DOI: 10.61782/fa.2025.0426

HARDWARE ACCELERATION OF CONVOLUTIONAL NEURAL NETWORK FOR LUNG ULTRASOUND SEGMENTATION

A. Rubio[1,2,*] **M. Muñoz**[1,2] **G. Cosarinsky**[1]
J. F. Cruza[1]

[1] Institute for Physical and Information Technologies, Spanish National Research Council,
28006 Madrid, Spain
[2] Electronic Department, Universidad de Alcalá, 28805 Alcalá de Henares, Spain

ABSTRACT

Lung Ultrasound (LUS) imaging is a valuable diagnostic technique for lung condition evaluation, due to its non-ionizing and portable nature. However, its complex interpretation can be enhanced by Machine Learning (ML) tools, yet traditional solutions often fail to meet the speed demands of real-time applications. This paper presents a Field Programmable Gate Array (FPGA)-based hardware solution for real-time segmentation of lung ultrasound images using a Convolutional Neural Network (CNN), achieving a throughput of 80 inferences per second.

Keywords: *Machine Learning (ML), Lung ultrasound (LUS), Field Programable Gate Array (FPGA), hardware acceleration, edge computing.*

1. INTRODUCTION

Artificial Intelligence (AI) has revolutionized many scientific fields. In medicine, AI tools have aided healthcare professionals in more accurate and effective patient diagnoses [1, 2].
LUS is a non-invasive imaging technique increasingly used for evaluating respiratory diseases [3]. Although its acquisition is fast, LUS interpretation requires expertise and can result challenging for less experienced clinicians [4] .
Muñoz et al. [5] developed a Machine Learning (ML) model based in the U-NET architecture [6], trained on a video dataset from 30 patients, acquired and labeled at video level by an expert physician. That manual tagging was extended to frame level by a semiautomatic labeling tool developed in the same work. The model processes the B-Scans and generates four outputs (pleura, consolidation, B-line, and A-line) representing the probability of each artifact per pixel. While achieving high performance, its GPU implementation (Mini-PC with integrated GTX-2060) limits its usability in low-power consumption portable devices and clinical enviroments.
Field Programmable Gate Arrays (FPGAs) are integrated circuits that can be reprogrammed after manufacturing, showing promising results in accelerating ML models [7], due to their versatility, low power consumption, and small size.

This work proposes a high-throughput implementation using Vitis-AI, an open source framework that provides the tools to implement AI models it a cost-effective FPGA real-time solution, demonstrating its viability for ML hardware acceleration.

2. METHODS

2.1 Neural Network

Convolutional Neural Networks (CNNs) are widely used for segmentation tasks due to their ability to identify both

Corresponding author: adrian.rubio@csic.es.

local features and global patterns. The U-Net architecture has skip connections that incorporate spatial information at different scales, thus preserving structural details. In this work, attention gates, used in the skip connections of [5] were removed from the implemented model due to limitations with Vitis-AI on FPGA fabric. One of the objectives of this work is to analyze how the performace is affected by this modification.

A simplified version of the architecture is shown in Fig. 1. The 256x128 input image first goes through the encoder, where increasingly abstract information is extracted, while spatial resolution is reduced. The decoder processes and upsamples the extracted features thanks to skip-connections, which help maintain crucial spatial information. This results in four 256x128 output images, each segmenting a relevant artifact (Pleura, consolidations, B-lines and A-lines).
This model underwent the same training as the original and serves as the baseline for comparing subsequent modifications.
Python implementations of ML are typically optimized for CPU or GPU architectures, which differ significantly from the architecture of FPGAs. The original model needs to go through some transformations to make it compatible with FPGA constraints. As a consequence, both model size and inference time decrease. The processes required to implement the model in FPGA are pruning (removal of weights with values close to zero and retraining to recover accuracy) and quantization (conversion of model parameters from floating-point to fixed-point representations). For this specific implementation only quantization was required.

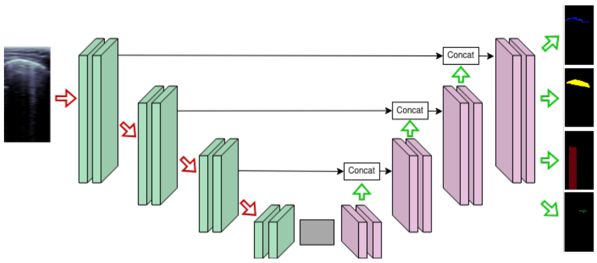


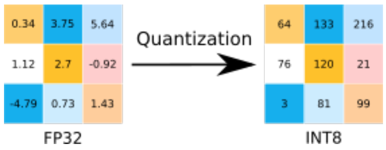
Figure 1. U-NET architecture. The first half encodes and reduces the dimensionality of the input, while the second half decodes and expands.

2.1.1 *Training*

The Python model was trained with 9624 images (70 % train, 30 % validation), using a learning rate of 0.0005. The loss function employed was binary cross-entropy, with a batch size of 64 images. Training was conducted for a maximum of 100 epochs, with early stopping activated if the validation loss did not decrease for 6 consecutive epochs. Upon completion of the training process, the model was prepared for quantization.

2.1.2 *Quantization*

FPGAs are not well-suited for the 32-bit floating-point precision commonly used in Python models. Instead, their computational capabilities are optimized for 8-bit fixed-point operations. This process, known as quantization, involves retraining the original model with a subset of the dataset, readjusting parameters and activations, and rounding decimals beyond a specific point (Fig. 2). The quantized model weighs less than a quarter of the original model.



0.34	3.75	5.64
1.12	2.7	-0.92
-4.79	0.73	1.43

64	133	216
76	120	21
3	81	99

Figure 2. Quantization simple example.

2.2 FPGA architecture

The model is implemented on the Kria KV260 Vision AI Starter Kit, an FPGA particularly well-suited for real-time computer vision and edge computing applications. Its low cost and limited power consumption make it a suitable solution for a portable IA-assisted diagnosis tool.
The Deep Learning Processing Unit (DPU) is a programmable core for AMD FPGAs optimized for accelerating neural network computations through a specialized instruction set. The DPU processes the input data, provided with the correct dimensionality, performs the necessary computations (such as convolutions), and stores the segmentation results in the FPGA memory for further use.
Vitis-AI was used to retrain, modify, quantize, and tune the parameters of the model in an intuitive manner.

After successfully creating the hardware project, including the DPU and its bus connections with the embedded microprocessor and system memory, we generated a PetaLinux image configured to load the trained ML model and perform the segmentation process.
A dataset of 1358 validation images is transferred to the board. A Python script executed on the board processes the images to generate the segmentation masks. These masks are then transferred back to the host computer for analysis.

3. RESULTS

Fig. 3 and Fig. 4 show two segmentation examples. In each figure, the left image represents the tagged segmentation (ground truth) of the LUS artifacts, color-coded as follows: pleura (blue), consolidations (yellow), B-lines (red), and A-lines (green). The middle image shows the output of the original U-NET model (thresholded at 0.5), using the same color scheme. The right image shows the output of the FPGA-implemented model with the same threshold. Visually, the FPGA results are almost identical to the original model's.

Comparing the segmentations indicates that the model's performance was not significantly affected by the modifications required for FPGA implementation. Tab. 1 presents three evaluation metrics, calculated for all artifacts across the test dataset: the **Dice** coefficient (measuring the overlap between two sets), **precision** (the percentage of correctly identified positive pixels), and **recall** (the percentage of retrieved positive pixels). The FPGA-implemented model shows very similar performance to the original model in detecting all artifacts.

The model achieves a throughput of more than 80 inferences per second, compared to the the 50 per second reached in the previous GPU implementation.

4. DISCUSSION

This study successfully implemented a LUS segmentation model, originally developed in Python, onto an FPGA. The necessary modifications for FPGA compatibility were achieved with minimal impact on the model's performance. The results demonstrate that the FPGA-implemented model closely replicates the behavior of the original software model, as shown in Tab. 1.

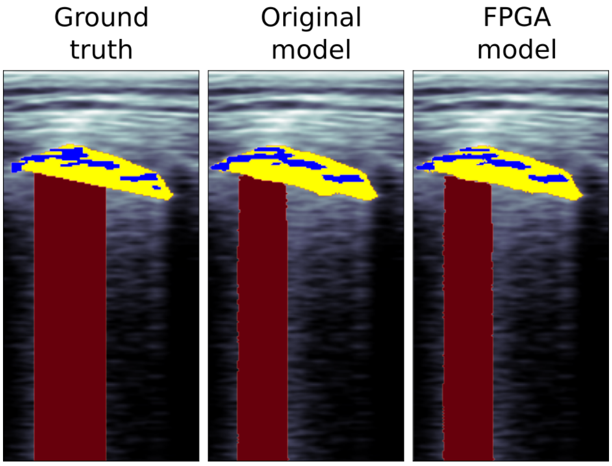


Figure 3. Pleura, B-line and Consolidation output example.

Table 1. Metrics for all artifacts and the two different models.

	Metric	Original	FPGA
Pleura	DICE	0.86	0.86
	Precision	0.85	0.88
	Recall	0.89	0.85
Consolidation	DICE	0.97	0.97
	Precision	0.97	0.98
	Recall	0.99	0.99
B-line	DICE	0.77	0.76
	Precision	0.90	0.90
	Recall	0.82	0.82
A-line	DICE	0.73	0.71
	Precision	0.78	0.77
	Recall	0.80	0.79

The achieved latency (12.5 ms), throughput, and power consumption demonstrate that FPGAs offer a viable alternative to traditional GPUs for hardware acceleration of AI-assisted LUS diagnostic tools. The compact size of the FPGA makes it particularly suitable for integration into small, portable ultrasound scanners.

The KV260 evaluation board (14cmx12cmx3.5cm) is smaller than the mini-PC with an integrated GPU

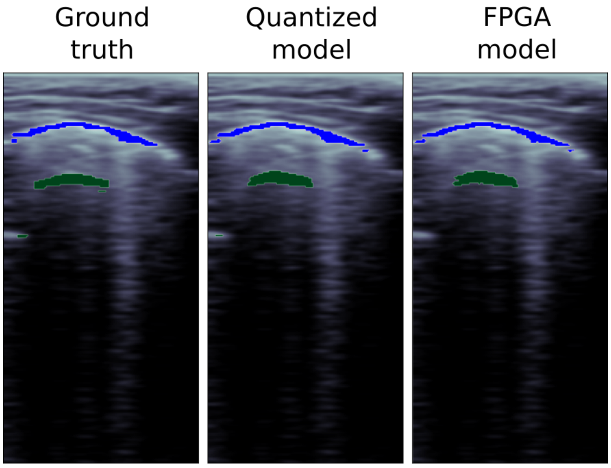


Figure 4. Pleura and A-line output example.

(21cmx19cmx5cm) employed in [5], and its use would eliminate the bottleneck caused by transferring the raw data to the PC for inference. Energy-wise, following the literature [8] FPGAs are more energy-efficient than GPUs, but a detailed study needs to be carried out to conclude that.

While the results are promising, this study was conducted using an offline test, processing all images in batch. Future work will focus on implementing frame-by-frame segmentation, integrating the neural network into the real-time data acquisition pipeline, and displaying the results to the physician on a screen.

5. CONCLUSIONS

Our findings demonstrate the suitability of FPGA acceleration for real-time LUS diagnosis due to its high inference speed, small size, and low power consumption with a cost-effective device. A key finding is that the quantization process bears almost no appreciable effect in the model performance, while reducing the size to a quarter of the original. Similarly, the removal of attention gates had neglible impact on the model's effectiveness.

The developed pipeline has been optimized for the specific characteristics of LUS segmentation. However, its core steps—model modification, quantization, and implementation—are adaptable and can be applied to other ML models and FPGA platforms. This underscores the potential of FPGAs in edge-computing by enabling rapid inference directly at the point of data acquisition.

6. ACKNOWLEDGEMENTS

Supported by the European Commission + NextGenerationEU, through Momentum CSIC Programme: Develop Your Digital Talent. The funding for these actions/grants and contracts comes from the European Union's Recovery and Resilience Facility-Next Generation, in the framework of the General Invitation of the Spanish Government's public business entity Red.es to participate in talent attraction and retention programmes within Investment 4 of Component 19 of the Recovery, Transformation and Resilience Plan. G. Cosarinsky staff is hired under the Generation D initiative, promoted by Red.es, an organisation attached to the Ministry for Digital Transformation and the Civil Service, for the attraction and retention of talent through grants and training contracts, financed by the Recovery, Transformation and Resilience Plan through the European Union's Next Generation funds.

Supported by the project PID2022-143271OB-I00, funded by MCIN/AEI /10.13039/501100011033/FEDER, UE.

7. REFERENCES

[1] M. L. Marinovich, E. Wylie, W. Lotter, H. Lund, A. Waddell, C. Madeley, G. Pereira, and N. Houssami, “Artificial intelligence (ai) for breast cancer screening: Breastscreen population-based cohort study of cancer detection,” *eBioMedicine*, vol. 90, Apr 2023.

[2] J. Liao, X. Li, Y. Gan, S. Han, P. Rong, W. Wang, W. Li, and L. Zhou, “Artificial intelligence assists precision medicine in cancer treatment,” *Frontiers in Oncology*, vol. 12, 2023.

[3] M. Beshara, E. A. Bittner, A. Goffi, L. Berra, and M. G. Chang, “Nuts and bolts of lung ultrasound: utility, scanning techniques, protocols, and findings in common pathologies,” *Critical Care*, vol. 28, p. 328, Oct 2024.

[4] J. L. Herraiz, C. Freijo, J. Camacho, M. Muñoz, R. González, R. Alonso-Roca, J. Álvarez-Troncoso, L. M. Beltrán-Romero, M. Bernabeu-Wittel, R. Blancas, *et al.*, “Inter-rater variability in the evaluation of

lung ultrasound in videos acquired from covid-19 patients," *Applied Sciences*, vol. 13, no. 3, p. 1321, 2023.

[5] M. Muñoz, A. Rubio, G. Cosarinsky, J. F. Cruza, and J. Camacho, "Deep learning-based algorithms for real-time lung ultrasound assisted diagnosis," *Applied Sciences*, vol. 14, no. 24, 2024.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[7] A. Nechi, L. Groth, S. Mulhem, F. Merchant, R. Buchty, and M. Berekovic, "Fpga-based deep learning inference accelerators: Where are we standing?," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 16, Oct. 2023.

[8] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey and benchmarking of machine learning accelerators," in *2019 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–9, 2019.