引用本文：孔一卜，黎海文，曾庆辉，等. 面向调制格式识别的稀疏CNN FPGA加速器设计[J]. 光通信技术，2026，50(2)：103-108.

面向调制格式识别的稀疏CNN FPGA加速器设计

孔一卜[1]，黎海文[2]，曾庆辉[2]，陆 叶[2,3*]

（1. 中国电子科技集团公司 第三十四研究所，广西 桂林 541004；
2. 广西师范大学 电子与信息工程学院，广西 桂林 541004；3. 广西高校光电信息技术工程研究中心，广西 桂林 541004）

摘要：为在逻辑与功耗资源受限的嵌入式场景中实现高效的调制格式识别，设计了一种面向调制格式识别的稀疏卷积神经网络(CNN)现场可编程门阵列(FPGA)的加速器。首先对基准CNN模型进行非结构化剪枝、8比特动态定点量化与层融合，显著压缩模型规模；随后设计基于ABM-SpConv算法的硬件加速架构，采用权重编码与单写多读缓存结构，优化并行卷积与数据访问效率。实验结果表明：在XC7A200 FPGA平台上，该设计以1.455 W的片上功耗实现90.2%的平均识别精度，每帧处理时间为142.48 μs，能效比达0.232 GOP/(s·W^{-1})，优于同任务下的中央处理器(CPU)与图形处理器(GPU)平台。

关键词：卷积神经网络；加速器；现场可编程门阵列；调制格式识别

中图分类号：TN29　**文献标志码：**A　**文章编号：**1002-5561(2026)02-0103-06

DOI：10.13921/j.cnki.issn1002-5561.2026.02.017

Design of sparse CNN FPGA accelerator for modulation format recognition

KONG Yibu[1], LI Haiwen[2], ZENG Qinghui[2], LU Ye[2,3*]

（1. The 34th Research Institute of CETC, Guilin Guangxi 541004, China;
2. School of Electronics and Information Engineering, Guangxi Normal University, Guilin Guangxi 541004, China;
3. Guangxi Colleges and Universities Optoelectronic Information Technology Engineering Research Center, Guilin Guangxi 541004, China）

Abstract: To achieve efficient modulation format recognition in embedded scenarios with limited logic and power resources, a sparse convolutional neural network（CNN） field-programmable gate array（FPGA） accelerator designed for modulation format recognition is presented. First, the baseline CNN model undergoes unstructured pruning, 8-bit dynamic fixed-point quantization, and layer fusion, significantly compressing the model size. Subsequently, a hardware acceleration architecture based on the ABM-SpConv algorithm is designed, employing weight re-encoding and a single-write multi-read cache structure to optimize parallel convolution and data access efficiency. Experimental results show that on the XC7A200 FPGA platform, the design achieves an average recognition accuracy of 90.2% with an on-chip power consumption of 1.455 W, a per-frame processing time of 142.48 μs, and an energy efficiency ratio of 0.232 GOP/(s·W^{-1}), outperforming central processing unit（CPU） and graphics processing unit（GPU） platforms for the same task. This provides a feasible path for deploying modulation format recognition in resource-constrained environments.

Key words: convolutional neural network, accelerator, field-programmable gate array, modulation format recognition

0 引言

光通信网络呈现出高速、异构与动态化的发展趋势，对光性能监测(OPM)模块提出了快速生成与灵活部署的要求。在数字信号处理中，调制格式信息的准确识别直接影响光纤传输系统的性能，因此，在不依赖发射端先验信息的前提下实现调制格式的自主识别，已成为OPM中的关键技术之一[1,2]。然而，传统OPM方法因依赖专用硬件，难以适应动态光网络的需求[3]。

与此同时，图像分类与识别作为人工智能领域的重要研究方向，已广泛应用于各类计算平台。卷积神经网络(CNN)作为其中的核心算法，在图形处理器

收稿日期：2026-01-31。

基金项目：中央引导地方科技发展资金项目(桂科ZY24212030)资助。

作者简介：孔一卜(1990—)，男，河南洛阳人，硕士，高级工程师。主要研究方向为数字光传输系统设计、IP分组交换网络等，具有丰富的科研经验与丰硕的研究成果。

***通信作者：**陆叶(1989—)，男，广西钦州人，广西师范大学高级工程师，主要研究方向为光通信技术。

（GPU）与中央处理器（CPU）等平台上取得了显著成果[3-5]。然而，随着网络模型深度的不断增加，其计算复杂度急剧上升，严重限制了CNN在资源受限场景下的部署能力。近年来，现场可编程门阵列（FPGA）因其具备高并行性、可配置数据精度以及低功耗等优势，在嵌入式与物联网设备中受到广泛关注[6-9]，成为实现高效、绿色计算的重要路径。尽管已有大量研究致力于FPGA上的CNN加速，但在调制格式识别这一特定领域中，相关硬件加速方案仍较为缺乏。

基于上述背景，本文结合CNN图像识别技术与FPGA硬件加速优势，设计一种面向调制格式识别的稀疏CNN FPGA加速器，通过剪枝与压缩技术减少模型规模与计算量[10-11]，旨在在资源受限环境下实现高能效、高识别率的实时信号处理。

1 调制格式识别与模型压缩

1.1 调制识别网络结构

调制格式识别网络结构如图1所示。识别网络结构共包含8层，即4个卷积层（C_1、C_2、C_3、C_4）、3个池化层（P_1、P_2、P_3）以及1个全连接层（FC）。该结构在识别精度和网络模型的大小方面取得了平衡。

卷积层的核心作用是进行局部特征探测。它通过滑动窗口与输入进行卷积运算，提取出信号中的初级特征，并组合成特征图。后续卷积层（C_2、C_3、C_4）则在此基础上，进一步提取调制样式本质的、更抽象的高级特征（如特定的星座图图案）。池化层主要进行下采样。它通过取局部区域最大值方式，压缩特征图的尺寸，减少了后续计算所需的参数量和计算量。C_4层输出的特征图随后被展平为一维特征向量。该向量被送入FC，以对提取的特征进行综合整合与判别，最终映射为各调制类别的概率分布P_{MFI}，从而完成识别任务。

1.2 模型剪枝与量化

本文采用软剪枝与非结构化剪枝相结合的方法对模型进行剪枝。剪枝过程主要分为3个阶段：第一阶段，对调制格式识别网络进行模型训练，得到一个非稀疏、全精度的基准网络；第二阶段，通过设定全局剪枝率，在基准模型上进行全局剪枝；第三阶段，对全局剪枝后的网络重新训练，以提高剪枝后模型的精度。

在CNN中，卷积层占大部分计算量，且权重通常以浮点数形式存储。为了在FPGA上高效部署网络，本文采用动态定点量化方案，使用8比特定点数值对网络中的权重参数进行量化，量化公式如下：

$$x_q=\begin{cases}-2^{B-1}, & x\leqslant -2^q\\ \mathrm{round}\left(x2^{F_{\mathrm{L}}}\right), & -2^q<x<2^q-1\\ 2^{B-1}, & x\geqslant 2^q-2^{-F_{\mathrm{L}}}\end{cases}\tag{1}$$

式中：q表示定点数的格式中整数部分的位数，$q=B-1-F_{\mathrm{L}}$，F_{L}表示十进制部分的长度，B表示数据的位宽；x表示权重位，round（·）表示四舍五入。例如，若权重值为2.5，采用8比特定点数值表示，且当前层小数部分长度为1，则$q=8-1-1=6$，可得2.5的8比特定点数值为round$(2.5\times 2)=5$。F_{L}的确定方式如下：

$$F_{\mathrm{L}}=B-1-\mathrm{ceil}\left(\log_2\max\left(W\right)\right)\tag{2}$$

式中：ceil（·）表示向上取整，max（·）表示求最大值函数，W表示每一层的权重值。

1.3 网络模型层融合

网络权重的量化消除了卷积层中的大部分浮点运算，但在其他层（如批量归一化层）中仍存在密集的浮点运算。批量归一化层在训练过程中能加快模型收敛、提高泛化能力并降低训练难度，但在前向推理过程中会增加计算延迟。为进一步压缩网络，本文采用层融合方法优化网络结构。融合后的权重W_{new}和数据位宽B_{new}分别表示为

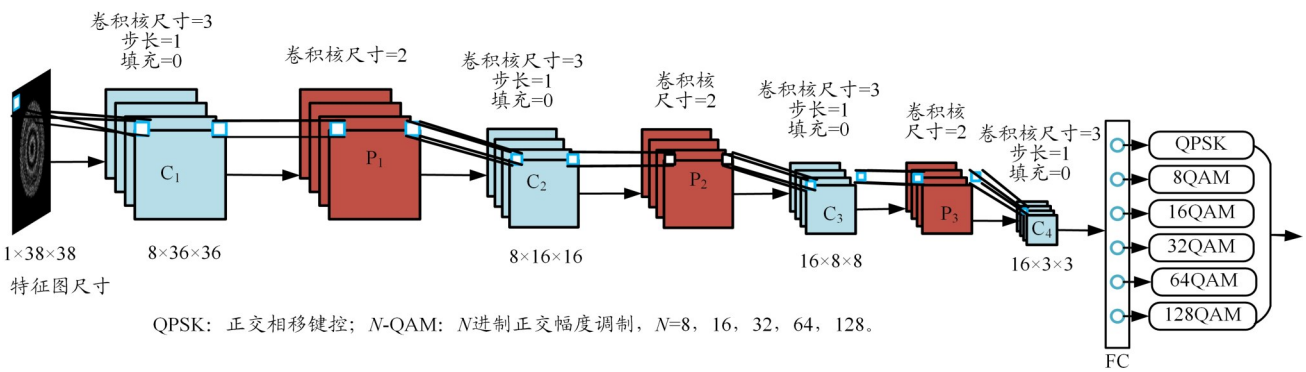

图1 调制格式识别网络结构

$$W_{new} = \frac{\gamma W}{\sqrt{\sigma^2 + \varepsilon}} \tag{3}$$

$$B_{new} = \frac{\gamma (B - \mu)}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{4}$$

式中：μ和σ^2是批量归一化的均值和方差，γ和β是比例参数和平移系数，ε是防止除零设置的一个极小值。

量化与层融合前，网络参数以浮点数形式存储，模型参数量较大，对内存带宽与存储空间提出较高需求。为此，本文采用层融合后的前向推理方法，示意图如图2所示。其核心包含2项关键技术：一是算子融合，将卷积、批归一化与激活函数层合并为单一计算内核，有效减少了内核启动开销及中间特征图的内存读写次数；二是离线量化，对网络权重、偏置及激活值进行低比特量化，使推理过程中的大部分浮点运算转化为整数运算，从而大幅降低计算复杂度与存储开销。二者协同作用，显著提升了卷积运算速度与整体推理效率。

1.4 压缩效果分析

模型剪枝与量化实验结果如表1所示。可以看出，调制格式识别网络经非结构化剪枝、8位动态定点量化及算子融合后，参数量最高压缩比达10.84，计算量最高压缩比达6.67，识别精度从91.6%降至90.2%（下降1.4%），精度损失在实际应用中可忽略不计。同时，网络中的浮点运算转换为8位定点运算，显著降低了存储开销并提升了推理效率。

当计算量压缩比从5.87提升至6.67时，识别精度从89.4%回升至90.2%。这主要得益于剪枝移除冗余参数、降低模型复杂度的正则化效应，结合微调策略优化了特征表示能力，使模型在更高压缩比下仍能保持甚至略微提升泛化性能。

表1 模型剪枝和量化结果

模型	参数量/10^3	计算量/（KOP）	参数量压缩比	计算量压缩比	识别精度/%
基准	29.7	336.64	1	1	91.6
剪枝（计算量压缩比为5）	8.10	67.33	3.70	5.0	90.0
剪枝（计算量压缩比为5.87）	6.91	57.23	4.30	5.87	89.4
剪枝（计算量压缩比为56.67）	6.08	50.47	4.88	6.67	90.2
量化+剪枝（计算量压缩比为5）	3.61	67.33	8.20	5.0	90.0
量化+剪枝（计算量压缩比为5.87）	3.19	57.23	9.58	5.87	89.2
量化+剪枝（计算量压缩比为6.67）	2.74	50.50	10.84	6.67	90.2

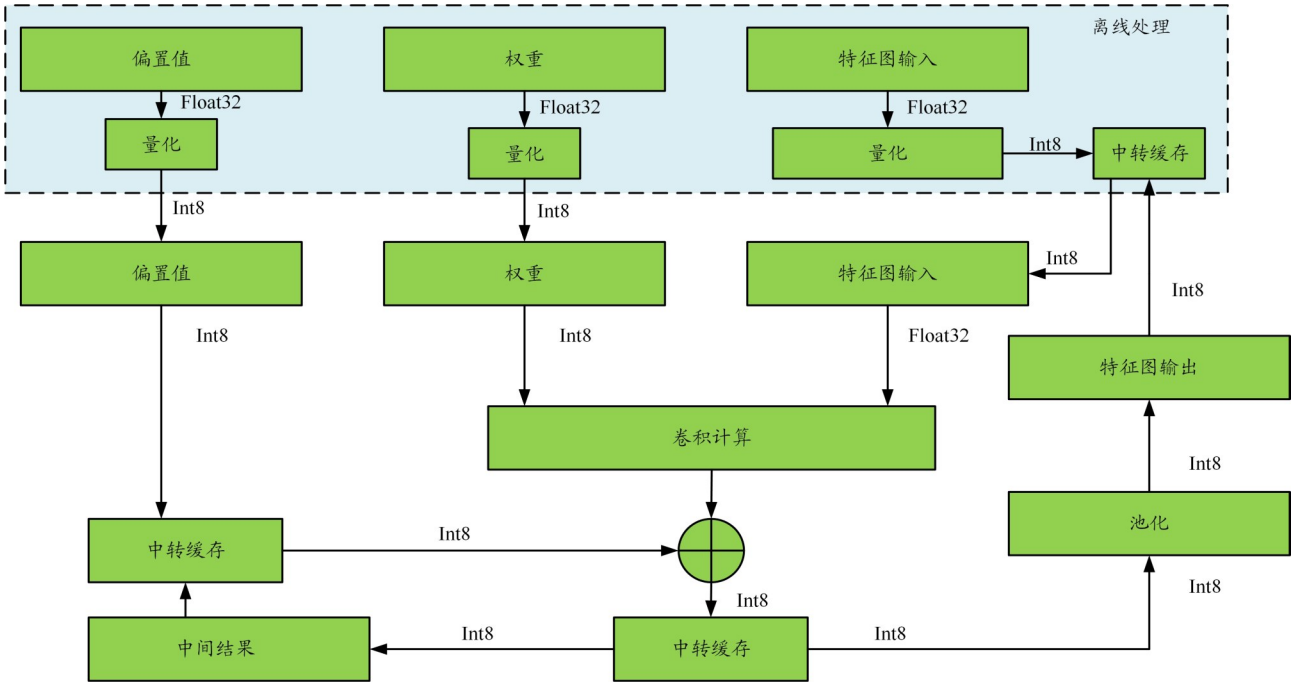

图2 层融合后的前向推理示意图

2 稀疏CNN FPGA加速器硬件设计

2.1 ABM-SpConv算法原理

CNN目标识别算法主要依赖卷积计算，其中包含大量乘法和加法运算。由于乘法运算耗时远高于加法运算，设计硬件加速电路的目标之一是尽量减少乘法运算的比例。ABM-SpConv是一种特殊的稀疏卷积算法，能有效平衡FPGA的逻辑资源和数字信号处理（DSP）资源使用。与其他算法不同，它将卷积运算中的加法和乘法分离为2个独立阶段，并利用卷积核中权重相同的特性，将乘法过程转化为加法过程。其公式推导如下：

$$F_{\mathrm{OUT}}=\sum_{n=0}^{N-1}\sum_{k=0}^{K-1}\sum_{k'=0}^{K-1}F_{\mathrm{IN}}W_{\mathrm{c}} \tag{5}$$

式中：n为输入特征图的通道数，k和k'分别卷积核的高度和宽度。

输入特征图像F_{IN}，卷积核W_{c}和输出特征图F_{OUT}的尺寸分别为$N\times R\times C$、$M\times N\times K\times K'$和$M\times R'\times C$。其中，$N$为输入特征图的通道数，$R$和$C$分别为输入特征图的宽度和高度，$M$为输出特征图的通道数、$K$和$K'$分别卷积核的宽度和高度，$R'$和$C'$分别为输出特征图的空间高度和宽度。经过剪枝和量化后，卷积网络在卷积核中保留了相同的权重，最多包含$Q\leqslant 2^{w}$个量化值，其中w代表量化位数。在第m个卷积核中仅存在Q个唯一的量化值，将这些唯一值表示为$\overline{W}_{p}$。提取公因式化简式(5)，可得

$$F_{\mathrm{OUT}}=\sum_{p=0}^{Q-1}\left(\overline{W}_{p}\sum_{w}I_{p}(w)\right) \tag{6}$$

式中：p为权重量化的数值，$I_{P}(w)$为输入特征图集合。

式(6)消除了卷积核中因权重相同而产生的冗余乘法运算，将原本耗时的乘累加操作转换为先加后乘运算，从算法层面优化了网络的前向推理时间。该公式是本文加速器硬件设计的核心算法基础。

2.2 硬件架构设计

神经网络经非结构化剪枝和8比特动态定点量化后，权重呈现显著的数据特征：一方面量化导致高频量化值聚集，另一方面剪枝引入大规模稀疏结构，这种双重特性降低了压缩存储效率。在CNN前向推理阶段，各层卷积核参数具有静态确定性，具体表现为稀疏结构固定、量化参数分布稳定等。这种先验知识使得在模型部署阶段可采用预编码方案，优化逻辑控制电路，减轻存储负担并提高数据读取频率。

权重编码方法采用索引表（Index_Table）和量化表（Q_Table）存储卷积核的全部相关信息：Index_Table记录权重的坐标信息，Q_Table存储权重的数值信息。该编码方法简化了电路查找结构，降低了存储需求，并有利于功耗最小化。

引入权重编码方案并结合ABM-SpConv算法设计的电路控制结构如图3所示。传统复杂的译码电路模块和权重缓存单元被量化和索引表取代，特征图的解码坐标由索引表提供，对应量化数值由量化表传递。

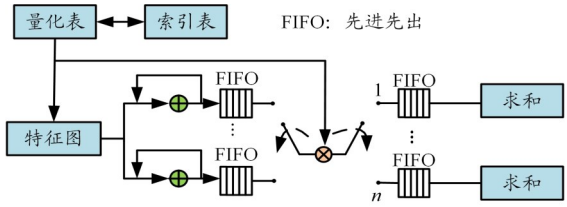

图3 ABM-SpConv算法电路控制图

2.3 并行计算与缓存优化

卷积层的运算本质是卷积核在输入特征图上滑动并进行卷积计算。本文利用FPGA的并行计算特性，在算法层面进行了微调设计，即通过在输入特征图纵向同时执行多个卷积操作，系统吞吐量得到进一步提升。图4为纵向并行优化后的“先加后乘”算法控制图，其在图3基础上引入地址变换模块，使得纵向卷积仅需在原有地址基础上增加一个固定偏移量即可获得特征图输出。

在列方向上卷积并行度N_{y}计算中，同步读取特征图数据容易引发访问冲突，影响并行效率。为解决该问题，本文提出单写多读端口缓存结构，如图5所示。图中，F_{x}和F_{y}分别表示特征图的宽度和高度，S表示卷积步长。每个单写多读端口由多个单读写窗口线性缓存组成，读取过程从左侧开始，按行高度依次扫描每个通道，随后将数据依次存入线性缓存。线性缓存的数量L_{n}由下式计算：

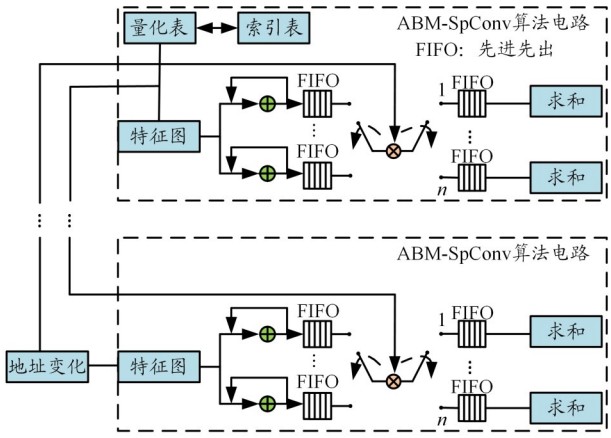

图4 纵向并行优化后的“先加后乘”算法电路控制图

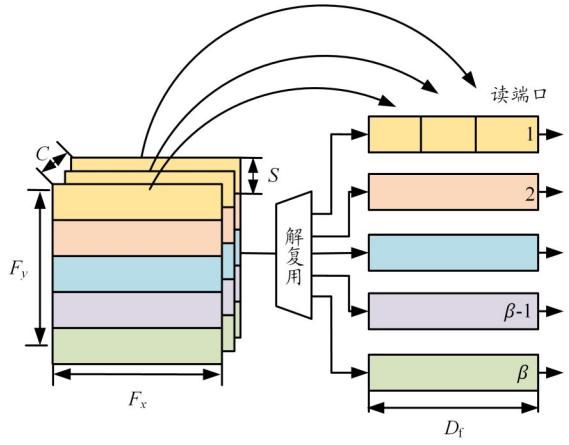

图5 单写多读缓存窗口

$$L_n = N_y + \text{ceil}\left(\frac{k-S}{S}\right) \tag{7}$$

每个线性缓存中存储的特征点数D_f为

$$D_f = F_x SC \tag{8}$$

纵向并行卷积时，特征图沿列方向同时输出N_y个滑动窗口数据，实现同一方向上的N_y次并行卷积。单写多读缓存结构的设计有效支撑了纵向并行卷积的硬件实现。

2.4 层融合硬件实现

传统卷积层后通常接有ReLU激活与最大池化操作，三者均需通过缓存Buffer传递中间数据。本文将卷积、ReLU与最大池化层设计为融合层，减少数据流动，形成高效流水线结构。优化前后的架构对比如图6所示，图6(a)为传统神经网络单元(优化前)的数据处理流程，图6(b)为优化后的融合处理流程。优化后的融合架构将ReLU与Max Pool层嵌入Conv层中，不仅减少了缓存Buffer的使用和数据读取时间损耗，还降低了关键路径延迟，在提高硬件利用率的同时提升了系统运行速度。

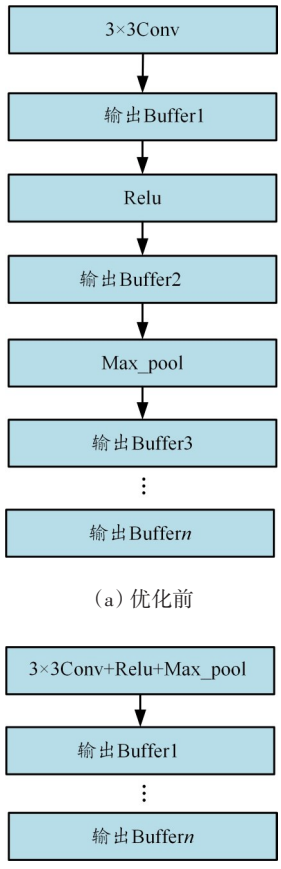

图6 优化前后融合层内部数据处理架构

3 实验与结果分析

3.1 实验环境

调制格式识别系统的硬件部署平台采用Xilinx Artix-7系列FPGA，型号为XC7A200。软件方面，开发环境基于PyCharm 2020.1.3集成开发工具，模型构建与训练采用PyTorch 1.x深度学习框架。主机实验平台配置为Intel® Core™ i7-7700HQ CPU与NVIDIA GeForce RTX 2080 Ti GPU。实验对比模型包括基准CNN模型及经非结构化剪枝、量化与算子融合压缩后的CNN-Pruned模型。

3.2 性能对比与分析

针对基准CNN模型与压缩后的CNN-Pruned模型，本文在Intel Core i7-7700HQ CPU、NVIDIA GeForce RTX 2080 Ti GPU及Xilinx Artix-7 XC7A200T FPGA 3种硬件平台上进行推理性能对比。表2与表3分别展示了压缩前后模型在功耗、推理延迟及能效方面的表现。可以看出，尽管CNN-Pruned模型平均分类精度略有下降(从91.6%下降到90.2%)，但其在FPGA上的能效比与推理延迟均优于基

表2 CNN模型不同平台性能对比

硬件平台	硬件型号	频率/MHz	功耗/W	推理延迟/μs	平均识别精度/%	吞吐量/(GOP/s)	能效比/(GOP·s^{-1}·W^{-1})
CPU	Intel® Core™ i7-7700HQ	2 800	25.11	1 081.87	91.6	0.311	0.012
GPU	NVIDIA GeForce GTX 2080 Ti	1 035	23.4	62.5	91.6	5.38	0.229
FPGA	XC7A200	200	1.543	227.96	91.6	1.476	0.91

表3 CNN_Pruned模型不同平台性能对比

硬件平台	硬件型号	频率/MHz	功耗/W	推理延迟/μs	平均识别精度/%	吞吐量/(GOP/s)	能效比/(GOP·s^{-1}·W^{-1})
CPU	Intel® Core™ i7-7700HQ	2 800	21.98	303.65	90.2	0.17	0.008
GPU	NVIDIA GeForce GTX 2080 Ti	1 035	22.9	52.64	90.2	0.96	0.043
FPGA	XC7A200	200	1.455	142.48	90.2	0.354	0.232

准CNN模型，分别为0.232 GOP/(s·W^{-1})、142.48 μs。功耗方面，FPGA平台功耗远低于GPU与CPU，约为后两者的1/14~1/13。

4 结束语

本文提出了一种面向调制格式识别的稀疏CNN FPGA加速器。通过剪枝、8比特动态定点量化与稀疏卷积优化，结合专用加速电路与缓存架构，在XC7A200 FPGA上实现高效部署。实验表明：参数量压缩比与计算量压缩比分别为10.84、6.67，平均识别精度仅从91.6%下降到90.2%；与CNN模型相比，压缩模型的片上功耗降至1.455 W，推理延迟从227.96 μs降至142.48 μs，功耗仅为前者的1/14~1/13。该方案兼具高性能与高能效，为边缘端调制格式识别提供了实用参考。

参考文献：

[1] Zhao Yilun, Yu Zhenming, Wan Zhiquan, et al. Low complexity OSNR monitoring and modulation format identification based on binarized neural networks[J]. Journal of Lightwave Technology, 2020, 38(6): 1314–1322.

[2] Dong Zhenhua, Khan F N, Sui Qi , et al. Optical performance monitoring: a review of current and future technologies[J]. Journal of Lightwave Technology, 2015, 34(2): 525–543.

[3] Thrane J, Wass J, Piels M, et al. Machine learning techniques for optical performance monitoring from directly detected PDM-QAM signals [J]. Journal of Lightwave Technology, 2017, 35(4): 868–875.

[4] 张家华. 一个基于FPGA的CNN加速器[D]. 成都：电子科技大学，2021.

[5] Wei Xin, Liu Wenchao, Chen Lei, et al. FPGA-based hybrid-type implementation of quantized neural networks for remote sensing applications[J]. Sensors, 2019, 19(4): 924-1–924-21.

[6] Li Weijia, He Conghui, Fu Haohuan, et al. A real-time tree crown detection approach for large-scale remote sensing images on FPGAs[J]. Remote Sensing, 2019, 11(9): 1025-1–1025-20.

[7] 周诗云，钱松荣，卫少东，等. 基于边缘部署低功耗的神经网络加速器[J]. 自动化与仪表，2024，39(7)：147–151，156.

[8] 牟云飞. 嵌入式系统的低功耗与可靠性技术研究[J]. 电子测试，2022，36(9)：132–134.

[9] 王婷，陈斌岳，张福海. 基于FPGA的CNN并行加速器设计[J]. 电子技术应用，2021，47(2)：81–84.

[10] Li Sicheng, Wen Wei, Wang Yu, et al. An FPGA design framework for CNN sparsification and acceleration[C]//IEEE. Annual International Symposium on Field-Programmable Custom Computing Machines. Napa: IEEE, 2017: 22–28.

[11] Wang Dong, Xu Ke, Jia Qun, et al. ABM-SpConv: a novel approach to FPGA-based acceleration of convolutional neural network inference [C]//IEEE. Proceedings of the 56th Annual Design Automation Conference . New York: IEEE, 2019: 1–6.