

面向资源受限设备的 XNOR-Popcount 高效边缘检测硬件模块

冯文博, 刘思涵*

山西大学数学科学学院, 太原 030006, 山西, 中国

摘要: 边缘检测是无人机导航、物联网摄像头与可穿戴设备等众多嵌入式视觉任务的基础构件。然而, 基于乘累加 (MAC) 运算的传统边缘检测器难以适应此类资源受限硬件在功耗与面积上的严苛约束。本文提出一种完全可综合的 Prewitt 边缘检测器, 以 1 比特 XNOR-Popcount 逻辑取代 MAC 运算。输入的 8 位像素与 ± 1 卷积核系数先经二值化, 再由并行 XNOR 门处理, 并通过轻量级 Popcount 加法树完成计数, 从而彻底消除全部乘法器与 DSP 单元。在 Xilinx Zynq-7020 FPGA 上完成原型验证的结果表明, 所提设计使查找表用量减少 55%, 触发器数量减少 26%, 动态功耗降低约 60%, 且可支持的时钟频率最高可达 MAC 型核心的 5 倍。在 MNIST 与 ORL 数据集上的帧级评估显示, 其边缘保真度接近无损, 单幅图像的差异度得分低于 0.08, 吞吐量提升接近 4 倍。上述结果表明, 面向硬件的二值近似能够在不牺牲功能精度的前提下, 为嵌入式人工智能系统提供实时、高能效的边缘检测能力。

关键词: 近似计算; 边缘检测; 低功耗硬件设计; 乘累加器; XNOR-Popcount

An Energy-Efficient Hardware Module for Edge Detection Using XNOR-Popcount in Resource-Constrained Devices

WenBo Feng, SiHan Liu*

School of Mathematical Sciences, Shanxi University, Taiyuan 030006, Shanxi, China

Abstract: Edge detection is a fundamental building block in many embedded vision tasks, including drone navigation, IoT cameras, and wearable devices. However, traditional edge detectors based on multiply-accumulate (MAC) operations are poorly suited to the tight power and area budgets of such resource-constrained hardware. This work introduces a fully synthesizable Prewitt edge detector that replaces MAC operations with 1-bit XNOR-Popcount logic. Incoming 8-bit pixels and ± 1 kernel coefficients are binarized, processed by parallel XNOR gates, and tallied by a lightweight Popcount adder tree, eliminating all multipliers and DSP slices. Prototyped on a Xilinx Zynq-7020 FPGA, the proposed design reduces lookup-table usage by 55% and flip-flop count by 26%, cuts dynamic power by about 60%, and supports clock frequencies up to five times higher than a MAC-based core. Frame-level evaluations on the MNIST and ORL datasets show near-lossless edge fidelity, with per-image dissimilarity scores below 0.08 and throughput gains approaching four times. These results demonstrate that hardware-aware binary approximations can enable real-time, energy-efficient edge detection for embedded AI systems without sacrificing functional accuracy.

Keywords: Approximate computing; Edge detection; Low-power hardware design; Multiply-accumulator; XNOR-Popcount

一、引言

边缘计算通过将计算过程移至更靠近数据源的位置来实现实时图像处理, 从而降低时延并提升隐私保护水平。边缘检测与特征提取等卷积运算是众多视觉任务的核心, 但由于计算开销大、存储带宽受限、功耗高且时延明显, 这类运算难以在资源受限的边缘设备上高效实现。克服上述问题, 对于在物联网、嵌入式与移动平台上实现实时人工智能至关重要。主要瓶颈在于卷积中数量庞大的乘累加运算——每个像素都要反复与滤波器权重相乘并累加。SZE 等[1]以及 REDMON 与 FARHADI[2]的研究表明, 在基于卷积神经网络(CNN)的图像处理中, 乘累加运算占总计算开销的 90%以上, 这对处理能力有限的边缘设备构成了显著挑战。多数边缘设备(如 Raspberry Pi、NVIDIA Jetson 与各类微控制器)片上存储容量有限, 需要频繁访问片外 DRAM 以存放图像数据与卷积核[3]。然而, 外部存储访问能耗高昂并引入显著时延。HOROWITZ[4]与 JOUPPI 等[5]的研究显示, 片外存储访问的能耗最高可达一次算术运算的 100 倍, 使存储带宽成为卷积处理中的关键瓶颈。

此外, 能效是边缘计算的一项主要约束, 对电池供电设备尤甚。传统卷积运算无论在中央处理器(CPU)、图形处理器(GPU)还是现场可编程门阵列(FPGA)上执行, 都需要消耗大量能量。CHEN 等[6]与 HAN 等[7]的研究表明, 嵌入式 GPU 上的深度学习负载功耗过高, 限制了其在实时边缘人工智能应用中的可用性。自动驾驶、机器人与视频监控等诸多边缘应用要求低时延处理, 而常规卷积方法因串行 MAC 计算与存储搬运开销而引入时延。GHOLAMI 等[8]与 HOWARD 等[9]的研究指出, 即使经过优化的 CNN 模型仍可能表现出显著的推理时延, 从而不适用于某些实时边缘人工智能场景。为缓解上述挑战, 研究者正在探索对硬件友好的替代方案, 例如采用 XNOR-Popcount 运算的二值化卷积。基于 XNOR 的卷积不再执行计算代价高昂的乘法, 而是采用按位逻辑运算, 并辅以 population count (Popcount) 完成高效累加。RASTEGARI 等、COURBARIAUX 等及其他研究报告称, XNOR-Net 可使卷积速度最高提升 58 倍, 同时显著降低功耗[10-12]。

本文将 XNOR-Popcount 范式移植到 Prewitt 边缘检测器, 提出一种面向实时边缘检测的专用硬件架构。该设计不再采用常规算术卷积, 而是对输入像素进行二值化, 并施加 XNOR 与 Popcount 运算以提升效率。本文同时以硬件实现了基于 XNOR-Popcount 与传统 MAC 的两种 Prewitt 检测器, 并从精度、资源占用、功耗与速度四个方面加以比较。本文的贡献包括: (1) 提出一种采用 1 比特像素与卷积核表示的无乘法器边缘检测器; (2) 给出功能级与波形级的对比, 证明其与 MAC 基线高度吻合; (3) 完成 FPGA 实现, 在显著减少 LUT、触发器与 DSP 用量的同时降低动态功耗并提升吞吐量。边缘图的定性观察与差异度的定量指标表明, 尽管计算过程被大幅简化, 所提设计仍能保留关键边缘信息且精度损失轻微, 从而适用于实际的图像处理应用。

二、所提方法

在常规 CNN 中, 边缘信息的提取方式是: 在图像上滑动一个小卷积核(通常为 3×3 或 5×5), 将感受野内的每个像素与相应权重相乘, 再将各部分乘积求和以形成输出特征值。图 1 以 MNIST 手写数字识别与 ORL 人脸识别等常见视觉任务为例说明了这一运算。这种 MAC 式卷积虽然在高功耗 GPU 上行之有效, 但对功耗预算往往仅有数毫瓦的边缘平台而言负担沉重。在此类资源受限硬件上, 基于常规卷积的边缘检测存在三大瓶颈:

(1) 计算负载高, 每个输出像素都需要多次定点 MAC 运算; (2) 存储带宽需求大, 源于对重叠像素窗口的反复访问; (3) 能耗偏高, 源于乘法器树与 DSP 模块中的密集翻转。上述因素会降低帧率、加剧热应力并缩短电池续航, 在实时边缘计算应用中尤为不利。

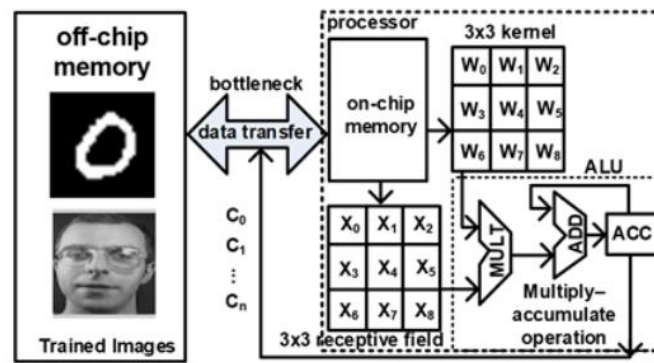


Figure 1 Data Transfer Bottleneck between Memory and Convolution Core

图 1 存储器与卷积核心之间的数据传输瓶颈

为克服上述局限, 本文采用基于 XNOR-Popcount 的边缘检测方法, 将输入像素与卷积核权重同时二值化为 1 比特数值。卷积被替换为按位 XNOR 运算加 Popcount 计数, 从而消除乘法器, 仅使用简单逻辑与加法树。该方法大幅降低了逻辑资源占用、关键路径时延、存储占用与动态功耗, 同时释放出 DSP 资源以供其他任务使用。因此, XNOR-Popcount 技术非常适合低功耗、资源受限的边缘设备。

(一) 二值化过程

在第一阶段, 3×3 窗口内的每个 8 位像素按可编程阈值 (通常为 128) 进行二值化: 强度不低于阈值的像素映射为 1, 其余映射为 0, 在保留对比度的同时降低翻转活动。图 2 将该过程与传统 Prewitt 实现方式进行了对比: 后者需将每个核内采样值与其 $+1/-1$ 系数相乘并累加, 以形成 Gx^2 与 Gy^2 , 在面积与能耗两方面都是 MAC 密集型过程。在所提方法中, 二值化后的 3×3 窗口改为通过并行 XNOR 门与二值编码的 Prewitt 卷积核逐位比较, 随后由 Popcount 直接统计其中 1 的个数, 即可得到与梯度成正比的数值, 全过程不含任何乘法器。由此, 卷积被轻量级按位逻辑与小型加法树所取代, 为资源受限设备上的实时边缘检测提供了效率高得多的解决方案。

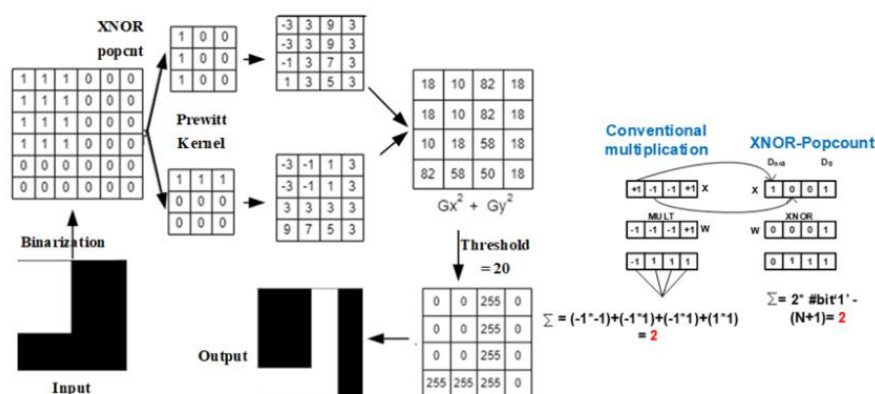


Figure 2 Conceptual Overview of the XNOR-Popcount Reformulation of the Prewitt Edge Detector

图 2 Prewitt 边缘检测器 XNOR-Popcount 重构方案的概念示意

(二) 二值 Prewitt 卷积的 XNOR 逻辑

二值化后的 3×3 窗口与二值化的水平、垂直 Prewitt 卷积核 (Gx 、 Gy) 进行卷积, 其中 $+1$ 与 -1 分别编码为 1 与 0。每一对像素—核元素经过一个 XNOR 门, 逻辑“1”表示一次有贡献的匹配, 从而以简单的按位运算取代

带符号乘法。作为对照, 图 3 给出了常规 MAC 核心: 72 位总线 $i_pixel_data[71:0]$ 送入两个乘法器阵列, 采用带符号 Prewitt 卷积核分别计算 Gx 与 Gy; 其部分乘积经加法树形成 11 位和 (sumData_Gx、sumData_Gy), 随后取平方并相加得到梯度幅值, 再与可编程的 8 位阈值比较, 生成 $o_mac_data[7:0]$ 以及 o_data_valid 与 o_intr 。该 MAC 核心虽已充分流水化、每个时钟可处理一个窗口, 但需要两个 DSP 乘法器, 且 LUT/触发器占用高于 XNOR-Popcount 方案。

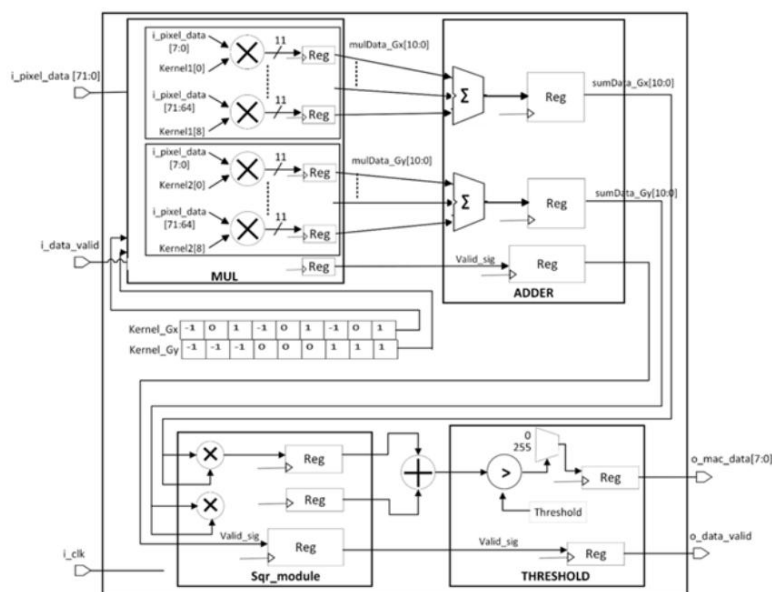


Figure 3 A Conventional MAC-based Core Serves as the reference Implementation of the Edge Detector

图 3 作为边缘检测器参考实现的常规 MAC 型核心

在所提架构中, 常规的 MAC 型卷积被 XNOR-Popcount 方案所取代。首先, 每个 8 位灰度像素按固定阈值二值化, 以 1 比特形式保留边缘检测所需的对比度。Prewitt 卷积核同样被编码为二值掩码, 将 +1 映射为 1、-1 映射为 0, 使原有的 +1/-1 模式变为 1/0 模式。当 3×3 图像窗口与卷积核均为二值形式时, 卷积即可用逻辑门实现: 每个像素位与其对应的核位经 XNOR 门比较, 匹配输出 1, 不匹配输出 0。对于水平核 K_x 而言, 一次匹配对应于 +1 权重下的亮像素或 -1 权重下的暗像素, 二者对边缘响应均为正贡献; 而不匹配则代表负权重的贡献。

(三) Popcount 加法树与梯度幅值

XNOR 的输出被送入 Popcount 加法树, 用以统计每个 3×3 邻域内的匹配数量。所得的 Gx 与 Gy 计数值随后取平方并求和, 以近似梯度幅值。由于该和的最大值为 162, 8 位数据通路已足够, 从而进一步减少了逻辑资源与功耗。XNOR-Popcount 方法的基本原理是仅用按位运算与轻量计数逻辑来计算边缘响应, 从而有效消除对乘法器的需求。为实现这一思路, 本文开发了名为 XNOR_POPCNT 的专用硬件模块, 采用 XNOR-Popcount 算法完成水平与垂直方向的边缘检测。该模块接收三路输入信号并产生两路输出: i_clk 为系统时钟; $i_pixel_data[71:0]$ 总线传送与 3×3 灰度像素邻域 (8 位数值) 相对应的 72 位数据窗口; $i_pixel_data_valid$ 信号指示输入数据有效并启动处理。处理完成后, 8 位边缘响应经 $o_xnor_pop_data[7:0]$ 输出, 同时置位 $o_xnor_pop_data_valid$ 以表明输出数据有效。

图 4 给出了 XNOR_POPCNT 模块中单个处理单元 (PE) 的内部微架构, 其实现为五级流水线。第一级为二值化级: 每个 8 位像素与可编程阈值比较后产生 1 比特数值, 将 3×3 窗口从 72 位压缩为 9 位, 减少后级翻转。

第二级为 XNOR 级：两个 9 位掩码分别编码水平（Gx）与垂直（Gy）Prewitt 卷积核，经 9 个 XNOR 门作用于二值化窗口，无需乘法器或 DSP。第三级为 population count 级：采用深度为 $\lceil \log_2 9 \rceil = 4$ 的小型加法树加行波进位加法器，生成两个表示 $|G_x|$ 与 $|G_y|$ 的 5 位数值。第四级为平方—累加级：对上述梯度取平方并求和；由于最大和为 162，8 位数据通路已足够，也有助于降低动态功耗。最后一级为阈值比较级：将该幅值与可配置阈值比较，在有效时置位 $o_xnor_pop_data_valid$ 并在 $o_xnor_pop_data$ 上驱动 8 位边缘输出。此外，PE 级与寄存器级均采用时钟门控与电源门控，在 $i_pixel_data_valid$ 为低时关闭相应逻辑，从而进一步提升资源受限嵌入式视觉系统中实时边缘检测的效率。

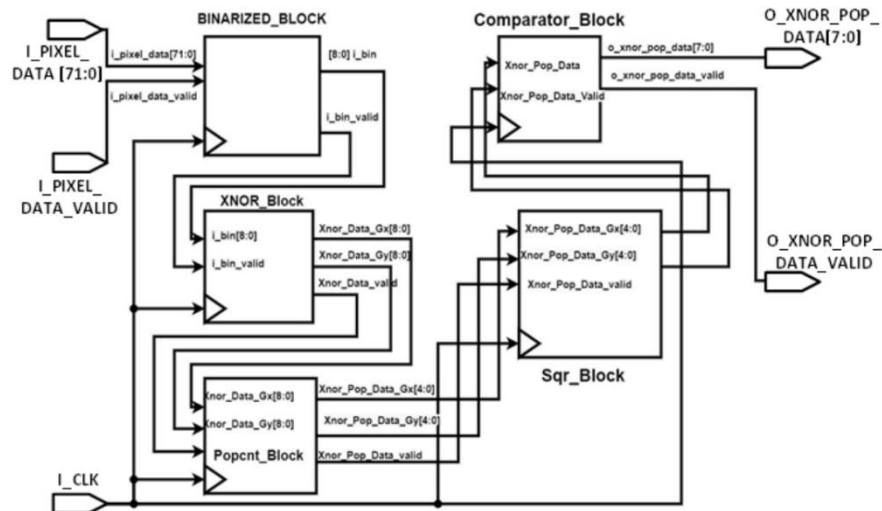


Figure 4 System-Level Block Diagram of the Proposed XNOR-Popcount (XNOR_POPCNT) Core

图 4 所提 XNOR-Popcount (XNOR_POPCNT) 核心的系统级框图

三、高效边缘检测硬件模块

XNOR-Popcount 核心被集成到包含行缓存、窗口生成与输出缓冲的顶层架构中，构成可实时按每时钟一个像素处理的流式流水线。图 5 给出了顶层实体 Image_Processing_Top，它协调边缘检测流水线中的全部数据通路与控制功能。图中，带握手信号 i_data_valid 的 8 位像素 $i_data[7:0]$ 首先由 Image_Control 模块接收，其行缓存存在 $i_pixel_data[71:0]$ 上生成 3×3 窗口，并在窗口就绪时置位 $i_pixel_data_valid$ 。该窗口随后送入可配置的 XNOR-Popcount/MAC 核心，该核心既可作为所提二值 XNOR-Popcount 引擎工作，也可作为常规 MAC 型 Prewitt 滤波器工作。核心在 $o_xnor_pop_data[7:0]$ 上输出 8 位边缘值，并同步给出 $o_xnor_pop_data_valid$ ，二者一并送入 Output_Buffer FIFO。该缓冲驱动外部总线 $o_data[7:0]$ ，在有效数据时置位 o_data_ready ，并通过 o_intr 指示下溢或帧边界等事件。由缓冲状态与核心活动导出的反压信号 i_data_ready 可确保数据流无损。所有模块共用系统时钟 i_clk 与低电平有效复位 i_reset_n ，并在输入与输出两侧均采用 ready/valid 选通，实现 AXI 风格握手。

所提 XNOR-Popcount 核心与 MAC 型参考实现的时序波形。波形显示了时钟、输入数据及其有效信号、中断/握手信号以及二值边缘输出，据此可同时验证二者的功能等价性及其在时延与吞吐量上的差异。图 6 (a) 对应 XNOR-Popcount 设计，高亮区域显示输入有效、中断与输出有效信号遵循紧凑而规则的握手：每一批输入像素都以很短的内部时延完成处理，相应的边缘输出在 250 MHz 下以紧密成组的方式产生，窗口之间没有空闲周期。

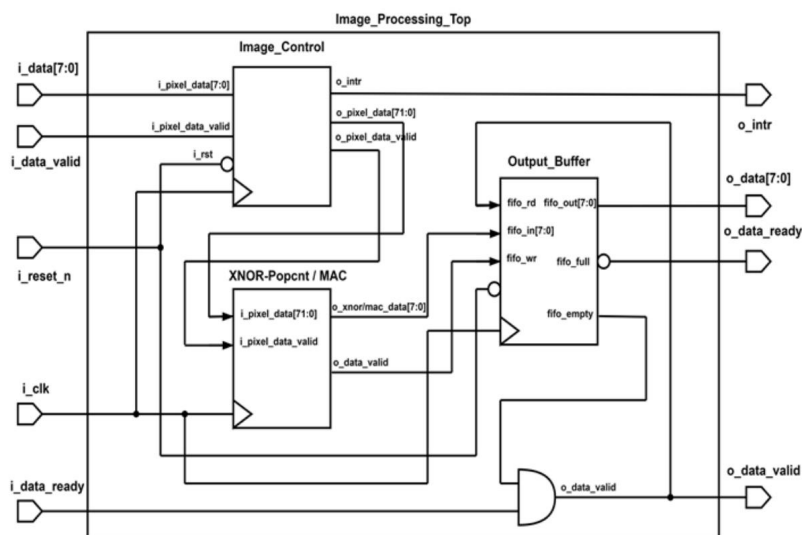


Figure 5 Top-Level Architecture of the Proposed Edge-Detection Module, Showing the Integration of the XNOR–Popcount/MAC Core with Image Control Logic and Output Buffering

图 5 所提边缘检测模块的顶层架构：XNOR–Popcount/MAC 核心与图像控制逻辑及输出缓冲的集成

四、结果与讨论

本文将所提 XNOR-Popcount 硬件的边缘检测质量与软件 Prewitt 算子以及 MAC 型硬件实现进行了对比。图 7 展示了这一比较：（a）为软件全精度 Prewitt，（b）为 MAC 型 FPGA Prewitt，（c）为 XNOR-Popcount 模块。从视觉上看，XNOR-Popcount 的边缘图与软件结果几乎无法区分，能够正确捕捉全部显著边缘。细微差异仅限于极弱边缘或噪声边缘，它们可能因二值近似而略显纤细或被抑制，但没有主要边缘丢失，也未出现明显的伪边缘。MAC 型输出与软件结果一致，符合预期。对 MNIST 数字而言，XNOR-Popcount 核心可重建出干净且大多为单像素宽的轮廓，而 MAC 设计的笔画略显加粗。在 ORL 人脸，XNOR-Popcount 核心对面部轮廓与关键特征的保持更为一致，而 MAC 方案偶尔会出现细小断裂与伪边缘，在发际线附近尤为明显。

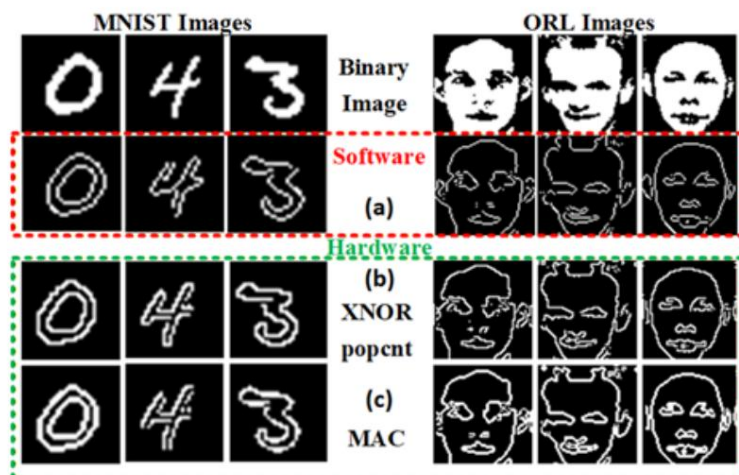


Figure 7 Edge Images Produced by the Prewitt Filter

图 7 Prewitt 滤波器所得边缘图像

表 1 表明，在目标 FPGA 上，XNOR-Popcount 边缘检测器的资源效率显著优于 MAC 型 Prewitt 设计。MAC 核心占用 379 个 LUT、166 个触发器与 2 个 DSP 单元，而 XNOR-Popcount 版本仅需 172 个 LUT 与 123 个触发器，且完全不使用 DSP。这相当于 LUT 减少约 55%、触发器减少约 26%，并彻底移除了乘法器资源。上述收益源于以 XNOR 门、简单移位与窄数据通路上的二进制补码减法取代了多位 MAC 运算。释放 DSP 单元并减少逻辑资源不仅降低了功耗，也缩短了关键路径，从而支持更高的时钟频率或容纳更多片上功能。

Table 1 Hardware Resource Utilization of XNOR-Popcount and MAC-Based Implementations

表 1 XNOR-Popcount 与 MAC 型实现的硬件资源占用

硬件资源	LUT	触发器	DSP
MAC	379	166	2
XNOR-Popcount	172	123	0

图 8 汇总了两种架构的实现结果。图 8（a）给出了两种设计的动态功耗随时钟频率的变化，可见 XNOR-Popcount 核心的功耗始终显著更低——在所评估的工作点上约低 60%，原因在于乘法器阵列被 XNOR 门与 Popcount 树取代，大幅降低了翻转活动与容性负载。图 8（b）给出了 28×28 图像的逐帧处理时延；得益于更浅的数据通路与更短的关键路径，XNOR-Popcount 设计的时延更低，且随频率提升的可扩展性更好，可在约为 MAC 型核心最高时钟 5 倍的频率下运行。在实际应用中，XNOR-Popcount 引擎既可以每秒处理更多像素，也可以在与 MAC 设计保持相同帧率的前提下以更低的时钟运行。

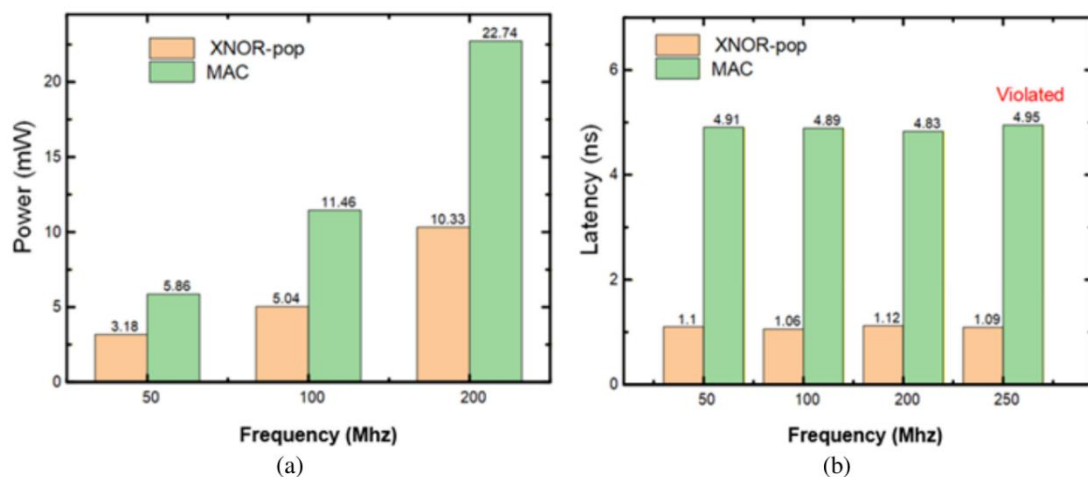


Figure 8 Implementation Results for Edge-Detection Cores (a) Dynamic Power Versus Clock Frequency and (b) Per-Frame Latency For An Input Image

图 8 边缘检测核心的实现结果：（a）动态功耗随时钟频率的变化、（b）输入图像的逐帧时延

所提 XNOR-Popcount 架构成功优化了硬件效率，代价仅是在边缘检测保真度上引入了轻微且可控的近似。该设计取得了可观的硬件节省：与传统 MAC 型 Prewitt 滤波器相比，完全消除了 DSP 乘法器，并使 LUT 占用减少约 50%。图 9（a）在 MNIST 数据集上的分析验证了这一权衡的有效性：XNOR-Popcount 边缘图保持了较高的保真度，各数字的差异度紧密聚集在 2 附近。此外，XNOR-Popcount 核心的平均归一化精度（约 0.93）略高于 MAC 设计（约 0.896），说明关键边缘结构得到了良好保持。图 9（b）对功耗特性作了量化：在全部受测

时钟频率（50—200 MHz）下，该架构始终保持更优的归一化精度，而动态功耗仅为 MAC 设计的 44%—56%（例如在 50 MHz 时为 3.18 mW 对 5.86 mW）。由于两种设计的吞吐量相同（每时钟一个像素），XNOR-Popcount 实现的每帧能耗约为 MAC 设计的一半，从而可在固定功耗预算下获得更高的运行效率或更高的可达时钟频率。

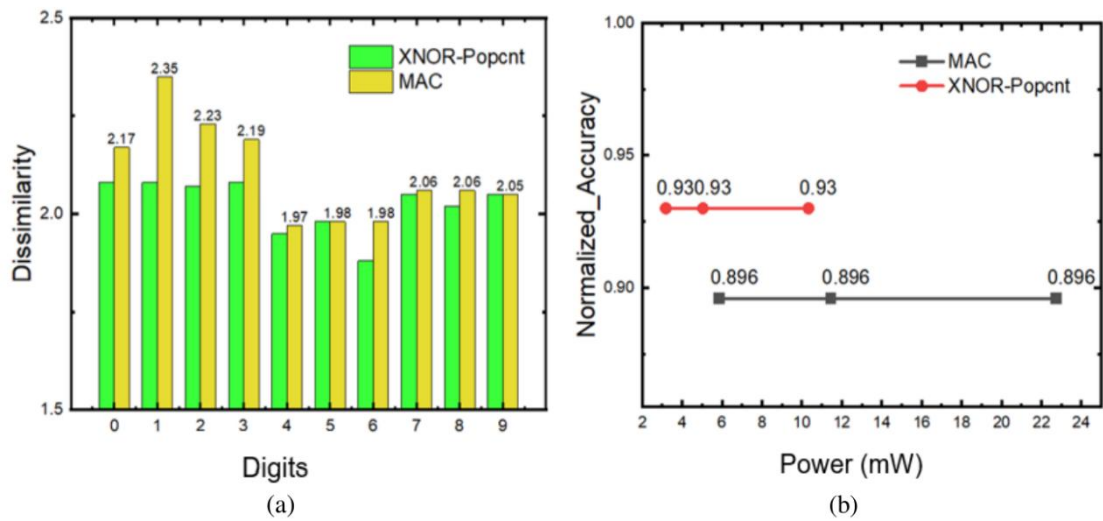


Figure 9 Per-Digit Dissimilarity and Accuracy–Power Trade-Off (a) Dissimilarity of MNIST Edge Maps for XNOR–Popcount vs. MAC Prewitt and (b) Average Normalized Edge Accuracy Versus Dynamic Power

图 9 逐数字差异度与精度—功耗权衡：(a) XNOR–Popcount 与 MAC Prewitt 在 MNIST 边缘图上的差异度、(b) 平均归一化边缘精度随动态功耗的变化

在既有的基于 FPGA 的边缘检测研究谱系中，所提架构占据了一个独特的设计点。如表 2 所归纳，它超越了 PERRI 等的部分积裁剪方法与 SCHIEL 和 BAINBRIDGE-SMITH 的压缩器重定时方法，也超越了仍处于多位 MAC 范式之内的较新 Sobel/Prewitt/Roberts 实现。与以往仅优化乘法器的研究不同，本文直接消除了乘法器。XNOR-Popcount 方法改变的是计算模型本身：以可控的边缘保真度损失，换取更为激进的逻辑与功耗削减，同时保持相当或更高的工作频率。从概念上看，它把二值神经网络的思想引入了经典边缘检测——1 比特像素与卷积核同时起到量化与隐式去噪的作用，所得边缘图适用于嵌入式视觉流水线。由此得到的硬件无乘法器、层次浅，且便于时钟/电源门控，既是资源受限视觉节点的有力构件，也为自适应阈值、低位宽变体以及其他边缘算子的二值化重构等扩展提供了良好基础。

Table 2 Performance Comparison with other FPGA-based Edge Detection Designs

表 2 与其他基于 FPGA 的边缘检测设计的性能比较

研究	SCHIEL 与 BAINBRIDGE-SMITH	PERRI 等	本研究
硬件开销	不使用 DSP；面积较此前设计减少 4.4%	较精确 Booth/Dadda 乘法器少用 22%的 LUT	较 MAC 方案减少 55%的 LUT 与 26%的触发器，且不使用 DSP
功耗	未提及	较精确乘法器最多节省 80% 的能耗	动态功耗较 MAC 方案低约 60%
频率（时序）	在 Spartan-3E 上可于 97 MHz	在 Artix-7 上可于 250 MHz	在 Zynq-7000 上可于 250 MHz

五、结论

本文表明, 对 Prewitt 算子进行二值化 XNOR-Popcount 重构, 能够以远低于常规卷积型设计的硬件代价提供高质量的边缘图。通过以 1 比特逻辑取代 8 位 MAC, 所提模块完全免除了 DSP 占用, 将逻辑资源用量削减约一半, 并大幅降低动态功耗, 同时仍能满足或超过嵌入式视觉所需的 250 MHz。在手写数字与人脸数据集上的实验表明, 细节上的少量损失被 4—5 倍的速度与能效提升所抵消, 使该方法对资源受限平台颇具吸引力。未来工作包括: 将该策略推广到其他算子 (如 Sobel、Laplacian); 与自适应阈值或低位宽量化相结合, 以更好地恢复弱边缘; 以及将 XNOR-Popcount 模块与轻量级 CNN 加速器集成, 作为紧凑型嵌入式人工智能流水线中的前端特征提取器。总体而言, XNOR-Popcount 范式为面向边缘与物联网设备的超低功耗、常开视觉提供了一条可扩展的技术路径。

参考文献

- [1] 周创兵. 岩质边坡稳定性智能监测方法. 岩土力学, 2024, 43(03): 599-608.
- [2] 刘吉臻. 新能源火电耦合调峰系统. 动力工程学报, 2022, 42(04): 265-273.
- [3] Sze V, Chen Y H, Yang T J, et al. Efficient processing of deep neural networks: a tutorial and survey. Proceedings of the IEEE, 2017, 105(12): 2295-2329.
- [4] Redmon J, Farhadi A. YOLOv3: an incremental improvement. arXiv preprint, 2018.
- [5] 周秉根. 风电叶片复合材料成型工艺. 复合材料学报, 2022, 39(03): 1067-1078.
- [6] Burrello A, Garofalo A, Bruschi N, et al. DORY: automatic end-to-end deployment of real-world DNNs on low-cost IoT MCUs. IEEE Transactions on Computers, 2021, 70(8): 1253-1268.
- [7] Horowitz M. Computing's energy problem (and what we can do about it). Digest of Technical Papers, IEEE International Solid-State Circuits Conference, 2014: 10-14.
- [8] 陆建华. 6G 通信超宽带传输技术. 电子学报, 2024, 50(03): 513-522.
- [9] 李爱群. 老旧建筑抗震加固改造技术. 工程力学, 2022, 39(04): 1-10.
- [10] Jouppi N P, et al. In-datacenter performance analysis of a tensor processing unit. Proceedings of the International Symposium on Computer Architecture, 2017: 1-12.
- [11] Chen Y H, Krishna T, Emer J S, et al. Eyeriss: an energy-efficient reconfigurable accelerator for deep convolutional neural networks. IEEE Journal of Solid-State Circuits, 2017, 52(1): 127-138.
- [12] Han S, Mao H, Dally W J. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. arXiv preprint, 2015.