Vol.14 No.12
Dec. 2012

基于OpenCV在Android平台下实现人脸识别

聂鹏鹏[1]，王二伟[2]，刘敏丰[1]，张昊堃[1]
（西安电子科技大学，西安 710071）

摘 要：本文提出了使用OpenCV（Open Source Computer Vision Library）在Android平台下实现人脸识别的方法。首先，本文介绍了OpenCV 2.4的人脸识别算法类FaceRecognizer的使用以及在Windows平台下的实现，接着详细的阐述了利用Android下的JNI（Java Native Interface），并结合Android NDK（Native Development Kit）调用和编译OpenCV下的相关函数，生成共享函数库。经过实验的在Android下人脸识别效果良好。

关键词：人脸识别，人脸归一化；Android；OpenCV；NDK；JNI

Face Recognition Based on OpenCV in Android System

NIE Peng-peng[1]，WANG Er-Wei[2]，LIU Min-feng[1]，ZHANG Hao-kun[1]
（Xidian University, Xi’an, 710071）

Abstract: This paper proposes a way of face recognition based on OpenCV in Android System. Firstly, this paper introduces the use of face recognition struct FaceRecognizer and implements this struct in Windows System. Then the invoking and compiling of OpenCV functions with JNI (Java Native Interface) and Android NDK (Native Development Kit) is introduced in detail and shared library is got. Finally, face recognition in Android works well in the experiment.

Keywords: face recognition, face normalization; Android; OpenCV; NDK; JNI

人脸识别是指利用人脸的视觉特征信息来进行身份鉴别的计算机技术。该技术广泛应用于娱乐、信息安全、法律监控等领域，这些应用就包括虚拟现实、人机交互、门禁、CCTV控制等[1]。目前人脸识别技术在计算机上得到了很好的应用。但随着移动设备的普遍使用，使得移动设备的安全也得到了高度的关注。

收稿日期：2012-10-21

Android系统是一个以Linux为基础的半开源系统，由Google和Open Handset Alliance（OHA，开放手持设备联盟）持续领导与开发中[2]。

OpenCV是基于BSD许可证授权发布的实时计算机视觉的函数库，它是有C++、C和Python，并且很快就有了Java接口，能够运行在Windows、Linux、Android和Mac平台，函数主要包括一般的图像处理，图像分割，机器学习、特征识别、跟踪等500多个函数[3]。在OpenCV 2.4版本中，推出

了FaceRecognizer类，实现了三种人脸识别方法Eigenfaces、Fisherfaces和Local Binary Patterns Histograms。

1 在Windows平台下实现人脸识别功能

在Android平台开发前，需要先在Windows下将所需要算法的C/C++函数先进行开发并测试好。这是因为结合Visual Studio开发工具，可以能够很好的调试C/C++代码。需要实现的功能模块包括人脸归一化处理和识别功能。

1.1 人脸归一化处理

在新来了一张图片后，首先需要做的是检测出是否有人脸，并做归一化处理。归一化处理的原因是为了尽可能减小光照和姿势对人脸识别的影响。新来一张图片后，需要的处理流程如图1所示。

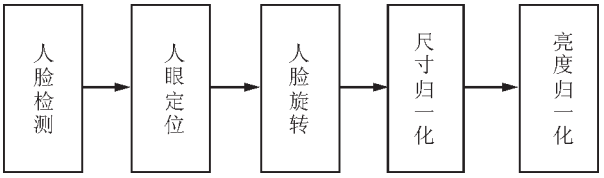

图1 人脸归一化流程

人脸检测和人眼定位是使用OpenCV自带的库函数进行实现的，关于具体如何配置Visual Studio环境下的OpenCV的开发，不再赘述。在OpenCV的中文官网“ http://www.opencv.org.cn/index.php/人脸检测”有实现的例程，其中用到的关键函数为：

CVAPI (CvSeq*) cvHaarDetectObjects (const CvArr* image, CvHaarClassifierCascade* cascade, CvMemStorage* storage, double scale_factor CV_DEFAULT (1.1) , int min_neighbors CV_DEFAULT (3) , int flags CV_DEFAULT (0) ,CvSize min_size CV_DEFAULT (cvSize (0,0)) , CvSize max_size CV_DEFAULT (cvSize (0,0))) ;

人脸旋转的目的是把有旋转角度的人脸变换成正立的人脸。实现手段是计算左右眼的中心连成线与水平线所成的角度，然后将图片中所有的像素点的坐标进行旋转。

人脸尺寸归一化是需要将人脸从原始图片中截取出来，制作成统一大小的图片。假定所需要的人脸图片的宽度是w，则两个人眼中心的距离为w/2。因此得到一张人脸图片后，计算其人眼中心的距离与要求的距离的比值，即为整个图片的缩放比例系数如下图2所示。

图2 人脸归一化处理效果

2.2 人脸识别实现

在利用OpenCV进行人脸识别算法前，需要从网上下载libfacerec库，该库已经给出了实例。我们需要做的是理解三种算法Eigenfaces，Fisherfaces和LBPH (Local Binary patterns Histograms) ，以及如何调用底层库来实现。这里以Eigenfaces为例，对其实现的过程做出解释。

首先需要准备好需要测试的人脸图片库。函数库推荐的是AT&T人脸数据库，从网络下载下来后，还需制作一个文件路径的文本文档，命名为csv.txt，每一行的格式如下为/path/to/image.ext;label，其中/path/to/image.ext是图片的绝对路径或相对路径，label必须是整数，同一个人的label是一样的如下图3所示。

现解释其中重要的函数：

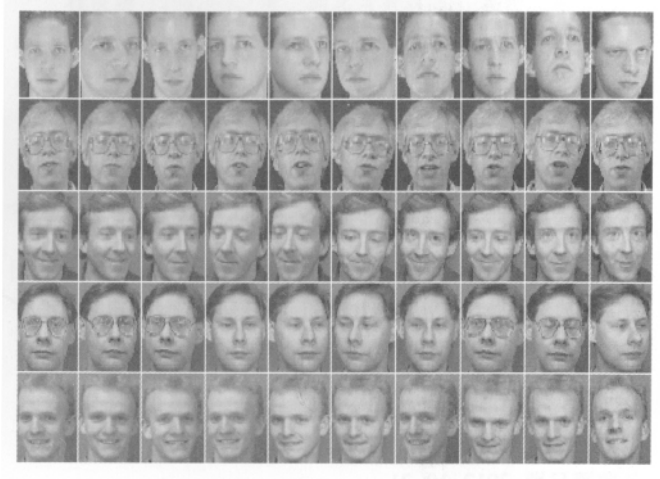

图3 AT&T人脸数据库一部分

Ptr<FaceRecognizer> createEigenFaceRecognizer (int num_components, double threshold) num_components是特征值的个数，也就是说在做PCA分析时，选取的主成分向量的个数，也可以不设定，直接使用默认值。Threshold是人脸识别的阈值，当设定为0值，人脸预测时返回的标号为-1时，则表明人脸库没有相应的人脸。

训练的图片，也就是你想训练的人脸，数据格式为vector<Mat>labels:对应图片的标号，数据格式为：vector<int>;int Eigenfaces::predictsrc:需要进行识别的图片。返回值是对应的标号。

3 搭建Android下的JNI开发环境

3.1 环境设置

Android在的开发语言是JAVA，所使用的集成开发环境是Eclipse。在标准的Android 开发环境下开发OpenCV，还需要准备Android NDK，Cygwin和OpenCV Android。

首先从网上下载准备好三个库文件，下载网址分别为：http://developer.android.com/tools/sdk/ndk/index.html，http://www.cygwin.cn/site/install/和http://opencv.org/downloads.html下载各自的包。

然后安装Cygwin，它主要是提供了Linux的编译的命令，提供了Windows平台下的Linux环境。在配置好系统环境变量，需要对/cygwin/home/username/bash_profil文件进行修改，在文件的末尾加上“ NDK=" /cygdrive/<path to Android NDK>"”和“ export NDK”。第一句是配置Android NDK的路径。假如下载并解压缩的Android NDK路径为“ D:/android-ndk-r8b”，则第一句应该为：“ NDK=" /cygdrive/d/ android-ndk-r8b”。

接着需要配置OpenCV的开发环境，在Eclipse中导入OpenCV Android-2.4文件里所有的Android工程，导入后会发现有几个OpenCV sample和OpenCV Tutorial几个例程，最重要的是还有一个OpenCV Library 2.4.2的工程。这时候，开始建立一个空的标准Android工程，并在工程目录下新建一个文件夹，取名为jni，所有的C/C++文件都放在该目录。在这个工程中，进行如下操作Project->Properties->Android->Library->Add，这时候就会看需要导入的OpenCV Library 2.4.2。然后是配置jni文件下的C/C++文件自动编译的问题，Project->Properties->Builders->New->Program，这是进入了编译器的配置选项卡。选择Main选项页，在Location框中输入cygwin\bin\bash.exe的绝对路径，在Working Directory框中输入cygwin\bin\的绝对路径，在Arguments输入“ --login -c " cd /cygdrive/<path to project> && $NDK/ndk-build"”，其中<path to project>是要输入该工程的路径，至于后半部分是调用ndk-build命令；选择Build options选项页下将Run the builder下的都选中，这是表示在制定的情况下调用编译命令，在specify working set of relevant resources下指定jni文件的路径即可。这样就配置完了jni下文件的编译器。编译文件的规则是根据jni目录下的Android.mk和Application.mk文件，这将在后面介绍编写。

4 实现程序过程

在建立好了编译环境后，需要进行源代码开发，主要有两部分需要开发，一部分是Android程序的JAVA平台的开发，主要涉及到UI界面和程序的逻辑流程；另一部分是Jni接口的开发，也就是用来链接JAVA与C/C++的部分如下图4所示。

4.1 建立UI界面

在界面上需要两个布局，一个是用于人脸识别的界面，另一个是用于人脸注册的界面。在res/

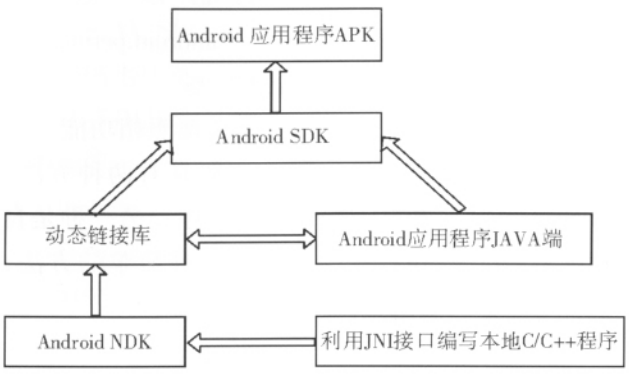

图4 Android应用程序构成图 [5]

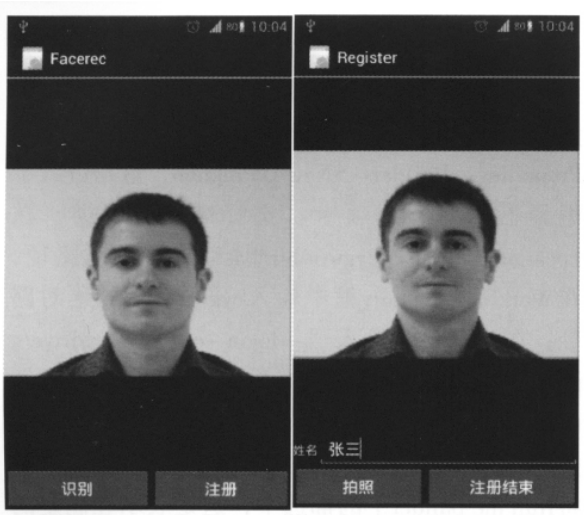

图5 两个界面组件

layout文件夹，新建两个xml布局文件，分别命名为main.xml和register.xml。在main.xml中主要两个Button，分别是“识别”和“注册”按键，以及一个imageview组件。在register.xml中有两个Button，分别是“拍照”和“注册结束”按键，一个Edittext输入框，用于接收姓名，以及一个imageview组件。

4.2 JAVA平台程序开发

为上述两个UI界面编写两个Activity，分别命名为FacerecActivity.java和register.java，对应着FacerecActivity和register两个Activity。为此需要在AndroidManifest.xml里组成这两个Activity，同时需要加入允许程序读写SD卡，因此加入权限语句：

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

在两个Activity里面都需要实现照相功能，获取图片。实现照相机功能，一般是有两种方法：一种是使用内置的摄像头应用程序；另一种是自定义照相Activity。在这里使用一种简单的方法，既使用内置的摄像头程序。步骤如下：

> 构建摄像头Intent：

ntent cameraIntent = new Intent (MediaStore.ACTION_IMAGE_CAPTURE) ;

> 启动摄像头的Intent：startActivityForResult (cameraIntent,0);

> 接受Intent结果：在应用程序中设置onActivityResult () 方法，用于接受摄像头intent返回的数据。

首先实现FacerecActivity.java文件，它作为主界面，主要是响应上述的main.xml中各个组件。分别实现对“识别”和“注册”两个按钮的监听，在“识别”按键的响应为调用JNI的识别算法。该人脸识别函数的原型应该接受的参数是一张照片，返回值应该是该照片中人脸对应的姓名或者找不到该人。故其原型为：

public static native String Facerec (int [] buf, int w, int h) ;

“public static native”即声明为本地的静态函数，告诉编译器要调用JNI层的函数；形参buf为图像的整型数据，w和h分别对应为图像的宽度和高度；返回值对应的一串字符。至于如何实现，在JNI层具体介绍。在“注册”按键的相应位调用class register，进入注册界面，具体实现方法为：

Intent.setClass (FacerecActivity.this, register.class) ;

FacerecActivity.this.startActivity (intent) ;

接着实现register.java函数，在Edittext中接受姓名字符串；分别监听“拍照”和“注册结束”两个Button。“拍照”的相应为调用内置相机程序，对返回的照片传送给JNI层的人脸归一化函数，并返回归一化后的图片，最后将其保存在SD目录中，并将文件路径和分配的标号写入上述的csv.txt文档。该人脸归一化函数的原型为：

public static native int FaceNormalization (int [] org, int w, int h, int [] res) ;

形参org为传递的图像数据，形参res为返回归一化的图像。“注册结束”的响应为调用人脸库的重新训练，其函数原型为：

public static native boolean FaceTrain ()

该函数不需要传递任何参数，它会直接读取csv.txt文档来训练指定的图片，并保存训练的结果。

在调用了JNI层函数的这两个JAVA函数，都

需要指定加载特定的动态链接库函数，加入编译后的动态链接库的名字为libfacerec.so，则需要加入语句：

static { System.loadLibrary ("facerec") ;}

4.3 JNI层函数接口

首先需要确定Native方法所对应的C/C++函数的申明，这可以通过JDK所提供的javah工具来生成函数声明的头文件。在上述文件的Java文件中已经申明了Java层的函数原型，已经所在的动态库的名字。打开命令窗口，进入工程的src文件所在的路径，输入如下命令：

Javah -classpath . -jni package_name.class_name则会在src目录下生成一个头文件，该文件定义了JNI层所对应的函数形式，只需要将其拷贝到新建的C++文件，并实现其函数体。

然后将下载的bytefish库中src和include目录下所有的文件都复制到工程的jni目录下，并新建一个JNI接口C++文件，命名为facerec_JNI_interface.cpp，该文件实现上述生成的头文件的内容，三个接口函数以及关键代码分别为如下：

> JNIEXPORT jint JNICALL Java_com_embed_FacerecActivity_FaceNormalization (JNIEnv *env, jclass obj, jintArray org, jint w, jint h, jintArray res) ;

Jint *cbuf = env->GetIntArrayElements (org, 0) ; //获取传递的人脸图像

Mat org_pic (h,w,CV_8UC4, (unsigned char*) cbuf) ; //传递给矩阵

剩下的代码就是该矩阵传递给在windows平台下写的函数即可。

> JNIEXPORT jboolean JNICALL Java_com_embed_FacerecActivity_FaceTrain (JNIEnv *env, jclass obj) ;

Ptr <FaceRecognizer > model = createEigenFaceRecognizer () ; //建立特征人脸识别器

Model->train (images, labels) ; //训练人脸图片，iamges：读入的图片，labels：标签model->save ("Train_model.xml") ; //保存训练的模型，这样下一次直接调用即可

> JNIEXPORT jstring JNICALL Java_com_embed_FacerecActivity_Facerec (JNIEnv *env, jclass obj, jintArray buf, jint w, jint h) ;

Jint *cbuf = env->GetIntArrayElements (buf, 0) ; //获取传递的人脸图像

Mat pic (h,w,CV_8UC4, (unsigned char*) cbuf) ; //传递给矩阵

Ptr <FaceRecognizer > model = createEigenFaceRecognizer () ;

model->load ("Train_model.xml") ; //加载训练的模型

int predict = model->predict (pic) ; //预测人脸

4.4 编写脚本文件

在jni目录下，还需要编写两个脚本文件Android.mk和Application.mk，给出编译信息和编译规则。在Android.mk文件需要配置如下信息：

LOCAL_PATH:=$ (call my-dir) //定位源代码文件的目录

Include $ (CLEAR_VARS) //清楚所有的变量

OPENCV_INSTALL_MODULES: =on//复制OpenCV动态链接库到工程的链接库文件中

OPENCV_CAMERA_MODULES: =off//跳过OpenCV库中与摄像头相关的库文件

OPENCV_LIB_TYPE:=STATIC//静态链接

Include ../OpenCV-2.4.4-android-sdk/sdk/native/jni/OpenCV.mk // 包含OpenCV.mk文件，该文件的路径根据实际情况设定

LOCAL_MODULE: =facerec//库的名字，与在JAVA文件中申明的一样

LOCAL_SRC_FILES:= <all c/c++ files> //等号右边要列出所有的C/C++文件，不用包含头文件

LOCAL_LDLIBS+= -llog - ldl //包含的一些其他本地库

Include $ (BUILD_SHARED_LIBRARY)

在Application.mk文件申明一些必要的信息：

APP_STL:=gnustl_static

APP_CPPFLAGS:=-frtti - fexceptions

APP_ABI:=armeabi-v7a //指定应用目标的平台

5 应用程序实际测试

人脸识别的核心算法在Windows平台和Android手机平台分别作了测试。使用的电脑的性能为Intel Core i3-2100 CPU, 3.1GHz.Android手机是型号为ME525+，搭载1GHz的OMAP3620处理器。测试的图片是AT&T的人脸图像库，每张图片的大小为92*112，假设每个人注册三张图片，改变注册人的数量，比较时间，如下表1所示。

表1 Windows和Android平台下
人脸训练时间和识别时间对比

	Windows		Android	
注册人数	训练时间(ms)	识别时间(ms)	训练时间(ms)	识别时间(ms)
2	1.5	0.08	64.2	4.39
4	8.7	0.27	443.3	8.24
8	21.8	0.36	985.5	12.60
16	82.4	1.64	3508.9	21.24
32	282.4	1.72	11791.5	39.12

从上表中可以发现，在Windows平台下训练时间和识别时间完全可以接受，特别是在训练完后，可以做到视频的实时处理。而在Android平台下，当人数超过8人后，其训练时间开始超过1秒，有明显的等待时间，因此在人数上升时，需要有足够的等待时间。但从识别时间完全可以接受。因为该程序在实际使用中，应该为先注册，再识别，所以使用中实时性不是问题。

6 结束语

本文介绍了在Android平台下，利用OpenCV的算法库，结合Android NDK进行人脸识别的开发。它利用了OpenCV成熟的算法程序，能在开发中做到高效性。而且，Android NDK的灵活开发方式，可以允许我们在Windows平台下做完所有核心算法的测试，方便的移植到Android平台下，将界面开发和算法开发分割开来，加快程序的编写。在实时性方面，目前市场上普通的智能机完全能够达到。而且，随着技术的发展，高频和多核手机越来越普遍，复杂的应用的程序能完美的运行在这些智能手机平台上。

参考文献

[1] W.Zhao,R. CHELLAPPA,P. J. PHILLIPS, A. ROSENFELD. Face Recognition: A Literature Survey [C] . ACM Computing Surveys, Vol. 35, No. 4, December 2003, pp. 399-458

[2] Android Wikihttp://zh.wikipedia.org/wiki/Android

[3] OpenCV Wiki http://opencv.willowgarage.com/wiki/

[4] Gary R. Bradski, Adrian Kaehler（著），于仕琪，刘瑞祯（译）. 学习OpenCV（中文版） [M] . 清华大学. 2009.10

[5] 王宏彬等. 利用OpenCV实现在Android系统下的人脸检测 [J] . 中国科技论文在线. 2011, (10)

(上接第60页)

有一定的实用价值。

参考文献

[1] 刘和平等.数字信号控制器原理及应用-基于TMS320F2808 [M] .北京：北京航空航天大学出版社，2011.

[2] Texas Instruments. TMS320F2808 Digital Signal Processor Data Manual（SPRS230）, 2012.

[3] 宁改娣，曾翔君，骆一萍.DSP控制器原理与应用.北京：科学出版社，2009.

[4] 彭启琮，李玉柏，管庆.DSP技术的发展与应用.北京：高等教育出版社，2004.

作者简介

尹维春（1973-），男，硕士，工程师，从事DSP原理及应用、自动控制原理等课程的实验教学工作。