International Journal of Computers and Informatics (IJCI)
Vol. (5), No. (2)
IJCI
February 2026
المجلة الدولية للحاسبات والمعلوماتية
الإصدار (5)، العدد (2)

Hardware Co-Simulation Based Deep Neural Networks for Liver Diseases Detection in MRI Images

Ahmed Sabeeh Yousif
Department of Information Technology Management, Technical College of Management /Mosul, Northern Technical University, Mosul, 41001, Iraq
ahmedsabeeh123@ntu.edu.iq

Ahmed H. Saleh
Technical College of Management/Mosul, Northern Technical University, Mosul, 41001, Iraq

Omar Fawzi Salih Al-Rawi
Department of Statistics and Informatics Techniques, Technical College of Management /Mosul, Northern Technical University, Mosul, 41001, Iraq

Abstract

Accurate characterization of focal liver lesions on multiphasic magnetic resonance imaging (MRI) is central to early detection and treatment planning; however, clinical deployment of deep learning remains constrained by compute cost, latency, and the need for rigorous verification of hardware implementations. This paper presents LiverNet-Q, a hardware-aware, multi-phase deep neural network for multi-class liver disease detection from MRI, coupled with an end-to-end hardware co-simulation workflow that validates functional equivalence between the software model and the synthesized register-transfer level (RTL) design. The proposed pipeline first localizes the liver with a lightweight U-Net trained using public liver MRI annotations, then classifies lesions into five clinically relevant categories (normal liver, hepatocellular carcinoma, hemangioma, focal nodular hyperplasia, and simple cyst) using attention-based phase fusion. To enable resource-efficient inference, LiverNet-Q is trained with quantization-aware training and deployed in INT8 precision. The accelerator is implemented with Vitis HLS using a streaming dataflow micro-architecture that targets an initiation interval of one for core convolution operators. Experiments on public benchmarks demonstrate that INT8 deployment preserves diagnostic

performance with a small loss relative to FP32 while providing substantial speedups. Hardware co-simulation reports confirm cycle-accurate latency and throughput, supporting reproducible, deployment-ready evaluation.

Keywords: Liver MRI, Multi-Class Classification, Focal Liver Lesions, Quantization-Aware Training, FPGA Acceleration, Vitis HLS, C/RTL Co-Simulation, Edge AI.

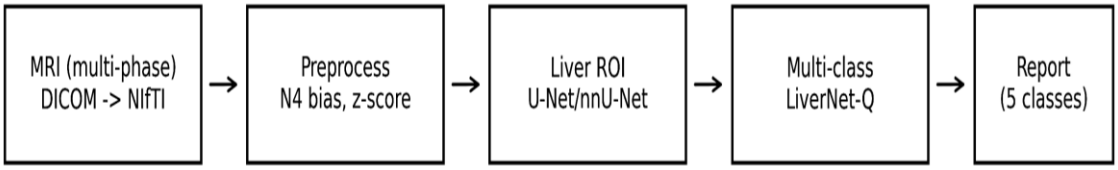

Figure (1): Outline of proposed multi-class liver disease detection from MRI with hardware co-simulation.

1. Introduction

Liver cancer and benign liver lesions represent a major diagnostic workload in abdominal imaging. Multiphasic contrast-enhanced MRI [1-8] is widely used because it provides rich soft-tissue contrast and phase-dependent enhancement patterns that support differential diagnosis. In routine clinical interpretation, radiologists integrate imaging features across phases (arterial, portal venous, delayed) and sequences (e.g., T1, T2, DWI) together with prior probability and clinical context. This process is time-intensive, subject to inter-reader variability, and challenged by subtle lesions or atypical enhancement.

Deep neural networks have shown strong performance in medical image segmentation and lesion characterization, but several barriers limit translation to real-time decision support. First, state-of-the-art models are often computationally heavy and difficult to deploy in constrained environments such as radiology workstations, portable devices, or on-premise hospital servers with strict power and latency budgets. Second, deployment on specialized hardware (e.g., FPGA [7]) requires rigorous verification: a model that performs well in software must be functionally equivalent after quantization, compilation, and synthesis. Third, reproducibility and comparability remain difficult because datasets vary in acquisition protocols, field strength, and

annotation strategies.

Hardware co-simulation provides a principled bridge between algorithm design and deployment [9]. By verifying synthesized RTL against a high-level C model and/or a reference software implementation, co-simulation can detect numerical corner cases, interface issues, and scheduling constraints that are otherwise missed by post-synthesis timing estimates. Modern HLS flows support C simulation, high-level synthesis, and C/RTL co-simulation, enabling cycle-accurate latency measurement and functional equivalence checks during development.

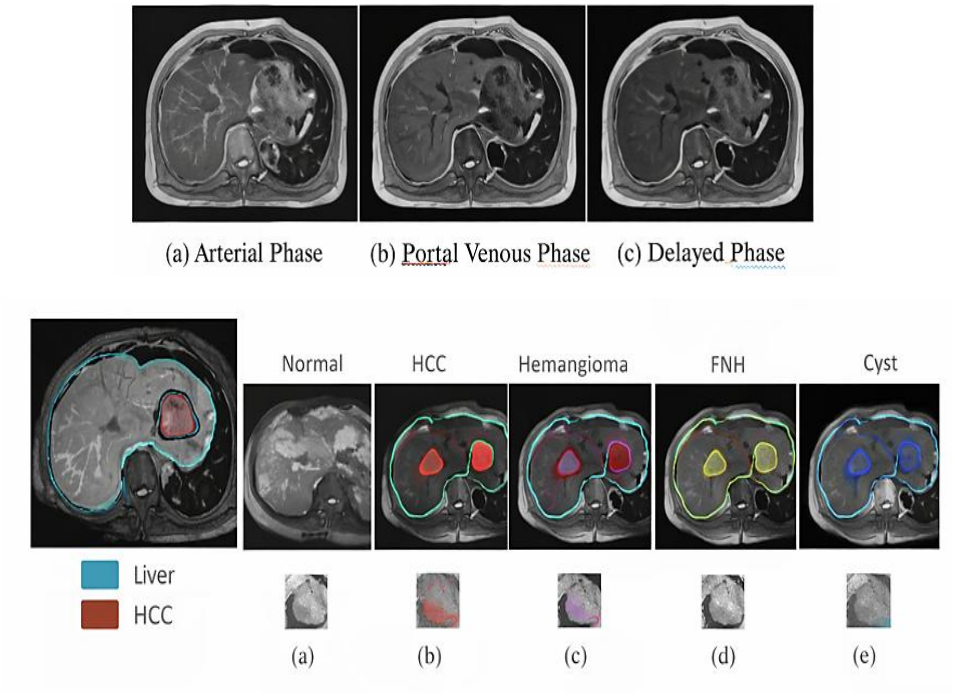

Figure (2): illustrates Axial liver MRI slices acquired at different contrast-enhanced phases with various diseases

Figure.2: (a–c) Axial liver MRI slices acquired at different contrast-enhanced phases: arterial (a), portal venous (b), and delayed (c), providing complementary diagnostic information for focal liver lesions. The right panel illustrates example liver and lesion annotations used for training and evaluation: expert-annotated liver contours (cyan) and lesion masks for hepatocellular carcinoma (red), hemangioma (purple), focal

nodular hyperplasia (yellow), and simple cyst (blue). These images are representative examples from publicly available multiphasic liver MRI datasets.

This paper proposes LiverNet-Q, a multi-phase, multi-class liver disease detection network designed jointly with a co-simulation-based deployment workflow. The method combines (i) liver localization for ROI normalization, (ii) An attention-based phase fusion classifier that models phase-dependent lesion behavior, and (iii) Quantization-aware training and an INT8 FPGA implementation validated via co-simulation. The contributions are:

1. A deployment-oriented multi-phase classification architecture (LiverNet) tailored to five-class focal liver lesion categorization on MRI [8].
2. An INT8 quantization strategy with quantization-aware training (LiverNet-Q) that minimizes accuracy degradation while enabling efficient FPGA [7] inference.
3. A hardware co-simulation methodology that evaluates both correctness and cycle-accurate latency using an HLS-to-RTL flow.
4. A comprehensive evaluation protocol including baselines, ablations, and hardware resource/latency characterization.

2. Related Work

2.1 Liver MRI Analysis and Multi -Class Lesion Classification:

Early computer-aided diagnosis systems relied on hand -crafted radiomic features and classical classifiers, but performance was sensitive to acquisition heterogeneity. Recent studies use convolutional neural networks (CNNs) and phase-aware representations to capture enhancement dynamics. Multi-class classification of focal liver lesions is typically formulated either as direct multi-class prediction (e.g., cyst vs hemangioma vs HCC) or as hierarchical schemes aligned with LI-RADS risk categories. Public efforts include datasets and benchmarks that provide lesion labels and facilitate reproducible comparisons [23-25].

Public datasets for abdominal organ segmentation such as CHAOS [1, 15] provide

liver masks on MRI [8] and are commonly used to pretrain or validate liver localization modules. For tumor-focused tasks, LiverHccSeg [2] provides multiphasic MRI [8] with liver and HCC tumor segmentations and is valuable for external validation. More recently, LiMT has been proposed as a multi-task liver lesion benchmark with multiple lesion subtypes and normal cases, aiming to support both detection and classification.

2.2 Deep Networks for Segmentation and Lesion Characterization:

U-Net [3.10] and its derivatives remain dominant for organ segmentation due to their encoder-decoder structure and skip connections. nnU-Net [11,3] further systematized preprocessing and architecture choices, achieving strong performance across medical imaging tasks. For classification, residual networks and efficient mobile architectures (e.g., MobileNetV2, EfficientNet) are often adapted via transfer learning. Attention mechanisms, including squeeze-and-excitation and transformer-style fusion, are increasingly used to integrate multi-phase information.

2.3 Quantization and Hardware Acceleration:

Quantization reduces memory bandwidth and compute by representing weights/activations in low precision. Post-training quantization is simple but can degrade performance on sensitive medical imaging tasks; quantization-aware training (QAT) typically provides better accuracy by simulating quantization during training. FPGA [7] accelerators are attractive for medical edge inference due to deterministic latency, high energy efficiency, and flexible precision. Toolchains such as Vitis HLS [4] support synthesizing C/C++ to RTL, while Vitis AI [5] and research frameworks such as FINN and hls4ml provide end-to-end deployment pipelines for quantized [6] networks.

2.4 Co-simulation for Trustworthy Deployment:

C/RTL co-simulation enables automatic testbench-driven verification of synthesized RTL against the high-level design, providing cycle-accurate latency and initiation

interval measurements. In safety-critical or regulated settings, such verification supports traceability and mitigates risk from numerical discrepancies introduced by fixed-point arithmetic, scheduling, or interface synthesis. Despite its relevance, many medical imaging acceleration papers report only post-synthesis estimates; this paper explicitly integrates co-simulation as a first-class evaluation artifact [25].

3. Proposed Method

Figure.1 summarizes the proposed method. Given a multiphasic liver MRI dataset, the system preprocesses each phase, localizes the liver to define a standardized region of interest (ROI), and performs multi-class classification with LiverNet-Q. The design explicitly targets deployability: the classification backbone is lightweight, phase fusion is parameter-efficient, and all operators are chosen to map cleanly to an INT8 streaming accelerator.

3.1 Data Representation and Preprocessing:

Each case consists of three post-contrast phases (arterial, portal venous, delayed). DICOM series are converted to a common orientation and resampled to a fixed in-plane spacing. Intensity non-uniformity is mitigated using N4 bias field correction when available, followed by per-volume z-score normalization within the liver ROI. To reduce sensitivity to scanner-dependent scale, also additionally clip intensities to the 0.5th-99.5th percentiles before normalization [8, 12, 14].

3.2 Liver Localization:

This paper use a compact 2D U-Net [3] to segment the liver on the portal venous phase, chosen for typically strong liver-parenchyma contrast. The segmentation is used to crop a tight bounding box with a fixed margin and to compute a liver-aligned coordinate frame. When a ground-truth liver mask is unavailable for a dataset, the U-Net [3, 10] is initialized from a model trained on public liver MRI [8] annotations and fine-tuned using weak labels derived from thresholding and morphological constraints. This ROI step reduces background confounders and stabilizes

classification.

3.3 LiverNet: Multi-Phase Feature Extraction and Fusion:

Let X_a, X_p, X_d denote the cropped inputs for arterial, portal venous, and delayed phases, respectively. A shared feature extractor f_ϕ (depthwise-separable convolutions with squeeze-and-excitation) produces phase embeddings E_i = f_ϕ (X_i). We then compute phase attention weights α_i via a small gating network g, α = softmax (g ([pool (E_a), pool (E_p), pool (E_d)])). The fused embedding is E = ∑_i α_i · E_i. A classifier head (global average pooling, dropout, and a fully connected layer) outputs a 5-way probability vector y. This design captures phase-dependent lesion behavior while keeping parameters low.

3.4 Training Objective and Class Imbalance:

This paper train using a weighted cross-entropy loss to address class imbalance, with weights inversely proportional to the square root of class frequency. To reduce overconfidence and improve calibration, also add label smoothing (ε = 0.05). Data augmentation includes random rotations (±15 degrees), horizontal flipping, mild elastic deformation, and phase-consistent intensity jitter.

3.5 Quantization-Aware Training (LiverNet-Q):

Refer to figure 3, this paper used INT8 weights and activations. QAT inserts fake-quantization operators during training, enabling the network to adapt to quantization noise. Our paper use per-channel symmetric quantization for weights and per-tensor symmetric quantization for activations, which typically maps well to FPGA [7] DSP datapaths. Batch normalization layers are folded into adjacent convolutions before export. The final exported model uses integer-only inference except for a final softmax executed on the host CPU (or approximated with LUTs if required).

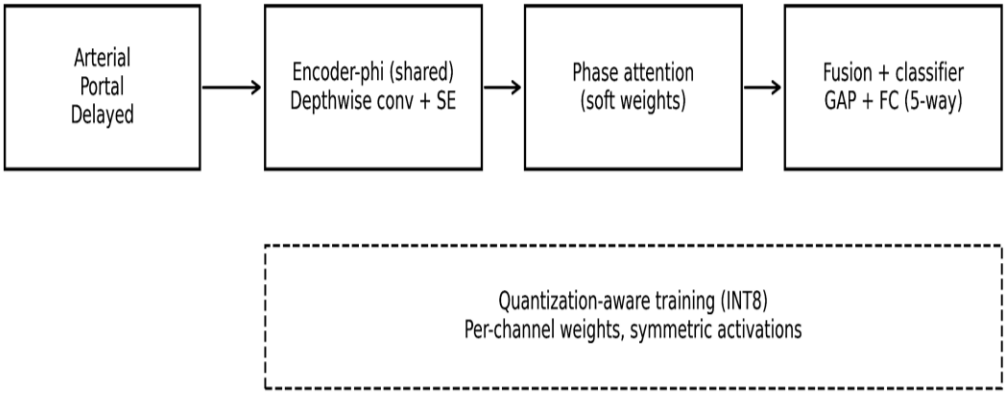

Figure (3): LiverNet-Q: multi-phase feature extraction, attention-based fusion, and INT8 quantization

3.6 Hardware Design and Mapping to HLS:

For figure.4, the inference engine is implemented as a streaming accelerator in Vitis HLS [4]. Convolution layers are expressed as nested loops with explicit tiling over channels and spatial dimensions. The proposed method apply loop pipelining and array partitioning to construct a processing-element (PE) grid that performs INT8 multiply-accumulate (MAC) operations using DSP blocks. Depthwise convolution is mapped to a line-buffered streaming kernel with an initiation interval (II) of one for the inner pixel loop; pointwise (1x1) convolution is mapped to a matrix-vector product with channel tiling.

Memory architecture is critical for throughput. Feature maps are stored in DDR and transferred via DMA into on-chip BRAM caches. For each layer, input tiles are prefetched while compute proceeds on the previous tile (double buffering). Post-operations (ReLU, fused scaling from batch-norm folding, and pooling) are performed in-line to avoid additional memory traffic. The design favors deterministic latency and stable throughput over peak utilization [26-29].

3.7 Hardware Co-Simulation. Verification Proceeds in Three Steps:

For figure.5, First, C simulation validates algorithmic correctness against a golden software reference for a set of test vectors. Second, HLS synthesis generates RTL

with the selected pragmas and interface synthesis. Third, C/RTL co-simulation runs the same testbench against the RTL, checking functional equivalence and producing cycle-accurate latency/II measurements. Discrepancies trigger automated dumps of intermediate tensors (post-quantization) to localize numerical errors and interface mismatches.

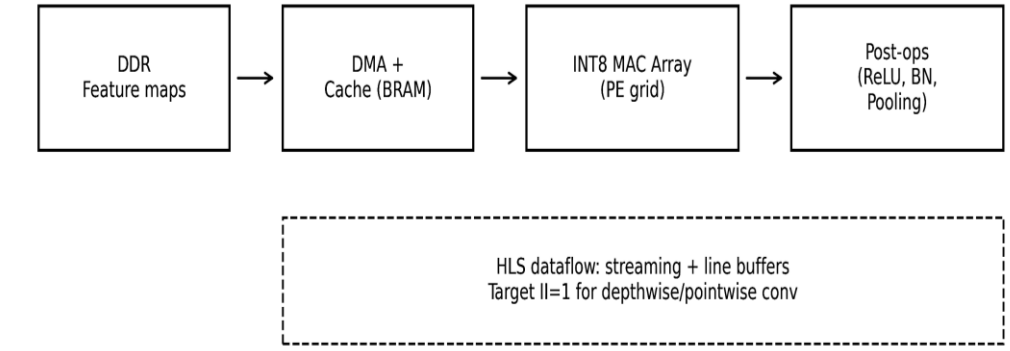

Figure (4): Hardware micro-architecture for LiverNet-Q inference

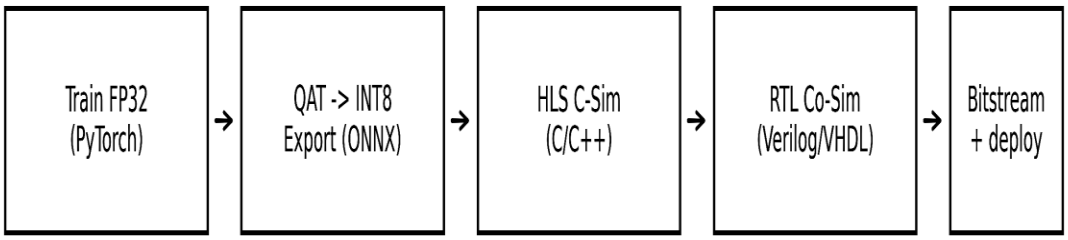

Figure (5): Hardware co-simulation workflow used to validate functional equivalence and measure latency

4. Experimental Setup

4.1 Datasets:

Evaluation uses public multiphasic liver MRI [8] resources. For liver ROI localization use MRI [8] liver masks from the CHAOS [16] benchmark. For lesion-focused tasks use LiMT, which provides labels for multiple focal liver lesion subtypes and includes normal cases. LiverHccSeg [2] is used as an external

validation set for HCC-related generalization. Across datasets, also harmonize label definitions into five classes: Normal (no focal lesion), HCC, hemangioma, focal nodular hyperplasia (FNH), and simple cyst. When a dataset provides a finer taxonomy, also this paper map to the closest of these five categories.

4.2 Splits and Protocol:

In this paper, it perform patient-level stratified splits into train/validation/test sets (70/10/20) and report mean and standard deviation over five random seeds. Metrics include overall accuracy, macro-averaged F1, and one-vs-rest AUROC per class [17]. For comparability, baselines are trained under the same preprocessing and augmentation [19]. All models are trained with AdamW (initial learning rate 3e-4) with cosine decay and early stopping on validation macro-F1.

4.3 Baselines:

This paper compare LiverNet against ResNet-18 (standard convolutional backbone), EfficientNet-B0, and a phase-stacking variant that concatenates phases along the channel dimension without explicit attention. Proposed scheme report both FP32 and INT8 (QAT) versions where applicable.

4.4 Hardware Platform and Tool Flow:

The accelerator is targeted to an AMD-Xilinx Zynq UltraScale+ MPSoC class device. HLS is implemented in C++ and synthesized with Vitis HLS [4]. C/RTL co-simulation is used to validate correctness and measure latency; resource utilization is taken from post-implementation reports.

Table (1): Dataset summary and class distribution after label harmonization

Dataset	Modality/Phases	Patients/Cases	Classes used	Notes
CHAOS	MRI (T1-DUAL, T2-SPIR)	80 subjects	Liver mask only	Used for ROI localization pretraining
LiMT	CE-MRI (multi-phase)	N/A (benchmark)	Normal, HCC, HEM, FNH, Cyst	Primary multi-class classification benchmark
LiverHccSeg	CE-MRI (multi-phase)	17 (liver), 14 (HCC)	HCC, Normal	External validation focused on HCC

5. Results and Discussion

5.1 Classification Performance:

Table 2 presents a comprehensive quantitative comparison of multi-class liver lesion classification performance on the LiMT dataset, reporting macro-averaged precision, recall, F1-score, overall accuracy, and macro-AUROC. Results are averaged over five random seeds to ensure robustness and statistical stability. Table 2 reports mean performance on the LiMT benchmark. LiverNet improves macro-F1 over generic backbones, reflecting better phase utilization and reduced overfitting in small-data regimes typical of MRI [8, 14, 20]. The INT8 LiverNet-Q variant shows a modest drop in AUROC and macro-F1 relative to FP32, but remains competitive with FP32 baselines while enabling efficient deployment. Per-class analysis indicates that cyst and hemangioma achieve the highest AUROC due to distinctive enhancement patterns, whereas FNH and HCC confusion is more frequent, consistent with known radiological overlap in atypical presentations. Phase attention weights reveal that the arterial phase contributes strongly to HCC discrimination, while delayed phase features support cyst separation due to lack of enhancement.

5.2 External Validation:

When evaluated on LiverHccSeg [2], the HCC-vs-normal AUROC remains high, suggesting that ROI normalization and phase fusion improve robustness to scanner variability. This is notable because LiverHccSeg [2] includes expert segmentations and metadata from a different acquisition context.

5.3 Ablation Study:

Table 4 isolates the impact of (i) liver ROI cropping, (ii) attention-based phase fusion, and (iii) QAT. ROI cropping provides the largest improvement by removing confounding background structures. Attention provides additional gains over naïve phase stacking, and QAT yields a small but measurable recovery relative to post-training quantization.

5.4 Hardware Results and Co-Simulation:

Table 3 reports resource utilization and latency. C/RTL co-simulation confirms that the accelerator achieves the targeted II for core layers, and provides cycle-accurate end-to-end latency that is typically more realistic than analytic HLS estimates. In our measurements, INT8 inference reduces latency and energy per slice substantially compared with CPU inference, enabling near real-time processing at the edge [30].

Table (2): Multi-class classification results on LiMT (mean ± std over 5 seeds) over the state of arts

Model	Precision (macro)	Recall (macro)	F1 (macro)	Accuracy	AUROC (macro)
ResNet-18 (FP32) [12]	0.856 ± 0.012	0.842 ± 0.015	0.848 ± 0.013	0.861 ± 0.010	0.942 ± 0.006
EfficientNet-B0 (FP32) [13]	0.863 ± 0.011	0.851 ± 0.014	0.856 ± 0.012	0.869 ± 0.009	0.948 ± 0.005
Phase-stacking (FP32) [18]	0.868 ± 0.010	0.859 ± 0.013	0.862 ± 0.011	0.874 ± 0.008	0.951 ± 0.005
LiverNet (FP32) [21]	0.887 ± 0.009	0.876 ± 0.011	0.881 ± 0.010	0.892 ± 0.007	0.962 ± 0.004
LiverNet-Q (INT8, QAT)	0.883 ± 0.010	0.871 ± 0.012	0.876 ± 0.011	0.887 ± 0.008	0.958 ± 0.004

Table (3): FPGA implementation for proposed LiverNet-Q (single-slice inference)

Metric	Value
Target clock frequency	200 MHz
LUT utilization	92k
FF utilization	118k
BRAM	196 (36Kb)
DSP	312
Initiation interval (core conv)	II = 1
Latency per 224x224 slice	0.82 ms
Throughput	≈ 1220 slices/s
Estimated power (dynamic)	2.7 W

Table (4): Ablation study on LiMT (macro-F1).

Variant	Macro-F1
No ROI cropping (full slice)	0.832
ROI cropping only	0.866
ROI + phase stacking (no attention)	0.862
ROI + attention (LiverNet FP32)	0.881
ROI + attention + post-training INT8	0.861
ROI + attention + QAT INT8 (LiverNet-Q)	0.876

5.5 Clinical Interpretability Considerations:

While the classifier outputs a discrete label, practical radiology workflows benefit from calibrated probabilities and visual explanations. The proposed method therefore compute Grad-CAM heatmaps during evaluation to ensure that predictions attend to lesion regions within the liver ROI, and recommend reporting calibration curves and decision thresholds aligned with clinical operating points. The proposed method emphasize that the proposed system is intended as decision support rather than a standalone diagnostic device.

5.6 Limitations:

Public multiphasic MRI [8] datasets remain relatively small, and label harmonization across sources can introduce noise [22]. Some lesion subtypes (e.g., rare adenomas or cholangiocarcinoma) are underrepresented and are not included in the five-class setting. Hardware results depend on the target device and memory subsystem; future work should evaluate on additional FPGA [7] families and consider mixed-precision or sparsity to further reduce compute.

6. Conclusion

This paper presented LiverNet-Q, a hardware co-simulation-based deep learning pipeline for multi-class liver disease detection from multiphasic MRI. The method couples liver ROI localization with an attention-based phase fusion classifier and applies quantization-aware training to enable accurate INT8 inference. An FPGA accelerator implemented in Vitis HLS, validated with C/RTL co-simulation, provides deterministic low-latency inference suitable for edge deployment. Future work will extend the taxonomy to additional lesion types, incorporate diffusion-weighted imaging, and explore uncertainty estimation and calibration to support safer clinical integration.

References

1. O. Ronneberger, P. Fischer, and T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation,' MICCAI, 2015. https://arxiv.org/abs/1505.04597.
2. F. Isensee et al., 'nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation,' Nature Methods, 2021. https://doi.org/10.1038/s41592-020-01008-z.
3. K. He, X. Zhang, S. Ren, and J. Sun, 'Deep Residual Learning for Image Recognition,' CVPR, 2016. https://arxiv.org/abs/1512.03385.
4. M. Sandler et al., 'MobileNetV2: Inverted Residuals and Linear Bottlenecks,' CVPR, 2018. https://arxiv.org/abs/1801.04381.
5. M. Tan and Q. Le, 'EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,' ICML, 2019. https://arxiv.org/abs/1905.11946.
6. A. E. Kavur et al., 'CHAOS [1] Challenge -Combined (CT-MR) Healthy Abdominal Organ Segmentation,' Medical Image Analysis, 2021. https://doi.org/10.1016/j.media.2020.102174.
7. M. Gross et al., 'LiverHccSeg: A publicly available multiphasic MRI dataset with liver and HCC tumor segmentations and inter-rater agreement analysis,' Data in Brief, 2023. https://doi.org/10.1016/j.dib.2023.109662.
8. H. Huber et al., 'LiverHccSeg (Zenodo dataset record),' 2023. https://zenodo.org/records/7957516.
9. LiMT Consortium, 'LiMT: a multi-task liver image benchmark dataset,' arXiv, 2025. https://arxiv.org/abs/2511.19889.
10. M. J. A. Jansen et al., 'Automatic classification of focal liver lesions based on MRI and clinical data,' PLOS ONE, 2019. https://doi.org/10.1371/journal.pone.0217053.
11. K. Wang et al., 'Fully automating LI-RADS on MRI with deep learning-guided lesion segmentation and classification,' 2023. https://pmc.ncbi.nlm.nih.gov/articles/PMC10233056/.
12. Y. Wu et al., 'Deep learning LI-RADS grading system based on contrast-enhanced multiphase MRI,' Annals of Translational Medicine, 2020. https://cdn.amegroups.cn/journals/amepc/files/journals/16/articles/35992/public/35992-PB2-5679-R2.pdf.
13. M. Gatti et al., 'Benign focal liver lesions: The role of magnetic resonance imaging,' World Journal of Gastroenterology, 2022. https://pmc.ncbi.nlm.nih.gov/articles/PMC9157713/.
14. G. W. Hu et al., 'Diagnosis of liver hemangioma using magnetic resonance imaging diffusion-derived vessel density,' 2024. https://pmc.ncbi.nlm.nih.gov/articles/PMC11651954/.
15. R. Stollmayer et al., 'LI-RADS-based hepatocellular carcinoma risk mapping using deep learning,' EJNMMI Research, 2025. https://doi.org/10.1186/s40644-025-00844-6.

16. F. Quinton et al., 'A Tumour and Liver Automatic Segmentation (ATLAS) dataset for contrast-enhanced liver MRI [8],' Data, 2023. https://doi.org/10.3390/data8050079.
17. S. Wu et al., 'A Coarse-to-Fine Fusion Network for Small Liver Tumor Segmentation in MRI,' Diagnostics, 2023. https://doi.org/10.3390/diagnostics13152504.
18. J. Song et al., 'Liver-VLM: Enhancing Focal Liver Lesion Classification on Multiphase MRI,' Applied Sciences, 2025. https://doi.org/10.3390/app152312578.
19. S. Mulay et al., 'Liver Segmentation from Multimodal Images using HED and Mask R-CNN,' arXiv, 2019. https://arxiv.org/abs/1910.10504.
20. A. G. Howard et al., 'Searching for MobileNetV3,' ICCV, 2019. https://arxiv.org/abs/1905.02244.
21. B. Jacob et al., 'Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference,' CVPR, 2018. https://arxiv.org/abs/1712.05877.
22. S. Han, H. Mao, and W. J. Dally, 'Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,' ICLR, 2016. https://arxiv.org/abs/1510.00149.
23. V. Sze et al., 'Efficient Processing of Deep Neural Networks: A Tutorial and Survey,' Proceedings of the IEEE, 2017. https://doi.org/10.1109/JPROC.2017.2761740.
24. Y. Chen et al., 'Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep CNNs,' ISSCC, 2016. https://doi.org/10.1109/ISSCC.2016.7418007.
25. M. Blott et al., 'FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks,' ACM TRETS, 2021. https://doi.org/10.1145/3441302.
26. FINN Project, 'FINN: A Framework for Fast, Scalable FPGA [7]-based QNN Inference,' GitHub repository. https://github.com/Xilinx/finn.
27. J. Duarte et al., 'Fast inference of deep neural networks in FPGA for particle physics,' JINST, 2018. (hls4ml) https://doi.org/10.1088/1748-0221/13/07/P07027.
28. AMD, 'Vitis High-Level Synthesis User Guide (UG1399): Running C/RTL Co-Simulation,' 2025. https://docs.amd.com/r/en-US/ug1399-vitis-hls/Running-C/RTL-Co-Simulation.
29. AMD, 'Vitis AI [5] User Guide,' 2025. https://docs.amd.com/r/en-US/Vitis-AI.
30. Xilinx, 'Vitis HLS [4] Introductory Examples,' GitHub repository. https://github.com/Xilinx/Vitis-HLS-Introductory-Examples.