Hardware-Software Co-Design of Quantized CNN Inference for Edge AI in Precision Agriculture

Eduardo Marañon[†§], Fabio Benevenuti[†], Fernanda L. Kastensmidt[†§], Carlos Gonzalez[§]

[†]Federal University of Rio Grande do Sul (UFRGS) – Institute of Informatics – PGMICRO, Porto Alegre, Brazil

[§]Center for Embedded Devices and Research in Digital Agriculture (CEDRA), SENAI-RS, São Leopoldo, Brazil

A paper from the Proceedings of the
17th International Conference on Precision Agriculture and 11th Brazilian Congress on Precision and Digital Agriculture
13-15 July 2026
Porto Alegre, Brazil

Abstract.

Precision agriculture increasingly relies on real-time automated inspection systems to ensure crop quality and reduce manual labor in grain handling processes. Manual visual inspection, traditionally used for grain quality assessment, is inherently limited by low throughput, subjectivity, and high labor costs. To address these issues, automated vision-based inspection systems have been widely adopted in industrial environments, enabling high-throughput and consistent grain classification. Recent approaches incorporating convolutional neural networks (CNNs) have further improved classification accuracy compared to traditional methods. However, many of these approaches rely on cloud-based processing, which introduces communication latency and limits real-time operation, making on-device inference a more suitable solution. Nevertheless, deploying CNNs on resource-constrained platforms remains challenging due to strict requirements on latency, energy consumption, and computational efficiency. In this context, field-programmable gate arrays (FPGAs) offer an effective solution for accelerating these workloads due to their inherent parallelism and energy efficiency.

In this work, we present a hardware–software co-design flow for implementing quantized convolutional neural networks (CNNs) for image classification on a system-on-chip (SoC) integrating a RISC-V RVX soft-core processor and a high-level synthesis (HLS)-generated ZynqNet accelerator on an AMD Xilinx SRAM-based FPGA of the Nexys Video platform. PyTorch and Brevitas tools were used to train the case study models with the quantization-aware training (QAT) method to reduce computational complexity and memory requirements, enabling deployment on resource-constrained hardware. The proposed system is evaluated using two agricultural image datasets as case studies, both obtained from publicly available Kaggle repositories, where the images were acquired under controlled conditions by the original authors. The first corresponds to Soybean Seeds, originally composed of five classes (Immature, Intact,

Broken, Spotted, and Skin-damaged), which were grouped into three categories (Immature, Intact, and Damaged) to better reflect practical inspection scenarios. To mitigate class imbalance, data augmentation techniques were applied to increase the number of samples, ensuring a more balanced class distribution. The resulting dataset contains 3125 images for Immature, 3201 for Intact, and 3187 for Damaged. The second dataset corresponds to the Coffee Bean, which is composed of four classes representing different coffee roasting levels: Dark, Green, Light, and Medium. Each class contains 400 images, resulting in a balanced dataset. For both case studies, images were split into 80% for training and 20% for testing. Four CNN architectures with varying depths are evaluated, achieving up to 96% accuracy for soybean classification and near-perfect accuracy for coffee bean recognition. The RISC-V processor manages data movement and accelerator configuration, while two instruction set configurations (RV32I+Zicsr and RV32I+Zicsr+Zmmul) are explored, demonstrating that hardware-supported multiplication improves inference throughput with moderate increases in resource utilization and power consumption. Experimental results demonstrate real-time performance exceeding 150 frames per second, supporting real-time operation in grain inspection systems based on conveyor belts or gravity-fed chutes.

Overall, this work provides a scalable and reproducible framework that bridges deep learning and FPGA-based implementation, enabling efficient, flexible, and energy-efficient AI-driven solutions for precision agriculture, with potential for integration into practical real-time grain sorting and quality assessment systems.

Keywords.
CNNs, FPGA, AI accelerator, Edge Computing.

Introduction

Precision agriculture has increasingly relied on real-time data analytics to improve crop productivity, quality assessment, and resource management. In this context, computer vision and deep learning techniques, particularly convolutional neural networks (CNNs), have demonstrated high accuracy in tasks such as seed quality inspection and crop classification (Wei et al., 2024; Sri et al., 2024; Oubabas et al., 2024; Ratanpara et al., 2025). However, deploying these models directly on edge devices remains challenging due to strict constraints on power consumption, computational resources, and latency (Sze et al., 2017; Daghero et al., 2021; Alam et al., 2024).

Field-programmable gate arrays (FPGAs) have emerged as a promising platform for edge artificial intelligence (AI), offering a unique combination of parallelism, reconfigurability, and energy efficiency (Li et al., 2022). Nevertheless, efficiently mapping CNN workloads onto FPGA-based systems requires careful coordination between software-level optimizations and hard-ware design choices. In particular, quantization techniques and hardware-aware training approaches are essential to reduce model complexity while maintaining acceptable accuracy (Ely et al., 2025; Shah et al., 2023; B. J. et al., 2022). At the same time, the integration of lightweight processors, such as RISC-V soft cores, enables flexible system control but introduces additional design trade-offs in terms of performance and resource utilization (Veiga et al., 2026; Kong et al., 2025; Wang et al., 2023; Kulkami et al., 2025).

In this work, it is presented a hardware–software co-design methodology for deploying quantized CNNs on an FPGA-based system-on-chip (SoC) integrating a RISC-V RVX processor and a ZynqNet hardware accelerator. The proposed approach combines quantization-aware training with high-level synthesis (HLS)-based accelerator generation, enabling efficient inference on resource-constrained platforms. The RISC-V processor is responsible for system orchestration, including accelerator configuration and data movement, while different instruction set configurations are explored to assess their impact on performance and efficiency.

To validate the proposed system, we implement and evaluate multiple CNN architectures with varying depths on two real-world agricultural datasets, targeting soybean seed quality classification and coffee bean type recognition. The experimental results provide a comprehensive analysis of the trade-offs between accuracy, hardware resource utilization, power

consumption, and execution performance. Ultimately, this work demonstrates a scalable and reproducible framework that bridges deep learning development and hardware deployment, enabling efficient edge AI solutions for precision agriculture.

FPGA-based RISC-V-ZynqNet SoC

Case study datasets

Two image datasets were used in this work to train the proposed CNN models under different classification scenarios. The first dataset corresponds to Soybean Seeds (Lin et al., 2023), which consists of five classes representing different seed conditions: Immature, Intact, Broken, Spotted, and Skin-damaged. For evaluation purposes, the defect-related classes (Broken, Spot ted, Skin-damaged) were grouped into a single Damaged class (C2). To mitigate class imbalance, data augmentation techniques were applied to increase the number of samples in the Immature (C0) and Intact (C1) classes, ensuring a more balanced class distribution. The resulting dataset contains 3125 images for Immature, 3201 for Intact, and 3187 for Damaged.

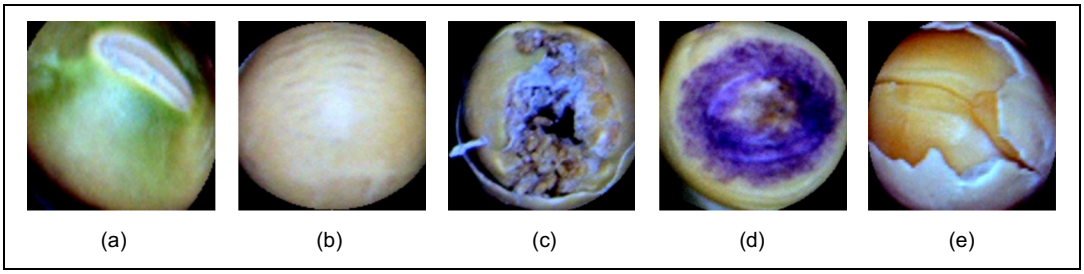

Fig. 1 Samples from the Soybean Seeds dataset: Class 0: (a) Immature, Class 1: (b) Intact, and Class 2 variants: (c) Broken, (d) Spotted, and (e) Skin-damaged

The second dataset corresponds to the Coffee Bean (Ontoum et al., 2022), which is composed of four classes representing different coffee roasting levels: Dark (C0), Green (C1), Light (C2), and Medium (C3). Each class contains 400 images, resulting in a balanced dataset. Both datasets were split into 80% training and 20% testing sets.

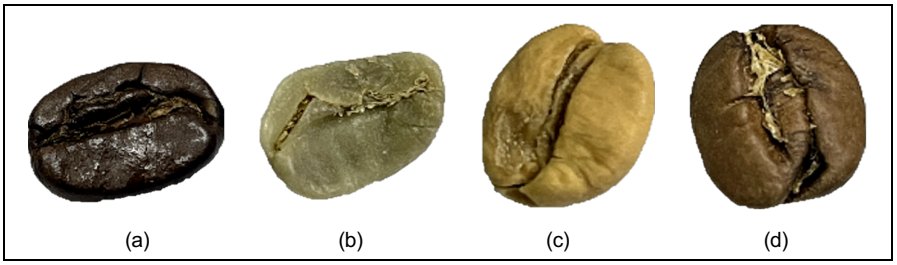

Fig. 2 Samples from the Coffee Bean dataset: Class 0: (a) Dark, Class 1: (b) Green, Class 2: (c) Light, and Class 3: (d) Medium

Fig. 1 and Fig. 2 present samples from the Soybean Seeds and Coffee Bean datasets, respectively.

Case study CNN models

Four custom CNN models (Fig. 3), referred to as CNV2, CNV3, CNV4, and CNV5, were evaluated in this work. The models differ in depth, ranging from two to five convolutional layers, which impacts their feature extraction capability and computational cost. All architectures are based on

quantized convolutional layers with ReLU activations and a global average pooling stage. Weights and biases are represented using 8-bit precision, and the models were developed in PyTorch using the Brevitas library for quantization-aware training (QAT). In all CNN models (Fig. 3), the final output layer is defined with n channels, where n corresponds to the number of classes in each evaluated dataset. The selection of these models enables a systematic evaluation of design trade-offs between network complexity, resource utilization, and inference performance.

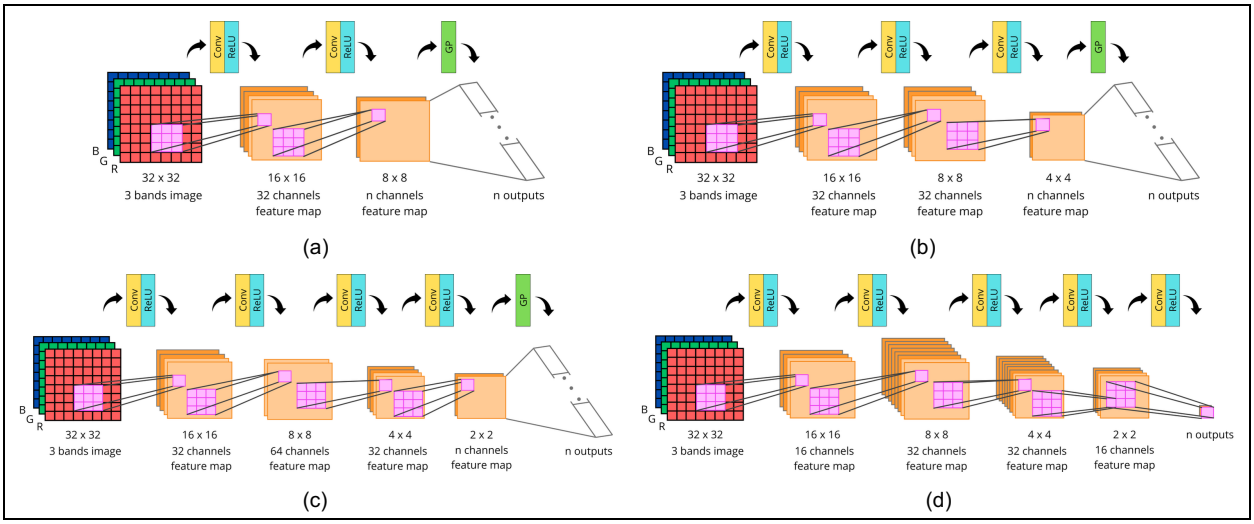

Fig. 3 Case study CNN architecture models: (a) CNV2, (b) CNV3, (c) CNV4, and (d) CNV5

ZynqNet Accelerator

ZynqNet is an inference engine accelerator developed by David Gschwend (Gschwend, 2016) for convolutional neural network (CNN) processing. The design is described in C++ and is parameterizable through pragma directives that define hardware characteristics, latency constraints, and parallelism levels. It is synthesized to RTL using high-level synthesis (HLS) tools. The accelerator relies on a processor for run-time configuration, which defines key parameters such as input and output channels, kernel size, stride, and memory addresses for weights, inputs, and outputs. After configuration, it operates autonomously by accessing data directly from external memory. This configuration-execution process is repeated for each CNN layer while reusing the same hardware resources. Its architecture is composed of five main components: processing elements (PEs) that perform MACC operations followed by bias addition and ReLU activation; a memory controller for external memory communication; AMBA AXI interfaces, including an AXI master for memory access and an AXI-Lite interface for configuration; cache memories, which include the input cache (ICache) for pixels and feature maps, the weight cache (WCache) partitioned per PE for storing the model weights, the output cache (OCache) for intermediate feature maps, and the global cache (GCache) for accumulating results of global pooling operations; and an address generator, responsible for memory address computation. In this work, the accelerator is configured with two PEs, uses 8-bit integer representation for inputs and weights, and 16-bit representation for final outputs. Fig. 4 illustrates the location of the ZynqNet accelerator within the proposed SoC architecture.

RISC-V RVX processor

The RISC-V RVX processor is a lightweight and open-source soft-core IP described in Verilog at the register-transfer level (RTL). The RVX implements the RV32I integer in struction set and supports the Zicsr (Calçada, 2025) and Zmmul (Calçada, 2026) extensions. The Zicsr extension enables access to control and status registers (CSRs) for system-level management, while Zmmul provides hardware support for integer multiplication, improving the performance of

arithmetic-intensive operations. It relies on an interconnection bus for both instruction and data transfer, supporting communication with various peripherals such as UART, GPIO, SPI, I2C, timers, and memory interfaces. Additionally, an AXI-Lite communication module can be incorporated to simplify its integration into system-on-chip (SoC) platforms. In this work, two RVX configurations are evaluated in bare-metal mode: a baseline version supporting RV32I+Zicsr, and an extended variant that incorporates the RV32I+Zicsr+Zmmul instruction set. Within the proposed system, the RVX processor is executing at 40 MHz and acts as the main control unit, coordinating data transfers, configuring the hardware accelerator, and managing the execution flow of convolutional neural network (CNN) inference tasks.

RISC-V RVX-ZynqNet SoC

The proposed system integrates a RISC-V RVX processor with the ZynqNet accelerator within an FPGA-based system on-chip (SoC), as illustrated in Fig. 4. The design is implemented on an AMD Xilinx Artix-7 FPGA (xc7a200tsbg484-1), available on the Nexys Video platform. A Xilinx Clock Manager (MMCM) is used to generate two clock domains: 40 MHz for the RISC-V RVX core bus domain and 100 MHz for the AXI4 AMBA domain, the latter integrating two Xilinx AXI4 Crossbar IPs to manage communication between the ZynqNet accelerator, the shared Xilinx BRAM memory, and the processor. The first crossbar connects the ZynqNet accelerator to the shared memory, enabling low latency data access during CNN processing. The second crossbar provides interconnection between the RVX processor and the accelerator, while also enabling the processor to access the shared memory through the interconnection of both crossbars. The RVX processor operates over the internal bus for instruction, data, and peripheral communication, whereas the accelerator and memory subsystem communicate through the AXI4 domain. The shared memory stores the weights and bias parameters of the implemented CNN model, a set of 18 input images of size 32×32×3 used for evaluation, and the final classification results.

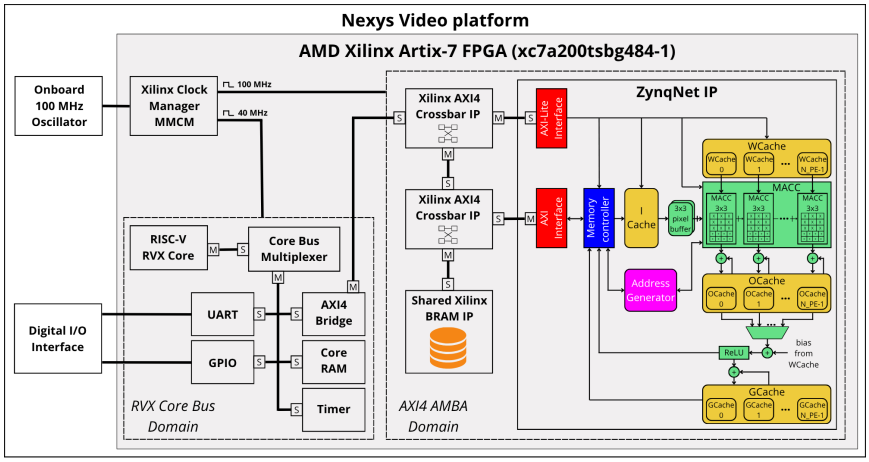

Fig. 4 RISC-V RVX-ZynqNet SoC hardware architecture

Experimental Results and Discussion

CNN model training

The CNN models were trained using PyTorch and the Brevitas library, enabling quantization-aware training (QAT). The training process was performed using the following configuration: Adam optimizer, cosine learning rate scheduler, Cross-Entropy loss function, a learning rate of 10^{-4}, batch size of 100, and 2000 epochs. Among all training epochs, the model that achieved the highest classification accuracy was selected for subsequent evaluation. The quantization scheme is inherently defined as part of the model architecture through the use of quantized convolutional layers, allowing control over weight and activation bit-widths for training. In this work, an 8-bit

quantization scheme is adopted, and no scaling factor is applied to the ReLU activation function.

The classification performance of the evaluated CNN models is summarized in Tables 1 and 2, with the corresponding confusion matrices shown in Fig. 5 and Fig. 6. For the Soybean Seeds dataset, CNV2 achieves an accuracy of 0.74, showing limitations in the Damaged class (recall of 0.62). As model complexity increases, accuracy improves in CNV3 and CNV4, reaching 0.86 and 0.95, respectively, while CNV5 achieves 0.96, indicating better accuracy with a reduced computational cost in terms of parameters and MACCs. The confusion matrices in Fig. 5 show a reduction in misclassifications as the depth increases. For the Coffee Bean dataset, CNV2 achieves 0.87 accuracy but shows low recall for the Dark class (0.58). CNV3 improves accuracy to 0.93, while CNV4 reaches near perfect classification accuracy (~1.00), as shown in Fig. 6. The corresponding confusion matrices show also a consistent reduction in class-level misclassifications across models. For both datasets, CNN models with an accuracy higher than 90% were considered for FPGA implementation.

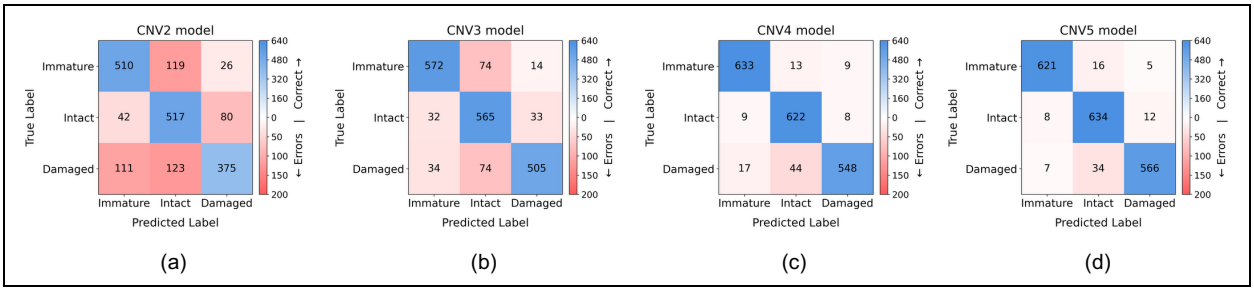

Fig. 5 CNN confusion matrices from Soybean Seeds dataset: (a) CNV2 model, (b) CNV3 model, (c) CNV4 model, and (d) CNV5 model

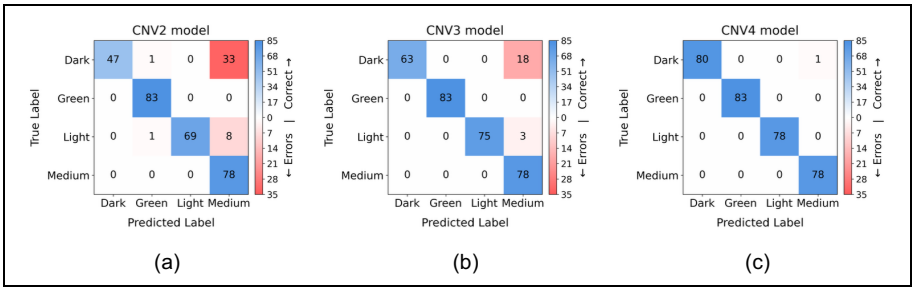

Fig. 6 CNN confusion matrices from Coffee Bean dataset: (a) CNV2 model, (b) CNV3 model, and (c) CNV4 model

Results on SRAM-based FPGA implementation: Area, Performance and Power

The hardware results are summarized in Tables 3, 4, and 5, showing a clear trade-off between hardware cost, power, accuracy and performance across CNN models, RISC-V ISAs, and datasets employed. From the FPGA area results (Table 3), the Zmmul extension increases DSP usage due to the inclusion of hardware multiplication support. CNV4 requires more Block RAMs (BRAMs) resources than CNV3 and CNV5, mainly due to its higher number of parameters. The number of Look-Up Tables (LUTs) and D flip-flops (DFFs) remains relatively similar across all configurations. In power consumption (Table 4), the inclusion of the Zmmul extension, which increases DSP utilization, leads to a slight increase in dynamic power across all configurations. Regarding performance (Table 5), for the Soybean Seeds dataset, CNV5 with Zmmul achieved the best processing time with 157.8 frames per second (FPS) and 151 correct FPS (CFPS), while also achieving the highest accuracy (0.957). For the coffee bean dataset, CNV3 with Zmmul provides the highest CFPS (115.2) with competitive accuracy (0.93). Although CNV4 achieves the highest accuracy (~1.00), CNV3 offers a better balance between inference speed and computational cost for real-time deployment.

Table 3. FPGA area utilization.

DS[(a)]	Model	RISC-V ISA	LUT	DFF	BRAM	DSP	EB[(b)]
S	CNV4	RV32I+Zicsr	12020	13797	86.5	23	2.96E+06
S	CNV5	RV32I+Zicsr	11439	13483	76.5	20	2.80E+06
S	CNV5	RV32I+Zicsr+Zmmul	11648	13658	76.5	32	2.89E+06
C	CNV3	RV32I+Zicsr	11485	13350	77.5	20	2.78E+06
C	CNV3	RV32I+Zicsr+Zmmul	11697	13525	77.5	32	2.88E+06
C	CNV4	RV32I+Zicsr	12020	13797	86.5	23	2.96E+06
C	CNV4	RV32I+Zicsr+Zmmul	12231	13972	86.5	35	3.05E+06

Notes: DS[(a)]: S = Soybean Seeds, C = Coffee Bean; EB[(b)]: Essential bits.

Table 4. FPGA power consumption.

DS[(a)]	Model	RISC-V ISA	Dynamic Power (W)
S	CNV4	RV32I+Zicsr	0.294
S	CNV5	RV32I+Zicsr	0.300
S	CNV5	RV32I+Zicsr+Zmmul	0.325
C	CNV3	RV32I+Zicsr	0.333
C	CNV3	RV32I+Zicsr+Zmmul	0.363
C	CNV4	RV32I+Zicsr	0.294
C	CNV4	RV32I+Zicsr+Zmmul	0.347

Notes: DS[(a)]: S = Soybean Seeds, C = Coffee Bean.

Table 5. Performance evaluation.

DS[(a)]	Model	RISC-V ISA	Time (ms)	FPS	Accuracy	CFPS
S	CNV4	RV32I+Zicsr	12.75	78.42	0.947	74.30
S	CNV5	RV32I+Zicsr	6.85	146.01	0.957	139.72
S	CNV5	RV32I+Zicsr+Zmmul	6.34	157.80	0.957	151.00
C	CNV3	RV32I+Zicsr	8.79	113.77	0.93	106.30
C	CNV3	RV32I+Zicsr+Zmmul	8.11	123.29	0.93	115.20
C	CNV4	RV32I+Zicsr	12.73	78.58	1.00	78.33
C	CNV4	RV32I+Zicsr+Zmmul	11.78	84.92	1.00	84.65

Notes: DS[(a)]: S = Soybean Seeds, C = Coffee Bean.

Conclusion and Future work

Experimental results using Soybean Seeds and Coffee Bean classification tasks demonstrated that the proposed approach achieves high accuracy while maintaining a favorable trade off in terms of area, power consumption, and execution time. The evaluation of different CNN depths and RISC-V ISA con figurations revealed important design considerations, showing that modest architectural enhancements, such as the inclusion of hardware multiplication support, can significantly improve inference throughput with limited overhead. These results reinforce the importance of joint optimization across software and hardware when targeting efficient edge AI systems. Overall, this work contributes a reproducible and scalable framework that bridges deep learning model design and FPGA-based system implementation using open-source RISC-V architectures. The proposed methodology is not limited to agricultural applications and can be extended to a wide range of embedded vision tasks. As future work, we plan to investigate dynamic precision scaling, support for more complex neural network models, and the incorporation of fault-tolerance mechanisms to enable reliable operation in harsh environments,

further expanding the applicability of FPGA-based edge AI systems.

Acknowledgments

This work was financed in part by the Center for Embedded Devices and Research in Digital Agriculture (CEDRA) under Grant 056/2023; Office of Naval Research Global (ONR) under Grant N629092212014; Coordination for the Improvement of Higher Education Personnel, Brazil (CAPES)-Finance Code 001, Council for Technological and Scientific Development, Brazil (CNPq); and Rio Grande do Sul Research Foundation, Brazil (FAPERGS).

References

Alam, S., C. Yakopcic, et al. 2024. Survey of Deep Learning accelerators for edge and emerging computing. *Electronics* 13(15). doi:10.3390/electronics13152988.

B, J. and L.S. Kumar. 2022. Implementation of convolution layer in FPGA for disease classification in tomato leaves. In: *2022 IEEE 19th India Council International Conference (INDICON)*. pp. 2023–2028.

Calçada, R. 2025. RVX: RISC-V soft-core processor (RV32I+Zicsr support). Available online: https://github.com/rafaelcalcada/rvx.

Calçada, R. 2026. RVX: RISC-V soft-core processor (RV32I+Zicsr+Zmmul support). Available online: https://github.com/rafaelcalcada/rvx/tree/dev.

Daghero, F., D.J. Pagliari and M. Poncino. 2021. Chapter eight - energy-efficient Deep Learning inference on edge devices. In: S. Kim and G.C. Deka, editors, *Hardware Accelerator Systems for Artificial Intelligence and Machine Learning*, ser. *Advances in Computers*. Elsevier. vol. 122, p. 247–301.

Ely, A.F., A. Carlos, S. Beck, et al. 2025. Implementation of an FPGA-based CNN accelerator for soybean seed quality analysis. In: *2025 IEEE Latin Conference on IoT (LCIoT)*. pp. 286–289.

Gschwend, D. 2016. ZynqNet: An FPGA-accelerated embedded Convolutional Neural Network. Master's thesis. Available online: https://github.com/dgschwend/zynqnet.

Kong, H., Z. Wei, et al. 2025. A low-power RISC-V based reconfigurable CNN accelerator for on-chip inference in wireless communication edge devices. In: *2025 3rd International Conference on Ubiquitous Communication (Ucom)*. pp. 272–276.

Kulkarni, P.R., M. M, et al. 2025. FPGA-based hardware accelerator with RISC-V processor for real-time CNN inference on ECG signals. In: *2025 9th International Conference on Computational System and Information Technology for Sustainable Solutions (CSITSS)*. pp. 1–6.

Li, X., H. Huang, et al. 2022. A hardware-efficient computing engine for FPGA-based deep Convolutional Neural Network accelerator. *Microelectronics Journal* 128: 105547. doi:10.1016/j.mejo.2022.105547.

Lin, W., Y. Fu, et al. 2023. SoybeanSeeds. *Mendeley Data*, V5. doi:10.17632/v6vzvfszj6.6.

Ontoum, S., T. Khemanantakul, et al. 2022. Coffee Roast Intelligence. *arXiv preprint arXiv:2206.01841*. Available online: https://arxiv.org/abs/2206.01841.

Oubabas, S., H.A. Issad, et al. 2024. Pests classification based on Convolutional Neural Networks in smart agriculture. In: *2024 International Conference of the African Federation of Operational Research Societies (AFROS)*. pp. 56–61.

Ratanpara, J. and K. Vaghela. 2025. Deep Learning in agriculture: Cotton leaf disease identification using CNNs. In: *2025 International Conference on Automation and Computation (AUTOCOM)*. pp. 1452–1456.

Shah, P., G. Rathod, et al. 2023. Plant leaf disease classification using Convolutional Neural Network on FPGA. In: *2023 International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*. pp. 307–311.

Sri, B.R. and T.S. Balakrishnan. 2024. Classification of pests in agricultural farms using Convolutional Neural Network compared to Artificial Neural Network. In: *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. pp. 9692–9695.

Sze, V., Y.-H. Chen, et al. 2017. Efficient processing of Deep Neural Networks: A tutorial and survey. *Proceedings of the IEEE* 105(12): 2295–2339.

Veiga, V.K., F. Benevenuti, et al. 2026. Power-performance analysis of a softcore RISC-V SoC with CNN accelerators for in-orbit computing. In: *2026 IEEE 17th Latin America Symposium on Circuits and System (LASCAS)*. pp. 1–5.

Wang, H., D. Li and T. Isshiki. 2023. Reconfigurable CNN accelerator embedded in instruction extended RISC-V core. In: *2023 6th International Conference on Electronics Technology (ICET)*. pp. 945–954.

Wei, C. and W. Feng. 2024. A review of Convolutional Neural Networks in the field of agricultural pest and disease recognition. In: *2024 14th International Conference on Information Technology in Medicine and Education (ITME)*. pp. 121–124.