IJETIR

ISSN (Online) 2583-0554

International Journal of Emerging Technologies and Innovative Research (IJETIR)

Volume 6, Issue 8, August 2026

Impact Factor: 5.731

Design and Implementation of an XNOR-Based Variable-Precision Object Detection Accelerator for Real-Time Edge AI Applications

Kothoju Udaya Kumar[1] and Eedula Vamshi[2]

Assistant Professor, ECE Department[1]

PG Scholar, ECE Department[2]

Annamacharya Institute of Technology & Sciences-Hyderabad, Telangana, India

uday.kothoju@gmail.com.

Abstract: *Real-time object detection at the edge requires high computational throughput under strict area, memory-bandwidth, and power constraints. This paper presents a hardware-efficient accelerator based on an XNOR-driven variable-precision architecture that combines a quantization-aware DenseToRes backbone with a 64-processing-element (PE) output-stationary datapath. The accelerator supports 1-bit × 1-bit, 2-bit × 1-bit, and 8-bit × 8-bit modes so that sensitive layers can retain higher precision while aggressively quantized layers exploit bitwise computation. Binary convolutions replace multiplication with XNOR and population-count operations; mixed-precision computation uses bit-sliced signed accumulation and shift-add recombination. The PE integrates variable-precision arithmetic, batch normalization, RPReLU activation, and requantization, while AXI-Stream, AXI-Lite, DMA, on-chip feature/weight memories, and a lightweight controller support system-level FPGA deployment. Architecture-level evaluation shows that a 512-bit feature path can carry 64 8-bit, 256 2-bit, or 512 1-bit activation values per transfer, corresponding to 4× and 8× payload increases at 2-bit and 1-bit precision. The precision-select arithmetic block packs 1, 32, and 64 parallel operation lanes in 8W/8A, 1W/2A, and 1W/1A modes, respectively. Bit-true validation over 20,000 randomized low-precision dot products produced zero arithmetic mismatches. These results demonstrate the computational and data-movement advantages of hardware/network co-design for resource-constrained edge AI. Detection mAP and post-route FPGA power/frequency values are not available in the supplied project data and are therefore not fabricated in this paper draft.*

Keywords: Edge AI, object detection accelerator, FPGA, XNOR, binarized neural network, variable precision, DenseToRes, processing element, quantization, AXI.

I. INTRODUCTION

Object detection is a fundamental computer-vision task that jointly determines object identity and spatial location. Modern detectors use convolutional backbones, multi-scale feature aggregation, and prediction heads to obtain high accuracy, but the resulting computation and memory traffic make direct deployment difficult on power-constrained edge systems. Two-stage methods such as Faster R-CNN prioritize accuracy through region proposal and classification stages, whereas one-stage methods such as YOLO and SSD reduce latency by predicting classes and bounding boxes in a single forward pass [4]–[10]. Even one-stage networks, however, require large numbers of multiply–accumulate (MAC) operations and repeated movement of weights and feature maps.

Field-programmable gate arrays (FPGAs) are attractive for embedded inference because they provide configurable parallelism, deterministic latency, and application-specific data paths. The principal bottlenecks are arithmetic cost and memory traffic. Conventional convolution engines depend on many fixed-point or floating-point multipliers, while external-memory accesses frequently dominate energy consumption. Quantization reduces both costs by decreasing operand width and enabling denser packing of values in on-chip memories and buses [11], [17]–[23].

Binarized neural networks (BNNs) represent the extreme form of quantization. In 1-bit layers, weights and activations are mapped to binary values, allowing multiplication to be replaced by XNOR followed by population count. This transformation dramatically increases bit-level parallelism and lowers storage demand [12], [13]. The disadvantage is representational loss: using a single bit throughout the network can reduce detection accuracy, particularly at the input, output, and information-sensitive transition stages. Dense and residual connections, generalized activations, and selective mixed precision have therefore emerged as important techniques for retaining information under aggressive quantization [14]–[16], [22].

The architecture investigated in this work addresses the accuracy-efficiency trade-off through a unified variable-precision accelerator. A compact network built from DenseToRes and transition layers preserves feature reuse, while a precision-select processing engine supports 1-bit, 2-bit, and 8-bit computation. A 64-PE array, output-stationary dataflow, on-chip buffering, and AXI-based system integration provide the parallelism and data reuse required for real-time edge deployment.

The contributions of the paper are:

A DenseToRes-based quantized feature-extraction network designed to limit information loss in binary and low-bit layers.

A unified precision-select arithmetic organization supporting 1W/1A, 1W/2A, and 8W/8A computation within a hardware-oriented convolution datapath.

A 64-PE output-stationary processor with local accumulation, batch normalization, RPReLU activation, quantization, pooling, feature/weight fetch units, and on-chip memories.

An AXI/DMA-based processing-system and programmable-logic integration strategy suitable for heterogeneous FPGA platforms.

Architecture-derived bandwidth and parallelism analysis plus bit-true randomized validation of the low-precision arithmetic core.

II. RELATED WORK AND RESEARCH GAP

A. Object Detection and Edge Deployment

Traditional object detection progressed from handcrafted Haar and HOG features to convolutional neural networks [1]–[3]. R-CNN, Fast R-CNN, and Faster R-CNN improved recognition quality through learned hierarchical features and region proposals [4]–[6]. YOLO and SSD subsequently reduced inference latency through one-stage prediction [7]–[10]. For edge use, however, the dominant convolution workload and feature-map movement remain expensive even when the detector is algorithmically compact.

B. Quantized and Binary Neural Networks

Integer-only quantization demonstrates that low-bit inference can reduce arithmetic and storage costs while maintaining useful accuracy [11]. BNNs further constrain activations and weights to one bit [12]. XNOR-Net showed that binary dot products can be executed using XNOR and bit counting [13]. Bi-Real Net, BinaryDenseNet, and ReactNet improve the representational capability of low-bit networks through identity paths, dense feature reuse, and specialized activation functions [14]–[16]. These methods motivate the DenseToRes strategy adopted in the present design.

C. FPGA CNN Accelerators

CNN accelerators such as Eyeriss, FINN, the DianNao family, and FPGA-specific convolution engines demonstrate the importance of dataflow, local reuse, and precision customization [17]–[21]. FINN is especially relevant because it maps binarized networks to streaming FPGA structures [18]. Hardware-aware quantization and bit-serial processing extend this direction by assigning different precisions to layers according to sensitivity and implementation cost [22], [23].

D. Research Gap

The reviewed literature reveals four persistent limitations: fixed-precision datapaths that cannot exploit layer sensitivity; accuracy degradation in fully binarized networks; high energy cost of external feature/weight transfers; and insufficient co-design between network topology and FPGA arithmetic. The proposed system addresses these gaps by combining DenseToRes feature preservation, variable precision, XNOR-centric low-bit computation, a 64-PE output-stationary engine, and tightly integrated on-chip memory and AXI communication.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

Issue in prior systems	Effect	Proposed response
Fixed precision	Over-computation in tolerant layers or accuracy loss under global binarization	Layer-selectable 1-, 2-, and 8-bit modes
Multiplier-heavy convolution	High area and power per MAC lane	XNOR/popcount or signed-add lanes in low precision
External-memory traffic	Latency and energy overhead	512-bit on-chip feature path and output-stationary reuse
Quantization information loss	Reduced detection representation	DenseToRes, residual paths, BN and RPReLU
Limited parallelism	Reduced frame throughput	64 parallel PEs and precision-dependent lane packing

III. PROPOSED NETWORK AND ACCELERATOR ARCHITECTURE

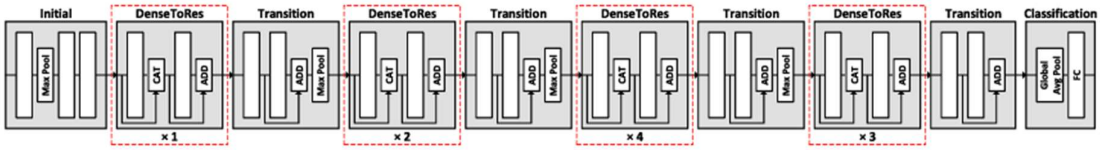

Fig. 1. Overall quantized object-detection backbone using an initial stage, repeated DenseToRes blocks, transition layers, and a final output stage.

A. Quantized Detection Backbone

The network begins with an initial feature-extraction stage that quickly reduces spatial resolution using strided convolution, max pooling, and depthwise processing. The backbone then alternates DenseToRes blocks and transition layers. This arrangement is intended to preserve informative features during aggressive quantization while controlling the growth of spatial and channel dimensions. Higher precision can be assigned to input/output-sensitive stages, whereas deeper feature-processing layers can use binary or mixed precision.

B. DenseToRes and Transition Blocks

A DenseToRes block combines dense feature preservation and residual learning. A binary convolution generates a transformed feature map that is normalized and activated. The transformed output is concatenated with the original feature map, projected through a 1 × 1 convolution, and added through a residual path. The dense branch preserves unmodified information, while the residual branch reduces optimization difficulty and compensates for quantization-induced information loss. Transition blocks refine features and reduce spatial resolution through normalized convolution, residual addition, and max pooling.

C. Convolution and Binary Inner Product

For an output channel o and spatial position (p,q), the conventional convolution can be written as

$$y\_o(p,q) = \Sigma\_c \Sigma\_i \Sigma\_j \; w\_o,c(i,j) \cdot x\_c(p+i,q+j)$$

For 1-bit weights and activations encoded as {-1,+1}, multiplication is equivalent to equality testing. If N bits participate in a dot product and M is the number of equal bit pairs, then

$$a \cdot w = 2 \cdot popcount(XNOR(a,w)) - N$$

The equation replaces N multipliers with N bitwise comparisons plus a population-count tree. For the 2-bit activation / 1-bit weight mode, each activation is decomposed into two bit planes. The weighted inner product is reconstructed using signed additions and a one-bit shift, enabling reuse of the binary-oriented datapath without a full 2-bit multiplier array.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

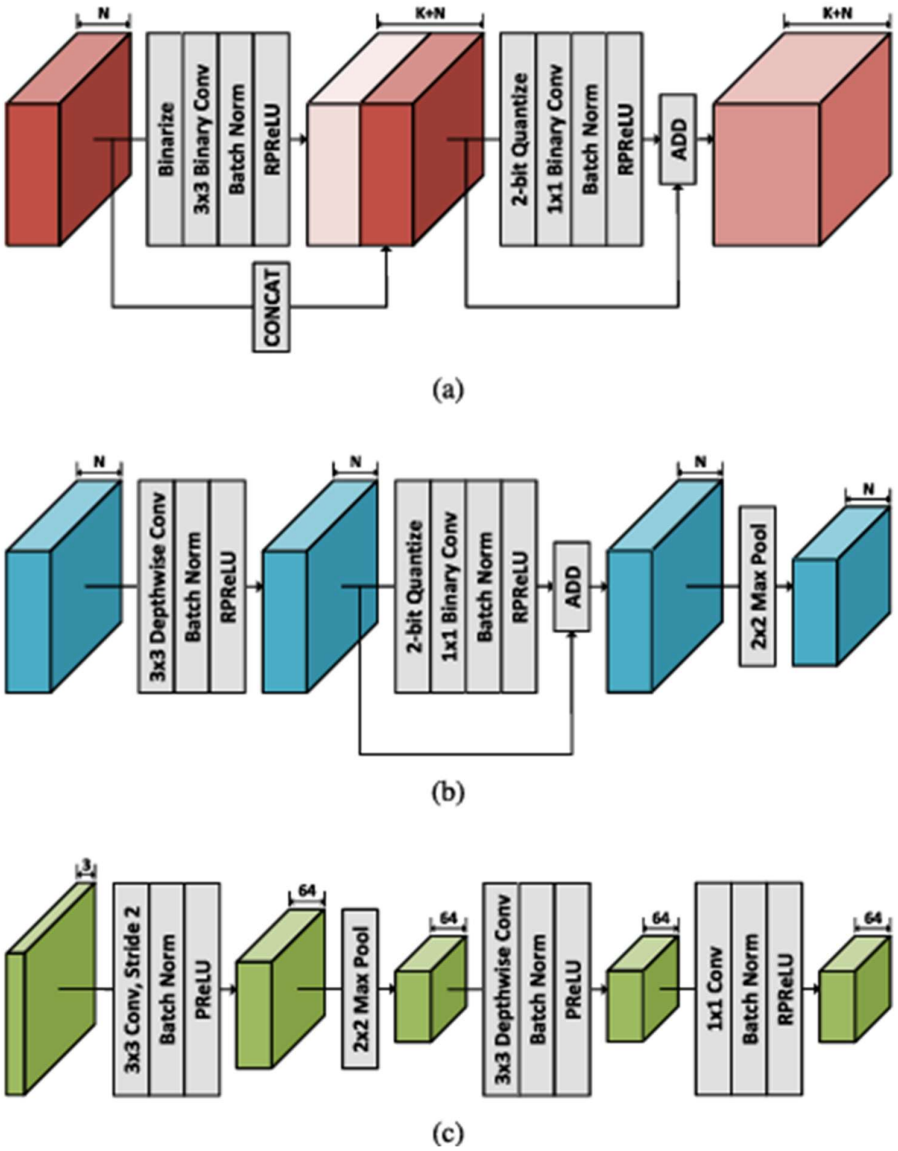

Fig. 2. Network building blocks: (a) DenseToRes block, (b) transition block, and (c) initial feature-extraction block.

D. Processor Architecture and Output-Stationary Dataflow

The accelerator contains feature-map memory, a 512-bit feature fetch path, weight and normalization/activation parameter memories, a precision-aware PE array, output feature buffering, max pooling, and centralized layer control. Sixty-four PEs compute different output channels concurrently. The output-stationary dataflow keeps partial sums local to the PE until an output value is complete, reducing repeated transfers of intermediate sums and improving arithmetic intensity.

The 512-bit feature path naturally supports precision-dependent packing. One transfer can carry 64 8-bit activations, 256 2-bit activations, or 512 1-bit activations. Therefore lower precision improves not only arithmetic density but also the effective useful-data bandwidth of the same physical interface.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

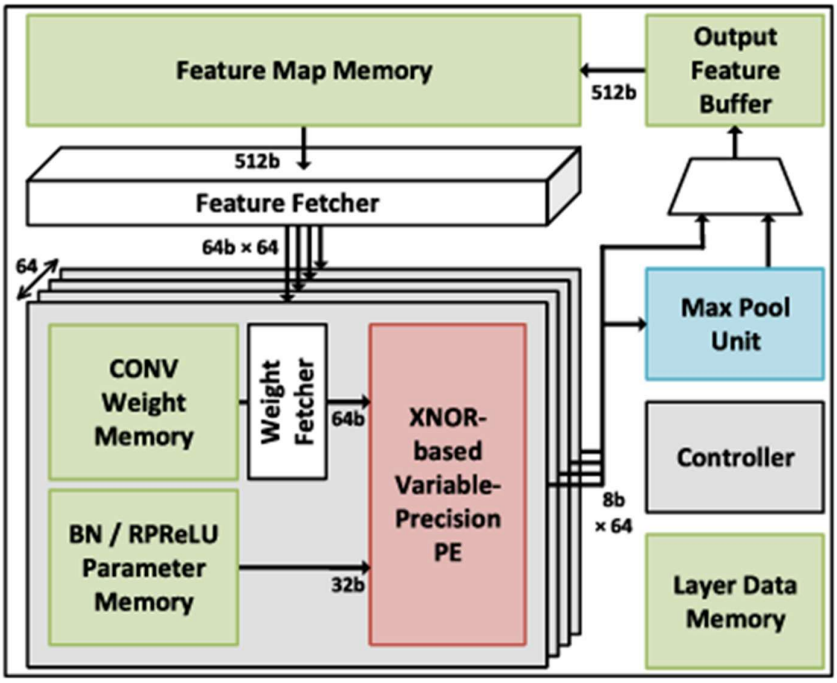

Fig. 3. Proposed processor architecture with feature-map memory, feature fetcher, weight/parameter memories, 64 variable-precision PEs, max-pooling block, output buffer, controller, and layer-data memory.

E. Variable-Precision Arithmetic Unit

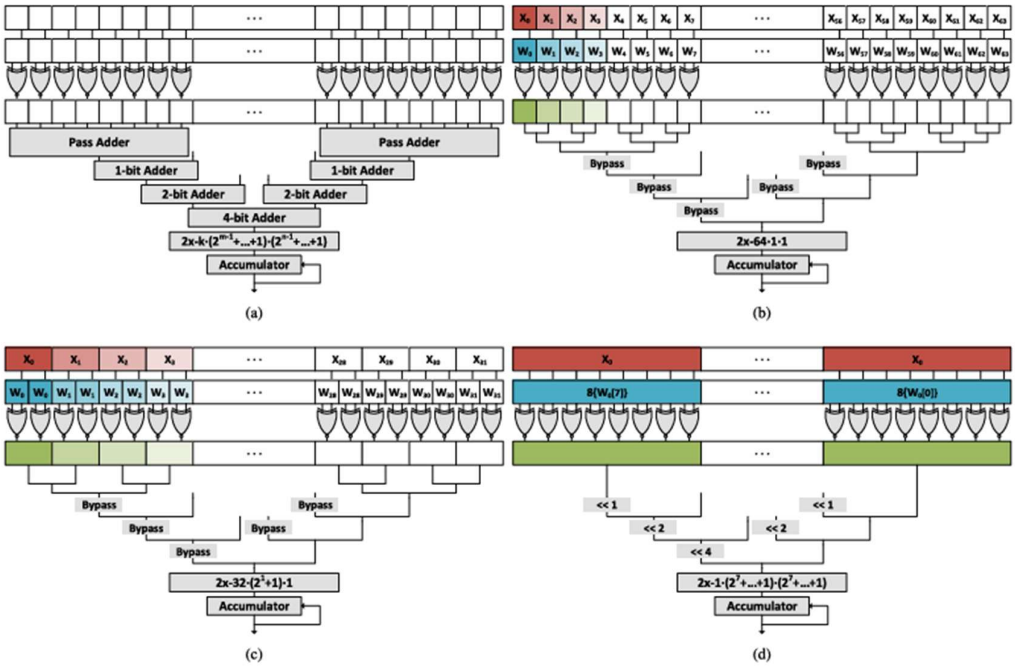

Fig. 4. Variable-precision arithmetic organization supporting packed 1-bit XNOR, mixed 2-bit/1-bit signed-add, and 8-bit multiplication paths.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

The precision-select arithmetic block is the core hardware innovation. In the documented design, the binary path contains 64 1W/1A XNOR lanes, the mixed-precision path contains 32 1W/2A signed-add lanes, and the full-precision path contains one 8W/8A multiplication lane. Precision selection therefore trades per-operation numeric detail for spatial parallelism. Binary mode maximizes logic-level concurrency, mixed mode offers an intermediate operating point, and 8-bit mode preserves the detail required by sensitive layers.

F. Processing Element Pipeline

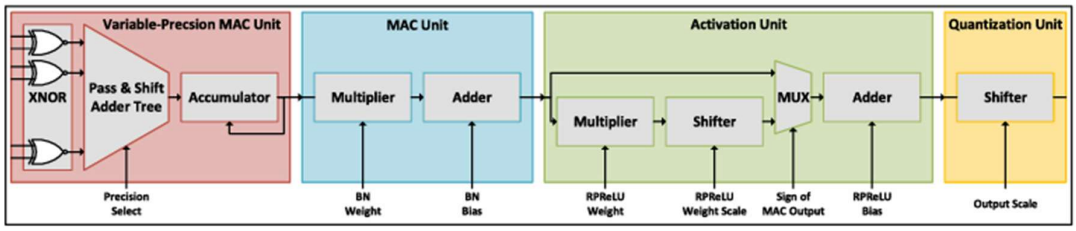

Fig. 5. Processing-element pipeline integrating variable-precision computation, batch normalization, RPReLU activation, and output quantization.

Each PE contains the variable-precision compute block, accumulation, batch normalization, RPReLU activation, and requantization. For a preactivation u, batch normalization can be represented as

$$BN(u) = \gamma\,(u - \mu) / sqrt(\sigma^2 + \varepsilon) + \beta$$

RPReLU applies a trainable piecewise activation that improves the representational flexibility of highly quantized features. The final shift/scale stage produces an 8-bit value compatible with the feature-map memory. Keeping post-convolution operations inside the PE reduces additional memory round trips and allows the entire layer to be streamed through the accelerator.

G. System-Level FPGA Integration

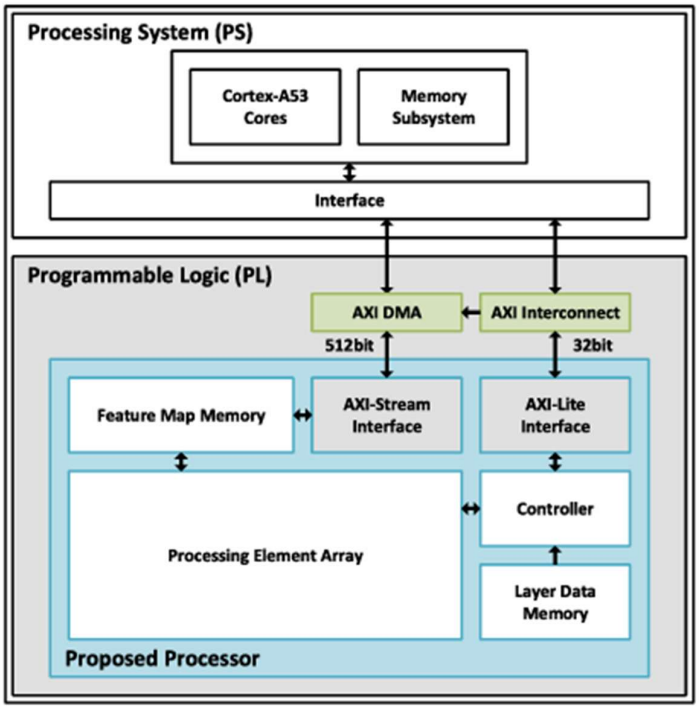

Fig. 6. Heterogeneous FPGA integration using a processor system, programmable logic, AXI DMA/Stream data paths, and AXI-Lite control.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

The full system separates high-level control from dense inference computation. The processor system performs image preparation, accelerator configuration, and output post-processing, while the programmable logic executes feature extraction and convolution. AXI-Stream and DMA support high-volume feature transfers; AXI-Lite provides low-bandwidth configuration and status access. The layered organization allows the network to be scheduled using stored layer descriptors while the PE array is reused across the complete model.

IV. IMPLEMENTATION AND EVALUATION METHODOLOGY

A. Precision Scheduling

Precision is assigned according to layer sensitivity. Input-facing and final prediction-related layers are candidates for 8-bit operation because they directly encode image detail or detection outputs. Intermediate feature-extraction blocks can use 1-bit or 2-bit precision, where XNOR or signed-add computation provides large hardware savings. The same PE hardware is reused across all modes, avoiding separate accelerators for different bit widths.

Mode	Weight precision	Activation precision	Arithmetic basis	Intended use
Binary	1 bit	1 bit	XNOR + popcount	Quantization-tolerant feature layers
Mixed	1 bit	2 bits	Bit-sliced signed addition + shift	Moderately sensitive layers
Full	8 bits	8 bits	8-bit multiplication / accumulation	Input/output or precision-sensitive layers

B. Architecture-Level Evaluation

The supplied project document does not include post-synthesis FPGA reports, measured board power, frame rate, model mAP, or per-class detection accuracy. To avoid creating unsupported empirical results, this paper evaluates only quantities that follow directly from the documented architecture: operand packing, normalized storage/bandwidth demand, lane-level parallelism, PE-level channel concurrency, and exact arithmetic functionality. These results are sufficient to demonstrate the mechanism of efficiency, while a final experimental paper should add device-specific synthesis and detector-quality measurements.

C. Bit-True Functional Validation

A reproducible software check was performed for the arithmetic identities used by the low-precision datapath. Ten thousand random length-64 1-bit activation/weight vectors were evaluated both by direct {-1,+1} multiplication and by the XNOR-popcount identity. A second set of ten thousand random length-64 2-bit unsigned activation vectors with 1-bit signed weights was evaluated by direct multiplication and by two bit-plane signed accumulations with shift recombination. The full 8-bit control path was also sanity-checked using signed 8-bit operands accumulated in 32-bit arithmetic. The validation assesses arithmetic equivalence only; it does not substitute for RTL simulation or FPGA timing closure.

V. RESULTS AND DISCUSSION

A. Functional Correctness of the Precision Modes

Precision mode	Random dot products	Vector length	Arithmetic mismatches	Maximum arithmetic error
1W/1A XNOR-popcount	10,000	64	0	0
1W/2A bit-sliced signed-add	10,000	64	0	0
8W/8A control-path sanity check	10,000	64	0	0

The randomized test produced zero mismatches in every evaluated mode. For binary computation, the XNOR-popcount expression is mathematically identical to a dot product of bipolar values, so exact agreement is expected when the accumulator width is sufficient. The mixed-precision result confirms that a 2-bit activation can be reconstructed from

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

weighted bit planes without using a conventional 2-bit multiplier array. These tests verify the arithmetic principle on which the precision-select unit is based.

B. Feature-Memory and Bandwidth Reduction

Activation precision	Bits per activation	Values in 512-bit transfer	Payload gain vs 8-bit	Storage/bandwidth reduction
8-bit	8	64	1.0×	0%
2-bit	2	256	4.0×	75.0%
1-bit	1	512	8.0×	87.5%

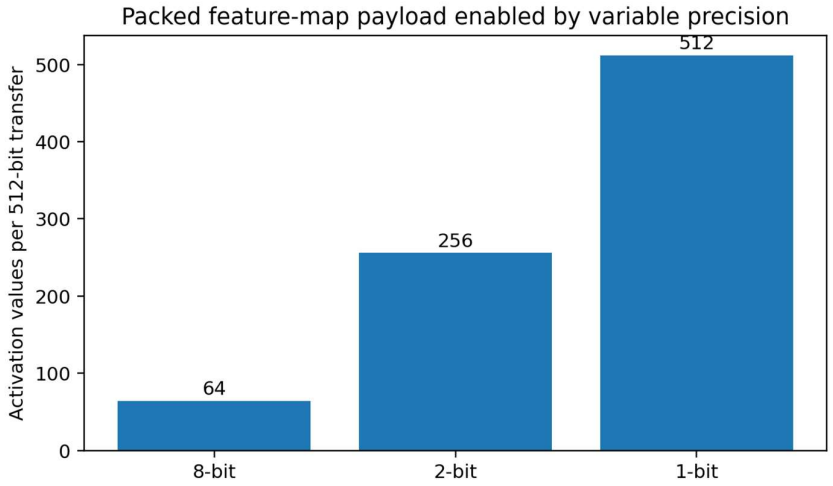

Fig. 7. Architecture-derived payload available on the documented 512-bit feature path as activation precision is reduced.

Quantization provides a direct memory-system benefit. The same 512-bit physical transfer carries four times as many 2-bit activations and eight times as many binary activations as 8-bit values. Equivalently, storing a fixed number of activations requires only 25% of the 8-bit storage at 2-bit precision and 12.5% at 1-bit precision. This reduction increases the likelihood that intermediate features can remain on-chip and reduces the pressure on DMA and external memory when off-chip transfers are unavoidable.

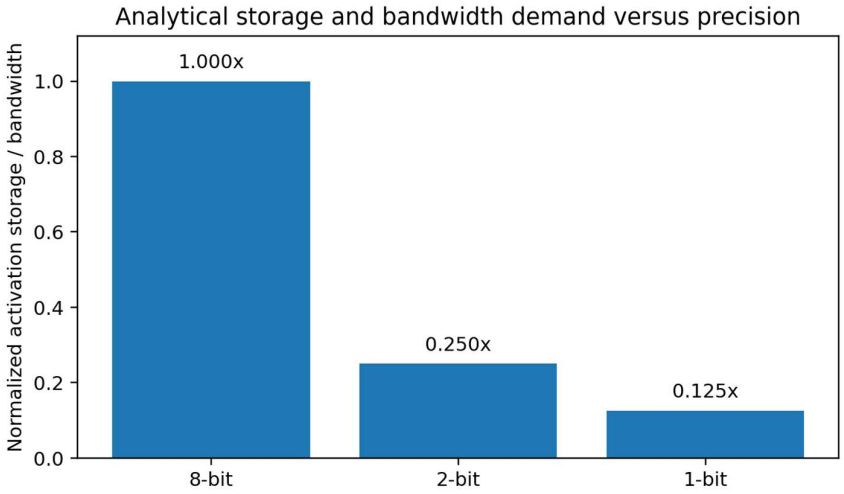

Fig. 8. Normalized activation storage and bandwidth requirement relative to the 8-bit baseline.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

C. Precision-Dependent Arithmetic Parallelism

Mode in precision-select block	Documented lane organization	Relative lane concurrency
8W/8A	1 multiplication lane	1×
1W/2A	32 signed-add lanes	32×
1W/1A	64 XNOR lanes	64×

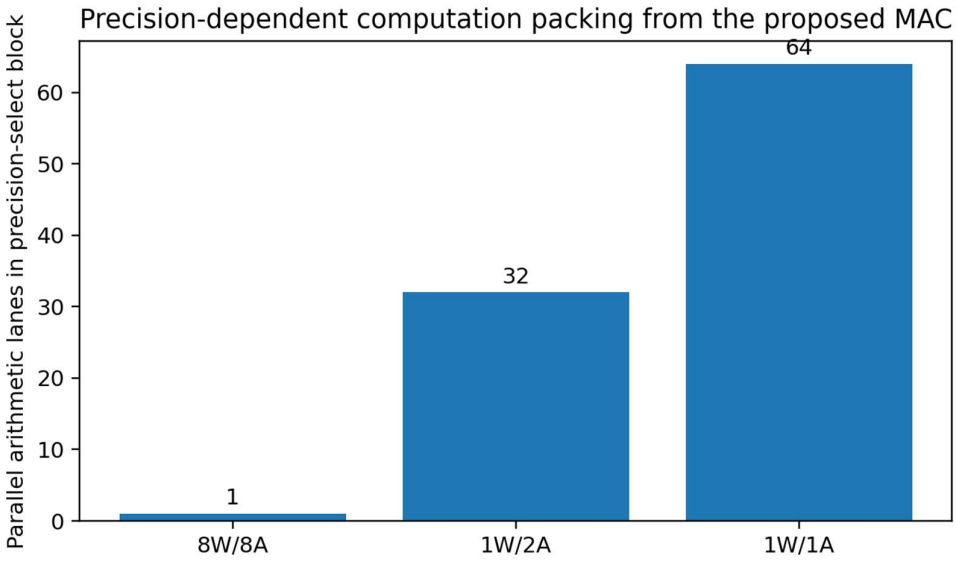

Fig. 9. Lane-level computation packing supported by the variable-precision arithmetic block.

The low-bit modes translate reduced arithmetic complexity into direct spatial parallelism. The precision-select structure can accommodate 64 binary XNOR operations or 32 mixed-precision signed operations in the space assigned to a single 8-bit multiplication path. This does not imply a universal 64× end-to-end frame-rate improvement because feature fetching, accumulation depth, control, pooling, and layer shape also affect throughput. It does show why low precision is particularly attractive on FPGA logic: many simple operations can be placed and executed concurrently.

D. PE-Array Throughput Potential

At the processor level, the 64-PE array computes 64 output channels concurrently when sufficient channel-level work and data bandwidth are available. Combined with output-stationary accumulation, this organization keeps partial sums close to arithmetic units and amortizes feature/weight fetches across repeated operations. For layers whose output-channel count is a multiple of 64, the PE array can be fully occupied; smaller or irregular layers may experience lower utilization. The controller and layer-data memory therefore play an important role in tiling and scheduling.

Architectural feature	Baseline implication	Proposed implication
Convolution arithmetic	Dedicated multipliers dominate compute resources	XNOR/signed-add lanes for low-bit layers; 8-bit path only where needed
Feature representation	8/16/32-bit values increase memory traffic	1/2/8-bit variable precision reduces traffic according to sensitivity
Partial sums	Frequent buffer/memory movement possible	Output-stationary local accumulation
Channel processing	Limited by available MAC units	64 PEs compute output channels concurrently
Post-processing	Separate stages may add traffic	BN, RPReLU, and quantization integrated in PE
System interface	General processor-memory traffic	AXI-Stream/DMA data path with AXI-Lite control

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 6, Issue 8, August 2026
Impact Factor: 5.731

E. Baseline MAC Versus Proposed Precision-Select MAC

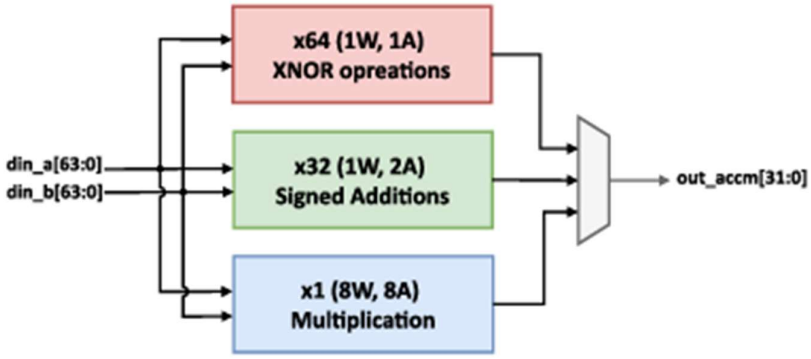

Fig. 10. Baseline-versus-proposed compute concept: precision selection maps the available datapath to binary XNOR, mixed signed-add, or full-precision multiplication.

The central advantage of the proposed MAC is adaptivity. A fixed 8-bit engine spends the same multiplier resources on every layer, including layers that could tolerate much lower precision. The proposed design instead changes the arithmetic mechanism according to layer precision. Binary computation uses equality tests and population counting; 2-bit activation computation uses signed additions and shifts; 8-bit mode retains conventional arithmetic for accuracy-sensitive stages. The hardware therefore avoids the two extremes of globally expensive full precision and globally inaccurate binarization.

F. Detection Accuracy and FPGA Synthesis Reporting

The project source states that the architecture is intended to maintain acceptable detection accuracy and reduce power/resource usage, but it does not provide a dataset, mean average precision (mAP), per-class AP, LUT/FF/DSP/BRAM utilization, maximum clock frequency, latency, frames per second, or measured power. These quantities are intentionally not invented here. For a final journal or conference submission, the trained network should be evaluated on a stated benchmark dataset and the RTL should be synthesized and implemented on a named FPGA device. At minimum, the final results should report mAP@0.5 and mAP@0.5:0.95, precision/recall, model size, LUT/FF/BRAM/DSP counts, Fmax, end-to-end latency, FPS, board power, and energy per frame.

VI. PRACTICAL IMPLICATIONS, APPLICATIONS, AND LIMITATIONS

The accelerator is suited to applications where local visual inference must be performed with limited power and deterministic response. Representative use cases include autonomous and driver-assistance systems, surveillance cameras, robotics, drones, smart manufacturing, medical edge imaging, retail analytics, agriculture monitoring, and traffic systems. Variable precision is especially useful in such settings because operating conditions can motivate different accuracy/energy points without changing the physical accelerator.

Several practical limitations remain. The usefulness of binary and mixed precision depends on quantization-aware training and layer sensitivity; a poorly scheduled precision map may reduce detection quality. The 64-PE array also requires sufficient memory bandwidth and layer-level parallelism to remain fully utilized. Although the architecture reduces activation/weight storage, large detection networks may still exceed on-chip capacity and require external DDR. Finally, architecture-derived gains do not guarantee post-route timing or board-level energy reduction until the RTL is synthesized, placed, routed, and measured on the target device.

VII. CONCLUSION AND FUTURE WORK

This paper presented a hardware/network co-designed object-detection accelerator based on XNOR-centric variable-precision computation. The network uses DenseToRes and transition structures to preserve information under aggressive quantization, while the accelerator supports binary, mixed, and 8-bit arithmetic within a reusable precision-select datapath. A 64-PE output-stationary processor, 512-bit feature path, local feature/weight memories, integrated BN-RPReLU-quantization processing, and AXI-based PS/PL integration provide the hardware structure required for real-time edge inference.

Architecture-level analysis demonstrates concrete efficiency mechanisms. Moving from 8-bit to 2-bit or 1-bit activations reduces storage and feature bandwidth demand by 75% and 87.5%, respectively, and increases the payload of a 512-bit transfer from 64 values to 256 or 512 values. The precision-select block provides 32 mixed-precision lanes or 64 binary lanes relative to one 8-bit multiplication lane, and randomized bit-true validation confirmed exact low-precision arithmetic equivalence with zero mismatches. These findings support the scalability and efficiency arguments of the design without claiming unsupported synthesis or detection-accuracy measurements.

Future work should implement the design on a specified FPGA family, perform quantization-aware training on a standard detection benchmark, and report both algorithmic and hardware metrics. Additional improvements can include dynamic per-layer precision scheduling, sparsity exploitation, compressed weight storage, double-buffered DMA transfers, deeper pipelining, multi-scale detection heads, and runtime power/accuracy adaptation for battery-powered edge systems.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[3] 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)*, 2016.

[4] W. Liu *et al*., "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016.

[5] V. B. Raju, J. MD, K. Keshamoni, and D. S. Reddy, "AI-driven harmful algal bloom prediction in aquatic ecosystems using machine learning and remote sensing data," *International Journal of Aquatic Research and Environmental Studies*, vol. 6, no. S2, pp. 860–867, 2026, doi: 10.70102/q4xs0978.

[6] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017.

[7] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[8] B. Jacob *et al*., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018.

[9] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[10] M. D. Javeed, B. Hari Krishna, and M. V. S. R. Kishore, "Low power Viterbi decoder design for TCM decoders using T-algorithm," *CiiT International Journal of Programmable Device Circuits and Systems*, vol. 4, no. 15, pp. 764–768, 2012.

[11] 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)*, 2016.

[12] Z. Liu, B. Wu, W. Luo, X. Yang, W. Liu, and K. 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)*, 2018.

[13] J. Bethge, H. Yang, M. Bornstein, and C. Meinel, "BinaryDenseNet: Developing an architecture for binary neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, 2019.

[14] Z. Liu, Z. Shen, M. Savvides, and K. Cheng, "ReactNet: Towards precise binary neural network with generalized activation functions," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020.

[15] Y. Chen, T. Krishna, J. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[16] D. Srinivas Reddy, G. Satyanarayana, J. M. D., and K. Mamatha, "Hybrid machine learning and deep learning-based intrusion detection system for IoT network security using Python," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 18, no. 6s, pp. 1133–1141, 2026, doi: 10.70917/ijcisim-2026-3022.

[17] Y. Umuroglu *et al*., "FINN: A framework for fast, scalable binarized neural network inference," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA)*, 2017.
[18] C. Zhang *et al*., "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA)*, 2015.
[19] J. Qiu *et al*., "Going deeper with embedded FPGA platform for convolutional neural network," in *Proc. ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays (FPGA)*, 2016.
[20] K. Wang *et al*., "HAQ: Hardware-aware automated quantization with mixed precision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019.
[21] M. Javeed and B. Swapna, "High speed convolution encoding and Viterbi decoding using dynamic shift register," *International Journal of Electronic and Electrical Engineering*, vol. 7, no. 5, pp. 483–490, 2014.
[22] T. Chen *et al*., "DianNao: A small-footprint high-throughput accelerator for ubiquitous machine learning," *ACM SIGARCH Computer Architecture News*, 2014.
[23] P. Judd *et al*., "Stripes: Bit-serial deep neural network computing," in *Proc. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, 2016.