Android 系统下 OpenCV 的人脸检测模块的设计

公衍宇[1]，郭 琦[1]，于 超[2]
（1. 河北工业大学 信息工程学院，天津 300380；2. 河北工业大学 机械工程学院，天津 300380）

摘要：针对解决 OpenCV 人脸检测模块在 Android 平台编译和移植的问题，提出一种利用 JNI 技术（Java Native Interface）调用 OpenCV 以及采用 Android NDK（Native Development Kit）生成共享库的目标检测方法。文中从分析利用 Android NDK 编译 Android 平台所需要的 OpenCV 静态库的问题入手，详细阐述了利用 JNI 调用 OpenCV 相关函数的具体步骤。经过多次试验，证明该人脸检测模块的平均检测时间为 1 280 ms，具有较高的检测速度和检测精度。

关键词：Android；OpenCV；人脸检测；JNI；NDK
中图分类号：TP311 **文献标识码**：A **文章编号**：1674-6236(2012)20-0052-03

Design of face detection module based on OpenCV in Android system

GONG Yan-yu[1], GUO Qi[1], YU Chao[2]
(1. *School of Information Engineering, Hebei University of Technology, Tianjin* 300380, *China*;
2. *School of Mechanic Engineering, Hebei University of Technology, Tianjin* 300380, *China*)

Abstract: Aiming at the problem of the OpenCV face detection module compiling and transplant in the Android platform, a JNI technology (Java Native-Interface) call OpenCV and Android NDK (Native Development Kit) to generate a shared library object detection method is proposed. Starting from analysis of compile the OpenCV static libraries needed by Android platform use Android NDK, Implementation steps of using JNI to call the relative functions of OpenCV are explained in detail. After numerous experiments to prove that the face detection module, the average detection time 1 280 ms, a high detection speed and detection accuracy.

Key words: Android; OpenCV; human-face detection; JNI; NDK

Android 是 Google 开发的基于 Linux 平台的开源手机操作系统，本意为“机器人”。它包括操作系统、用户界面和应用程序——移动电话工作所需的全部软件，而且不存在任何以往阻碍移动产业创新的专有权障碍。OpenCV 于 1999 年由 Intel 建立，现在由 Willow Garage 提供支持。OpenCV 是一个基于 BSD 许可证授权（开源）发行的跨平台计算机视觉库，可以运行在 Linux、Windows、Mac OS 和 Android 操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++类构成，同时提供了 Python、Ruby、MATLAB 等语言的接口，实现了图像处理和计算机视觉的接口，实现了图像处理和计算机视觉方面的很多通用算法[1-2]。Android 应用程序是通过 Android SDK（Software Develo-pment Kit）利用 Java 编程语言进行开发的，此虚拟机支持 JNI，同时伴随着 Android NDK 的发布，使开发者利用第三方 C/C++库协助编写 Android 程序成为可能，如 OpenCV 库便可方便地应用在 Android 系统中[3]。

1 OpenCV 移植到 Android

1.1 软件开发环境的搭建

OpenCV 从 2.2 版本以后支持 Android，本设计的编译环境 PC 端为 ubuntu 10.10，所用 OpenCV 版本为 2.2 1。OpenCV 只提供 C/C++、Python 接口，这就涉及到 Java 与 C/C++ 混合编程、Java 程序与 C/C++ 程序互相调用的问题，而 Android NDK 很好地解决了这一问题，它提供了一系列的工具，帮助开发者快速开发 C（或 C++）的动态库，并能自动将 so 和 java 应用一起打包成 apk。同时还集成了交叉编译器，并提供了相应的 mk 文件隔离 CPU、平台、等差异，开发人员只需要简单修改 mk 文件（指出“哪些文件需要编译”、“编译特性要求”等），就可以创建出 so。

官方提供的 Android NDK R4 版本对于有些 OpenCV 中使用的一些 C++概念不支持 Cryta x 为我们提供了一种支持 OpenCV 的 Android NDK 修订版。本设计 Android NDK 所使用的版本为 android-ndk-r4-linux-cry-tax-4 在开始编译安装 OpenCV 库之前，需要在 PC 机 Ubuntu 操作系统下安装如：cmake、buid-essential 等编译 OpenCV 库需要用到的工具。在此环境下编译并安装 OpenCV 库。

1.2 利用 NDK 编译生成 OpenCV 静态库

OpenCV 现已官方支持在 Android 环境下的开发，OpenCV2.2 源文件包中包含一 android 文件夹，在此文件夹中包括一些官方提供的编译文件，利用 NDK 并借助这些编译

收稿日期：2012-07-04 **稿件编号**：201207027
作者简介：公衍宇（1986—），男，山东泰安人，硕士研究生。研究方向：通信与测控系统。

文件对 OpenCV 源码进行编译，就可生成适用于在 Android 环境下运行的静态库。在编译之前需要解压缩 android-ndk-r4-linux-x86-crytax-4.tar.bz2，完成修订版本 NDK 的安装。进入 OpenCV2.2 内的 android 文件夹下，执行 “mkdir build”指令，完成创建 build 文件夹，然后执行 “cd build” 指令进入 build 文件夹，随后执行“cmake ..”指令生成 makefile 编译文件，最后执行“make”指令对其进行编译，编译完成后将生成 OpenCV 静态库，用于在 Android 环境下基于 OpenCV 的 Android 应用软件开发。

2　设计整体框架

在 Android 系统下基于 OpenCV 的人脸检测实现需通过 Android NDK 工具集将利用 JNI 编写的本地代码组件嵌入到 Android 应用程序中，所以整个实现过程分为两个步骤首先，通过 JNI 与 OpenCV 接口编写本地 C/C++代码，并利用 Android NDK 对其进行编译生成 Java 可调用的共享库，接下来利用 Android 应用程序框架编写 Java 端代码，最后通过 Android SDK 生成 Android 应用程序，整体设计框架如图 1 所示。

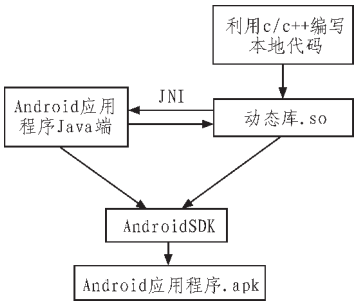

图 1　整体设计框架

Fig. 1　Overall design framework

打开 eclipse，新建一个 workspace，在 Window->Preferences 中设置好 Android SDK 的路径。然后新建一个 Android 项目，Build Target 选择 Android2.2，命名为"facedetect"，活动名改为“facedetect”，Package name 中填写 com.testopencv.facedetect，最后点击 finish。

2.1　利用 JNI 编写 C++OpenCV 代码

首先在工程目录下新建 jni 文件夹，新建 C++源代码文件并引入此头文件，编写函数实现代码。

分类器文件与图片文件的路径分别在 Java 端利用 String 静态变量进行存储，在本地代码端需要得到此 String 变量的 UTF-8 形式用来加载分类器和图片文件，这就需要通过 JNI 来实现。关键代码如下：

```
jclass cls = env->GetObjectClass(obj);//得到代表 Java 类的 jclass 对象。
jfieldID fid1 = enc->GetStaticFieldID(cls,"cstr","Ljava/lang/String");//得到 Java 端保存分类器路径 String 静态变量的 jfieldID。
jsting jst1 = (jstring)env->GetStaticObjectField(cls,fid1);//得到 Java 端的 String 实例 jstring。
const char* cstring = env->GetStringUTFChars(jst1,NULL);//得到利用 UTF-8 编码的 C/C++字符串，此字符串为存储分类器的路径。
env->ReleaseStringUTFChars(jst1,cstring);//释放拷贝的内存。
```

以上代码可得到分类器的绝对路径，图片文件的存储绝对路径也可以此方式得到[4]。

2.2　人脸识别函数的实现

人脸检测是指在输入图像中将人脸从庞杂的背景中提取出来，并返回人脸在图像中的位置、大小等信息。首先将彩色图像转换为灰度图像，然后对图像进行直方图均衡化，最后利用分类器对图像中某一区域是否为人脸进行检测。OpenCV 将分类器信息利用 xml 文件进行存储，同时 OpenCV 自带分类器训练程序：haartrfimng.exe，可以使用它来方便的进行 haar-like 特征分类器的训练。将分类器 xml 文件放在 Java 工程目录下的 asset 文件夹内，当安装应用时，利用流操作将此文件写入 sd 卡上的指定路径下，同时将此途径存储在一静态变量 String，人脸检测函数实现关键代码如下：

```
Mat img = CV.imread(istring);// 加载图片得到 RGB 彩色模型 Mat 数据。
CascadeClassifier cascade;
cascade.load (cstring);//得到分类器 CascadeClassifier 数据。
Mat gray,smallImg(cvRound(img.rows/scale)),CV_8UC1);
cvtColor(smallImg, gray,CV_RGB2GRAY);//对图像数据进行灰度化。
equalizeHist(smallImg, smallImg);// 进行直方图均衡化。
cascade.detectMultiScale(smallImg,faces,1.1,2.0|CV_HAAR_SCALE_IMAGE,Size(30,30));// 进行人脸检测。
```

检测到的人脸位置信息将会存入数据类型为 vector<RECT>的 faces 中，最后通过迭代将检测到人脸的矩形信息分别换算成空心圆信息存入一个 int 数组，此空心圆信息包括圆心的 x 坐标、y 坐标以及半径。

2.3　设置 Java 端人脸位置信息

在 C/C++端得到人脸位置信息后，需要将此信息传入 Java 端进行 UI 更新，这依然需要通过 JNI 来完成。在 IntentService 类中新建一 int 数组用来存放检测到的人脸位置信息，当本地人脸检测操作执行完成后对其进更新与读取，关键代码如下：

```
jfieldID fid3 = env->GetFieldID(cls,"faceArray","[I");
jintArray tmp1=(jintArray)env->GetObjectField(obj,fid3);// 得到 Java 端的 int 数组实例。
jintArray tmp2 = env->NewIntArray (faces.capacity());// 新建一个长度与 C/C++端 int 数组长度相同的 Java 端 int 数组。
jintArray env->SetIntArrayRegion (tmp2,0,faces.capacity()),facearrays);// 通过 C/C++ 端存取人脸位置信息的 int 数
```

组设置。

jintArray env->SetObjectField(obj,fid3,tmp2);//通过 tmp2 设置 Java 端 int 数组的内容,此时 Java 端就可根据此内容完成 UI 更新操作[5-7]。

2.4 脚本文件编写

有两个脚本文件需要编写,分别是 Android.mk 和 Application.mk 文件。Android.mk 文件描述了需要 NDK 进行编译的源文件,以及所要形成的组件,Application.mk 文件描述一些关于本应用的辅助编译信息,这个编译文件是可选的。

在 Java 工程目录下的 jni 文件夹下新建 Android.mk 文件。打开此文件,首先定 LOCAL_PATH:= $(call my-dir)",用来定位源代码文件所在目录;其次 "include $(CLEAR_VARS)",用来清除许多 LOCAL_XXX 变量,因为所有编译控制文件都是在一个 GNU Make 执行上下文中进行解析,而此时所有变量都是全局可见的,所以需要这一操作;由于利用第三方库 OpenCV 协助开发,需要指定此库通过 NDK 编译好的静态版本的头文件、静态库文件的搜索路径与连接标记,所以编写如下如下 6 条语句:

```
include ../includeOpenCV.mk
ifeq ("$(wildcard $(OPENCV_MK_PATH)","")
include $(TOOLCHAIN_PREBUILT_ROOT)/usr/share/OpenCV/OpenCV.mk
else
include $(OPENCV_MK_PATH)
endif
```

再者,需要定义源文件后缀名、源文件、所产生的组件,编写如下两条语句:

```
LOCAL_MODULE:=facedetect
LOCAL_SRC_FILES:=ImgFun.cpp
```

最后需说明所要产生的是静态库还是动态库,由于动态库为所需,故编写如下代码:

```
include $(BUILD_SHARED_LIBRARY)
```

在 Java 工程目录下的 jni 文件夹下新建 Application.mk 文件,由于需要在 ARM-V7 构架的 CPU 上运行,所以通过定义"APP_ABI:= armeabi-v7a"来指定。打开 Cygwin,进入本 Android 工程目录下的 jni 文件中,执行"$NDK/ndk-build"命令,执行完毕后将生成共享文件 libface detect.so。

3 结 论

人脸检测的实现充分说明了在 Android 系统下利用 OpenCV 进行计算机视觉研究与开发的可行性与实用性,虽然由于 Android NDK 工具还处于初级发展阶段,不够成熟,以及硬件配置的差距,致使最后人脸检测操作的运行速度相对较慢,但随着 NDK 的发展,利用其进行编译的本地代码在 Android 系统中的运行将更加高效,同时越来越强大的硬件配置也将给予有力支持。Android 系统如今发展迅速,许多第三方库现已提供 Android 通用编程接口,这为 Android 应用软件开发人员提供极大的便利,OpenCV 也会在下一版本提供此接口,但对于计算机视觉算法研究人员以及对软件深度开发人员来说,利用 JNI 编写本地代码,NDK 编译本地代码的方式还是必须的,并且这一方式也更加灵活。

参考文献:

[1] Gray Bradski,Adrian Kaebler. Learning OpenCV:Computer Vision with the OpenCV Library[M]. USA:O Reilly media,2008.

[2] Willow Garage. OpenCV wiki [EB/OL]. [2012-05] (2012-07).http://opencv.willowgarage.com/wiki/Welcome.

[3] Google. Android Guide[EB/OL]. [2012-02] (2012-07).http://developer.android.com/guide/index.html.

[4] 张莹,李勇平,敖新宇. 基于OpenCV的通用人脸检测模块设计[J]. 计算机工程与科学,2011(1):97-101.
ZHANG Ying,LI Yong-ping,AO Xin-yu. Common face detection module design based on OpenCV[J]. Computer Engineering and Science,2011(1):97-101.

[5] 韩露,李祖枢,陈东义.一种Java与OpenCV结合实现的目标检测模块[J].计算机应用,2008(3):773-775.
HAN Lu,LI Zu-shu,CHEN Dong-yi. A Java and OpenCV combination of the goal detection module[J]. Computer Applications,2008(3):773-775.

[6] 任俊伟,林东岱. JNI技术实现跨平台开发的研究[J]. 计算机应用研究,2005(7):180-184.
REN Jun-wei,LIN Dong-dai. JNI technology to achieve cross-platform development[J]. Computer Applications Research,2005(7):180-184.

[7] 陈勇飞, 刘新明. 基于肤色和类Harr特征的人脸图像的人脸检测[J]. 计算机工程与应用,2008(33):174-180.
CHEN Yong-fei,LIU Xin-ming. Face images based on skin color and class Harr features face detection[J]. Computer Engineering and Applications,2008(33):174-180.

[8] 张靠社,张增强,杨宝杰. 基于Hamilton能量函数含TCSC的电力系统非线性控制[J]. 陕西电力,2009(3):23-26.
ZHANG Kao-she,ZHANG Zeng-qiang,YANG Bao-jie. Nonlinear control for power system with TCSC based on hamiltonian energy function[J]. Shaanxi Electric Power,2009(3):23-26.

[9] 孟洪波,王亚军,方涛. 气相色谱法检测无水肼中水含量的不确定度评定[J]. 火箭推进,2011(3):68-72.
MENG Hong-bo,WANG Ya-jun,FANG Tao. Uncertainty evaluation of detecting the water content in anhydrous hydrazine with gas chromatography[J]. Journal of Rocket Propulsion,2011(3):68-72.