

PYNQ 实现基于 OpenCV 的人脸识别

一、实验目的

1. 了解 PYNQ 的开发过程
2. 熟悉用 Vidado 等软件实现 HLS 硬件加速的流程

二、实验设备

1. MicroPhase Mizar Z7010 开发板；
2. HYUNDAI HYS-005 摄像头
3. 笔记本电脑

三、实验原理

HLS 是一种将高级语言描述的算法转换为硬件描述语言的自动化过程。通过使用 HLS 工具，可以实现对特定算法的硬件加速，从而提高其执行速度和性能。

HLS 加速的原理可以概括为以下几个步骤：

1. 算法描述：使用高级语言将要加速算法的功能、数据流和控制流等方面进行描述。
2. 优化和分析：使用 HLS 工具对算法进行分析，识别和优化计算密集型的代码部分，优化数据传输和存储，减少资源占用，提高性能和资源利用效率。
3. 控制流转换：HLS 工具将高级语言描述的控制流转换为硬

件描述语言。

4. 数据流转换：HLS 工具将高级语言描述的数据流转换为硬件数据路径和处理单元。

5. 综合和优化：HLS 工具将硬件描述语言综合为逻辑门级电路，并对电路进行优化，减少逻辑延迟、最小化资源使用和增加运行速度。

6. 与处理器系统集成：生成的硬件模块可以与处理器系统集成，这样就可以通过处理器与硬件模块之间的接口进行通信协作，实现复杂的功能。

HLS 加速的关键在于高级语言级别上对算法进行优化和转换，从而可以更轻松地对硬件加速器进行设计和实现。它提供了一种更高级别的抽象，使开发人员可以专注于算法的功能和性能，而不需要深入了解底层的硬件实现细节。可以根据具体的硬件平台和设计需求，选择合适的优化策略和技术，以达到最佳的性能和资源利用效率。

总之，HLS 加速的原理是：通过将高级语言描述的算法进行优化、转换和综合，并与处理器系统集成，以实现硬件加速器的设计和实现。

四、实验步骤

1. 在 PYNQ 上使用 Vivado HLS 来加速人脸识别任务，步骤操作如下：

- (1) 定义并导出加速的函数：根据人脸识别算法，确定可以使用硬件加速的部分。将这部分代码提取到一个单独的函数中，并使用 Vivado HLS 进行优化和导出。
- (2) 创建加速器 IP：使用 Vivado HLS 工具创建加速器 IP，该 IP 包含加速函数的硬件描述。
- (3) 在 PYNQ 项目中包含 IP 核：将生成的加速器 IP 核文件添加到 PYNQ 项目中。
- (4) 设计板级连接：使用 PYNQ 的 Overlay 功能，在 Python 环境中创建一个 Overlay 对象，并将加速器 IP 核实例化为一个硬件加速器。
- (5) 在 Jupyter Notebook 中修改人脸识别代码，使用加速器进行硬件加速。
- (6) 运行代码和评估性能：执行修改后的代码，并使用 PYNQ 和加速器进行人脸识别任务。使用计时器具来比较加速前后的性能提升。

2. 使用 PYNQ 和 Jupyter Notebook 的组合可以为人脸识别任务提供交互式 and 可视化的环境。实验步骤如下：

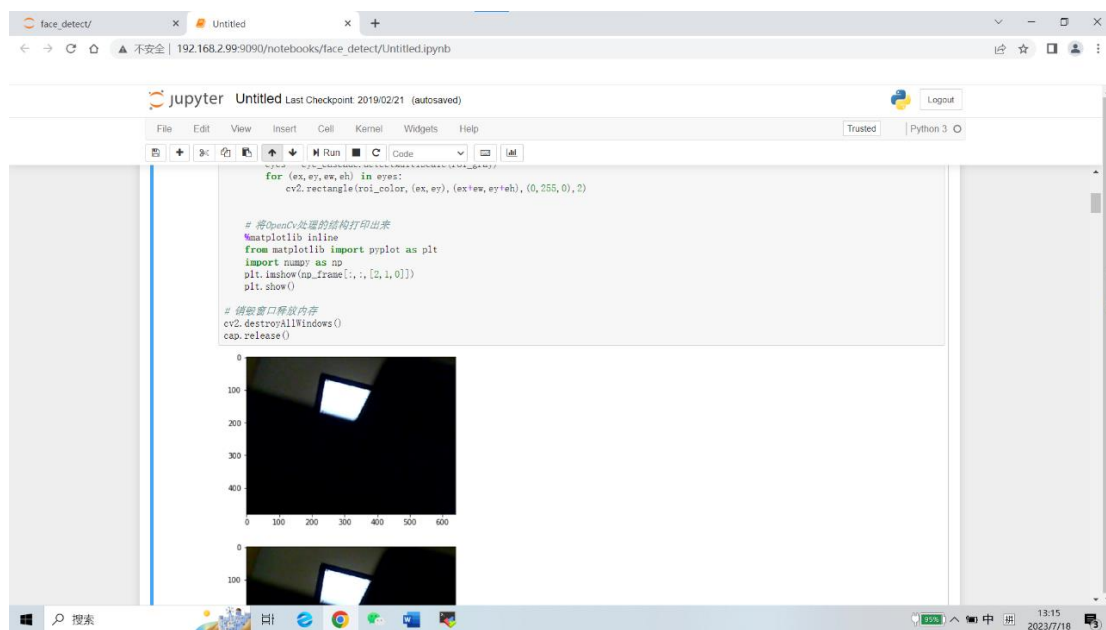
- (1) 设置开发板：确保 PYNQ 开发板已连接到电脑，通过网页浏览器访问 PYNQ 的 Jupyter Notebook 界面。
- (2) 创建新的 Jupyter Notebook：在 Jupyter Notebook 界面上创建一个新的 Notebook 文件，

并命名。

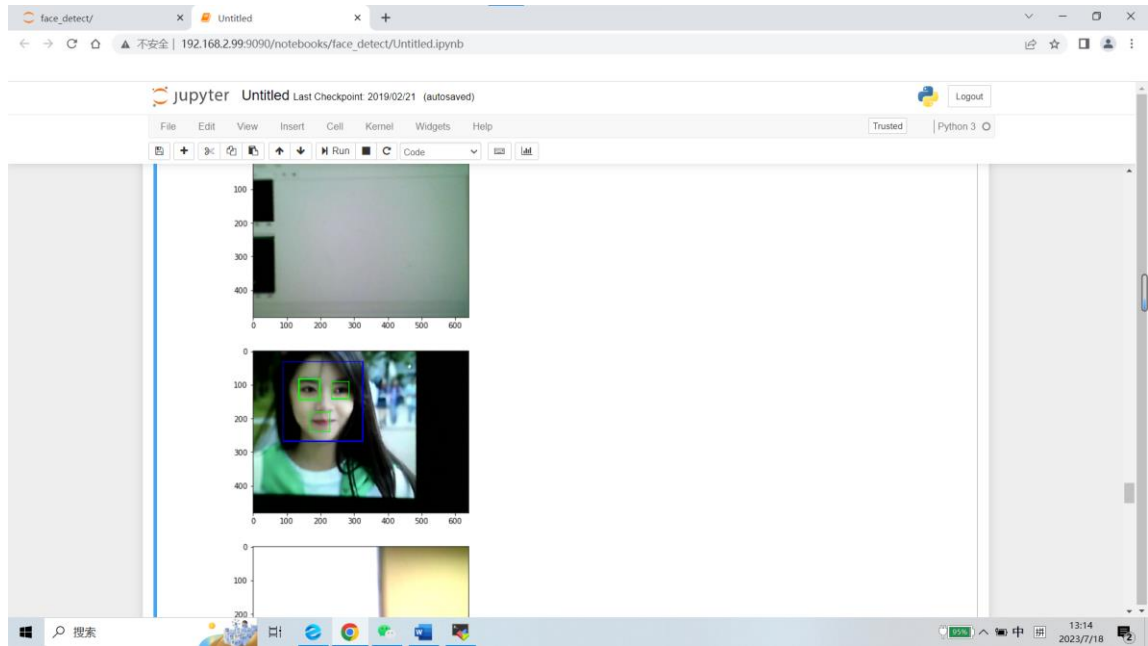
- (3) 导入需要的库：在 Notebook 的第一个代码单元格中，导入所需的 Python 库，包括 OpenCV。
- (4) 加载人脸识别模型和数据：根据人脸识别模型，从 PYNQ 论坛下载预训练模型和必要的文件，在代码单元格中加载模型和数据。
- (5) 开启摄像头并进行人脸识别：在代码单元格中编写代码，打开摄像头并应用人脸识别算法。包括获取输入图像、进行人脸检测、通过识别模型进行人脸识别、绘制人脸框和结果。
- (6) 执行代码：观察识别结果。

五、仿真结果

1. 程序开始运行，摄像头启动



2. 检测出稳定的人脸，并选中



3. 更换照片，仍然能够较好的检测到人脸

