Computer Science and Application 计算机科学与应用, 2023, 13(3), 601-607
Published Online March 2023 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2023.133059

Hans汉斯

CNN手写数字识别系统的ZYNQ实现

李　晓，高树静

青岛科技大学信息科学技术学院，山东　青岛

收稿日期：2023年2月23日；录用日期：2023年3月24日；发布日期：2023年3月31日

摘　要

针对卷积神经网络手写数字识别系统在软件平台上运行速度慢，功耗高的问题，同时更好地满足便携性的需求，利用FPGA并行计算的优势，在ZYNQ平台的逻辑端对卷积神经网络中的卷积层和池化层进行硬件加速，使用MT9V034摄像头采集图像通过屏幕实时显示识别后的数字。与软件平台相比较，处理一帧图像的速度提高了至少178倍，综合后的总片上功耗为1.969 W，逻辑资源的使用量为9.260 K，实现了对手写数字的低功耗快速识别。

关键词

手写数字识别，硬件加速，软硬件协同设计

CNN Handwritten Digit Recognition System ZYNQ Implementation

Xiao Li, Shujing Gao

College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao Shandong

Received: Feb. 23rd, 2023; accepted: Mar. 24th, 2023; published: Mar. 31st, 2023

Abstract

In order to solve the problems of slow running speed and high power consumption of the convolutional neural network handwritten digit recognition system on the software platform, and to better meet the needs of portability, this paper uses the advantages of FPGA parallel computing to accelerate the convolutional layer and pooling layer in the convolutional neural network on the logical end of ZYNQ platform. MT9V034 camera is used to collect images and display the recognized numbers in real time through the screen. Compared with the software platform, the speed of processing a frame image is increased by at least 178 times, the total on-chip power consumption is 1.969 W, and the usage of logical resources is 9.260 K, realizing the low power consumption and

文章引用: 李晓, 高树静. CNN 手写数字识别系统的 ZYNQ 实现[J]. 计算机科学与应用, 2023, 13(3): 601-607.
DOI: 10.12677/csa.2023.133059

fast recognition of hand written digits.

Keywords

Handwritten Digit Recognition, Hardware Acceleration, Soft and Hard Collaboration

1. 引言

随着人工智能以及大数据的发展，卷积神经网络在语音识别、图像处理、自然语言处理等领域取得了突破性的进展[1] [2] [3] [4] [5]。目前，手写数字识别技术在 CPU 和 GPU 平台上已经实现了部署，但是随着卷积神网络的模型层数以及参数数量增加、模型精度的提高，传统的 CPU 和 GPU 已经无法满足对手写数字识别实时性的要求，因此如何提高计算速度，是目前的热门研究方向。

近些年来，采用高层次综合语言 HLS 加速的方法[6]得到了快速发展，虽然降低了开发难度但是也存在综合后面积大，具体电路无法修改的问题。在软件平台，采用 ARM 处理器(Advanced RISC Machines)的嵌入式平台对卷积神经网络进行处理虽然体积小，功耗低，灵活性高，方便部署和迁移[7]，对硬件相关知识要求相对较低，但是传统 ARM 处理器算力低，在一些应用场景中不能满足实时性的要求。

本文采用赛灵思 zynq7010 系列开发板设计了基于卷积神经网络的手写数字识别系统，采用 MT9V034 摄像头实时采集图像，将卷积神经网络中浮点数运算以及串行计算慢等问题，利用 FPGA 并行加速的优势，采用流水线处理的方法对卷积神经网络的卷积层和池化层进行硬件加速设计，将设计后的 IP 核通过 AXI 总线与 PS 端交互，最后在屏幕上实时显示识别后的数字。

2. 本文 CNN 模型

本文所采用的卷积神经网络模型结构如图 1 所示。该结构一共有 7 层，其中第一层和最后一层分别为输入层和输出层，输入层通过采用 MT9V034 摄像头实时采集的图像作为测试图像，输出层为检测到的识别后的数字并且通过 HDMI 显示器输出结果。第二四层为卷积层，步长为 1 填充为 0，采用 5 × 5 卷积核，第二层卷积核的通道为 1 卷积核的个数为 6，第四层的卷积核通道为 6 卷积核的个数为 16。第三五层为池化层，步长为 2 填充为 0，采用最大池化算子。

卷积神经网络模型的运算大部分集中在卷积层[8]。所以对卷积神经网络的加速主要是针对卷积层和池化层进行加速。

3. 系统设计

3.1. 硬件环境

本文选用赛灵思 ZYNQ7010 平台，由两个部分组成其中可编程逻辑(Programable Logic)即 FPGA 部分包含：逻辑单元(logic cells) 23k、存储单元(block ram) 2.1Mbit、寄存器 35k；处理系统(processing system)由一个双核 ARM Cortex-A9 处理器组成最大频率为 666Mhz。Zynq-7000 系列平台可以满足复杂嵌入式系统的高新能、低功耗和多核处理能力等要求。

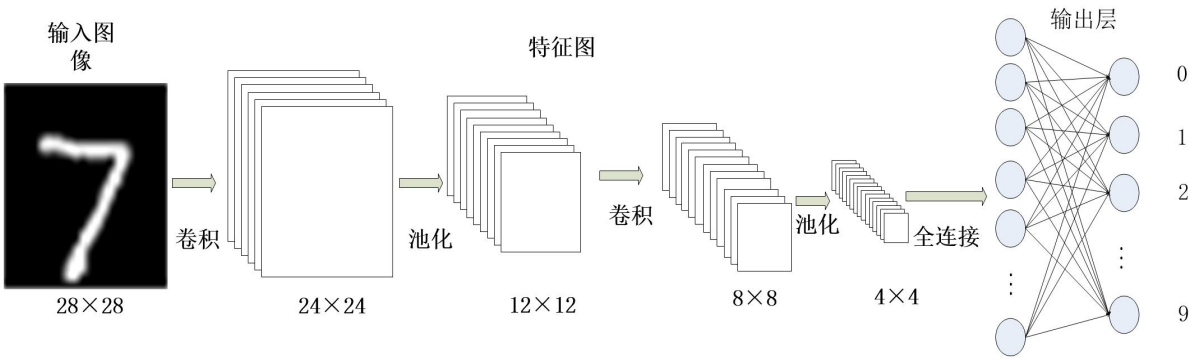

Figure 1. Convolutional neural network model is used in this paper
图 1. 本文所用卷积神经网络模型

3.2. 总体系统架构

系统总体设计如图 2 所示，由 MT9V034 摄像头完成图像的采集，采集的图像为灰度图像分辨率为 640 × 480 深度为 8 bit 并将数据传输到 ZYNQ，PL 部分接收摄像头采集到的灰度图像，并将其进行二值化处理，然后将处理后的图像数据传输到卷积和池化模块，利用 FPGA 的并行计算特性进行硬件加速。将处理后的数据通过 VDMA 传输到 PS 端口，通过 PS 端将数据缓存到 DDR3 模块，之后将图像数据读出使用 PS 端进行隐藏层和全连接层的数据处理，最后写回 VDMA 通过 PL 端的 HDMI 模块驱动显示屏，将识别后的手写数字通过显示器显示。

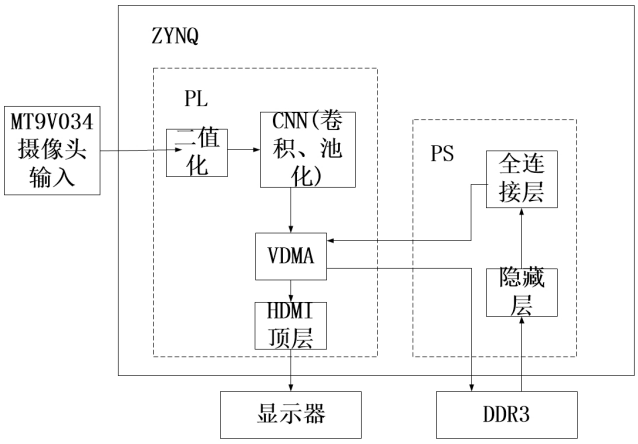

Figure 2. Overall system architecture
图 2. 总体系统架构

3.3. 卷积层硬件加速设计

卷积层是整个卷积神经网络中计算量最多的一层，因此对卷积层的加速尤为重要。卷积层主要包含输入数据缓存模块、权重参数模块以及乘法和加法模块。数据缓存模块主要用来存储 28 × 28 的输入图像数据，权重参数模块主要是将 5 × 5 的卷积核的参数通过 coe 文件存储到 ROM 中用来做卷积运算，乘法和加法模块通过读取输入的缓存数据以及权重数据进行加法和乘法运算。具体的实现流程如下：

1) 乘加模块分别读取数据缓存模块中的数据生成 5 × 5 的卷积滑窗和权重缓存模块中数据；

2) 将得到的 5 × 5 的卷积滑窗与卷积核做乘法运算得到 25 个值；

3) 延迟一个时钟周期后将这 25 个数值以及偏置做加法运算得出卷积后的值。整个过程一共用了 25 个乘法器和 25 加法器，第二次的卷积运算可以通过分时复用乘法器和加法器来实现卷积功能减少资源的使用量。结构如图 3 所示。

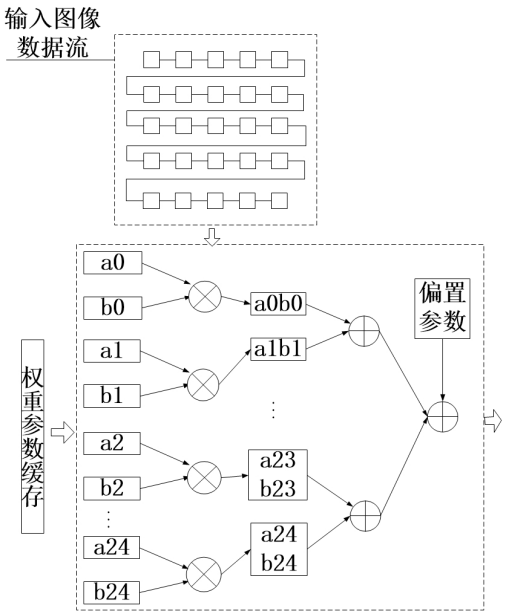

Figure 3. Hardware design of convolutional module
图 3. 卷积模块硬件设计

在硬件实现时，卷积滑窗的提取是关键环节，由于输入图像为尺寸 28 × 28，在提取 5 × 5 卷积滑窗时，可以采用流水线的方式提高计算速度。选取输入图像的行作为信号的标志，由于输入的是二值化图像，因此可以将输入图像做拼接处理，比如行标志为 1 时，将第一到第五行的每一列数据拼接在一起(图 4 绿色方框部分)，一共可以生成 28 个数据，共使用 28 的时钟周期，之后将提取的数据做打拍四拍。从第五个时钟周期就依次生成了所需要的 5 × 5 的卷积滑窗，具体波形如图 4 所示。乘加模块直接采用 25 个乘法器和加法器做并行运算，之后将得到数据与权重参数相加，整个流程用了 3 个时钟周期，提高了计算速度。

第二次的卷积运算可以通过分时复用乘法器和加法器来实现卷积功能，通过分时复用虽然会牺牲部分运算速度但是会减少资源的使用量，适合逻辑资源有限开发平台。

3.4. 池化层硬件加速设计

池化层采用的是最大池化算子，池化窗口的大小为 2 × 2，步长为 2，通过上一级卷积处理后第一次池化的输入尺寸大小为 24 × 24。池化层实现流程如下：

1) 将输入数据的第一行打一拍，把输入数据和延迟一拍后的数据在同一上升沿作比较，将大的数据写入 fifo；

2) 由于步长为 2，所以采用行列数据计数最后一位以及行有效标志信号作为写入 fifo 的标志位，即行数据最后一位为 1 列数据最后一位为 0 行有效标志信号为高时将比较得到的最大数 max1 写入 fifo；

3) 读取写入 fifo 的数据与第二行数据作比较取大值 max2，在下一个时钟周期时将 max2 与第二行数据中的下一个数据作比较取最大值 max。第三行到第二十四行的操作重复上面的步骤。

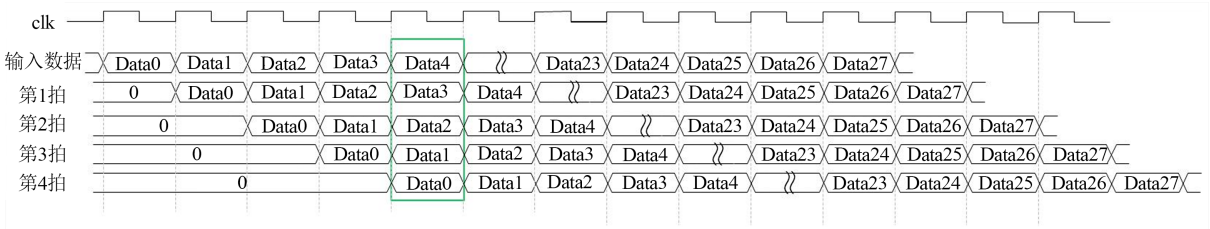

Figure 4. Waveform was extracted by 5 × 5convolution sliding window
图 4. 5 × 5 卷积滑窗提取波形图

池化层硬件实现比较简单，采用了三个比较器在4~5个时钟周期内就可以完成一个2 × 2区域的池化，在进行数据比较时要注意时序对齐，否则将会导致数据比较出错影响系统整体功能的实现。硬件设计框图，如图 5 所示。

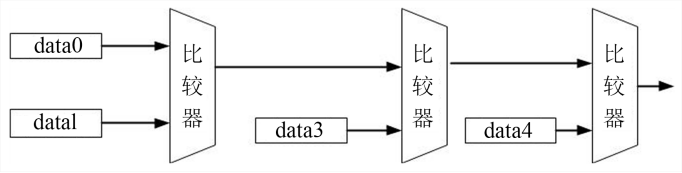

Figure 5. Pool layer block diagram
图 5. 池化层框图

4. 系统测试与分析

本次实验使用 VIVADO 2019.2 对卷积层和池化层进行设计并完成综合以及布局布线生成比特流，PS 部分是使用赛灵思 vitis 进行编程。采用赛灵思 XC7Z010CLG400-1 开发板，板卡上有最大数据采样率为 1006 Mbpa 的 512 MB 的 DDR3 SDRAM。

4.1. 检测结果

为了节约 FPGA 上的资源和解决 FPGA 不擅长浮点数运算的问题，本文将权值数据由浮点数字转化成 16 bit 的定点数，由文献[8]可知采用 16 bit 的定点数可以获得与单精度浮点数相近的运算结果。在 zynq 平台下摄像头采集手写数字图像经过二值化后如图 6 中左下角所示，系统正确的将手写数字进行了二值化处理，处理后的二值化图像经过卷积层、池化层、全连接层以及输出层的处理后将识别到的数字通过显示器显示在右上角(如图 6)，实验表明系统在 zynq 平台下快速准确识别到了手写数字。

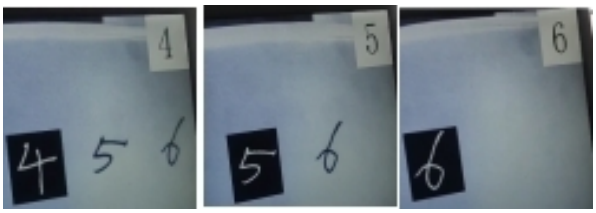

Figure 6. Real-time monitoring results
图 6. 实时监测结果

4.2. 资源及功耗分析

实验设计综合后，整体的资源使用情况如图 7 所示。结果显示 LUT、BRAM、MMCM 资源使用率

比较高。由第三节分析卷积层一共使用了 28 个缓存 ram，池化层使用了一个 fifo，卷积核权值的缓存以及检测到结果后用作显示识别的.coe 文件占用了比较多的储存资源从而导致了 BRAM 资源使用较多。MMCM 资源使用了 100%这是由于整个系统中需要用到 2 个不同的时钟，包括：摄像头采集时钟和 HDMI 显示时钟。乘加器采用的是 LUTRAM 资源所以 DSP 的资源使用率是 0，在 LUTRAM 资源有限的情况下可以通过 vivado 软件调用开发板 DSP 模块实现乘加功能来节省 LUTRAM 的资源。LUT 部分采用的是 6 输入查找表方式，摄像头驱动、硬件加速模块以及 HDMI 驱动模块的实现都会通过 LUT 来布局布线生成电路。LUT 一共使用了 9260 个，占总资源的 52.61%。触发器使用 8928 个，占用 25.36%这。IO 口等其他资源的使用率均未超过一半。通过以上分析可以得到系统的总体资源使用量较低，主要是由于采用分时复用等方法，实现了速度和面积的互换。

Resource	Utilization	Available	Utlization %
LUT	9260	17600	52.61
LUTRAM	346	6000	5.77
FF	8928	35200	25.36
BRAM	29	60	48.33
IO	21	100	21.00
BUFG	8	32	25.00
MMCM	2	2	100.00

Figure 7. Resource consumption
图 7. 资源消耗

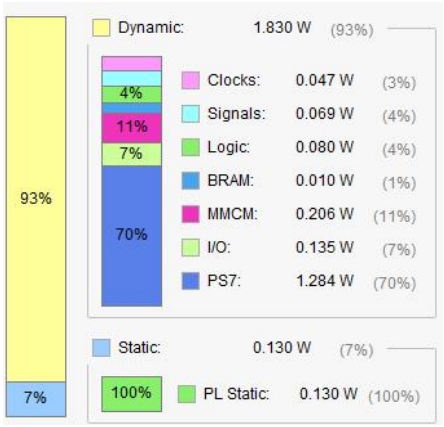

Figure 8. On-chip power consumption
图 8. 片上功耗

系统的功耗如图 8 所示，动态功耗占据系统功耗主要地位，其中静态功耗仅为 0.130 W，占系统总功耗的 7%，动态功耗为 1.830 W 占系统总功耗的 93%，总片上功耗为 1.969 W。动态功耗主要是由于逻辑门的开关导致，由图 8 可知动态功耗主要集中在 PS 端为 1.284 W，占动态功耗的 70%，这主要是因为 ARM 硬核工作频率高于 PL 端的工作频率。在 PL 端动态功耗主要集中在 MMCM 以及 IO 口，分别占动态功耗的 11%和 7%。

表 1 为与其他平台以及 FPGA 实现卷积神经网络的比较，通过仿真波形可以得到本文卷积神经网络加速模块的预测时间，将预测到的时间与其他的文献进项比较。

Table 1. Performance comparison under different platforms
表 1. 不同平台下性能对比

平台	预测时间	频率	LUT
ARM	0.041 s	666 MHz	---
文献[10]	0.047 s	100 MHz	23.258 K
文献[9]	20.23 us	200 MHz	21.235 K
本文	0.23 ms	100 MHz	9.260 K

文献[9]采用的是赛灵思 PYNQ 开发板，使用高层次综合语言 HLS 进行 IP 核的设计，与之相比本文模型总的使用时间为 0.23 ms 文献[9]的总时间为 20.3 us，但是本文采用的工作时钟频率为 100 MHZ 文献[9]使用的时钟频率为 200 MHZ，本文为了节省资源牺牲了部分速度换取资源，因此本文 LUT 资源使用量远少于文献[9]。在 ARM 平台下，采用 ARM Cortex-A9 处理器最大工作频率为 666 Mhz，由于 ARM 在计算数据时为串行工作方式，因此模型总的预测时间为 0.041 s 速度较慢，本文速度与之相比提高了 178 倍，说明经过 FPGA 并行处理速度得到了较大的提升。与文献[10]比较速度以及资源消耗都有明显的提高，这是由于本文所使用的网络模型和文献[10]有所不同。

5. 结语

本文在 zynq 平台下，完成卷积神经网络手写数字识别系统设计，重点介绍了卷积层和池化层硬件描述语言的实现方法。实验结论表明在软硬协同的设计下，识别速度比软件平台下分别提升了 178 倍，ZYNQ 平台下逻辑资源的使用量减少了一半以上，节约了硬件资源，减少了成本。

参考文献

[1] 刘晓阳. 基于忆阻的原位学习神经网络电路设计研究[D]: [硕士学位论文]. 武汉: 华中科技大学, 2021.

[2] Shawahna, A., Sait, S.M. and El-Maleh, A. (2019) FPGA-Based Accelerators of Deep Learning Networks for Learning and Classification: A Review. *IEEE Access*, **7**, 7823-7859. https://doi.org/10.1109/ACCESS.2018.2890150

[3] 吕浩, 张盛兵, 王佳, 刘硕, 景德胜. 卷积神经网络 SIP 微系统实现[J]. 计算机工程与应用, 2021, 57(5): 216-221.

[4] 齐延荣, 周夏冰, 李斌, 周清雷. 基于 FPGA 的 CNN 图像识别加速与优化[J]. 计算机科学, 2021, 48(4): 205-212.

[5] 刘之禹, 李述, 王英鹤. 基于 ZYNQ 的深度学习卷积神经网络加速平台设计[J]. 计算机测量与控制, 2022, 30(12): 264-269.

[6] 尹震宇, 徐光远, 张飞青, 徐福龙, 李兴滢. 面向 Zynq 平台的卷积神经网络单元设计与实现[J]. 小型微型计算机系统, 2022, 43(2): 231-235.

[7] Arredondo-Velazquez, M., Diaz-carmo-na, J., Barranco-Gutierrez, A.-I., *et al*. (2020) Review of Prominent Strategies for Mapping CNNs onto Embedded Systems. *IEEE Latin America Transactions*, **18**, 971-982. https://doi.org/10.1109/TLA.2020.9082927

[8] 焦李成, 孙其功. 深度卷积神经网络 FPGA 设计与实现[M]. 西安: 西安电子科技大学, 2020.

[9] 李慧. 手写体数字识别的卷积神经网络研究与 FPGA 实现[D]: [硕士学位论文]. 西安: 西安石油大学, 2021. https://doi.org/10.27400/d.cnki.gxasc.2021.000839 .

[10] 刘彬峰. 一种卷积神经网络加速电路的设计与 FPGA 实现[D]: [硕士学位论文]. 南京: 东南大学, 2019.