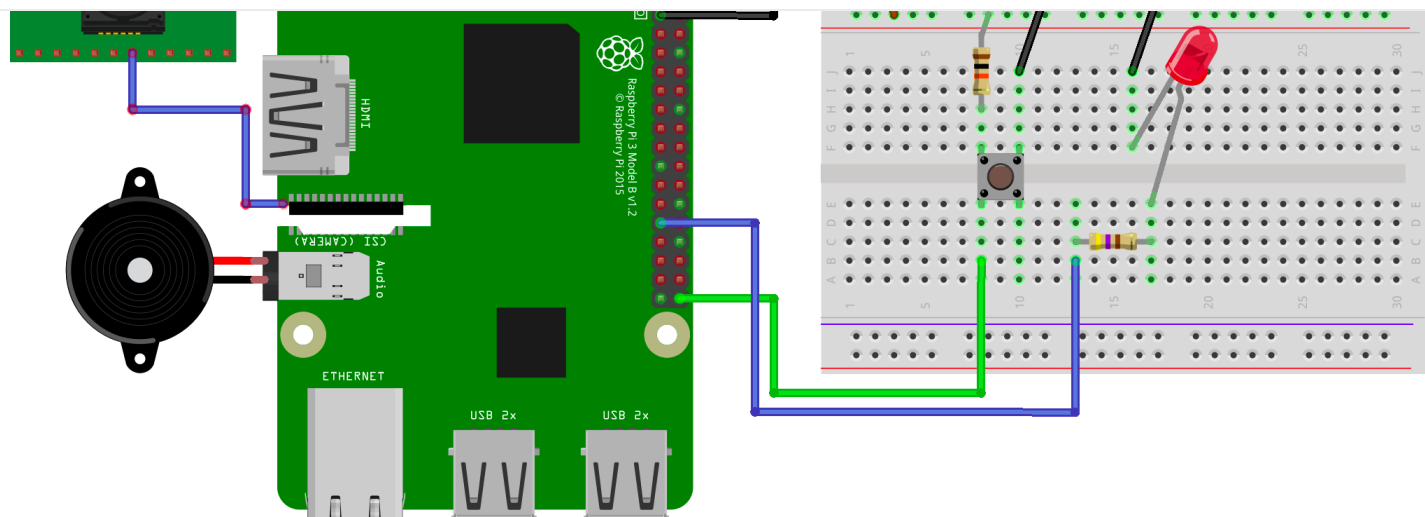
<http://blog.csdn.net/wfing>

硬件连接

接下来便是硬件如何连接了。

站长推





<http://blog.fritzing.org>

硬件清单如下：

Android Things 兼容的开发板，比如 Raspberry Pi 3

Android Things 兼容的摄像头，比如 Raspberry Pi 3 摄像头模块

元器件：

1 个按钮，见面包板

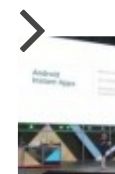
2 个电阻，这块儿必须要说明一下：由于图片是接的 5V 的电压，一般来说 GPIO 和 led 的承压能力是 3 V，有些 GPIO 是兼容 5V 的，所以中间需要串联 100 ~ 200 欧的电阻。当然，为了保险，建议用 3.3V 的电压。

1 个LED灯

1 个面包板

杜邦线若干

站长推



连完了硬件，我们这时候就要理解操作流程了。

操作流程

按照前面讲解的内容，用 Andorid Studio，连接 ADB，配置好开发板的 Wi-Fi，然后把应用加载到开发板上。

操作流程如下：

重启设备，运行程序，直到 LED 灯开始闪烁；

把镜头对准猫啊，狗啊，或者一些家具；

按下开关，开始拍摄图片；

在 Raspberry Pi 3 中，一般在 1s 之内，可以完成图片抓拍，经 Tensorflow 处理，然后再通过 TTS 放出声音。在运行的过程中 LED 灯是熄灭的；

Logcat 中会打印出最终的结果，如果是有显示设备连接的话，图片和结果都会显示出来；

如果有扬声器或者是耳机的话，会把结果语音播报出来。

由于代码的结构特别简单，注意一下几段关键的操作即可。想必图形、摄头的操作在Android 的编程中大家都会了，所以不做讲解了。

代码流程

主要是看 LED 的初始化操作：

站长推




```
17
18 import ...
51
52 public class ImageClassifierActivity extends Activity implements ImageReader.OnImageAvailableListener {
53     private static final String TAG = "ImageClassifierActivity";
54     private static final int PERMISSIONS_REQUEST = 1;
55
56     private static final String BUTTON_PIN = "BCM21";
57     private static final String LED_PIN = "BCM6";
58
59     private ImagePreprocessor mImagePreprocessor;
60     private TextToSpeech mTtsEngine;
61     private CameraHandler mCameraHandler;
62     private TensorFlowImageClassifier mTensorFlowClassifier;
63
64     private HandlerThread mBackgroundThread;
65     private Handler mBackgroundHandler;
66
67     private ImageView mImage;
68     private TextView[] mResultViews;
69
```

<http://blog.csdn.net/wfing>

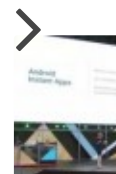
有必要说一下，ImageClassifierActivity.Java 是应用唯一的 Activity 的入口。在 Manifest 中已经有定义，它初始化了 LED, Camera, TensorFlow 等组件。其中，我们用到的 Button 是 BCM32 这个管脚，用到的 LED 是 BCM6 管脚，相关的初始化在这个 Activity 中已经完成。

```
95 private void init() {
96     try {
97         mButtonDriver = new ButtonInputDriver(BUTTON_PIN, Button.LogicState.PRESSED_WHEN_LOW,
98             KeyEvent.KEYCODE_ENTER);
99         mButtonDriver.register();
100         PeripheralManagerService service = new PeripheralManagerService();
101         mReadyLED = service.openGpio(LED_PIN);
102         mReadyLED.setDirection(Gpio.DIRECTION_OUT_INITIALLY_LOW);
103     } catch (IOException e) {
104         Log.w(TAG, "Could not open GPIO", e);
105     }
106
107     mBackgroundThread = new HandlerThread("BackgroundThread");
108     mBackgroundThread.start();
109     mBackgroundHandler = new Handler(mBackgroundThread.getLooper());
110     mBackgroundHandler.post(mInitializeOnBackground);
111 }
```

<http://blog.csdn.net/wfing>

这部分代码是捕捉按键按下的代码。当按下按键时，摄头开始捕捉数据。

站长推



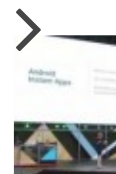

```

184 public void onImageAvailable(ImageHeader reader) {
185     final Bitmap bitmap;
186     try (Image image = reader.acquireNextImage()) {
187         bitmap = mImagePreprocessor.preprocessImage(image);
188     }
189
190     runOnUiThread(() -> { mImage.setImageBitmap(bitmap); });
191
192     final List<Classifier.Recognition> results = mTensorFlowClassifier.recognizeImage(bitmap);
193
194     if (mTtsEngine != null) {
195         // speak out loud the result of the image recognition
196         if (Math.random() < 0.3) {
197             mTtsEngine.setPitch(0.2f);
198             mTtsEngine.speak("I see dead people...", TextToSpeech.QUEUE_ADD, null, "ID");
199             mTtsEngine.setPitch(1);
200             mTtsEngine.speak("just kidding...", TextToSpeech.QUEUE_ADD, null, "ID");
201         } else {
202             mTtsEngine.setPitch(1f);
203             mTtsEngine.setVoice(mTtsEngine.getDefaultVoice());
204         }
205         String message;
206         if (results.isEmpty()) {
207             message = "I don't understand what I see. Am I using drugs?";
208         } else if (results.size() == 1 || results.get(0).getConfidence() > 0.4f) {
209             message = String.format(Locale.getDefault(), "I see a %s",
210                                     results.get(0).getTitle());
211         } else {
212             message = String.format(Locale.getDefault(),
213                                     "This is a %s or maybe a %s",
214                                     results.get(0).getTitle(), results.get(1).getTitle());
215         }
216         mTtsEngine.speak(message, TextToSpeech.QUEUE_ADD, null, "ID");
217     } else {
218         // if theres no TTS, we don't need to wait until the utterance is spoken, so we set
219         // to ready right away.
220         setReady(true);
221     }
222 }

```

把摄像头拍摄的数据转成 Bitmap 文件之后，我们会调用 TensorFlow 来处理图像。

站长推

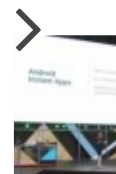


```
184 public void onImageAvailable(ImageReader reader) {
185     final Bitmap bitmap;
186     try (Image image = reader.acquireNextImage()) {
187         bitmap = mImagePreprocessor.preprocessImage(image);
188     }
189
190     runOnUiThread(() -> { mImage.setImageBitmap(bitmap); });
191
192     final List<Classifier.Recognition> results = mTensorFlowClassifier.recognizeImage(bitmap);
193
194     if (mTtsEngine != null) {
195         // speak out loud the result of the image recognition
196         if (Math.random() < 0.3) {
197             mTtsEngine.setPitch(0.2f);
198             mTtsEngine.speak("I see dead people...", TextToSpeech.QUEUE_ADD, null, "ID");
199             mTtsEngine.setPitch(1);
200             mTtsEngine.speak("just kidding...", TextToSpeech.QUEUE_ADD, null, "ID");
201         } else {
202             mTtsEngine.setPitch(1f);
203             mTtsEngine.setVoice(mTtsEngine.getDefaultVoice());
204
205             String message;
206             if (results.isEmpty()) {
207                 message = "I don't understand what I see. Am I using drugs?";
208             } else if (results.size() == 1 || results.get(0).getConfidence() > 0.4f) {
209                 message = String.format(Locale.getDefault(), "I see a %s",
210                     results.get(0).getTitle());
211             } else {
212                 message = String.format(Locale.getDefault(),
213                     "This is a %s or maybe a %s",
214                     results.get(0).getTitle(), results.get(1).getTitle());
215             }
216             mTtsEngine.speak(message, TextToSpeech.QUEUE_ADD, null, "ID");
217         } else {
218             // if theres no TTS, we don't need to wait until the utterance is spoken, so we set
219             // to ready right away.
220             setReady(true);
221         }
222     }
223 }
```

<http://blog.csdn.net/wfing>

这个函数调用了 TensorFlow 进行处理，最后把结果输出到 logcat 中。如果代码中调用了 tts 引擎，那么则把结果转成语音读出来。看来，最重要的就是 TensorFlowClassifier 类的 recognizeImage() 这个接口了。我们继续往下看。

站长推



```
151 // Preprocess the image data from 0-255 int to normalized float based
152 // on the provided parameters.
153 for (int i = 0; i < intValues.length; ++i) {
154     final int val = intValues[i];
155     floatValues[i * 3] = (((val >> 16) & 0xFF) - imageMean) / imageStd;
156     floatValues[i * 3 + 1] = (((val >> 8) & 0xFF) - imageMean) / imageStd;
157     floatValues[i * 3 + 2] = ((val & 0xFF) - imageMean) / imageStd;
158 }
159
160 // Copy the input data into TensorFlow.
161 inferenceInterface.fillNodeFloat(
162     inputName, new int[]{1, inputSize, inputSize, 3}, floatValues);
163
164 // Run the inference call.
165 inferenceInterface.runInference(outputNames);
166
167 // Copy the output Tensor back into the output array.
168 inferenceInterface.readNodeFloat(outputName, outputs);
169
170 // Find the best classifications.
171 PriorityQueue<Recognition> pq = new PriorityQueue<~>(3,
172     (Comparator) (lhs, rhs) -> {
173         // Intentionally reversed to put high confidence at the head of the queue.
174         return Float.compare(rhs.getConfidence(), lhs.getConfidence());
175     });
176
177 for (int i = 0; i < outputs.length; ++i) {
178     if (outputs[i] > THRESHOLD) {
179         pq.add(new Recognition("'" + i, labels.get(i), outputs[i], null));
180     }
181 }
182
183 ArrayList<Recognition> recognitions = new ArrayList<~>();
184 int recognitionsSize = Math.min(pq.size(), MAX_RESULTS);
185 for (int i = 0; i < recognitionsSize; ++i) {
186     recognitions.add(pq.poll());
187 }
188
189 return recognitions;
```

<http://blog.csdn.net/wfing>

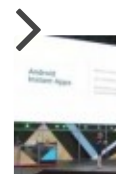
这是最后的一步，调用 TensorFlow 进行图像识别：

把 RGB 图像转成 TensorFlow 能够识别的数据；

把数据拷到 TensorFlow 中；

识别出图像，给出结果。

站长推



www.tensorflow.org/tutorials/image_recognition

有一点需要提示，TensorFlow 的图像识别分类可以用网络提交到服务器识别，也可以离线的数据识别。可以先把 200M 左右的识别数据放在本地，然后提交后识别。现在大概能分出 1000 个类别的图像，哪 1000 个类别呢？项目代码中已经包含了哦。

是不是运用 TensorFlow 来处理物联网的数据会特别简单，不光是 TensorFlow, Firebase 也可以用到 Android Things 中来。这功能，强大的没话说了！

今天提到的这个项目，来源于 Google 在 GitHub 上维护的项目，项目的地址是 github.com/androidthings/sample-tensorflow-imageclassifier

当然，GitHub 上还有很多 Android Things 的代码可以参考。

是不是迫不及待的自己写一个应用呢？实际上，这个项目稍加改动便能有新的玩法。例如加上一个红外感应器，一旦有生物在附近就马上拍图片，并且识别。

大开你的脑洞吧

本文地址：<http://www.dandroid.cn/?p=2066>

关注我们：请关注一下我们的微信公众号：扫描二维码，头条号：大安卓源码

温馨提示：文章内容系作者个人观点，不代表大安卓源码网对观点赞同或支持。

版权声明：本文为转载文章，来源于 谷歌开发者，版权归原作者所有，欢迎分享本文，转载请保留出处！

👍 3 人喜欢

🎁 赞赏

分享：

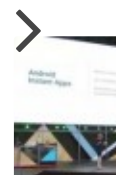


PREVIOUS:

NEXT:



站长推



老薛主机
LAOXUEHOST.COM

- ◆ PHP版本随意换，兼容更多程序
- ◆ 功能强大的cPanel中文面板
- ◆ 企业QQ技术支持，售后有保障
- ◆ 每日异地备份，网站数据更安全
- ◆ 精选香港、日本亚洲优化线路

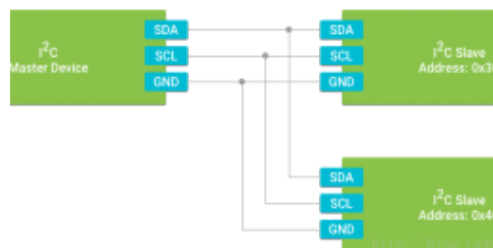
优惠码：
80off

稳定 快速 免备案

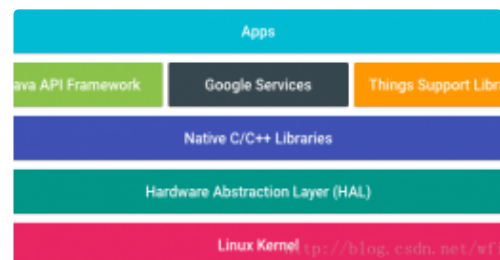
相关文章

关键词：

roidth



Android Things Developer Preview 3发布 Android Thing专题5 I2C



Android Things 专题4 SDK框架



Android Things专题3 开发环境搭建



Android Things专题2 硬件介绍

roidth

Android Things专题 1.前世今生

站长推



昵称（必填）

邮箱（必填）

网址

提交评论

☒ 有人回复时邮件通知我

表情

站点相关

- AndroidDevTools
- github
- android developer
- google开发者平台
- 泡在网上的日子
- Android Studio
- 优质美国空间-老薛...

Copyright © 大安卓源码网 | Theme by Nana | Powered by WordPress |

欢迎您关注我们



我的QQ空间

我的新浪微博

站长推

