

# 轻量化 CNN 在 FPGA 上的手写字符识别系统设计

杨 洁

贵州航天控制技术有限公司 贵州 贵阳 550025

**【摘要】**：针对嵌入式边缘场景中手写字符识别系统存在的算力不足、功耗偏高及实时性不足等问题，本文提出一种基于轻量化卷积神经网络（CNN）与现场可编程门阵列（FPGA）协同实现的手写字符识别系统。通过对经典 CNN 模型进行结构精简与参数优化，采用卷积核压缩、冗余层裁剪及定点量化等方法，构建适用于硬件部署的轻量化模型；在此基础上结合 FPGA 并行计算与可重构特性，设计卷积、池化及激活等核心运算模块的流水线并行加速架构，实现高效推理计算。系统以 MNIST 手写数字数据集为测试对象，完成模型训练、硬件综合及板级验证。实验结果表明，该系统识别准确率可达 98.4%，单帧字符识别延迟低于 1.2ms，整体功耗为 3.6W，资源占用率适中。所提方法在保证识别精度的同时显著降低计算复杂度与功耗，适用于边缘端实时手写字符识别应用。

**【关键词】**：轻量化 CNN；FPGA；手写字符识别；硬件加速；定点量化

DOI:10.12417/2705-0998.26.04.028

## 1 引言

手写字符识别作为模式识别与计算机视觉领域的重要研究方向，在工业检测、人机交互及智能终端等场景中具有广泛应用。随着物联网与嵌入式技术的快速发展，相关应用逐步从云端部署转向边缘端落地，对识别系统的实时性、功耗、集成化及资源占用提出了更高要求。

传统手写字符识别多依赖 CPU 或 GPU 实现，其中 CPU 受限于串行计算，难以满足高实时性需求；GPU 虽具备强劲的算力，但其功耗过高、成本偏高，不适用于资源受限的嵌入式场景。因此，探索低功耗、高效率的硬件加速方案具有重要意义。

FPGA 凭借其并行计算能力强、功耗低及可重构性高等优势，逐渐成为边缘端深度学习推理加速的重要实现平台。但标准 CNN 模型存在参数量大、计算复杂度高的问题，直接部署于 FPGA 易导致资源占用过高及时序不收敛，因此必须对模型进行轻量化处理。

针对上述问题，本文基于经典 LeNet-5 模型，结合结构剪枝与定点量化技术，设计一种轻量化 CNN 模型；同时，基于 FMQL20S484 平台构建 CNN 硬件加速架构，实现卷积、池化及激活运算的并行流水线处理。通过算法与硬件的协同优化，在保证识别精度的前提下有效降低系统功耗与延迟。实验结果表明，所设计系统在性能与资源利用之间取得了良好平衡，具有较高的工程应用价值。

## 2 系统总体设计

本文设计的手写字符识别系统采用软硬件协同设计方法，整体架构由算法层与硬件实现层构成，算法层负责模型构建与优化，硬件层负责模型推理加速与系统实现，两者通过统一接口实现高效协同。

## 2.1 系统工作流程

系统工作流程分为四个阶段：第一阶段为数据输入与预处理，采集 28×28 尺寸的手写字符灰度图像，对原始数据进行归一化、二值化及去噪处理，以剔除图像干扰信息，并统一输入数据格式，从而适配轻量化 CNN 模型输入要求。第二阶段为轻量化 CNN 推理计算，将预处理后的图像数据输入优化后的神经网络模型，依次完成卷积特征提取、池化降维、全连接分类等核心运算，输出字符分类结果；第三阶段为 FPGA 硬件加速，利用 FPGA 并行计算能力，对卷积、池化及激活等核心运算进行硬件实现，压缩推理延时；第四阶段为通过外设接口输出识别结果，完成整个识别流程。

## 2.2 软硬件协同架构

系统基于 FMQL20S484 FPGA 平台实现，该芯片集成四核 ARM Cortex-A9 处理器（PS）与可编程逻辑单元（PL），兼顾软件控制与硬件加速能力。其中，ARM 处理系统负责数据采集、预处理调度、结果输出及外设控制等上层任务；可编程逻辑部分用于搭建神经网络硬件加速引擎，重点实现卷积、池化及激活等计算密集型操作。PS 与 PL 之间通过 AXI 总线进行数据交互，从而实现高效的数据流转与任务协同。

该架构充分发挥了 FPGA 在并行计算与低功耗方面的优势，同时结合处理器的灵活控制能力，实现了系统性能与资源利用的平衡。

## 3 轻量化 CNN 模型设计与优化

针对手写字符识别任务输入特征维度较低的特点，本文基于经典 LeNet-5 模型进行轻量化改进，结合参数剪枝、定点量化及结构精简等方法，构建适配 FPGA 部署的轻量化 CNN 模型，在保证识别精度的前提下，大幅降低参数量与计算量，有效降低模型复杂度。

### 3.1 基础模型结构改进

传统 LeNet-5 模型虽然结构相对简单,但在嵌入式硬件部署中仍存在参数冗余问题。为降低模型复杂度,本文对其进行针对性优化:首先,在卷积层设计中缩减卷积核数量,第一层卷积层采用 8 个卷积核,第二层卷积层采用 16 个卷积核,相较于标准模型减少 40%卷积核数量。其次,仅保留一层全连接层实现分类输出,并将神经元数量精简至 64 个。最后,采用 ReLU 激活函数替代传统 Sigmoid 函数,在避免梯度消失的同时简化硬件计算逻辑,提升运算速度。优化后的轻量化 CNN 模型结构如表 1 所示。

表 1 优化后的轻量化 CNN 模型结构

网络层类型	核心参数	输入尺寸	输出尺寸	参数量
输入层	单通道灰度图	28×28×1	28×28×1	0
卷积层 1	8 个 3×3 卷积核,步长 1, 填充 1	28×28×1	28×28×8	80
池化层 1	最大池化, 2×2 核, 步长 2	28×28×8	14×14×8	0
卷积层 2	16 个 3×3 卷积核,步长 1, 填充 1	14×14×8	14×14×16	1168
池化层 2	最大池化, 2×2 核, 步长 2	14×14×16	7×7×16	0
全连接层	64 个神经元	7×7×16	64	50240
输出层	Softmax 分类, 10 类字符	64	10	650

优化后的模型在保持特征提取能力的同时显著降低参数规模,模型总参数量仅 5.2 万左右,相较于标准 LeNet-5 模型减少 65%以上,计算量降低 70%,大幅减轻 FPGA 硬件部署的资源压力,同时保留核心特征提取能力,保证识别精度不受大幅影响。

### 3.2 模型量化与压缩

为适应 FPGA 对定点运算的硬件特性,本文将模型训练后的 32 位浮点型权重与偏置参数量化为 16 位定点数,通过缩放与偏移处理最大限度保留参数精度,避免量化误差导致识别准确率大幅下降。

量化后模型存储容量压缩 50%,数据位宽的减小有效提升了运算效率。同时,采用结构化剪枝技术,剔除模型中权重值趋近于零的冗余参数,进一步压缩模型体积,提升硬件推理的流畅性。

### 3.3 模型训练与验证

模型基于 TensorFlow 框架完成训练,选用 MNIST 手写数字数据集作为实验数据。该数据集包含 6 万张训练样本与 1 万张测试样本,均为 28×28 单通道灰度图。

训练过程采用 Adam 算法,设置初始学习率为 0.001,批次大小为 32,训练轮次设置为 20 轮,以避免模型过拟合。训练完成后,模型在测试集上的识别准确率达到 98.6%;经过量化与剪枝后,准确率为 98.4%,仅下降 0.2%,满足实际应用的精度要求。

上述结果表明,所设计的轻量化模型在降低复杂度的同时仍具备较高识别精度。

## 4 FPGA 硬件加速系统设计

本文在国产 FPGA 平台 FMQL20S484 上实现硬件加速系统设计。通过对 CNN 推理计算进行实现卷积、池化、激活等核心运算的硬件加速。系统由数据调度、卷积运算、池化激活与存储控制四个模块组成,通过片上总线构成高效数据流处理通路。

### 4.1 核心运算模块设计

卷积运算是 CNN 的核心,占总计算量 80%以上。本文采用并行卷积架构,在 FPGA 中构建多卷积核并行处理单元,实现单时钟周期的通道卷积运算。在数据访问上,引入行缓存技术实现输入特征图的数据复用,减少片外访存次数与带宽压力;卷积累加采用加法树结构,提升计算速度与时序性能。池化模块采用最大池化,通过比较运算实现特征降维,硬件开销低,可与卷积模块同步运行。激活模块采用 ReLU 函数,仅需符号判断与数据选择逻辑,将负值置零、正值直通,简化硬件实现。

各模块通过流水线结构连接,使卷积、池化与激活并行执行,显著提高整体吞吐率。

### 4.2 存储与数据调度设计

FPGA 片内存储资源有限,系统采用片内 BRAM 与片外 DDR 协同存储架构。片内 BRAM 用于缓存输入特征图、卷积核权重与中间运算结果,保证高速访问;片外 DDR 用于存储完整模型参数与批量输入数据,提供大容量支撑。

数据调度模块通过 AXI4 总线接口实现 PS 与 PL 端的数据交互,采用突发传输模式提升效率,并优化数据存储顺序以适配卷积运算的数据流特性,减少数据冲突与等待。

此外,在流水线执行过程中,通过合理设计数据缓存策略

与读写控制逻辑,实现计算与数据传输的重叠,提高系统吞吐量,保证硬件加速模块持续高效运行。

## 5 系统测试与结果分析

为验证系统的整体性能,从识别精度、硬件资源占用、实时性及功耗四个维度进行测试。搭建基于 FMQL20S484 开发板的测试平台,完成模型部署与板级验证,测试环境为 Vivado 2018.3 开发工具,时钟频率设置为 100MHz。

### 5.1 硬件资源占用测试

系统硬件资源占用情况是衡量硬件架构合理性的核心指标,本次测试通过 Vivado 工具综合报告,统计 FMQL20S484 内各类型硬件资源的占用情况,具体结果如表 2 所示。

表 2 FPGA 硬件资源占用情况

硬件资源类型	总资源数量	占用数量	占用率
查找表 (LUT)	53200	19790	37.2%
触发器 (FF)	106400	34620	32.5%
DSP 单元	220	63	28.6%
BRAM 存储单元	140	58	41.3%

测试数据表明,系统硬件资源占用合理,未出现资源过载或时序违例问题。LUT、FF、DSP、BRAM 的占用率分别为 37.2%、32.5%、28.6%和 41.3%。DSP 占用率远优于未优化的 CNN 部署方案,验证了模型轻量化与硬件架构适配的有效性;BRAM 的合理分配既满足了特征图与权重的缓存需求,又保留了中间结果存储空间,保障了流水线运算的连续性。

### 5.2 识别性能与实时性测试

在 MNIST 测试集上选用 1 万张样本进行识别测试,系统最终识别准确率为 98.4%,与软件端量化后模型精度基本一致,

表明硬件实现未引入明显精度损失,验证了硬件加速的可靠性。

实时性测试结果显示,单张 28×28 手写字符图像的识别延迟为 1.18ms,对应处理速度约为 847 帧/s。对比传统 CPU 实现方案,本系统识别速度提升 12 倍,功耗降低 85%,体现出 FPGA 并行计算架构的显著优势。

### 5.3 功耗测试

系统稳定运行时,通过开发板功耗检测模块测得整体功耗为 3.6W,远低于 GPU 方案的 50W 以上功耗,也低于传统嵌入式 ARM 方案的 6-8W 功耗,具备极强的低功耗优势。结合国产 FPGA 平台的低功耗特性,所设计系统在满足实时性的同时具备良好的能效比,适用于对功耗敏感的边缘计算场景。

### 5.4 对比分析

与现有手写字符识别方案相比,本文系统在在识别精度、实时性、功耗、资源占用四个维度实现了均衡优化,既保证了高精度识别,又满足边缘端低功耗、高实时性的要求,综合性能优于传统软件方案与未优化的硬件部署方案,具备更强的工程实用性。

## 6 结论

本文设计并实现了一种基于轻量化 CNN 与国产 FPGA 平台协同工作的手写字符识别系统。通过对神经网络模型进行结构优化与定点量化处理,并结合 FPGA 并行计算架构,实现了高精度、低延迟、低功耗的手写字符识别。实验结果表明,该系统的识别准确率达 98.4%,单帧识别延迟低于 1.2ms,整体功耗仅 3.6W,资源占用适中,完全满足嵌入式边缘场景的应用需求。

此外,本文基于国产 FPGA 平台完成系统实现,对提升嵌入式智能系统的自主可控能力具有一定参考意义。后续可从三方面优化:拓展识别类型至手写字母与简单汉字,提升系统适用性;采用更先进的量化技术与并行计算策略,降低资源占用与识别延迟;集成图像采集外设,实现端到端的一体化手写字符识别终端,拓宽应用场景。

## 参考文献:

- [1] 李珍琪,王强.轻量化卷积神经网络硬件加速设计及 FPGA 实现[J].计算机工程与科学,2024,46(03):421-428.
- [2] 李宏伟.基于 FPGA 的手写数字识别系统设计[J].电子技术应用,2023,49(07):89-93.
- [3] 李阳.轻量化 CNN 模型优化及嵌入式硬件部署研究[D].哈尔滨工业大学,2023.
- [4] 李志强.基于 ZYNQ 的神经网络硬件加速系统设计[J].微电子学与计算机,2022,39(11):76-82.
- [5] 李雪梅.嵌入式边缘端低功耗深度学习推理系统设计[J].计算机工程与应用,2024,60(02):231-237.