RYv8_BNH: A 10 μs Ultra-Low-Power Binarized YOLOv8 Accelerator for Smart Home Edge AI

Ching-Han Chen*
Department of Computer Science and Information Engineering, National Central University
Taoyuan, Taiwan
pierre@g.ncu.edu.tw

Bo-Jia Chen
Department of Computer Science and Information Engineering, National Central University
Taoyuan, Taiwan
vincent19980324@gmail.com

Yu-An Shen
Department of Computer Science and Information Engineering, National Central University
Taoyuan, Taiwan
scckate444@gmail.com

***Abstract*—Real-time visual sensing is critical for smart home applications such as elderly fall detection and security monitoring, where edge devices must operate under strict constraints on latency, memory, and power consumption. This paper presents RYv8_BNH, an ultra-low-power binarized YOLOv8-based object detection accelerator optimized for resource-constrained edge AI systems. The proposed model integrates ReActNet-inspired binarization, structured pruning, and a lightweight detection head, reducing the model size to 0.46M parameters while maintaining competitive accuracy. A single-scale (P4-only) detection strategy is further adopted to minimize computation and memory access overhead. The hardware accelerator is implemented on an Intel Agilex 5 FPGA, achieving 10.08 μs inference latency, representing over 300× speedup compared to software execution. Experimental results on the Pascal VOC dataset demonstrate 39% mAP50–95, while energy consumption is reduced to the order of tens of μJ per inference. These results demonstrate that RYv8_BNH achieves an effective balance among accuracy, latency, and energy efficiency, making it highly suitable for always-on smart home edge AI applications.**

Keywords—Edge AI, FPGA accelerator, binarized neural networks, YOLOv8, low-latency inference, energy-efficient computing, smart home monitoring.

I. INTRODUCTION

With the rapid advancement of smart home technologies, real-time visual sensing has become a fundamental component for applications such as security monitoring and elderly fall detection. These systems are typically deployed on resource-constrained edge devices, which must operate continuously under strict limitations on computation, memory, and power consumption. Therefore, achieving an effective balance among detection accuracy, latency, and energy efficiency is a critical challenge for edge AI systems [15].

Deep convolutional neural networks (CNNs) have significantly improved object detection performance in recent years. Early approaches such as R-CNN, Fast R-CNN, and Faster R-CNN adopted multi-stage pipelines that achieved high accuracy but suffered from high computational complexity and latency [1]–[3]. To address these limitations, the YOLO (You Only Look Once) framework introduced a one-stage detection paradigm, enabling real-time object detection with improved efficiency [4]. Recent developments such as YOLOv8 further enhance detection performance through architectural improvements and training strategies [5], [11]. However, these models still require substantial computational resources, making them difficult to deploy directly on low-power edge devices.

To overcome these constraints, model compression techniques such as pruning and quantization have been widely explored. In particular, binarized neural networks (BNNs) significantly reduce computational complexity by representing weights and activations with 1-bit precision, enabling efficient bitwise operations [6]. Representative works such as Bi-Real Net and ReActNet improve the representational capability of BNNs through advanced activation functions and architectural design [7], [8]. Despite these advancements, applying binarization to object detection remains challenging due to the need for precise spatial localization.

In parallel, FPGA-based accelerators have been investigated as a promising solution for low-latency and energy-efficient edge AI deployment. Prior works have demonstrated high-throughput implementations of YOLO-based models using customized dataflows and hardware optimizations [9], [10]. More recent studies have explored lightweight object detection models such as TinyissimoYOLO for ultra-low-power edge devices [12], [13]. However, most existing designs primarily focus on throughput optimization or classification tasks, while overlooking the stringent latency and energy constraints required for always-on smart home applications.

To address these challenges, this paper proposes RYv8_BNH, an ultra-lightweight binarized YOLOv8-based object detection architecture and its corresponding hardware accelerator. The proposed approach integrates binarization, structured pruning, and a novel lightweight detection head to significantly reduce model complexity while maintaining competitive accuracy. In addition, a single-scale detection strategy is adopted to minimize computation and memory access overhead, making it particularly suitable for indoor monitoring scenarios such as fall detection.

Unlike existing approaches, the proposed design emphasizes latency-driven edge AI optimization, targeting microsecond-level inference rather than conventional throughput metrics. By jointly optimizing model architecture and hardware execution, the proposed system achieves a new trade-off point in terms of accuracy, latency, and energy efficiency, enabling real-time always-on operation in smart home environments.

979-8-3315-5062-2/26/$31.00 ©2026 IEEE

II. Related Work

This section reviews the key techniques related to the proposed system, including object detection architectures, binarized neural networks, and FPGA-based accelerators for edge AI deployment.

A. Object Detection Architectures

Deep learning-based object detection has evolved significantly over the past decade. Early approaches such as R-CNN, Fast R-CNN, and Faster R-CNN adopt multi-stage pipelines to achieve high detection accuracy, but suffer from high computational complexity and latency, making them unsuitable for real-time edge applications [1]–[3].

To address these limitations, the YOLO (You Only Look Once) framework introduces a one-stage detection paradigm that directly predicts bounding boxes and class probabilities, significantly improving inference speed [4]. Recent advancements such as YOLOv8 further enhance detection performance through architectural refinements and training strategies [5], [11]. Despite these improvements, the increasing model complexity and parameter size still pose challenges for deployment on resource-constrained edge devices.

Lightweight object detection models have also been explored to address these issues. For instance, TinyissimoYOLO demonstrates efficient deployment on ultra-low-power systems through model simplification and quantization techniques [12], [13]. However, such models often involve trade-offs between accuracy and efficiency, especially in complex detection tasks.

B. Binarized Neural Networks

Binarized neural networks (BNNs) have emerged as an effective approach to reduce computational complexity and memory usage by representing weights and activations with 1-bit precision. XNOR-Net is one of the earliest works that enables efficient convolution operations using bitwise logic [6]. Subsequent approaches such as Bi-Real Net and ReActNet improve the representational capability of BNNs through residual connections and advanced activation functions [7], [8].

Recent studies have further explored the application of BNNs in edge intelligence scenarios, demonstrating significant advantages in energy efficiency and hardware friendliness [14]. However, most existing BNN designs focus on classification tasks, while extending binarization to object detection remains challenging due to the need for accurate spatial localization and multi-scale feature representation.

C. FPGA-Based Edge AI Accelerators

FPGA-based accelerators have been widely studied for deploying deep learning models in edge environments due to their flexibility and energy efficiency. Prior works such as [9] and [10] demonstrate high-throughput implementations of YOLO-based models by exploiting parallelism and customized dataflows. These designs primarily focus on maximizing throughput and computational efficiency.

More recent research has explored lightweight and quantized object detection models for ultra-low-power edge devices, such as TinyissimoYOLO, which enables on-device inference with reduced energy consumption [12], [13]. Nevertheless, most existing FPGA-based designs emphasize throughput-oriented optimization (e.g., frames per second) rather than latency-critical scenarios. In smart home applications such as fall detection, systems must operate in an event-driven and always-on manner, where ultra-low latency and energy efficiency are more critical than throughput.

D. Research Gap

Based on the above discussion, existing works exhibit three major limitations. First, state-of-the-art object detection models provide high accuracy but remain computationally expensive for edge deployment. Second, although BNNs offer significant efficiency advantages, their application to object detection is still limited due to accuracy degradation. Third, FPGA-based accelerators often prioritize throughput rather than latency and energy efficiency required for always-on edge AI systems.

To address these challenges, this work proposes a binarized and hardware-efficient YOLOv8-based architecture that jointly optimizes model design and hardware execution. The proposed approach aims to achieve a balanced trade-off among accuracy, latency, and energy consumption, enabling practical deployment in smart home edge environments.

III. Proposed Method

This section presents the proposed RYv8_BNH architecture, a binarized and hardware-efficient object detection model designed for ultra-low-latency edge AI applications. The design emphasizes joint optimization of model structure and computational efficiency to meet the strict constraints of real-time smart home systems, including limited memory, low power consumption, and always-on operation.

A. Overview of RYv8_BNH Architecture

The overall architecture of RYv8_BNH is illustrated in Fig. 1. The model follows the general YOLO framework, consisting of three main components: Backbone, Neck, and Head. However, unlike standard YOLOv8, the proposed design introduces aggressive model simplifications and binarization strategies to reduce computational complexity.

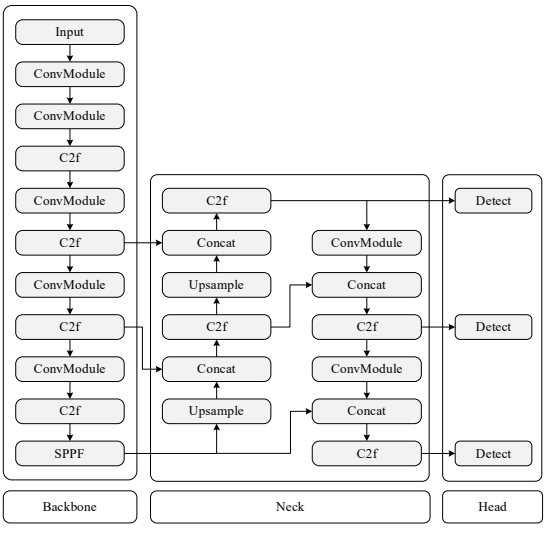


Fig. 1. RYv8_BNH overall architecture.

Specifically, the backbone is redesigned using binarized convolutional blocks inspired by ReActNet, while redundant layers in both the backbone and neck are removed through structured pruning. In addition, the multi-scale detection mechanism is simplified into a single-scale (P4-only) detection pipeline, significantly reducing feature map movement and memory access overhead.

This design is motivated by the observation that indoor smart home applications, such as fall detection, typically involve limited object scale variation, making full multi-scale detection unnecessary. By focusing on a single representative feature scale, the proposed architecture achieves a more efficient balance between accuracy and computational cost.

B. Binarized Backbone Design

To enable efficient computation on resource-constrained hardware, the backbone network is implemented using binarized neural network (BNN) techniques. As shown in Fig. 2, conventional convolutional layers are replaced with ReActNet-inspired BasicBlocks, where both weights and activations are quantized to 1-bit precision.

Compared to standard convolution, binarized operations can be implemented using bitwise XNOR and bit-count operations, significantly reducing computational complexity and memory bandwidth requirements. However, aggressive binarization often leads to performance degradation.

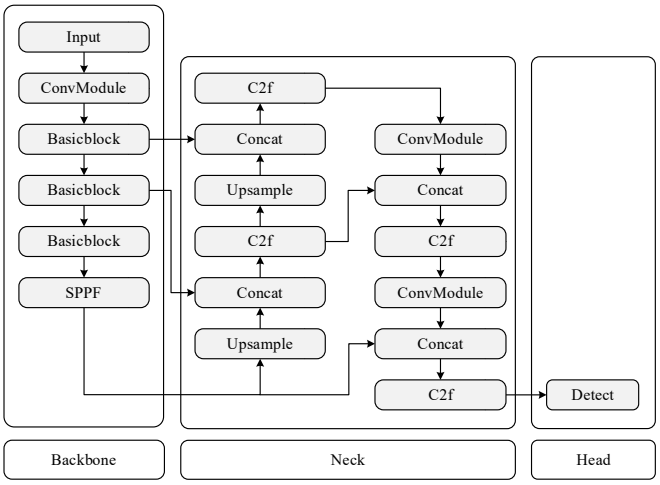


Fig. 2. Binarized backbone architecture of RYv8_BNH.

To mitigate this issue, the proposed design adopts modified BasicBlocks by removing the initial RSign operation and preserving residual connections, which helps maintain feature representation capability. In addition, pruning is applied to eliminate redundant channels and layers, further reducing model size while minimizing accuracy loss.

Through these modifications, the backbone achieves substantial compression while preserving sufficient feature extraction capability for object detection tasks.

C. Single-Scale Detection Strategy

Unlike conventional YOLO architectures that rely on multi-scale feature fusion (e.g., P3, P4, P5), the proposed model adopts a single-scale (P4-only) detection strategy.

This design choice is based on the target application scenario. In indoor environments, objects of interest (e.g., human bodies for fall detection) typically occupy a relatively consistent spatial scale. Therefore, using multiple detection scales introduces unnecessary computation and memory overhead without significant accuracy gains.

By retaining only the P4 feature map for detection, the proposed model reduces the number of convolution operations, memory accesses, and data movement between layers. This simplification is particularly beneficial for hardware acceleration, where memory bandwidth and latency are critical bottlenecks.

D. Lightweight Detection Head (RYv8_BNH_LConv)

The detection head is further optimized through the proposed RYv8_BNH_LConv module, as illustrated in Fig. 3.

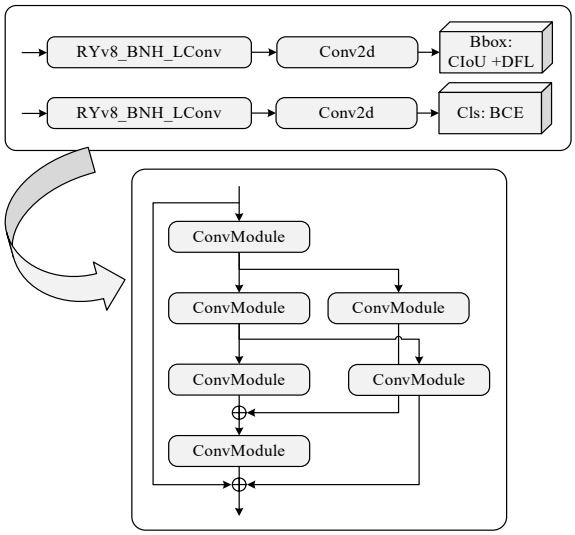


Fig. 3. RYv8_BNH_LConv Architecture

In standard YOLOv8, each detection branch typically consists of multiple convolution layers for feature refinement. These layers contribute significantly to computational cost, especially in edge deployment scenarios.

To address this issue, the proposed LConv module replaces multiple convolution layers with a compact structure combining 1×1 and 3×3 convolutions with shortcut connections. This design reduces the number of parameters and multiply–accumulate (MAC) operations while preserving feature transformation capability.

In addition, the use of shortcut connections improves gradient flow and stabilizes training, which is particularly important for binarized networks. The simplified head structure also reduces memory access frequency, further improving hardware efficiency.

E. Model Optimization Pipeline

The overall model is optimized through a combination of binarization, pruning, and architectural simplification. The design process consists of the following key steps:

1. Replacement of standard convolution layers with binarized BasicBlocks.

2. Removal of redundant modules in the backbone and neck through structured pruning.
3. Adoption of a single-scale detection pipeline to reduce computation.
4. Introduction of the lightweight LConv detection head.

The model is trained on the Pascal VOC dataset using standard backpropagation. After training, the model is exported and quantized for efficient deployment.

Through this co-design of architecture and computation, the proposed RYv8_BNH model achieves a compact structure with significantly reduced latency and energy consumption, making it suitable for real-time edge AI applications.

IV. Experimental Results

This section evaluates the proposed RYv8_BNH model in terms of model efficiency, detection accuracy, and hardware acceleration performance. The experiments are conducted to validate the effectiveness of the proposed design for real-time edge AI applications.

A. Experimental Setup

The software model is trained on the Pascal VOC dataset with an input resolution of 256×256. The training process is conducted using a standard deep learning framework with 1000 epochs to ensure convergence.

For hardware evaluation, the accelerator is implemented on an Intel Agilex 5 FPGA platform. The synthesized design is evaluated in terms of resource utilization, operating frequency, and inference latency.

To ensure fair comparison, all models are evaluated under consistent input settings and experimental conditions.

B. Model Efficiency and Accuracy

The efficiency of the proposed model is first evaluated in terms of parameter size. As shown in Table I, RYv8_BNH achieves a model size of only 0.46M parameters, which is significantly smaller than YOLOv5n (2.6M) and YOLOv8n (3.1M), representing reductions of approximately 82% and 85%, respectively.

Compared to lightweight models such as TinyissimoYOLO variants, the proposed model still achieves a further reduction in parameter size, demonstrating superior compression capability.

TABLE I. Model Size Comparison

Model	Parameters (M)
YOLOv5n	2.6
YOLOv8n	3.1
TinyissimoYOLO (v5-S)	0.62
TinyissimoYOLO (v8-S)	0.71
RYv8_BNH (Proposed)	**0.46**

In terms of detection performance, Table II presents the mAP50–95 comparison. Despite its extremely compact size, RYv8_BNH achieves 39% mAP50–95, which is comparable to TinyissimoYOLO (v8-S) and outperforms v5-S variants.

TABLE II. Detection Accuracy Comparison

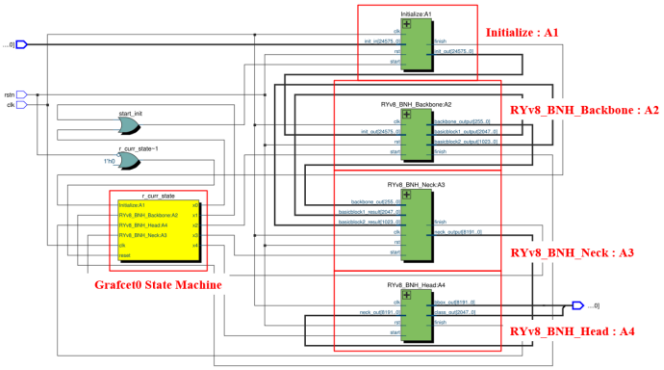
Model	mAP50–95 (%)
TinyissimoYOLO (v5-S)	35
TinyissimoYOLO (v8-S)	39
RYv8_BNH (Proposed)	**39**

These results indicate that the proposed model successfully maintains competitive detection accuracy while significantly reducing model complexity, demonstrating an effective accuracy–efficiency trade-off.

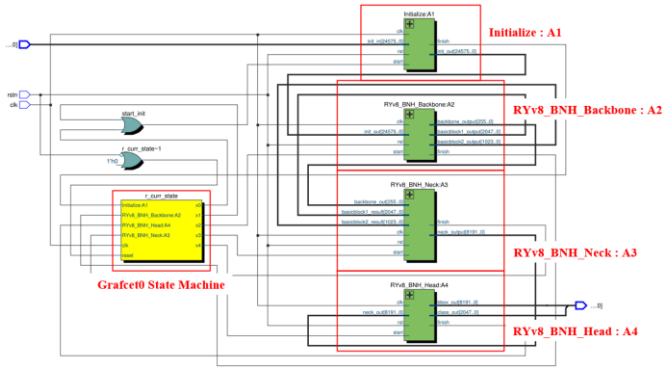
C. Hardware Resource Utilization

To further illustrate the hardware realization of the proposed design, Fig. 4 presents the synthesized hardware architecture of the RYv8_BNH accelerator. The major computation modules corresponding to the backbone, neck, and detection head are mapped to dedicated hardware blocks, forming a structured dataflow for feature extraction and prediction.

For clarity, control-related annotations and intermediate state labels from the original design are removed, highlighting the overall module organization and dataflow. The resulting architecture demonstrates a modular and parallel processing structure, enabling efficient execution of binarized convolution and feature processing operations on FPGA.



(a)



(b)

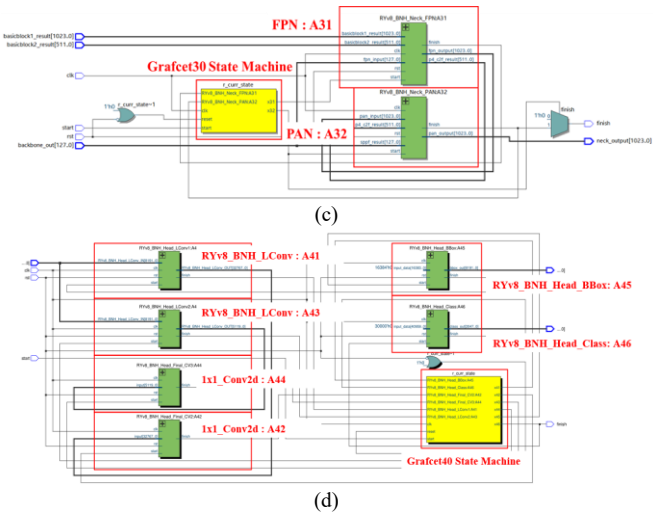


Fig. 4. Synthesized hardware architecture of the proposed RYv8_BNH accelerator, illustrating the modular organization and dataflow of backbone, neck, and detection head components (simplified for clarity). (a) RYv8_BNH circuit block (b) RYv8_BNH_Backbone circuit block (c) RYv8_BNH_Neck circuit block (d) RYv8_BNH_Head circuit block

The synthesized hardware resource utilization is summarized in Table III. The proposed design consumes 603,572 logic elements and 433,350 registers on the FPGA platform, operating at a maximum frequency of 25.31 MHz.

Although the resource usage is relatively high due to parallel processing, the design prioritizes latency reduction, which is critical for real-time edge applications.

TABLE III. HARDWARE RESOURCE UTILIZATION

Resource	Usage
Logic Elements (LEs)	603,572
Registers	433,350
Max Frequency	25.31 MHz

D. *Inference Latency and Acceleration Performance*

The inference latency comparison between software and hardware implementations is shown in Table IV. In the PyTorch environment, the model requires approximately 3400 µs per inference. In contrast, the FPGA-based accelerator completes inference in only 10.08 µs, corresponding to a speedup of over 300×.

TABLE IV. INFERENCE LATENCY COMPARISON

Execution Platform	Latency
PyTorch (CPU/GPU)	3400 µs
FPGA Accelerator (Proposed)	10.08 µs

This significant reduction in latency demonstrates the effectiveness of the proposed hardware-aware design. Unlike conventional throughput-oriented accelerators, the proposed system is optimized for ultra-low latency, enabling near real-time response in event-driven smart home scenarios.

E. *Qualitative Results*

Fig. 5 shows qualitative detection results in representative smart home scenarios, including doorbell monitoring and abnormal activity detection. The proposed model is able to correctly detect human presence and motion patterns under typical indoor conditions.



Fig. 5. Qualitative detection results of the proposed RYv8_BNH model in representative smart home scenarios: (a) visitor detection in a doorbell monitoring application, and (b) abnormal human activity detection for indoor safety monitoring.

These results further validate the applicability of the proposed system for practical smart home deployment, particularly in safety-critical applications such as elderly fall detection.

F. *Discussion*

The experimental results demonstrate that the proposed RYv8_BNH achieves a favorable trade-off among model size, detection accuracy, and inference latency.

Compared to existing lightweight detection models, the proposed approach provides:

1. significantly smaller model size,
2. comparable detection accuracy, and
3. substantially lower inference latency.

This combination of characteristics makes the proposed design particularly suitable for always-on edge AI systems, where both responsiveness and energy efficiency are critical.

In addition, the use of binarized operations and simplified architecture suggests strong potential for further optimization in energy consumption, which is an important direction for future work.

V. CONCLUSION

This paper presented RYv8_BNH, an ultra-lightweight and binarized YOLOv8-based object detection architecture and its corresponding FPGA accelerator for real-time smart home edge AI applications. By integrating binarized neural network design, structured pruning, and a lightweight detection head, the proposed model significantly reduces model complexity while maintaining competitive detection performance.

Experimental results demonstrate that the proposed approach achieves a compact model size of 0.46M parameters and maintains 39% mAP50–95 on the Pascal VOC dataset. Furthermore, the FPGA-based implementation achieves an inference latency of 10.08 μs, corresponding to over 300× speedup compared to software execution. These results indicate that the proposed design effectively balances accuracy, latency, and computational efficiency.

The proposed system is particularly suitable for always-on smart home applications, such as indoor monitoring and elderly fall detection, where real-time responsiveness and energy efficiency are critical.

Future work will focus on further reducing energy consumption, extending the design to more complex detection scenarios, and exploring deployment on emerging edge AI platforms.

ACKNOWLEDGEMENT

The authors would like to thank the National Science and Technology Council, Taiwan, R.O.C., for their financial support (Grant Nos. NSTC 112-2221-E-008 -029 -MY3).

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Columbus, OH, USA, Jun. 2014, pp. 580–587.

[2] R. Girshick, "Fast R-CNN," in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Santiago, Chile, Dec. 2015, pp. 1440–1448.

[3] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 779–788.

[5] Ultralytics, "YOLOv8," [Online]. Available: https://github.com/ultralytics/ultralytics

[6] M. Rastegari, V. Ordonez, J. Redmon and A. Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks," in Proc. Eur. Conf. Comput. Vis. (ECCV), Amsterdam, Netherlands, Sep. 2016, pp. 525–542.

[7] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu and K.-T. Cheng, "Bi-Real Net: Enhancing the Performance of 1-bit CNNs With Improved Representational Capability and Advanced Training Algorithm," in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, Sep. 2018, pp. 747–763.

[8] Z. Liu, Z. Shen, M. Savvides and K.-T. Cheng, "ReActNet: Towards Precise Binary Neural Network With Generalized Activation Functions," in Proc. Eur. Conf. Comput. Vis. (ECCV), Glasgow, UK, Aug. 2020, pp. 143–159.

[9] D. T. Nguyen, T. N. Nguyen, H. Kim and H.-J. Lee, "A High-Throughput and Power-Efficient FPGA Implementation of YOLO CNN for Object Detection," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 27, no. 8, pp. 1861–1873, Aug. 2019.

[10] Z. Wang, K. Xu, S. Wu, L. Liu, L. Liu and D. Chen, "Sparse-YOLO: Hardware/Software Co-Design of an FPGA Accelerator for YOLOv2," IEEE Access, vol. 8, pp. 116569–116585, 2020.

[11] Z. Fang et al., "YOLOv8n-RSDD: A High-Performance Low-Complexity Rail Surface Defect Detection Network," IEEE Access, vol. 12, pp. 196249–196265, 2024.

[12] J. Moosmann et al., "Flexible and Fully Quantized Lightweight TinyissimoYOLO for Ultra-Low-Power Edge Systems," IEEE Access, vol. 12, pp. 75093–75107, 2024.

[13] J. Moosmann, P. Bonazzi, Y. Li, S. Bian, P. Mayer, L. Benini, and M. Magno, "Ultra-Efficient On-Device Object Detection on AI-Integrated Smart Glasses with TinyissimoYOLO," in Computer Vision – ECCV 2024, Cham: Springer Nature Switzerland, 2025, pp. 262–280. doi: 10.1007/978-3-031-91989-3_17.

[14] F. Luo, S. Khan, Y. Huang and K. Wu, "Binarized Neural Network for Edge Intelligence of Sensor-Based Human Activity Recognition," IEEE Trans. Mobile Comput., vol. 22, no. 3, pp. 1356–1368, Mar. 2023.

[15] N. Kojima et al., "Real-Time Fall Detection in Smart Home Environments: A Review," IEEE Access, vol. 10, pp. 12245–12260, 2022.