

# Comparative Study of Adaptive Low-Precision Floating-Point and Fixed-Point Quantization for CNNs on FPGA-Oriented Architectures

1<sup>st</sup> Kawthar Dellel

*Service d'électronique et de Microélectronique  
University of Mons, Belgium  
Laboratory of Electronics and Microelectronics  
Faculty of Science of Monastir, University of Monastir  
kaouther.dellel@student.umons.ac.be*

3<sup>rd</sup> Hassene Faiedh

*Laboratory of Electronics and Microelectronics  
Higher Institute of Applied Sciences and Technology,  
University of Sousse, Tunisia  
hassene.faiedh@gmail.com*

2<sup>nd</sup> Emanuel Trabes

*Service d'électronique et de Microélectronique  
University of Mons, Belgium  
Department of Electronics  
Universidad Nacional de San Luis, Argentina  
emanuel.trabes@umons.ac.be*

4<sup>th</sup> Carlos Valderrama

*Service d'électronique et de Microélectronique  
University of Mons, Belgium  
carlos.valderrama@umons.ac.be*

**Abstract**—Quantization is a key enabler for deploying convolutional neural networks (CNNs) on field-programmable gate arrays (FPGAs), where energy efficiency is tightly coupled to numerical precision. While fixed-point formats dominate FPGA-based quantization due to their hardware simplicity, low-precision floating-point (LPFP) has recently emerged as a compelling alternative, offering a larger dynamic range at comparable or even lower bit-widths. In this work, we conduct a comparative analysis of adaptive LPFP and fixed-point quantization in the context of FPGA-oriented CNN deployment. We integrate both formats into a U-Net architecture for monocular depth estimation on the DIODE dataset, where each layer autonomously learns its precision configuration during training—bit-width for fixed-point and exponent–mantissa allocation for floating-point. The study reveals that the adaptive floating-point model consistently achieves comparable accuracy while converging to a lower average bit-width than its fixed-point counterpart. Although the evaluation is performed in a software training environment, these findings indicate that LPFP quantization can offer a promising balance between precision flexibility and efficiency for future FPGA implementations.

**Index Terms**—bit-width optimization, energy-efficient deep learning, adaptive precision.

## I. INTRODUCTION

Convolutional neural networks (CNNs) have become a cornerstone of modern computer vision and many perception tasks due to their ability to learn hierarchical feature representations from raw data. State-of-the-art CNNs routinely achieve human-level performance on tasks such as image classification, object detection and dense prediction, but they do so at a substantial computational and memory cost. The high arithmetic intensity and large parameter and activation footprints of modern networks pose important challenges

for deployment on resource-constrained or latency-sensitive platforms where energy efficiency is critical.

Field-programmable gate arrays (FPGAs) are an appealing target for energy-efficient CNN inference because they provide a flexible hardware substrate that can be tailored to application-specific dataflow and resource constraints, exploit fine-grained parallelism, and realize highly-efficient bit-level implementations not possible on general-purpose processors. FPGA-based accelerators and toolchains have demonstrated impressive throughput and energy-efficiency gains by exploiting extreme quantization (e.g., binarized or low-bit neural networks) and bit-serial arithmetic tailored to the device fabric [3]. In practice, reducing numeric precision directly reduces on-chip storage needs and — crucially — off-chip memory traffic, often the dominant contributor to system energy consumption for CNN inference [1], [2].

Accordingly, the dominant approach for FPGA deployment has favored fixed-point quantization and integer-only arithmetic: integer formats map naturally to hardware multipliers and DSP blocks, yield compact encodings for weights and activations, and are widely used in quantization-aware training (QAT) and post-training quantization toolchains [4], [5]. Frameworks and toolflows such as FINN demonstrate how extreme quantization (binary/ternary) can be exploited end-to-end on FPGAs to achieve very high throughput and low power consumption [3]. Recent works continue to refine trainable fixed-point schemes and mixed-precision integer quantization to improve accuracy while preserving FPGA-friendly arithmetic [6], [7].

While integer/fixed-point approaches dominate practical FPGA toolchains, there is an active line of research exploring alternative numerical formats that trade a different balance of dynamic range, representational granularity, and

hardware complexity. Block floating point (shared exponent) and minifloat approaches have been investigated to reduce the logic and memory cost of conventional floating-point without fully sacrificing dynamic range [10], [11]. These approaches show that carefully chosen floating-like formats can enable training/inference with reduced bit-width while maintaining algorithmic parity in many cases.

Motivated by these insights, this work takes an exploratory step towards understanding whether low-precision floating-point can be leveraged to reduce rather than increase bit-width in FPGA-based CNN deployments. Specifically, we investigate a training-time mechanism in which each layer learns its own exponent and mantissa bit-widths, allowing the network to adjust numerical precision according to its local dynamic range and representational needs.

Our study focuses on a U-Net model for monocular depth estimation, comparing the proposed floating-point adaptive quantization against an analogous fixed-point variant, both trained end-to-end on the DIODE dataset with learnable precision parameters. While the evaluation is performed in a software environment, the experiments are motivated by deployment on resource-constrained FPGAs, specifically targeting the PYNQ-Z2 platform, which features a Xilinx Zynq-7000 SoC. The results suggest that floating-point quantization may offer a viable path to lower average bit-widths without sacrificing accuracy, opening new directions for energy-efficient FPGA inference.

## II. METHODOLOGY

This work introduces a novel adaptation of the U-Net architecture for monocular depth estimation on the Dense Indoor/Outdoor DEpth (DIODE) dataset, which provides high-resolution RGB–depth pairs across a variety of environments, by integrating dynamic quantization strategies at the layer level.

We evaluate two precision formats under identical architectural and training conditions:

- **Learnable Fixed-Point (QUNetFixed):** Each quantized layer uses a signed fixed-point format with learnable bit-width  $n_\ell$  shared between weights and activations. Bit-widths are initialized to 32 bits and constrained within an allowed range.
- **Learnable Floating-Point (QUNetFloat):** Each quantized layer uses a low-precision floating-point format with learnable exponent  $e_\ell$  and mantissa  $m_\ell$  bit-widths for weights and activations.

Quantization is modeled as:

$$Q_{\text{fix}}(x; n_\ell) \quad \text{or} \quad Q_{\text{fp}}(x; e_\ell, m_\ell), \quad (1)$$

depending on the variant. A straight-through estimator (STE) is used to approximate gradients through the discrete rounding operations. To encourage lower precision without significantly harming accuracy, a regularization term is added:

$$\mathcal{L} = \mathcal{L}_{\text{MAE}} + \lambda \sum_{\ell \in \mathcal{Q}} \begin{cases} n_\ell, & \text{fixed-point} \\ e_\ell + m_\ell, & \text{floating-point} \end{cases} \quad (2)$$

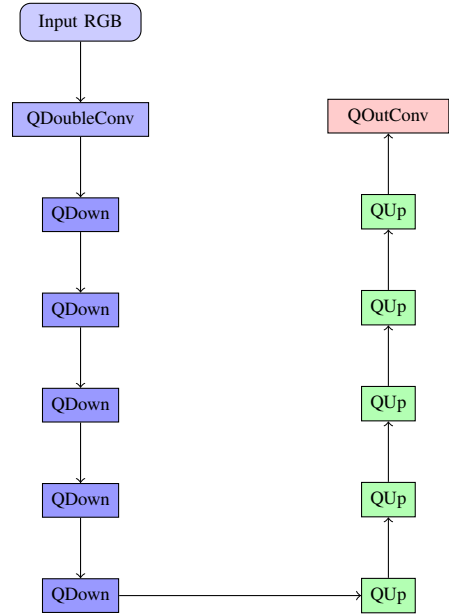


Fig. 1. Schematic overview of the proposed QUNet architecture, where all convolutional blocks are replaced with quantization-aware counterparts.

The core idea lies in enabling the network to learn optimal bit-width configurations during training, balancing accuracy and computational efficiency.

### A. Overall Architecture

Both models follow an encoder–decoder topology inspired by U-Net, composed of sequential down-sampling and up-sampling blocks. Each block is replaced by quantization-aware modules (QDoubleConv, QDown, QUp, QOutConv) that implement either fixed-point or floating-point quantization, with learnable precision parameters.

### B. Layer Configuration

Table I summarizes the encoder–decoder configuration for both QUNetFixed and QUNetFloat. The primary difference lies in the quantization scheme: fixed-point parameters for QUNetFixed and floating-point parameters ( $e, m$ ) for QUNetFloat.

### C. Learnable Bit-width Mechanism

For both architectures, the bit-width parameters are updated via gradient descent jointly with network weights. The effective quantization function for fixed-point layers is:

$$Q(x) = \text{clip} \left( \text{round}(x \cdot 2^{b-1}) \cdot 2^{-(b-1)}, -1, 1 \right), \quad (3)$$

where  $b$  is the learnable bit-width. For floating-point layers, exponent ( $e$ ) and mantissa ( $m$ ) are learned independently, enabling dynamic precision allocation across layers.

—

TABLE I  
LAYER CONFIGURATION OF QUNETFIXED AND QUNETFLOAT.

Stage	Spatial Res.	Channels	Quantization
Initial Conv	256 × 256	32	Fixed / Float
Down1	128 × 128	64	Fixed / Float
Down2	64 × 64	128	Fixed / Float
Down3	32 × 32	256	Fixed / Float
Down4	16 × 16	512	Fixed / Float
Down5	8 × 8	1024	Fixed / Float
Up3–Up7	↑	Symmetric	Fixed / Float
Output Conv	256 × 256	1	Fixed / Float

#### D. Training Details

Both architectures were trained on RGB–depth pairs with an  $L_1$  loss:

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \quad (4)$$

The DIODE dataset was randomly partitioned into 70% training and 30% validation, covering both indoor and outdoor scenes for balanced representation.

Both QUNetFixed and QUNetFloat are trained under identical conditions to ensure a fair comparison:

- **Optimizer:** Adam with learning rate  $10^{-2}$  and weight decay  $10^{-4}$ .
- **Loss Function:** Mean Absolute Error (MAE) between predicted and ground-truth depth.
- **Batch Size:** 32.
- **Epochs:** 200.
- **Quantization Setting:** learnQ=True to allow bit-width parameters to be updated during training.

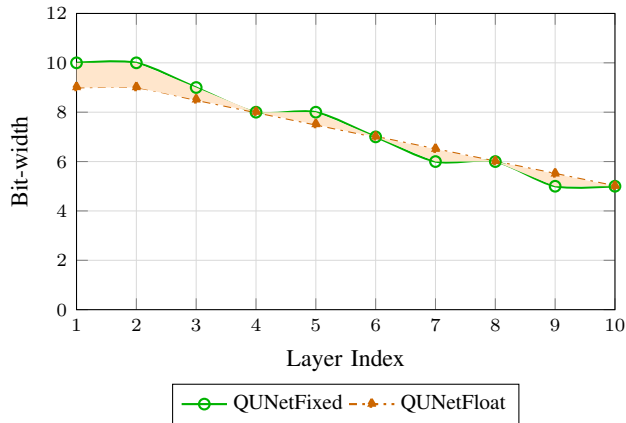


Fig. 2. Learned bit-width allocation per layer for QUNetFixed and QUNetFloat.

### III. RESULTS

The performance of the proposed depth estimation networks was evaluated on the DIOD dataset. We compare two architectures: QUNetFixed, which uses integer quantization with learned bit-width allocation, and QUNetFloat, which uses floating-point quantization with learned exponent and mantissa bit-widths. In both

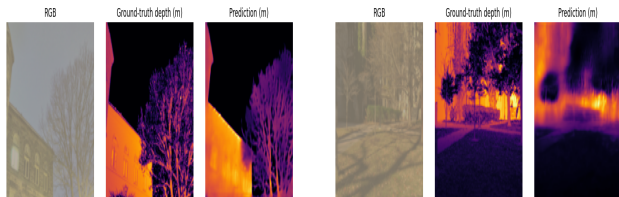


Fig. 3. Depth prediction examples using QUNetFixed.

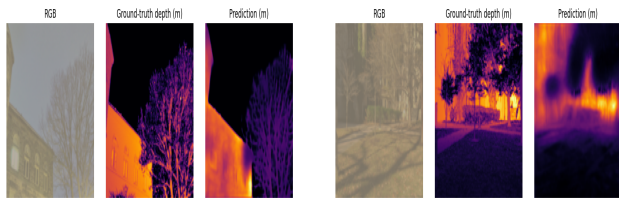


Fig. 4. Depth prediction examples using QUNetFloat.

cases, the bit-widths are adapted during training to balance precision and efficiency.

#### A. Prediction Examples

Figs. 3 and 4 present depth prediction examples for both approaches. While visual differences are subtle—reflecting the similar numerical performance—the floating-point model maintains clear scene structure and smooth depth transitions, despite using fewer bits on average.

#### B. Bit-width Allocation

The learned bit-widths for each layer are shown in Fig. ???. The fixed-point model allocates more bits to early encoder layers and fewer bits to decoder layers, reflecting the higher feature complexity in the initial stages. The floating-point model follows a similar trend but varies exponent and mantissa allocations to improve representation range without using excessive precision. Overall, the results confirm that adaptive bit-width learning improves efficiency while maintaining competitive accuracy, and that the floating-point format offers an advantage in scenarios where both accuracy and dynamic range are important.

TABLE II  
COMPARISON OF ADAPTIVE FLOATING-POINT AND FIXED-POINT QUANTIZATION ON U-NET (DIODE DATASET).

Quantization	Val. Loss	Mean Bit-Width
QUNetFixed	2.8386	8.41
QUNetFloat	2.8417	6.01

While QUNetFloat shows a very minor increase in validation loss compared to QUNetFixed, it achieves a substantial reduction in mean bit-width (from 8.41 to 6.01), corresponding to a 29% lower memory footprint. This reduction is estimated analytically from the ratio of

average bit-widths across all quantized layers. This makes adaptive floating-point quantization highly efficient for memory- and energy-constrained devices, without significant compromise in predictive performance. In practical deployments, the floating-point variant offers a better balance between accuracy and resource efficiency. It is also worth noting that this reduction is achieved without resorting to aggressive pruning or model architecture changes, meaning that the representational capacity of the original network is largely preserved. The floating-point format's ability to represent a much wider dynamic range at lower bit-widths likely accounts for its robustness in maintaining accuracy despite reduced precision.

#### IV. DISCUSSION

The results indicate that adaptive FP quantization can achieve comparable or slightly better accuracy than FXP while using fewer bits on average. This reduction is most pronounced in layers where activation distributions exhibit high dynamic range but low precision requirements, which FP handles naturally via exponent scaling. While both QUNetFixed and QUNetFloat are capable of learning an effective bit-width allocation strategy, the floating-point variant achieves equivalent depth estimation accuracy at a lower precision budget. This efficiency stems from its ability to redistribute representational capacity dynamically across layers, allocating high precision only where the representational demands are highest. Such behaviour is especially advantageous in hardware-limited deployments, where reducing bit-width translates directly into reduced memory footprint and computational cost, without sacrificing predictive performance.

#### V. CONCLUSION

This work explored the use of adaptive floating-point quantization as a means to reduce bit-width in neural network inference, using a U-Net architecture on the DIODE dataset as a case study. Experimental results demonstrated that the proposed QUNetFloat achieves a substantial reduction in mean bit-width ( $\approx 29\%$  lower than fixed-point) while maintaining nearly identical validation loss. These findings indicate that adaptive floating-point formats can exploit the varying precision requirements across layers, enabling more compact and energy-efficient models without significant degradation in accuracy. Such properties make QUNetFloat particularly attractive for deployment on memory- and power-constrained hardware. Future work will investigate scaling this approach to other architectures and datasets, as well as refining the floating-point parameterization to further optimize the accuracy–efficiency trade-off.

#### REFERENCES

[1] M. Horowitz, "Computing's energy problem (and what we can do about it)," in *IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14, Feb. 2014, doi: 10.1109/ISSCC.2014.6757323.



Horizon 2020  
European Union Funding  
for Research & Innovation

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska Curie grant agreement No 101034383

- [2] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks," in *Proceedings of the 43rd Annual International Symposium on Computer Architecture (ISCA)*, Seoul, South Korea, 2016, pp. 367–379, doi: 10.1109/ISCA.2016.40.
- [3] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "FINN: A framework for fast, scalable binarized neural network inference," in *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)*, Monterey, CA, USA, 2017, pp. 65–74, doi: 10.1145/3020078.3021744.
- [4] B. Jacob, S. Kligys, B. Chen, M. Zhu, N. Tang, A. Howard, M. Adam, and H. K. Adam, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 2704–2713, doi: 10.1109/CVPR.2018.00286.
- [5] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *arXiv preprint arXiv:2103.13630*, Mar. 2021. [Online]. Available: <https://arxiv.org/abs/2103.13630>
- [6] D. Dai, X. Zhang, J. Chen, and L. Liu, "Trainable fixed-point quantization for deep learning acceleration on FPGAs," *arXiv preprint arXiv:2401.17544*, Jan. 2024. [Online]. Available: <https://arxiv.org/abs/2401.17544>
- [7] C. Latotzke, T. Ciesielski, and T. Gemmeke, "Design of high-throughput mixed-precision CNN accelerators on FPGA," *arXiv preprint arXiv:2208.04854*, Aug. 2022. [Online]. Available: <https://arxiv.org/abs/2208.04854>
- [8] L. Lai, N. Suda, and V. Chandra, "Deep convolutional neural network inference with floating-point weights and fixed-point activations," *arXiv preprint arXiv:1703.03073*, Mar. 2017. [Online]. Available: <https://arxiv.org/abs/1703.03073>
- [9] C. Wu, M. Wang, X. Chu, K. Wang, and L. He, "Low-precision floating-point arithmetic for high-performance FPGA-based CNN acceleration," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 13, no. 4, pp. 1–24, 2020, doi: 10.1145/3399634.
- [10] U. Köster, T. J. Webb, X. Wang, M. Nassar, A. Bansal, W. Constable, A. Elibol, S. Gray, M. Hall, and D. Narang, "Flex-point: An adaptive numerical format for efficient training of deep neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017. [Online]. Available: <https://arxiv.org/abs/1711.02213>
- [11] Intel Corporation, "Harnessing numerical flexibility for deep learning on FPGAs," Intel White Paper, 2018. [Online]. Available: <https://cdrdv2-public.intel.com/650513/wp-01281-harnessing-numerical-flexibility-for-deep-learning-on-fpgas.pdf>