

第四届

全国大学生集成电路创新创业大赛

CICIEC

RISC-V 片上系统设计——身份识别应用 项目设计报告

参赛题目： RISC-V 子赛题 2

队伍编号： ALN114318

团队名称： Low Tech

目 录

1. 设计简介	1
2. 片上系统设计方案	1
2.1 使用 LowRISC 实现外设连接及图像处理功能	1
2.1.1 LowRISC 简介	1
2.1.2 外设添加及 LowRISC 系统构建	2
2.1.3 基础功能测试	3
2.2 使用 Instant SoC 实现 PMOD 输出和数码管显示	5
2.2.1 Instant SoC 简介	5
2.2.2 外设连接与使用	5
2.3 OpenCV 交叉编译与部署	5
3. 身份证数字识别方案	6
3.1 图像预处理	6
3.2 数字识别算法	7
4. 人脸识别方案	8
4.1 人脸检测	8
4.2 人脸识别	9
5. 总结与后续工作	10
5.1 阶段总结	10
参考文献	12

1. 设计简介

本作品实现了基于 RISC-V 架构的片上系统,开发板型号为 Digilent Nexys4 DDR。使用了基于 RISC-V 架构的 IP 核(LowRISC),使用板上资源实现了 RAM 和 ROM;对 LowRISC 进行了外设的添加,通过 LowRISC 软核控制 PMOD 八路数字输出显示出 RISC-V 字样和使用数码管显示 LowTech 队伍名。选用了 LowRISC0.6 版本所提供的 Debian 操作系统,该系统除可正确调用 VGA 视频输出、板上开关、LED、以太网外,还可与 FTP 服务器进行通信,且能实现图像的显示与处理。在片上系统中成功部署 OpenCV 开发环境,基于印刷体数字的统计特征实现了身份证数字识别功能,并使用局部二值模式算法对给定人脸图像数据集进行了模型训练和识别,获得了较好的速度和准确率。

2. 片上系统设计方案

2.1 使用 LowRISC 实现外设连接及图像处理功能

2.1.1 LowRISC 简介

LowRISC 是一个基于 rocket 的开源项目,包括了 RISC-V 软核、适用于 RISC-V 的 Debian 系统包、uboot 启动引导程序和工具链等,其系统框图如图 2-1 所示^[2]。LowRISC 系统目前发布的稳定版本为 0.6,测试版本为 0.7RC2。其系统构建也存在两种方式,即使用 buildroot 构建和使用完整 Debian 系统包构建。经过反复测试及调整,最终选用了 0.6 版本的完整 Debian 系统包作为运行在软核上的操作系统。

在测试过程中还发现,LowRISC 的 0.7RC2 版本若使用完整 Debian 系统包,会遇到网卡无法正常启动的问题。经过调试,发现是由于其发布版本中系统构建存在缺陷,缺少 DHCP 服务组件导致。安装缺少的软件包后功能正常,目前已向 LowRISC 开发者报告此问题及解决方案(<https://github.com/lowRISC/lowrisc-chip/issues/175>)。

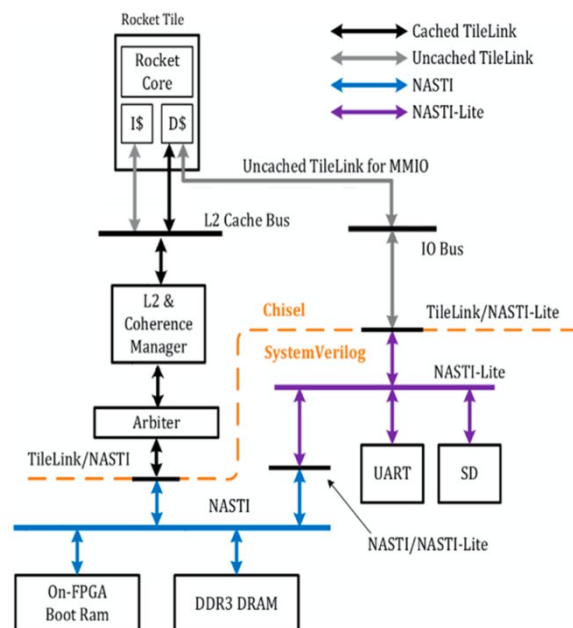


图 2-1 lowRISC 系统框图

2.1.2 外设添加及 LowRISC 系统构建

为了实现向 LowRISC 添加外设的功能，需要对其有关外设定义及调用的 `periph_soc.sv` 及 `chip_top.sv` 两个 System Verilog 源文件进行修改。为方便调试，使用 Vivado GUI 进行调试。首先使用 `make project` 命令生成 vivado 项目，然后使用 vivado GUI 打开 `lowrisc-chip-imp` 项目，进行 System Verilog 编程。通过定义新的外设及、在上层模块调用及通过 xdc 文件正确连线，实现了通过 PMOD JA 接口输出八位逻辑信号组成字样‘RISC-V’的功能，以及在数码管上显示‘LOVTECH’代表队伍名。经过在 vivado 内进行设计综合、生成比特流、烧录并测试，功能正常。

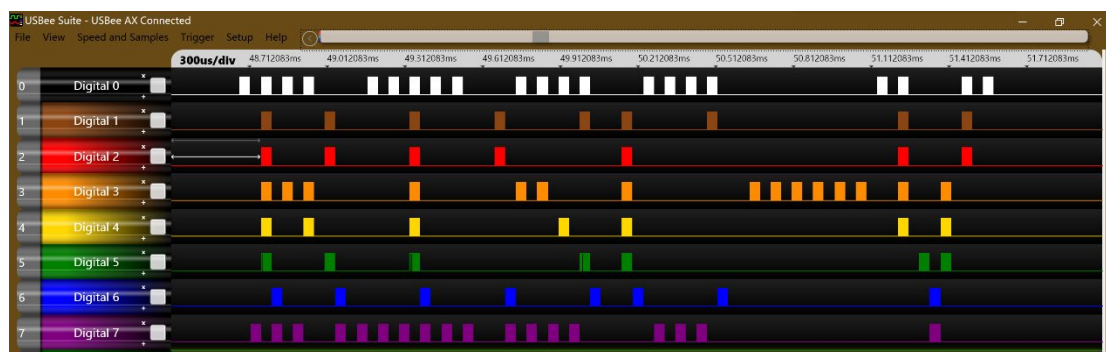


图 2-2 八位逻辑输出“RISC-V”字样

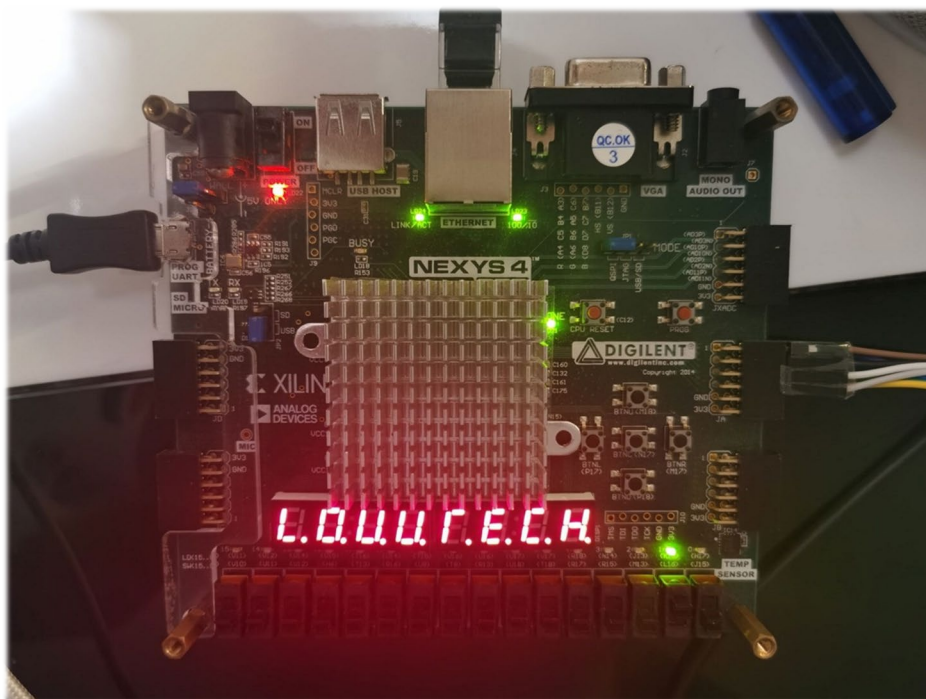


图 2-3 数码管显示“LOVVTECH”

功能测试正常后，需要将比特流写入 FLASH (QSPI Memory) 才可保证在板子重新上电后仍保留比特流二进制信息。在 vivado 内执行完毕 Generate Bitstream 后，使用 shell 执行 Makefile 命令：`make cfgmem && make program-cfgmem` 将比特流写入 QSPI Memory，至此完成 LowRISC 搭建。

2.1.3 基础功能测试

图 2-4 为在 LowRISC 上使用 cacaview 实现图片的显示功能展示。使用 `sudo apt-get install caca-utils` 安装 cacaview，然后执行 `cacaview Logo.jpg` 将大赛 Logo 通过 VGA 输出显示在显示器上。

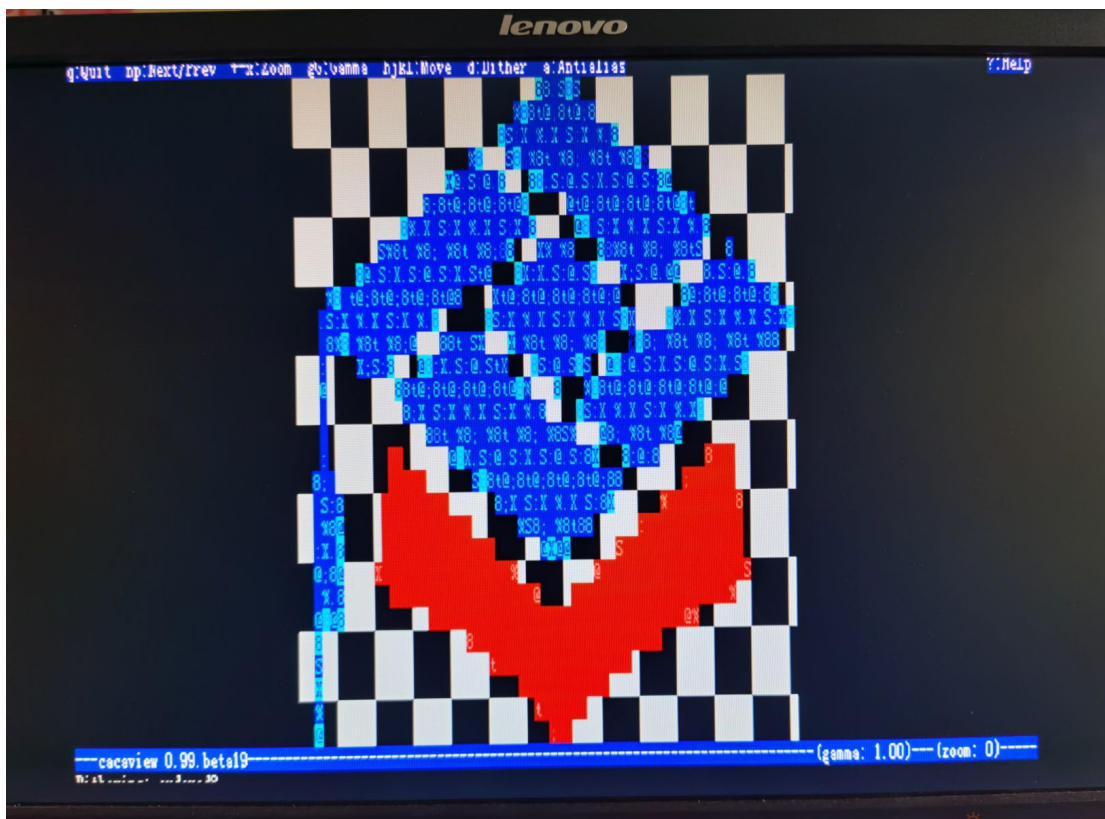


图 2-4 图片显示

图 2-5 为网络及 FTP 功能测试，FTP 功能是进阶功能依赖的功能，在进阶任务的执行脚本中均有体现。

```
Debian GNU/Linux buster/sid lowrisc tty1
lowrisc login: root
Password:
Last login: Tue Jun 16 08:08:51 BST 2020 from 192.168.123.109 on pts/0
Linux lowrisc 4.18.0-gc81ff0d #48 Thu Oct 18 16:00:24 BST 2018 riscv64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
[ 153.840000] systemd-logind[1511]: New seat seat0.
root@lowrisc:~# ping www.baidu.com
PING www.a.shifen.com (39.156.66.18) 56(84) bytes of data:
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=1 ttl=52 time=17.2 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=2 ttl=52 time=17.2 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=3 ttl=52 time=17.2 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=4 ttl=52 time=17.3 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=5 ttl=52 time=17.4 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=6 ttl=52 time=17.5 ms
64 bytes from 39.156.66.18 (39.156.66.18): icmp_seq=7 ttl=52 time=17.3 ms
^C
--- www.a.shifen.com ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 66ms
rtt min/avg/max/mdev = 17.162/17.297/17.466/0.155 ms
root@lowrisc:~# ftp 192.168.123.109
Connected to 192.168.123.109.
220 Microsoft FTP Service
Name (192.168.123.109:root): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 User logged in.
Remote system type is Windows_NT.
ftp> get logoword.jpg
local: logoword.jpg remote: logoword.jpg
200 PORT command successful.
125 Data connection already open; Transfer starting.
WARNING! 352 bare linefeeds received in ASCII mode
File may not have transferred correctly.
226 Transfer complete.
113074 bytes received in 0.53 secs (207.5273 kB/s)
ftp>
```

图 2-5 网络及 FTP 功能测试

2.2 使用 Instant SoC 实现 PMOD 输出和数码管显示^注

注：初赛中使用此方案实现 PMOD 和数码管外设连接，后经改进直接对 LowRISC 进行了外设的添加。决赛提交作品中未使用 Instant SoC。

2.2.1 Instant SoC 简介

Instant SoC 是一种将 C++ 代码编译为可综合 VHDL 代码的系统，其 VHDL 代码实现了包括基于 RISC-V 架构的软核 CPU、CPU 指令、存储器和外设的构建。其中 CPU 的外围设备，例如逻辑 I/O 输出、UART、SPI、I2C、AXI4 数据流被定义为 C++ 中的对象，可直接在调用时对其进行实例化。此外，VHDL 代码对生成的 RISC-V 指令进行了优化，只有使用过的指令才会被系统执行，并使用静态堆栈分析提高存储器的性能^[1]。

2.2.2 外设连接与使用

首先，使用系统库中的 FC_IO_In 和 FC_IO_Out 对板上开关和 LED 进行实例化，将编号为 1-15 的开关和 LED 进行一一对应的连接，当开关闭合时，对应的 LED 点亮。此外，PMOD 接口此时工作在 GPIO 输入输出模式，1-4 号和 7-10 号管脚可输出逻辑高低电平，使用编号为 0 的开关 sw0 用于控制 PMOD 输出字符，当该开关断开时，PMOD 不输出；当该开关闭合时，PMOD JA 接口将会输出 8 位数字信号，在逻辑分析仪上显示出“RISC-V”字样。其中字符的生成采用字模提取的方法，输入长宽比为 8*8 位的字母，通过控制点阵中每个像素点的电平值显示字符，因此理论上可显示任意字样。

其次，成功调用了板上的数码管模块，通过实例化系统库中的 FC_IO_SegmentDisplay 类，使用八位数码管显示“LowTech”字样。

2.3 OpenCV 交叉编译与部署

为了使身份证和人脸识别算法在片上系统中成功运行，需要使用 RISC-V gcc 工具链对 OpenCV 进行交叉编译，生成静态链接库以便程序调用。

首先需要在 Linux 系统上完成 RISC-V gcc 工具链的编译，本方案中的编译环境为 Ubuntu 18.04，RISC-V gcc 工具链版本信息如图 2-6 所示：


```
tbh@ubuntu: /opt/riscv/bin
File Edit View Search Terminal Help
tbh@ubuntu: /opt/riscv/bin$ riscv64-unknown-linux-gnu-gcc -v
Using built-in specs.
COLLECT_GCC=riscv64-unknown-linux-gnu-gcc
COLLECT_LTO_WRAPPER=/opt/riscv/libexec/gcc/riscv64-unknown-linux-gnu/9.2.0/lto-w
rapper
Target: riscv64-unknown-linux-gnu
Configured with: /home/tbh/Desktop/riscv-gnu-toolchain/riscv-gcc/configure --tar
get=riscv64-unknown-linux-gnu --prefix=/opt/riscv --with-sysroot=/opt/riscv/sysr
oot --with-system-zlib --enable-shared --enable-tls --enable-languages=c,c++,for
tran --disable-libmudflap --disable-libssp --disable-libquadmath --disable-nls -
-disable-bootstrap --src=../riscv-gcc --disable-multilib --with-abi=lp64d --wi
th-arch=rv64imafdc --with-tune=rocket 'CFLAGS_FOR_TARGET=-O2 -mcmodel=medlow'
'CXXFLAGS_FOR_TARGET=-O2 -mcmodel=medlow'
Thread model: posix
gcc version 9.2.0 (GCC)
tbh@ubuntu: /opt/riscv/bin$
```

图 2-6 RISC-V 交叉编译工具链

之后，使用 CMake 工具对 OpenCV 及其 contrib 模块源码进行交叉编译。编译时需要将 CMAKE_EXE_LINKER_FLAGS 设置为 `-lrt -lpthread` 以调用进程库支持。生成的动态链接库文件如图 2-7 所示。

```
tbh@ubuntu: ~/Desktop/cv/batch_proc/lib
File Edit View Search Terminal Help
tbh@ubuntu: ~/Desktop/cv/batch_proc/lib$ ls
cmake                                libopencv_gapi.so
libopencv_calib3d.so                libopencv_gapi.so.4.4
libopencv_calib3d.so.4.4            libopencv_gapi.so.4.4.0
libopencv_calib3d.so.4.4.0          libopencv_imgcodecs.so
libopencv_core.so                   libopencv_imgcodecs.so.4.4
libopencv_core.so.4.4               libopencv_imgcodecs.so.4.4.0
libopencv_core.so.4.4.0             libopencv_imgproc.so
libopencv_features2d.so              libopencv_imgproc.so.4.4
libopencv_features2d.so.4.4          libopencv_imgproc.so.4.4.0
libopencv_features2d.so.4.4.0        libopencv_photo.so
libopencv_flann.so                   libopencv_photo.so.4.4
libopencv_flann.so.4.4               libopencv_photo.so.4.4.0
libopencv_flann.so.4.4.0
tbh@ubuntu: ~/Desktop/cv/batch_proc/lib$
```

图 2-7 OpenCV 库文件的交叉编译结果

最后，调用 RISC-V gcc 编译器对识别程序进行交叉编译，生成可执行文件，输入参数即可完成身份证号和人脸的识别。

3. 身份证数字识别方案

3.1 图像预处理

图像预处理操作主要功能为对从 FTP 服务器上获取的身份证图片进行逐数字的分割，向数字识别模块传递单个数字图片进行识别。

本设计方案中主要使用 OpenCV 库以完成图像处理。由于给出的测试集中身份证为白底黑字，而需调用的轮廓查找函数 `cv2.findContours` 所接收的图像矩阵的背景色一般为黑色以获得更高的精度，因此将图像转换为灰度后使用 `bitwise_not` 函数进行黑白反转，调用 `threshold` 函数转换为二值图像。之后，

调用 `cv2.findContours` 函数依次找出每个数字的最小外接矩形并进行分割，将分割之后的图像进行识别。

3.2 数字识别算法

一般地，针对印刷体数字的识别方法有基于模版匹配的识别方法、基于神经网络的识别方法、基于数字特征的识别方法等。而本项目中所移植的 RISC-V 处理器主频为 50mHz，其性能不足基于满足模版匹配以及神经网络的识别方法，因此，可以根据数字的形状及其特征进行分析。在本系统中，选取了对数字的统计特征进行分析与提取的方法进行数字识别，即对图像进行二值化和分割操作之后，通过统计特定位置上的直线与数字交点的个数进行识别。根据身份证字体，所选择的直线位于数字高度的 $\frac{3}{10}$ 和 $\frac{2}{3}$ 处，以及数字宽度的 $\frac{1}{2}$ 处。以数字 6 为例，三条直线的特征示意图如图 3-1 所示。

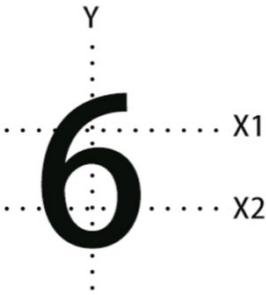


图 3-1 数字特征划分示意图

表 3-1 数字特征统计表

数字	与 Y 轴交点	与 X1 轴交点	与 X2 轴交点	类别
0	2	2	2	A
1	1	1	1	B
2	3	1	1	C
3	3	1	1	C
4	2	2	1	D
5	3	1	1	C
6	3	1	2	E
7	2	1	1	F
8	3	2	2	G
9	3	2	1	H

X	1	2	2	I
---	---	---	---	---

由上表可得，根据以上方法可将数字 0-9 以及字母 X 分为九类，可识别除数字 2、3、5 之外的所有字符。而对于数字 2、3、5，可以根据在 X1 和 X2 处的交点与 Y 的相对位置来进行判断，其关系如表 3-2 所示。

表 3-2 交点相对位置关系表

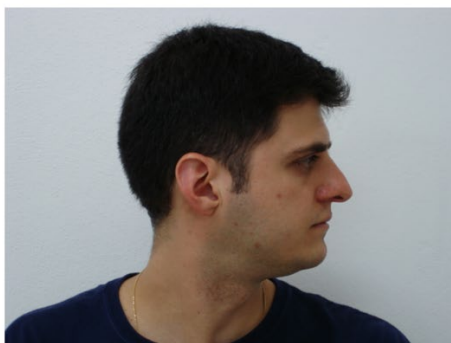
数字	X1 交点与 Y 关系	X2 交点与 Y 关系	类别
2	右侧	左侧	A
3	右侧	右侧	B
5	左侧	右侧	C

由上表可知，数字 2、3、5 的与线 Y 的交点关系互不相同，因此可通过该特征区别进行数字 2、3、5 的识别。

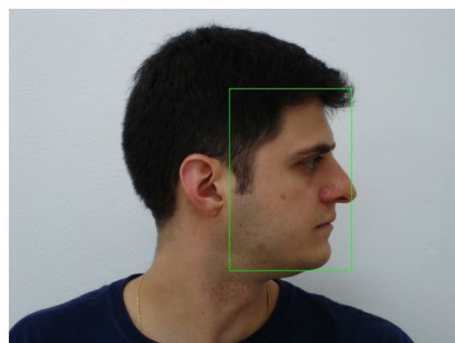
4. 人脸识别方案

4.1 人脸检测

实现人脸识别的首要工作是将所有图像样本进行人脸检测，即提取图像中的人脸区域，进行截取操作后传入人脸识别部分。OpenCV 中包含有基于 resnet10 和 SSD 模型的人脸检测预训练模型，在经过前期测试中，其识别精度和准确度较高，检测结果如图 4-1 中绿色矩形框区域所示，但在部署到 riscv 平台之后，其单张识别时间约为 40 秒-120 秒左右，原因可能在于网络模型过大，且输入图像为 3 通道，故不能满足快速检测的需求。



(a) s01_06.jpg



(b) detected.jpg

图 4-1 基于神经网络的检测结果

在观察给定人脸数据库图像特征之后，对图中左上坐标为(120, 70), 右下角坐标为(480, 390)的矩形区域进行了直接裁剪，之后进行灰度处理以提高运行速度，其输出结果如图 4-2 所示。

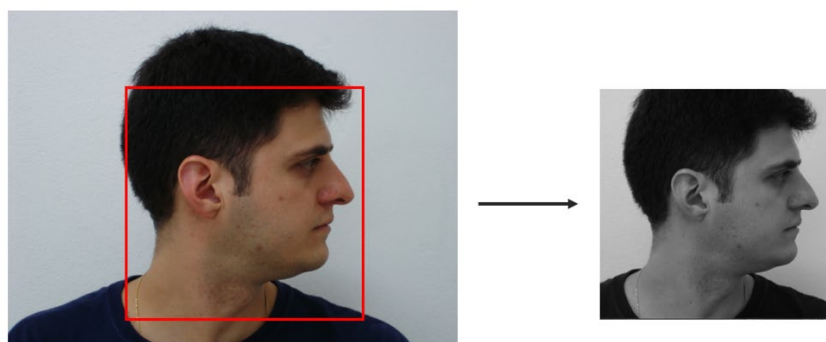


图 4-2 检测结果

4.2 人脸识别

本作品采用传统人脸识别算法——局部二值模式，构建了人脸识别系统，系统的输入为包含 20 张人脸的数据库和一张待检测目标人脸，输出为包含目标人脸序号的文本文件。

局部二值模式是一种用来描述图像局部纹理特征的算子，它具有速度快、旋转不变性和灰度级不变性等优点。首先由 T. Ojala, M. Pietikäinen, 和 D. Harwood 在 1994 年提出，用于图像局部纹理特征的提取。原始的 LBP 算子定义在像素 3×3 的邻域内，以邻域中心像素为阈值，相邻的 8 个像素的灰度值与邻域中心的像素值进行比较，若周围像素大于中心像素值，则该像素点的位置被标记为 1，否则为 0。如图 4-3，以 5 为中心的 3×3 邻域内的 8 个点经过比较可产生 8 位二进制数，将这 8 位二进制数依顺时针顺序排列形成一个二进制数字，这个二进制数字就是中心像素的 LBP 值，LBP 值共有 256 种可能。中心像素的 LBP 值反映了该像素周围区域的纹理信息。

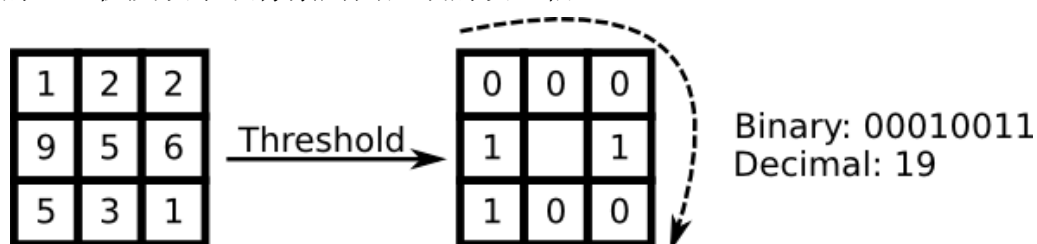


图 4-3 LBP 特征

LBP 特征算子可用公式化参数表示为：

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

其中 (x_c, y_c) 为中心点坐标， i_c 为中心点数值， i_n 为其周围像素点数值， $s(x)$ 为阶跃函数。如图 4-4，以该算子为基础，可进行图像中斑点、角点、边缘等的检测。

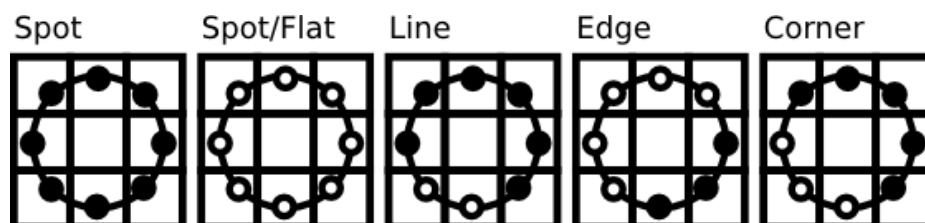


图 4-4 局部特征检测

在 OpenCV 的 Face 模块中，内置有基于该算法的人脸特征描述功能，我们基于 LBP 特征，使用给定 20 张图像训练人脸模型，其识别结果如图 4-5 所示。可见目标人脸检测结果与数据库中编号一致。在 riscv 平台上经多组测试后，其运行时间大约为 1 分钟左右，且准确度均为 100%。



图 4-5 识别结果

5. 总结与后续工作

5.1 阶段总结

目前阶段，基于 RISC-V 的片上系统构建已完成，并成功完成了身份证数字识别算法的构建与部署。片上系统能够实现外设调用、图像显示、ftp、图像处理等功能。数字识别算法在测试环境中取得了较高的准确率和处理速度。

所有的成果已发布至本队 Github 仓库：<https://github.com/Loow-Teech>

仓库项目对应列表:

1. lowrisc-chip: 增加外设后的 LowRISC
2. human-face-identification-on-RISC-V-arch: 人脸识别项目包
3. number-identification-on-RISC-V-architecture: 身份证号识别项目包
4. 8bit-PMOD-logic-output: 使用 Instant SoC 实现的 PMOD 输出和数码管显示(初赛成果)
5. libopencv-riscv: 交叉编译的 OpenCV 库

参考文献

- [1] FPGA Ethernet Cores, Connect your FPGA[EB/OL]. <https://www.fpga-cores.com>.
- [2] Zeynab Mohseni, Pedro Reviriego. Reliability characterization and activity analysis of lowRISC internal modules against single event upsets using fault injection and RTL simulation[J]. Microprocessors and Microsystems, 2019, 71.
- [3] 魏子奇. 基于 FPGA 的数字识别系统的研究与实现[D]. 东华大学,2016.
- [4] 高振斌,赵盼,王霞,陈洪波.印刷体数字识别系统的 FPGA 实现[J]. 重庆邮电大学学报(自然科学版), 2015, 27(02):213-218.
- [5] Robert Laganriere.OpenCV 计算机视觉编程攻略(第 3 版)[M]. 北京: 人民有带出版社, 2018.