



FPGA-Based ResNet-50 Acceleration for Automated Pneumonia Detection from Chest X-Ray Images

Vinaya V. Mane¹, Vishwanath R. Potdar², Nainika N. Patil³, P. H. Mane⁴

^{1,2,3}Department of Technology, Electronics & Telecommunication Engineering, Shivaji University, Kolhapur, Maharashtra, India.

⁴Project Guide, Professor, M.Tech, Electronic Engineering, Department of Technology, Shivaji University, Kolhapur, Maharashtra, India.

To Cite this Article: Vinaya V. Mane¹, Vishwanath R. Potdar², Nainika N. Patil³, P. H. Mane⁴, “FPGA-Based ResNet-50 Acceleration for Automated Pneumonia Detection from Chest X-Ray Images”, Indian Journal of Electronics and Communication Engineering, Volume 03, Issue 02, May-August 2026, PP: 40-46.

Abstract: Pneumonia remains a leading cause of morbidity and mortality worldwide, and timely radiological screening is essential for effective treatment, particularly in settings where specialist radiologists are scarce. Convolutional neural networks (CNNs) classify chest X-ray images as normal or pneumonic with high accuracy, but their computational cost limits deployment on portable, low-power clinical devices. This paper presents an end-to-end pipeline that trains a ResNet-50 network, using transfer learning, to classify chest radiographs and deploys the trained model on a Xilinx ZCU104 Zynq UltraScale+ FPGA board through the Vitis AI toolchain. Images from a public pediatric pneumonia dataset (5216 training, 624 test, 16 validation images) are resized, normalized, and evaluated under several classical filtering operations before being passed to the network, which combines residual convolutional blocks, ReLU activations, and a sigmoid output layer. The model is trained for 30 epochs with the Adam optimizer and binary cross-entropy loss, reaching a validation accuracy in the 90–92% range. After quantization and compilation for the on-board Deep-Learning Processing Unit, the FPGA implementation achieves an inference throughput of 1250 frames per second at 4.1 W, compared with 44 frames per second at 19.9 W on a general-purpose CPU baseline — approximately a 28-fold gain in throughput and a 138-fold gain in energy efficiency. These results indicate that SoC-FPGA platforms are a practical route to real-time, energy-efficient pneumonia screening at the point of care.

Key Words: Convolutional Neural Network, ResNet-50, FPGA, Zynq UltraScale+, Pneumonia Detection, Chest X-Ray Classification, Vitis AI, Edge AI.

1. INTRODUCTION

Pneumonia is an acute infection of the lung parenchyma and remains among the leading causes of hospitalization and mortality worldwide, particularly among young children, the elderly, and populations with limited access to specialist care. Chest radiography is the primary imaging modality used to confirm a clinical diagnosis of pneumonia, yet its interpretation still depends heavily on the availability and experience of a trained radiologist, a resource that is unevenly distributed across health systems. Convolutional neural networks (CNNs) have proven effective at learning discriminative visual patterns directly from chest X-ray images, and several studies have shown that CNN-based classifiers can approach the diagnostic performance of practicing radiologists on pneumonia screening tasks. Translating this accuracy into clinical impact, however, requires that inference run where the image is acquired — at the point of care — rather than solely on a networked server. General-purpose CPUs are widely available but offer limited throughput per watt for the dense matrix operations that dominate CNN inference, while GPU accelerators, though fast, draw power levels that are difficult to sustain in portable or field-deployable diagnostic equipment.

Field-Programmable Gate Arrays (FPGAs) offer an attractive middle ground. Their fine-grained, reconfigurable logic can be tailored to the multiply-accumulate-heavy structure of convolutional layers, and System-on-Chip (SoC) FPGA devices additionally integrate embedded ARM processors that simplify system-level software development. Realizing this potential in practice, however, requires navigating a nontrivial deployment pipeline: models must be quantized, compiled for a target Deep-Learning Processing Unit (DPU), and validated for accuracy loss introduced during conversion — steps that are rarely reported in end-to-end detail alongside a clinically relevant classification task.

This paper addresses that gap by presenting a complete pipeline that trains a ResNet-50 classifier for pneumonia detection from chest radiographs and deploys the trained model on a Xilinx ZCU104 Zynq UltraScale+ FPGA board using the Vitis AI toolchain, reporting both diagnostic and system-level performance. The main contributions of this work are:

- An end-to-end workflow combining ResNet-50 transfer learning with Vitis AI quantization and compilation for deployment on a Zynq UltraScale+ SoC-FPGA.
- A comparative evaluation of classical image-filtering operations (erosion, dilation, Gaussian blurring, Canny edge detection) as preprocessing candidates for chest radiograph classification.
- A quantitative throughput and power comparison between CPU and FPGA inference on the same trained model.
- A working demonstration of the deployed system through both a browser-based interface and on-board Jupyter inference on the ZCU104.

II. RELATED WORK

A. CNN-Based Pneumonia Classification

Deep learning approaches to pneumonia screening have progressed rapidly since Rajpurkar et al. introduced CheXNet, a 121-layer CNN that matched practicing radiologists on the ChestX-ray14 benchmark [1]. Much of this progress builds on the pediatric chest-radiograph corpus released by Kermany et al. [2], which established transfer learning from ImageNet-pretrained backbones as an effective strategy for medical image classification with limited labeled data. Subsequent work has explored a range of CNN backbones for the same task: Ansari et al. reported strong results with a ResNet-50 transfer-learning pipeline and noted that shallower networks such as VGG-16 struggled to capture the subtle pathology patterns present in chest radiographs [4]; Racic et al. and Szepesi and Szilagyi each proposed custom CNN architectures and reported the persistent difficulty of distinguishing early-stage pneumonic opacities from normal variation [5], [6]; and Shah et al. and Antin et al. examined lighter CNNs trained from scratch, trading some accuracy for reduced model complexity [7], [8]. Across this body of work, ResNet-style residual connections are consistently reported to ease optimization of the deeper networks needed for competitive accuracy, motivating the choice of ResNet-50 as the backbone in the present study.

B. FPGA Acceleration of CNNs

A separate line of work focuses on making CNN inference tractable on FPGA fabric. Huynh proposed a hardware-software co-design framework for mapping CNN layers onto reconfigurable logic and highlighted the latency and power advantages of FPGAs over CPU/GPU baselines for embedded vision applications [9]. Farrukh et al. addressed a specific bottleneck in this mapping — the limited number of hardened multiply-accumulate units on commodity FPGAs — by replacing conventional MAC-based processing elements with Wallace-tree multipliers, reducing both resource usage and critical-path delay in the convolution engine [10]. These studies establish hardware-level techniques that make FPGA-based CNN acceleration practical, but they are evaluated on generic vision benchmarks rather than a clinically motivated classification task.

C. Research Gap

The pneumonia-classification literature optimizes for diagnostic accuracy on general-purpose hardware, while the FPGA-acceleration literature optimizes for throughput and resource efficiency on generic CNN workloads; few studies report both diagnostic performance and system-level throughput/power figures for the same clinically relevant model deployed on a commercially available SoC-FPGA development board. This work targets that intersection, using a widely available development platform (ZCU104) and a public dataset to keep results reproducible. Table 1 summarizes this gap qualitatively against representative prior work.

Study	Domain	Reports Accuracy	HW Throughput/Power
Kermany et al. [2]	Medical CNN	Yes	No
Ansari et al. [4]	Medical CNN	Yes	No
Huynh [9]	FPGA CNN	No	Yes
Farrukh et al. [10]	FPGA CNN	No	Yes
This work	Medical CNN + FPGA	Yes	Yes

Table 1: Positioning relative to representative prior work

III. PROPOSED METHODOLOGY

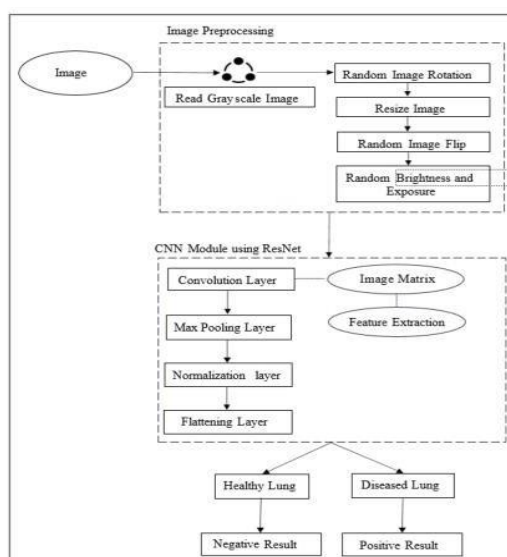


Fig. 1. Block diagram of the proposed image-preprocessing and CNN classification pipeline, from raw grayscale input through convolution, pooling, normalization, and flattening to the healthy/diseased decision.

Fig. 1 summarizes the overall pipeline: chest radiographs are preprocessed and augmented, passed through a ResNet-50 CNN for feature extraction and binary classification, and the trained model is subsequently quantized and deployed to the ZCU104 FPGA for accelerated inference. Algorithm 1 restates this pipeline as a compact procedural flow.

Algorithm 1: End-to-End Classification and Deployment Pipeline

1. **Input:** Grayscale chest radiograph.
2. Apply random rotation, resize, horizontal flip, and brightness/exposure jitter (training only).
3. Normalize pixel intensities to $[0, 1]$ and resize to $224 \times 224 \times 3$.
4. Forward-pass through ResNet-50 residual blocks; extract feature maps via convolution and max-pooling.
5. Flatten pooled feature maps and pass through the fully connected layer.
6. Apply sigmoid activation; threshold output probability to obtain Normal/Pneumonia label.
7. Quantize trained weights to INT8 using Vitis AI and compile to a DPU-targeted xmodel.
8. Deploy the compiled model to the ZCU104 DPU; serve predictions via Jupyter or the Gradio front end.

A. Dataset and Preprocessing

The classifier is trained and evaluated on a public pediatric chest radiograph corpus originally released by Kermany et al. [2] and widely used for pneumonia-screening benchmarks; each image is labeled as Normal or Pneumonia. Table 2 summarizes the partition used in this work, and Fig. 2 shows representative examples from each class.

Subset	Number of Images
Training	5216
Testing	624
Validation	16

Table 2: Dataset summary

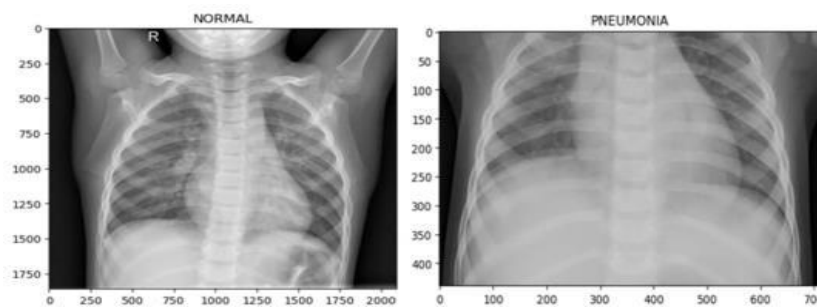


Fig. 2. Representative radiographs from the dataset: a structurally clear lung field labeled Normal (left) versus a radiograph with visible opacities labeled Pneumonia (right).

All images are resized to 224×224 pixels with three color channels and normalized to the $[0, 1]$ range before being passed to the network. Because the validation split is comparatively small, validation metrics reported in Section V should be read as indicative trends rather than statistically definitive estimates. Independently of the normalization pipeline, four classical filtering operations — morphological erosion, morphological dilation, Gaussian blurring, and Canny edge detection — were applied to representative images to assess their effect on lesion contrast and boundary visibility; Gaussian blurring and Canny edge detection in particular proved useful for visualizing opacity boundaries during preprocessing analysis, although the CNN itself was trained on the normalized, unfiltered images.

B. Network Architecture

The classification backbone is ResNet-50, a 50-layer convolutional network composed of 48 convolutional layers together with a max-pooling and a global average-pooling layer, as shown in Fig. 3. Its defining feature is the residual (skip) connection, which adds the input of a block directly to its output and thereby allows gradients to propagate through very deep stacks of layers without vanishing — a property that earlier, plain CNN architectures lack.

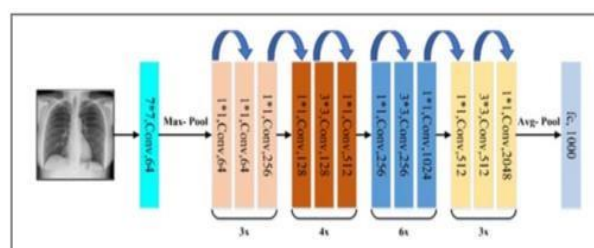


Fig. 3. ResNet-50 architecture used as the classification backbone, showing the initial 7×7 convolution and the four stages of bottleneck residual blocks (repeated 3, 4, 6, and 3 times, respectively) leading to the final fully connected classification layer.

Each residual block performs a convolution operation, illustrated schematically in Fig. 4, in which a learned kernel slides across the input feature map and computes a weighted sum at each spatial position to produce an output feature map that highlights localized patterns such as edges and opacity boundaries.

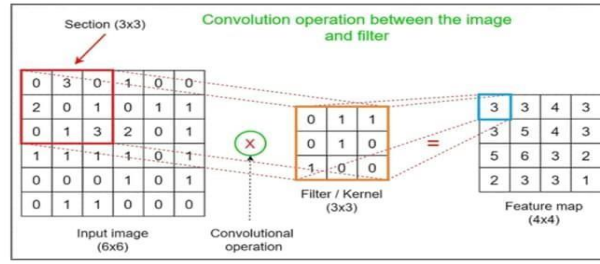


Fig. 4. Convolution operation between a local region of the input image and a learned 3×3 kernel, producing one entry of the resulting feature map.

Following each convolutional stage, a max-pooling operation (Fig. 5) down-samples the feature maps by retaining the maximum activation within each local window, which reduces spatial resolution and parameter count while preserving the strongest activations relevant to classification.

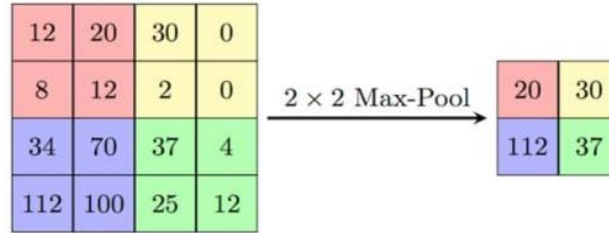


Fig. 5. 2×2 max-pooling operation reducing the spatial resolution of a feature map by retaining the maximum value within each non-overlapping window.

Each convolutional block is followed by a rectified linear unit (ReLU) activation,

$$f(x) = \max(0, x) \quad (1)$$

which introduces the non-linearity needed to learn complex decision boundaries while remaining computationally inexpensive relative to alternatives such as tanh at hidden layers. After the final residual stage, the feature maps are flattened and passed to a fully connected layer, and the network terminates in a sigmoid output unit,

$$\sigma(x) = 1 / (1 + e^{-x}) \quad (2)$$

which maps the final logit to a probability of the Pneumonia class, since the task is posed as binary classification.

C. Training Configuration

The network is trained end-to-end using the Adam optimizer [11] to minimize the binary cross-entropy loss,

$$L = -(1/N) \sum [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (3)$$

where y_i is the ground-truth label and $\hat{y}_i$ is the predicted probability for the i -th sample. Training was run for 30 epochs using TensorFlow/Keras with NumPy, Pandas, and Matplotlib for data handling and visualization, inside an Anaconda-managed Python environment. The trained network provides the reference model that is subsequently quantized and compiled for FPGA deployment, described next.

IV. FPGA-BASED HARDWARE IMPLEMENTATION

The target platform is the Xilinx ZCU104 evaluation board (Fig. 6), built around a Zynq UltraScale+ MPSoC (part XCZU7EV-2FFVC1156) that combines programmable logic with a quad-core ARM Cortex-A53 application processor and a dual-core Cortex-R5 real-time processor on a single die. The board additionally exposes HDMI input/output, DisplayPort, Gigabit Ethernet, USB/JTAG-UART, a PL-side SODIMM socket, SATA/M.2, and PMBus power-monitoring headers, which together support both the machine-learning workload and the peripheral I/O needed for a self-contained diagnostic demonstrator.

Deployment follows the standard Vitis AI flow. The trained Keras model is first quantized from floating-point to a fixed-point (INT8) representation using a calibration subset of the training data, then compiled by the Vitis AI compiler into an instruction graph (xmodel) targeting the on-chip Deep-Learning Processing Unit (DPU) IP core synthesized into the ZCU104's programmable

logic [12]. The compiled model is transferred to the board over Ethernet — the board is configured with a static IP address and accessed through a Xilinx-hosted Jupyter environment — after which inference can be triggered directly from a notebook running on the embedded ARM cores while the DPU executes the convolutional layers.

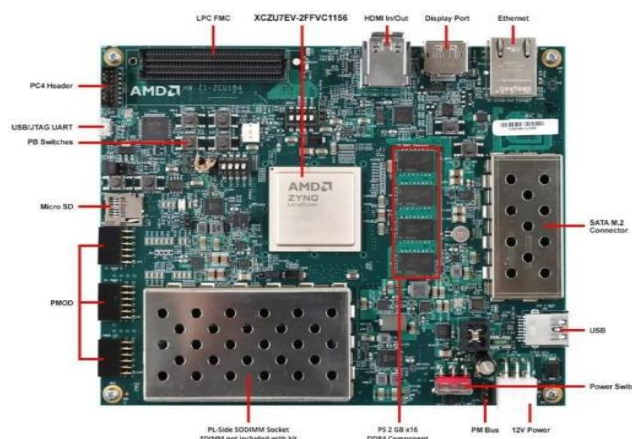


Fig. 6. Xilinx ZCU104 Zynq UltraScale+ MPSoC evaluation board used for on-board DPU inference, with key interfaces (HDMI, DisplayPort, Ethernet, PMOD, SATA/M.2, and power) labeled.

Two front ends were built around this deployment for demonstration and testing: a browser-based interface (using Gradio) that accepts an uploaded or camera-captured chest X-ray and returns a Normal/Pneumonia prediction, and a native on-board test path that runs inference directly from a Jupyter notebook on the ZCU104 using image files supplied by path. Both paths exercise the same compiled DPU model and were used to sanity-check predictions against the held-out test set described in Section III-A.

V. RESULTS AND DISCUSSION

A. Training Behavior

Training and validation accuracy both increased over the 30 training epochs (Fig. 7), converging to a validation accuracy in the neighborhood of 90–92%, with the binary cross-entropy loss decreasing correspondingly to below 0.2 by the final epochs. The validation curve fluctuates more than the training curve, which is expected given the 16-image validation split noted in Section III-A; the overall upward trend, rather than any single epoch's value, is the more reliable indicator of convergence.

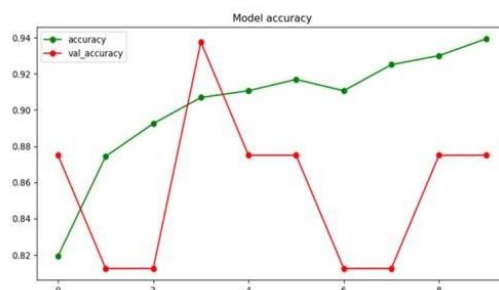
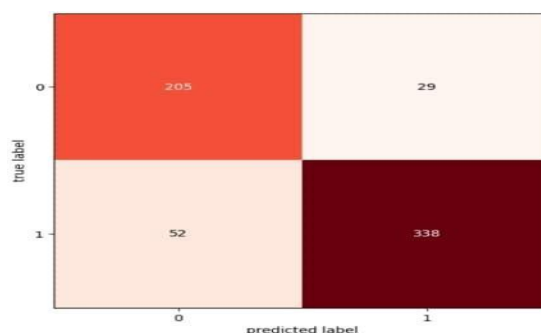


Fig. 7. Training and validation accuracy over 30 epochs. The validation curve (red) fluctuates due to the small validation split, while the training curve (green) rises steadily, indicating stable convergence without severe overfitting.

B. Classification Performance

The confusion matrix computed on the 624-image test set shows the model correctly classifying the large majority of both Normal and Pneumonia cases, with misclassifications concentrated in a comparatively small number of borderline cases (Fig. 8). This pattern is consistent with prior CNN-based pneumonia studies, which report that early-stage or subtle pneumonic opacities



are the most common source of disagreement between automated classifiers and ground-truth radiological labels [5], [6].

Fig. 8. Confusion matrix of the trained classifier on the 624-image held-out test set.

C. Hardware Performance

Table 3 compares inference throughput and power draw for the compiled model running on the ZCU104's DPU against a general-purpose CPU baseline executing the same network.

Metric	CPU	FPGA (ZCU104)
Throughput (fps)	44	1250
Power (W)	19.9	4.1
Efficiency (fps/W)	2.2	304

Table 3: CPU vs. FPGA inference performance

The FPGA implementation delivers approximately a 28-fold increase in raw throughput and, because it does so at roughly one-fifth the power draw of the CPU baseline, an approximately 138-fold improvement in energy efficiency (frames per second per watt). This gap follows directly from architectural differences between the two platforms: the DPU's array of parallel multiply-accumulate units is purpose-built for the convolution operations that dominate ResNet-50 inference, whereas the CPU executes the same operations largely sequentially at a much higher clock frequency and correspondingly higher power draw. For a battery-powered or field-deployed diagnostic device, the efficiency gain is arguably more consequential than the raw speedup, since it ultimately governs how long a portable unit can operate between charges.

D. Discussion

The results support the practicality of SoC-FPGA platforms for real-time, energy-constrained pneumonia screening: the combination of high throughput and low power draw is well suited to point-of-care devices operating outside a hospital's central IT infrastructure, and the reconfigurable fabric leaves room to update or extend the deployed model without a hardware redesign. Set against these advantages are limitations that should temper any clinical interpretation of the results. The validation split is very small, and the test set, while larger, is drawn from a single public source and a comparatively narrow (pediatric) population, so accuracy figures should not be extrapolated to a general adult clinical population without further validation. Quantization to a fixed-point representation is also a potential source of accuracy loss relative to the floating-point host model; the present work reports post-quantization functional accuracy but does not separately quantify this gap. Finally, the Vitis AI development flow and ZCU104 hardware carry a nontrivial upfront cost and toolchain learning curve relative to a CPU-only deployment, a practical consideration for smaller clinical or research settings. None of these limitations are specific to the ZCU104 platform, and the same pipeline is expected to generalize to newer Zynq UltraScale+ and Versal-family SoC-FPGAs as they become available.

VI. CONCLUSION

This paper presented an end-to-end pipeline for pneumonia detection from chest X-ray images that couples a ResNet-50 classifier with FPGA-based inference on a Xilinx ZCU104 board. Using Vitis AI to quantize and compile the trained model for the board's Deep-Learning Processing Unit, the deployed system achieved roughly 28 times the inference throughput and 138 times the energy efficiency of a CPU baseline running the same network, while validation accuracy remained in the 90–92% range. The system was demonstrated through both a browser-based interface and native on-board inference, indicating that SoC-FPGA acceleration is a practical route to low-power, real-time medical image screening at the point of care, with particular relevance to resource-constrained clinical settings. The qualitative comparison in Table 1 underscores that, unlike prior work which reports either diagnostic accuracy or hardware efficiency in isolation, this study reports both for the same deployed system, directly addressing the identified research gap.

VII. FUTURE WORK

- Validate the classifier on larger, multi-institutional, and demographically diverse chest radiograph cohorts.
- Apply quantization-aware training to narrow any accuracy gap introduced by fixed-point deployment.
- Extend the model to multi-class diagnosis, distinguishing bacterial and viral pneumonia and related thoracic conditions.
- Benchmark the ZCU104 deployment against other embedded accelerators, such as edge GPUs and dedicated neural processing units, under matched conditions.
- Investigate structured pruning and lower-bit quantization to further reduce latency and on-chip resource utilization.
- Report detailed DPU resource utilization (LUTs, BRAM, DSP slices) to guide porting to smaller SoC-FPGAs.
- Integrate the system with clinical imaging workflows and pursue the validation studies required for regulatory and clinical use.

REFERENCES

1. P. Rajpurkar et al., "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," arXiv:1711.05225, Nov. 2017.
2. D. S. Kermany et al., "Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.
3. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, 2016, pp. 770–778.
4. N. Ansari, A. R. Faizabadi, S. M. A. Motakabber, and M. I. Ibrahimy, "Effective Pneumonia Detection Using ResNet Based Transfer Learning," *Test Eng. Manage.*, vol. 82, pp. 15146–15153, 2020.
5. L. Racic, T. Popovic, S. Cakic, and S. Sandi, "Pneumonia Detection Using Deep Learning Based on Convolutional Neural Network," in *Proc.*

- 25th Int. Conf. Inf. Technol. (IT), 2021, pp. 1–4.
6. P. Szepesi and L. Szilagy, "Detection of Pneumonia Using Convolutional Neural Networks and Deep Learning," *Biocybern. Biomed. Eng.*, vol. 42, no. 3, pp. 1012–1022, 2022.
 7. S. Shah, H. Mehta, and P. Sonawane, "Pneumonia Detection Using Convolutional Neural Networks," in *Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, 2020, pp. 933–939.
 8. B. Antin, J. Kravitz, and E. Martayan, "Detecting Pneumonia in Chest X-Rays with Supervised Learning," Stanford Univ., CS229 course project report, 2017.
 9. T. V. Huynh, "FPGA-Based Acceleration for Convolutional Neural Networks," Faculty of Electron. and Telecommun. Eng., Univ. of Danang — Univ. of Sci. and Technol., Da Nang, Vietnam, 2016.
 10. F. U. D. Farrukh, T. Xie, C. Zhang, and Z. Wang, "Optimization for Efficient Hardware Implementation of CNN on FPGA," in *Proc. IEEE Int. Conf. Integr. Circuits, Technol. Appl. (ICTA)*, Beijing, China, 2018.
 11. D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015.
 12. Xilinx Inc., Vitis AI User Guide (UG1414), Xilinx/AMD, San Jose, CA, USA, 2023. [Online]. Available: <https://docs.xilinx.com>