



上海理工大學

机器视觉综合实践报告

专业：智能科学与技术 姓名：荀镕基 学号：1819660230

组别：同实验者：荀镕基

指导教师：巨志勇

实验日期：2019.7.1-7.5

课程成绩：

课程（论文）评分依据（必填）：

任课教师签字：

日期： 年 月 日

人脸识别系统

一、实验目的

1. 学习 LabVIEW 软件的基本操作
2. 使用 LabVIEW 实现人脸的实时采集、识别功能：
 - (1) 使用 VISION 库中 IMAQdx 模块实现通过摄像头实时采集图像，实时状态下对人脸进行检测、识别
 - (2) 读取本地图像文件，检测并截取图中人脸区域，进行人脸识别

二、实验设备

PC, win10, LabVIEW 2017(VISION 模块, NIVision-OpenCV 模块), MATLAB R2018b;

三、实验原理

1.主成分分析 PCA (K-L 变换)

1.1 方法一：计算所有模式的协方差矩阵

$$C = \frac{1}{n} \sum X_i * X_i^T - M * M^T$$

方法二：自相关矩阵 R

$$R = \frac{1}{n} \sum X_i * X_i^T$$

1.2 计算协方差矩阵 C，或自相关矩阵 R 的特征值 λ_i

$$|\lambda * I - C| = 0 \text{ or } \det(\lambda * I - C) = 0$$

1.3 根据需求，按不同顺序提取特征值

- 1) 单个类提取特征，从小达到选取 m 个特征值 λ_i (去除方差较大的分量)
 - 2) 多个类分类：从大到小选取 m 个特征值 λ_i (选方差较大分量，凸出可分性)
- 对选出的特征值 λ_i ，代入 $C * u_k = \lambda_k * u_k$ 公式计算特征向量

1.4 将特征向量 u_k 转置为行向量，作为变换矩阵 A 的行。通过变换矩阵 A 对样本 X 进行变换

$$X_i' = A * X_i$$

2.基于 PCA 的人脸识别

2.1 将图像样本展平为一列向量

因为样本 X 为 $n_1 * n_2$ 大小的图像灰度值矩阵，因此首先需要将图像样本 X 按列展开为一个 $(N=n_1 * n_2) * 1$ 大小列向量。

所有图像展平为列向量的样本 X，按列存入矩阵 T。

2.2 计算中心化矩阵 A、协方差矩阵 L

1) 计算 n 个样本列向量 X 的均值 m

2) 计算中心化矩阵 A ($N \times n$)

$$A(:, i) = T(:, i) - m$$

3) 计算矩阵 L (协方差矩阵, 注: L 矩阵为 $n \times n$ 维度 (n 为样本数))

$$L = A' * A$$

2.3 计算特征脸 EigenFace (降维 K 维, 则有 K 张特征脸)

1) 计算特征值、特征向量, 组合为变换矩阵 Lei

从大到小选取特征值 λ_i , 将对应特征向量组合为变换矩阵 Lei 。

2) 特征脸 EigenFace = $A * Lei$ (特征脸为 $N \times k$ 大小矩阵, 类似 KL 变换中的变换矩阵)

2.4 识别过程

1) 计算所有训练样本的映射向量 X' ($k \times 1$ 维列向量)

将样本 X 化为中心化后的列向量

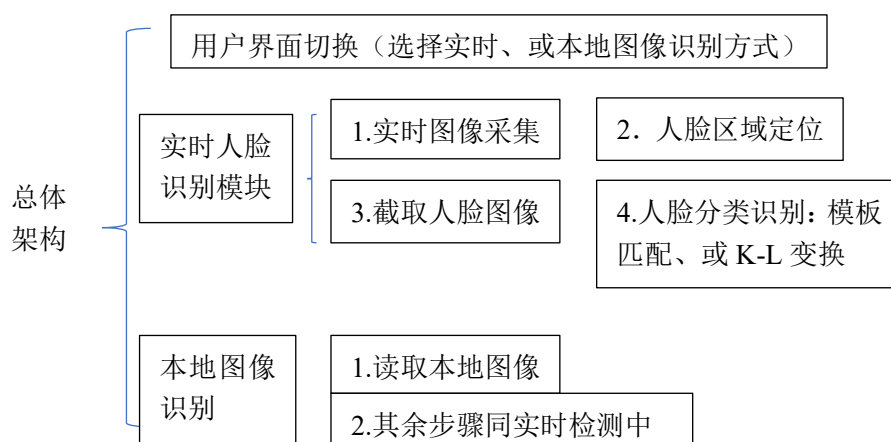
$$X' = E' * A(:, i)$$

2) 对测试样本预测

对所用测试样本进行中心化, 求解映射向量 $Test$ 。将 $Test$ 列向量与所有训练样本的映射向量求二范数距离, 将该测试样本预测为与距离最小的训练样本一类。

四、系统方案设计

本实验程序总体架构:



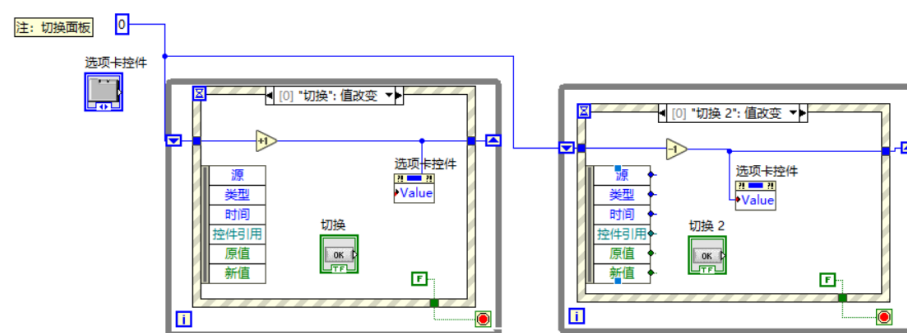
五、系统 VI 构建

1. 用户界面模块

用户界面使用 1 个选项卡控件，通过左上角“切换”按键来切换“实时识别”或“本地图像识别”

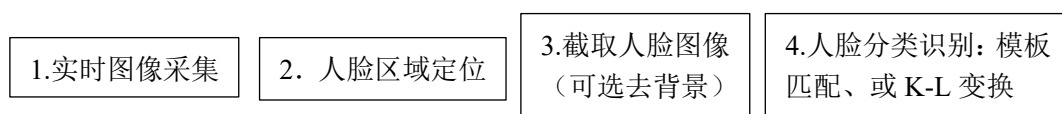


通过“切换”按键控制 2 个选项卡的切换：



2. 实时检测模块

实时人脸识别流程：

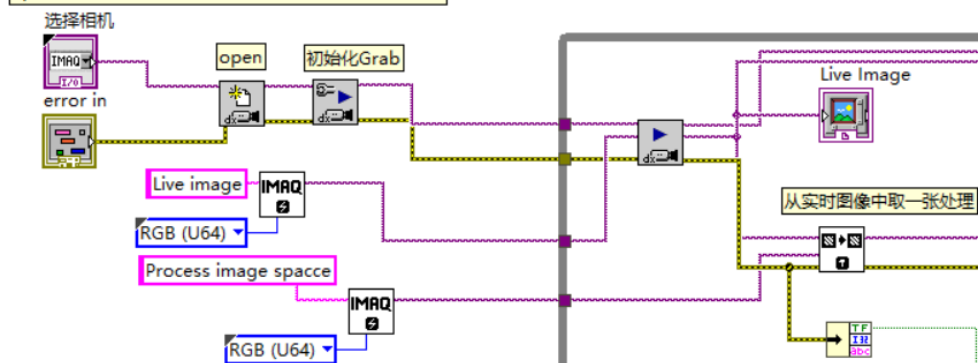


2.1 实时图像采集

(1) 摄像机设置

- 1) 调用 IMAQdx open.vi 打开摄像机，完成图像采集板卡的初始化；
- 2) 调用 IMAQdx Grab Setup 完成 Grab 操作的初始化；
- 3) 调用 IMAQ Create 创建图像的数据缓冲区；
- 4) 调用 IMAQdx Grab Acquire 完成快速采集图像数据

一、实时监测模块：获取实时图像
1) 对copy后的图像来处理，可以拖到while外面
2) 处理的时候加时延

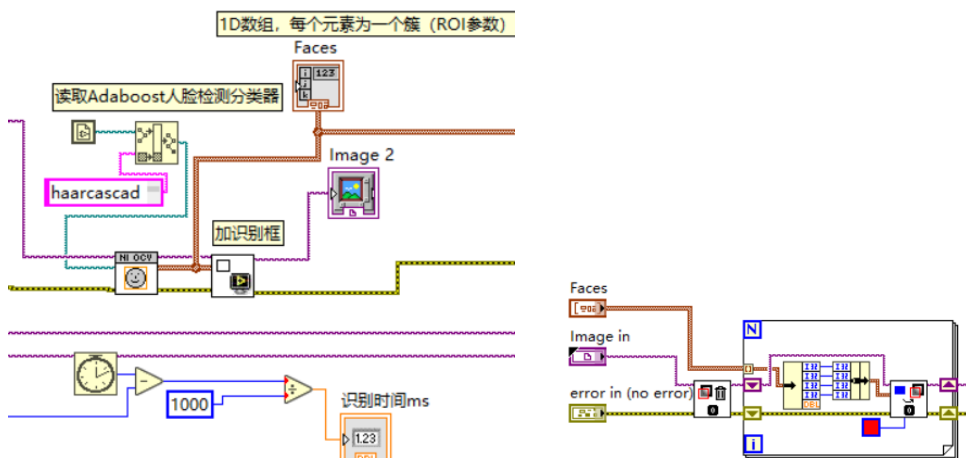


(2) 将 Grab 模块放入 while 循环中，对摄像头采集的每一帧最新的图像进行检测。并使用 Copy 模块复制每一帧最新的图像，对 Copy 后的图像进行后续处理。

2.2 人脸区域定位

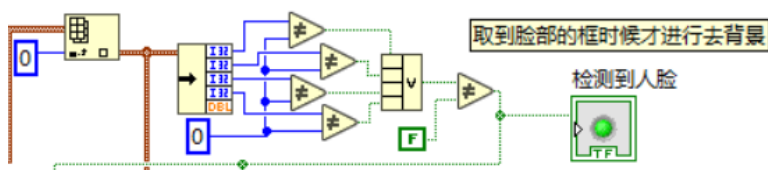
1) 调用 NIVision OpenCV 库中的 Facedetection 模块，实现人脸区域的定位，获取人脸区域的最小外接矩形（返回 4 位置个参数：Left、Top、Right、Bottom）

2) 编写子 vi：使用红色的框将人脸区域框出来。

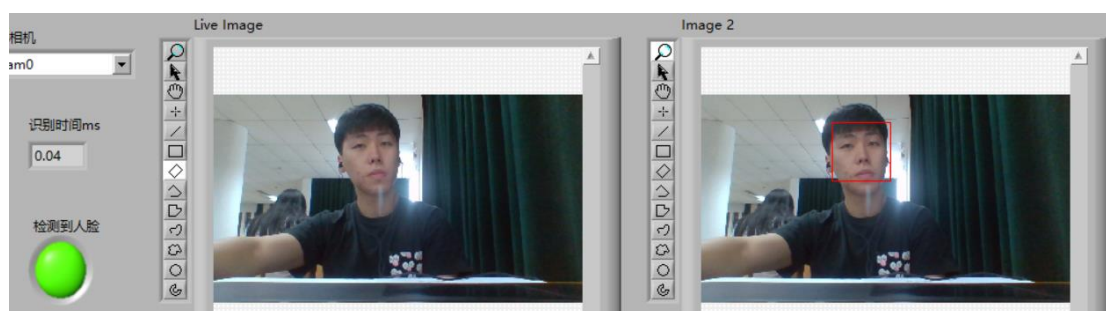


3) 判别是否此帧下存在人脸：

通过判断 Facedetection 模块返回的人脸最小外接矩形 4 个参数，若均为 0，则此帧下无人脸图像；判断出存在人脸后，再进行后续识别处理。



人脸定位效果：

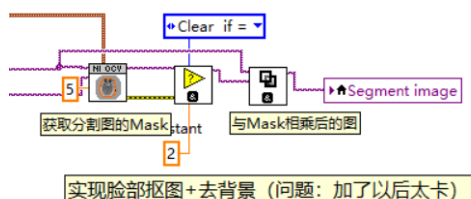


2.3 截取人脸图像

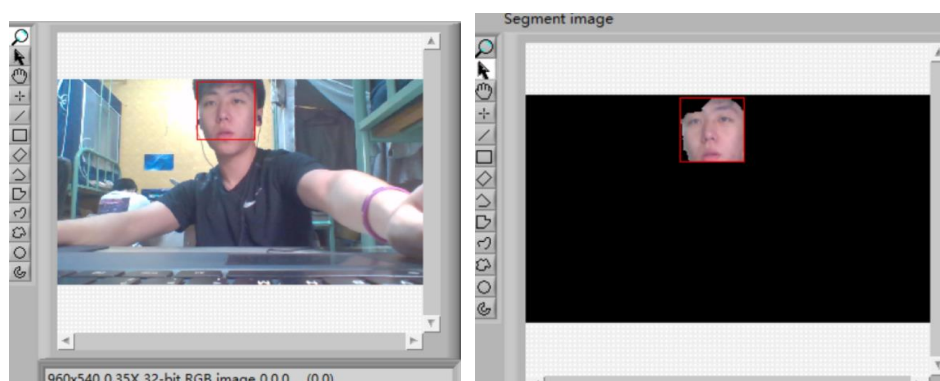
2.3.1 显示“去背景的人脸图像”

使用垂直摇杆开关，选择是否对人脸区域去背景。若开关和判断存在人脸均为 True，则调用 OpenCV GrabCut 模块获取人脸区域的 Mask，将原图与 Mask 相乘得到去背景的人脸图像。

问题：在实时检测中，若加入去背景操作，运算量太大，会造成严重卡顿的现象。

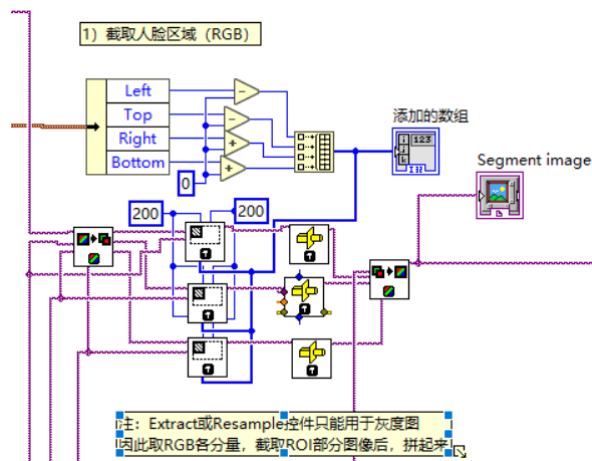


去背景效果：

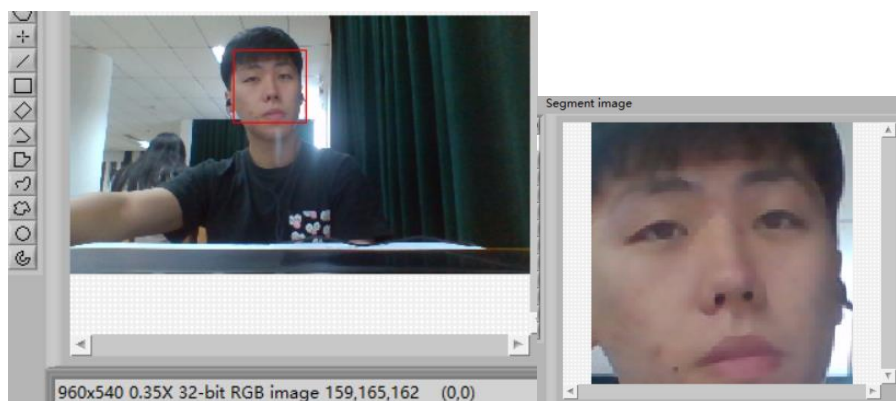


2.3.2 截取保留背景的图像

当“判断是否存在人脸”为 True, 且“去背景”开关为 False，则截取保留背景的人脸区域。获取最小外接矩形的 4 个位置参数，对 RGB 图像每个分量，分别选取出相应区域，并缩放到统一大小 200*200 便于进行 K-L 变换，最后再合起来。



截取效果:

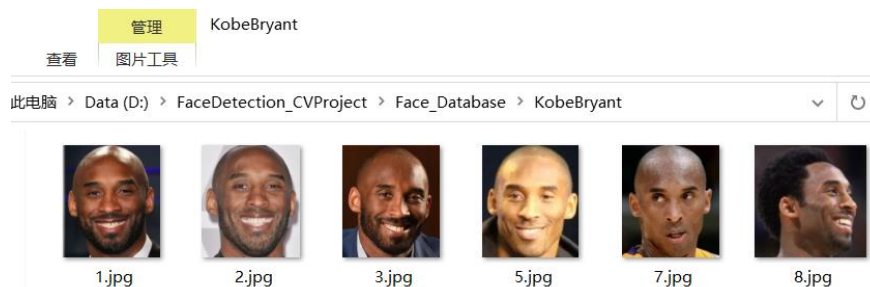


2.4 人脸分类识别

对截取出来并缩放到 200×200 的人脸图像, 使用模板匹配、K-L 变换方法, 与数据库内的图像进行匹配、识别。

2.4.1 数据库 Face_Database 构造:

将每个人的人脸图像存储于, 以其“人名”命名的文件夹中

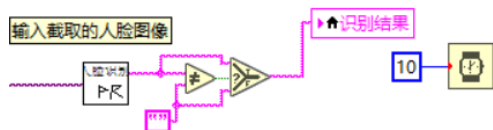


K-L 变换进行分类时, 只需存储所有训练样本投影后的向量矩阵, 均值向量, 变换矩阵 (特征脸矩阵), 和存储每个图像路径、名称的 Matlab struct 结构体即可。

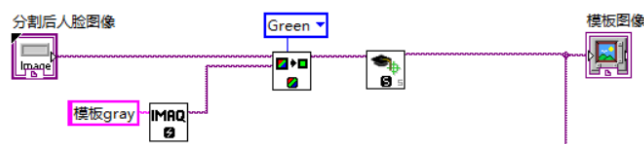
2.4.2 分类识别

(一)模板匹配法（封装到“人脸识别程序_2 调用 DB 匹配.vi”子 vi 中）

输入：截取的人脸图像；输出：识别结果字符串（人名）

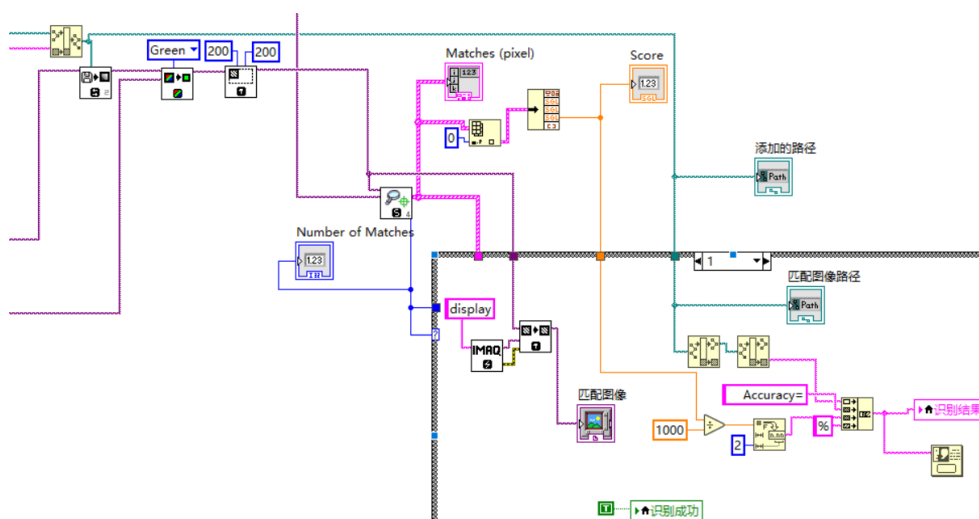


1) 调用 Learn Pattern 控件对截取的人脸图像（模板）进行学习



2) 模板匹配：调用 Match Pattern 控件

遍历 Face_Database 数据库中每个文件夹（每个人）的每张人脸图像，将截取大小为 200*200 人脸区域，与数据库中每一张图像进行匹配



模板匹配效果：分析：模板匹配常用于在一张图中选取一部分图像作为模板，并且根据此模板，在图中匹配大小，形状等近似的区域进行匹配。而实时采集中，截取的人脸图像大小很难与数据库图像中的人脸部分大小类似，因此识别效果不好。

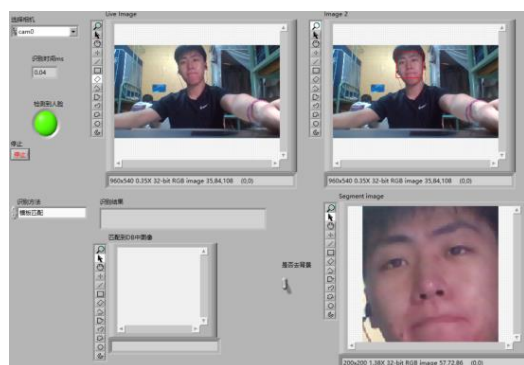


图 1 等待 10s 仍无法识别

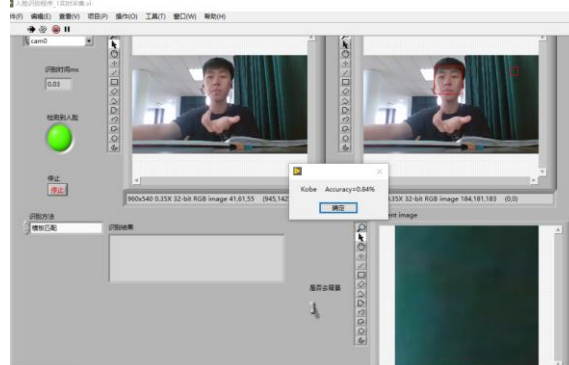


图 2 识别错误

(二) 调用 Matlab 下编写的 K-L 变换算法，进行人脸识别

1) 在 Matlab 下，编写 K-L 变换算法：

运行“EXP7_FaceRecognitionPCA.m”提取数据库图像的降维后的矩阵，均值向量，变换矩阵，训练图像的结构体（对应人名、图像路径），存储到“FaceDetect_DB_7_15.mat”

2) 最小距离分类

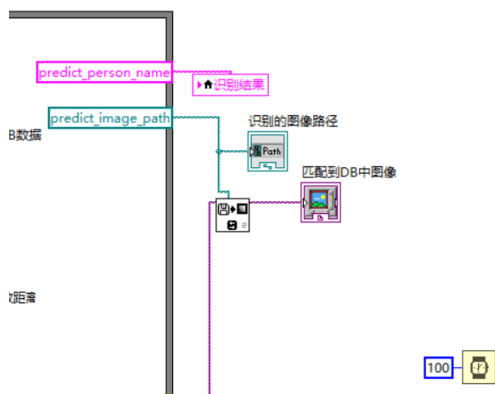
将“截取的人脸图像”与变换矩阵相乘，进行降维；计算与 DB 中降维后的矩阵每个降维图像的距离，选取距离最近的样本作为分类结果。

3) 将 2) 中 matlab 预测程序使用 MathScript 导入 LabVIEW 中

将截取的 RGB 人脸图像，取绿色分量（观察效果最好）作为灰度图，传入 MathScript 中，进行预测；返回分类结果（距离最近图像对应的人名，图像路径）



4) 显示分类结果（人名、距离最近的图像）



K-L 变换识别效果:



图 2-1 K-L 识别结果：识别正确且迅速

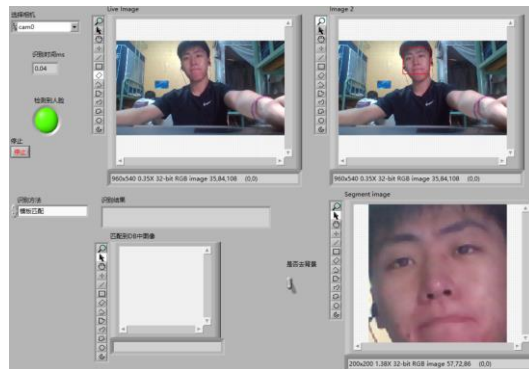


图 2-2 模板匹配结果：等待 10s 未能识别

分析：K-L 变换，通过计算图像的降维后的主成分向量之间的距离来进行识别，克服了模板匹配中需要高质量且极为近似模板的缺点，能够更快且准确地进行识别。

缺点：当截取的面部图像为侧脸、角度不同、面部阴阳脸（一部分颜色偏暗）识别易出现错误；并且识别的稳定性不高，实时检测中，面部略有晃动可能造成识别为不同类。



图 2-3 K-L 变换识别结果：识别正确

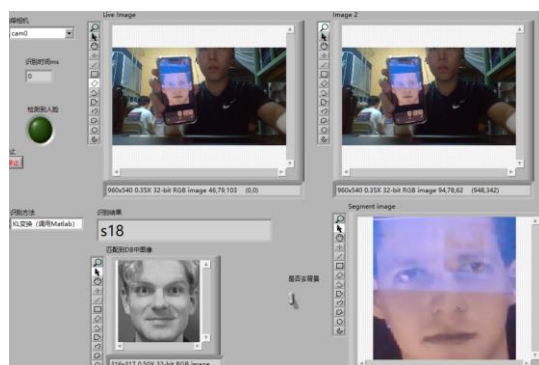


图 2-4 K-L 变换识别错误

3. 本地图像人脸识别模块

本地图像人脸识别，和实时识别，区别仅在于人脸图像加载的是本地图像。其余识别步骤与实时检测中相同。

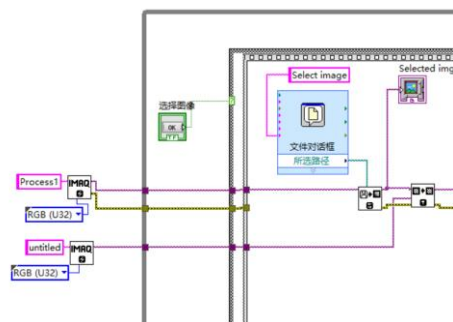


图 3-1 加载本地人脸图像

本地图像人脸识别效果：

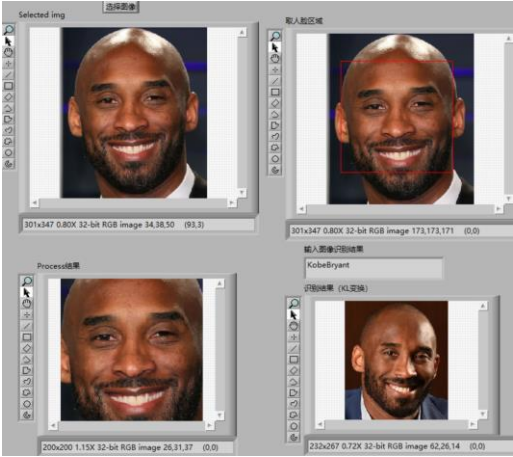


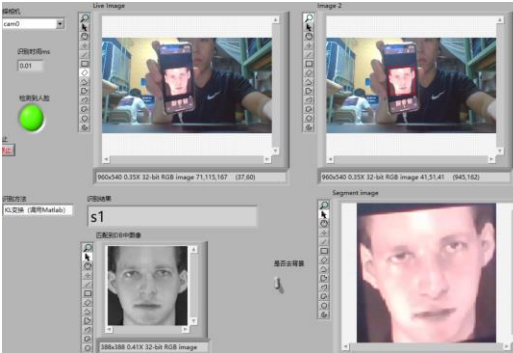
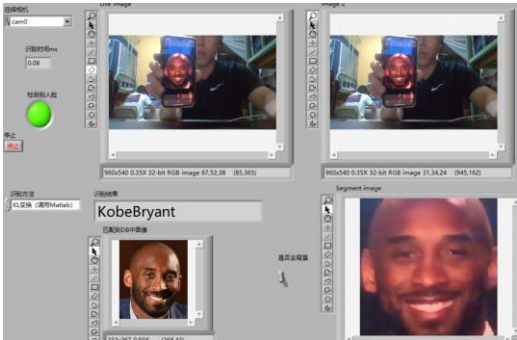
图 3-2 本地图像识别：识别正确



图 3-3 本地图像识别：无法识别

- 1) 当本地图像质量较好时，本地图像识别效果优于实时人脸识别，识别更为稳定。
- 2) 缺点：当本地图像无法识别时，无法像实时识别中可任意旋转角度，改变位置，从而获取到质量更好的人脸图像。

六、实验结果



七、实验小结

1) 通过本次短学期, 学习掌握了 LabVIEW 的基本操作, 例如: 前面板、程序框图的如何添加控件、数据结构(数组、字符串、Vision 图像等)的使用、程序结构(顺序、循环、事件结构实现点击按钮触发指定操作、公式结点来简化程序框图, 用使能结构添加注释)、子 vi 的编写与调用、读取图像等文件 I/O 操作,

2) 通过“人脸识别系统”实验, 学会了通过 Vision 模块来进行实时图像采集、并通过添加 Copy 控件实现对前一帧图像的处理; 练习了实验 LabVIEW 进行直方图均衡化、边缘检测等图像预处理操作;

并且练习掌握了通过 MathScripts 将 Matlab 程序导入 LabVIEW 进行计算的流程。

3) 练习了 LabVIEW 中模板匹配的识别方法、复习巩固了 K-L 变换算法的基本流程。