2019-11-15

OPEN AI LAB

人脸算法案例

变更记录  
(Reversion Record)

|  |  |  |  |
| --- | --- | --- | --- |
| 日期  (Date) | 版本  (Rev) | 说明  (Change Description) | 作者  (Author) |
| 2019.11.15 | V0.1 | 初稿 | 路明 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录(**catalog**)

[1 人脸算法案例演示介绍 3](#_Toc24740104)

[1.1 概述 3](#_Toc24740105)

[1.2 人脸识别的概念 3](#_Toc24740106)

[1.2.1 基本概念 3](#_Toc24740107)

[1.2.2 辨识（1:N） 5](#_Toc24740108)

[1.2.3 验证（1:1） 5](#_Toc24740109)

[2 人脸算法演示 6](#_Toc24740110)

[2.1 获取源码 6](#_Toc24740111)

[2.2 编译指引 6](#_Toc24740112)

[2.3 运行指引 6](#_Toc24740113)

[3 演示使用介绍 7](#_Toc24740114)

[4 FAQ 8](#_Toc24740115)

# 人脸算法案例演示介绍

## 概述

本案例基于EAIDK套件，展示人脸算法的概念和实际应用。

## 人脸识别的概念

### 基本概念

人脸识别，是基于人的脸部特征信息进行身份识别的一种生物识别技术。用摄像机或摄像头采集含有人脸的图像或视频流，并自动在图像中检测和跟踪人脸，进而对检测到的人脸进行脸部识别的一系列相关技术，通常也叫做人像识别、面部识别。

人脸识别系统主要包括四个组成部分，分别为：人脸图像采集及检测、人脸图像预处理、人脸图像特征提取以及匹配与识别。

人脸图像采集：不同的人脸图像都能通过摄像镜头采集下来，比如静态图像、动态图像、不同的位置、不同表情等方面都可以得到很好的采集。当用户在采集设备的拍摄范围内时，采集设备会自动搜索并拍摄用户的人脸图像。

人脸检测在实际中主要用于人脸识别的预处理，即在图像中准确标定出人脸的位置和大小。人脸图像中包含的模式特征十分丰富，如直方图特征、颜色特征、模板特征、结构特征及Haar特征等。人脸检测就是把这其中有用的信息挑出来，并利用这些特征实现人脸检测。主流的人脸检测方法基于以上特征采用Adaboost学习算法，Adaboost算法是一种用来分类的方法，它把一些比较弱的分类方法合在一起，组合出新的很强的分类方法。人脸检测过程中使用Adaboost算法挑选出一些最能代表人脸的矩形特征(弱分类器)，按照加权投票的方式将弱分类器构造为一个强分类器，再将训练得到的若干强分类器串联组成一个级联结构的层叠分类器，有效地提高分类器的检测速度。

人脸图像预处理：对于人脸的图像预处理是基于人脸检测结果，对图像进行处理并最终服务于特征提取的过程。系统获取的原始图像由于受到各种条件的限制和随机干扰，往往不能直接使用，必须在图像处理的早期阶段对它进行灰度校正、噪声过滤等图像预处理。对于人脸图像而言，其预处理过程主要包括人脸图像的光线补偿、灰度变换、直方图均衡化、归一化、几何校正、滤波以及锐化等。

人脸图像特征提取：人脸识别系统可使用的特征通常分为视觉特征、像素统计特征、人脸图像变换系数特征、人脸图像代数特征等。人脸特征提取就是针对人脸的某些特征进行的。人脸特征提取，也称人脸表征，它是对人脸进行特征建模的过程。人脸特征提取的方法归纳起来分为两大类：一种是基于知识的表征方法；另外一种是基于代数特征或统计学习的表征方法。基于知识的表征方法主要是根据人脸器官的形状描述以及他们之间的距离特性来获得有助于人脸分类的特征数据，其特征分量通常包括特征点间的欧氏距离、曲率和角度等。人脸由眼睛、鼻子、嘴、下巴等局部构成，对这些局部和它们之间结构关系的几何描述，可作为识别人脸的重要特征，这些特征被称为几何特征。基于知识的人脸表征主要包括基于几何特征的方法和模板匹配法。

人脸图像匹配与识别：提取的人脸图像的特征数据与数据库中存储的特征模板进行搜索匹配，通过设定一个阈值，当相似度超过这一阈值，则把匹配得到的结果输出。人脸识别就是将待识别的人脸特征与已得到的人脸特征模板进行比较，根据相似程度对人脸的身份信息进行判断。这一过程又分为两类：一类是验证，是一对一进行图像比较的过程，另一类是辨识，是一对多进行图像匹配对比的过程。

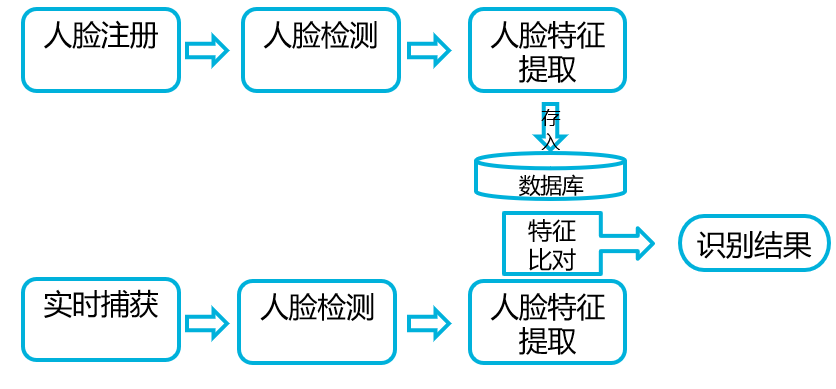


图1-1 人脸识别系统基本架构

### 辨识（1:N）

人脸辨识做的是1:N的比对，即系统采集了“我”的一张照片之后，从海量的人像数据库中找到与当前使用者人脸数据相符合的图像，并进行匹配，找出来“我是谁”。比如疑犯追踪，小区门禁，会场签到，以及新零售概念里的客户识别。

1:N人脸识别模式，同时具有动态比对与非配合两种特点。动态对比是指通过对动态视频流的截取来获得人脸数据并进一步比对的过程；而非配合性是指识别的过程表现出非强制性与高效性的特点，识别对象无需到特定的位置便能完成人脸识别的工作。由于这两个特性使1:N身份认证模式能迅速落地于公共安全管理与VIP客户人脸识别等场景，但其难度要远高于静态1:1，因为机器面临着曝光过度、逆光、侧脸、远距离等挑战。

### 验证（1:1）

人脸验证做的是1:1的比对，其身份验证模式本质上是计算机对当前人脸与人像数据库进行快速人脸比对，并得出是否匹配的过程，可以简单理解为证明你就是你。就是我们先告诉人脸识别系统，我是张三，然后用来验证站在机器面前的“我”到底是不是张三。  
这种模式最常见的应用场景便是人脸解锁，终端设备(如手机)只需将用户事先注册的照片与临场采集的照片做对比，判断是否为同一人，即可完成身份验证。

1:1作为一种静态比对，一般在金融、信息安全领域中应用较多。例如在高速路、机场安检时，受检人员手持身份证等证件，通过检查通道，同时对受检人员的外貌及身份证信息进行识别，此过程就是典型的1:1模式的人脸识别。

# 人脸算法演示

## 获取源码

请联系相关相关技术支持或者销售经理获取源码

## 安装依赖

安装qt5和qtcreator，及其它依赖库。

sudo dnf install qt5 qt-creator qt5-qtmultimedia-devel

sudo dnf install protobuf-devel

sudo dnf install opencv-devel

## 运行指引

在编译的目录build-eaidk\_visual\_embeded-Desktop-Debug，已预编译好生成eaidk\_visual\_embeded文件，可以直接执行。

./eaidk\_visual\_embeded

## 编译指引

如需自行编译运行，参数如下步骤：

1. EAIDK310 DDR规格是1G，可能会碰到本地编译内存不够的情况。

可以用物理磁盘扩大一点swap内存，操作内容如下：

可以在build-eaidk\_visual\_embeded-Desktop-Debug目录执行已写好的脚本：

sh make\_swap.sh

实际内容如下：

sudo dd if=/dev/zero of=/swapfile bs=64M count=16

sudo mkswap /swapfile

sudo chmod 600 /swapfile

sudo swapon /swapfile

可以通过free -m查看swap内存大小。

1. 编译依赖tengine，不过默认情况下EAIDK310已安装tengine1.7.1版本，如必要，可以输入如下命令安装:

sudo dnf install tengine-1.7.1\_EAIDK310 (默认可以忽略该步骤)

(3)开发板不带RTC电池，第一次上电开机时，系统时间会是默认时间2018-05-11，需要手动设置一下当前系统时间，参考命令如下：

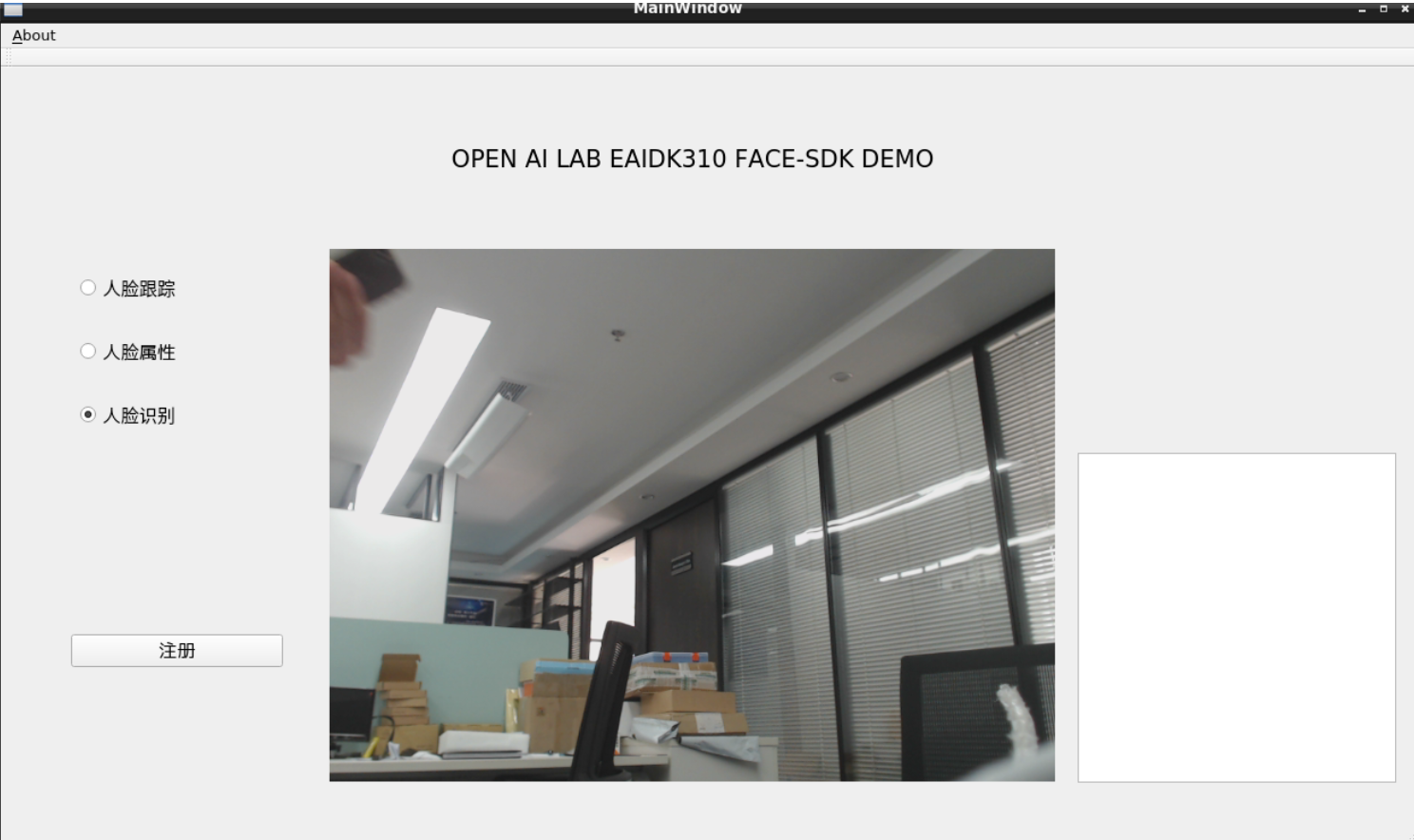
sudo date -s “2019-11-29 09:16:30” (该时间是当前北京时间)

(4)解压源码包，进入build-eaidk\_visual\_embeded-Desktop-Debug，直接输入make命令即可。

# 演示使用介绍

1，打开软件，左侧显示人脸算法的三个示例功能：人脸跟踪、人脸属性、人脸识别。

2，单项体验人脸识别时，左下方会有根据流程注册人脸，并比对，界面如下。



# 注意&FAQ

notice 1:

本次硬件配置是EAIDK310+USB单目摄像头即可，程序默认/dev/video0是USB摄像头的设备节点，暂时没有做自动探测。

notice 2:

1. 当前案例测试的USB摄像头型号是HD1080P logitech usb摄像头。



B. 后续规划会加上双目摄像头的调试。

notice 3：

案例源码中调用Face-SDK的接口，看参照另外提供的Face-SDK手册和Q&A手册。

notice 4：

当修改mainwindow.ui时，重新在build-eaidk\_visual\_embeded-Desktop-Debug目录通过命令

qmake-qt5 ../eaidk\_visual\_embeded/eaidk\_visual\_embeded.pro -o Makefile

然后再执行make

notice 5：

本案例包已包含Vision.SDK与相关文档，具体问题可以与相关技术支持联系。