

Lightweight Quantized CNN for Real-Time Histopathological Diagnosis on Edge Devices in Digital Pathology

Sohaib R. Awad*

sohaib.awad@uoninevah.edu.iq

Amar I. Daood***

amar.daood@uomosul.edu.iq

Amer T. Ali**

amer.ali@uoninevah.edu.iq

Ahmed M. Alkababji***

ahmedalkababji72@uomosul.edu.iq

*Computer and Information Engineering Department, Ninevah University, Mosul, Iraq

**Electronic Engineering Department, Ninevah University, Mosul, Iraq

***Computer Engineering, College of Engineering, University of Mosul, Mosul, Iraq

Received: September 4th, 2025 Received in revised form: November 8th, 2025 Accepted: Dec 28th, 2025

ABSTRACT

There is a goal to develop a fundamental deep learning system for histopathological image categorization, which operates in real-time while keeping a lightweight design to facilitate better cancer diagnostics in mobile health settings with limited resources. Accurate deep learning models are not well-suited for low-power edge devices, as they require high computational power and memory capacity and do not integrate well with user interfaces for real-time diagnostic applications. A CNN with optimization for inference latency reduction and power efficiency was achieved through a design approach that included quantization-aware network pruning to maintain performance and decrease model size. The proposed model implements grouped convolutional processing while decreasing its filter width, together with mixed-precision quantization, which enables INT8 convolutions combined with FP16 batch normalization and FP32 fully connected layer operation through a MATLAB graphical user interface that enables users to select images, along with automatic diagnostic report generation with label watermarking. The experimental results show that the system achieves 99.14% classification precision, with a model dimension of 263 KB and a runtime of 2.63 milliseconds, producing better or equivalent results compared to the lightweight ShuffleNet and MobileNetV2 architectures based on F1-score, throughput, and energy efficiency metrics. This work takes the concept of deep learning closer to practice by introducing an effective end-to-end prototype, but additional cross-dataset validation, on-edge device benchmarking, and interpretability to clinicians will be needed to make it clinically ready.

Keywords: Histopathological image classification, deep learning, lightweight CNN, quantization, real-time diagnosis, edge computing, digital pathology, energy efficiency, medical image analysis, graphical user interface (GUI).

This is an open access article under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://jamh.uomosul.edu.iq/index.php/rengj>

Email: alrafidain_engjournal3@uomosul.edu.iq

1. INTRODUCTION

The accurate diagnosis of cancer depends mainly on histopathological examination because it shows detailed information about cellular features and disease development. High-resolution whole-slide images (WSIs) can be acquired using automated computational techniques enabled by digital pathology technology [1]. Lately, artificial intelligence and machine learning have shown significant results across a range of applications, including pattern classification [2], biometric systems [3], surveillance techniques [4], human activity detection [5], complex object classification [6,7], and medical applications [8]. More specifically, convolutional Neural Networks CNNs from the category of deep learning operate as a primary component that successfully conducts histopathological image analysis and matches

diagnostic results to expert pathologist performance [9, 10]. Analysis of Deep Learning DL models is currently constrained for clinical applications, mainly in areas without sufficient resources. Traditional deep learning models need large memory capacity and powerful computational power; therefore, they cannot function properly on low-power devices like mobile medical equipment and embedded systems [11, 12]. Current resource requirements of high-performance algorithms create obstacles when trying to apply Artificial Intelligence AI in digital pathology practices. The need for energy-efficient DL solutions becomes urgent, as diagnostic accuracy must be maintained alongside installation capability in clinical environments of all types [13, 14].

Scientists have investigated compact CNN architectures because they seek processing efficiency without sacrificing diagnostic performance standards. Spatial pyramid pooling represents a method that lowers the number of parameters through effective feature extraction [15]. Research on the MobileNet architecture provides valuable insights into its performance in classifying pathological images in mobile computing environments. Most current solutions do not integrate into clinical tools as usable platforms, and they fall short of meeting operational needs, such as real-time diagnostic speed, inference time requirements, and report generation [16, 17]. The combination of image data with text annotations as additional modalities proves useful for both model interpretation and general flexibility. The development of Learned Resizing with Efficient Training (LRET) training protocols enables researchers to work with large histology patches while ensuring structural identification, so they can enhance both training rates and prediction precision [18, 19]. Mobile and edge devices need proper power management because they have limited energy and computational resources. In the specific case of AI-powered edge devices with battery power, advanced solutions are required to achieve efficient performance and extended operational life [20]. Current research has produced various lightweight CNNs with optimization strategies. Yet, there remains a major need for an accurate, deployable histopathological AI solution that supports real-time operation via a user-friendly interface, is power-efficient, and has a compact architecture. The majority of historical research focuses on efficient network design and accurate results but fails to consider the practical implementation required for operational diagnostic workflows, especially in low- and mobile-healthcare environments [21, 22].

This study addresses the previously mentioned challenge by developing a CNN model that combines pruning and quantization methods to improve edge-device utilization in histopathological image classification systems. The proposed model delivers efficient performance with reduced implementation requirements, supporting high accuracy. The system integrates with a MATLAB-based GUI for batch image selection, with real-time classification, embedded watermark labeling, and automatic report generation that includes diagnostic summaries and timestamps. The complete system provides a practical AI diagnostic tool suitable for remote clinical applications, telemedicine systems, and mobile pathology

laboratories. The study offers many significant contributions as explained below:

1. Built a lightweight CNN through architectural development, followed by pruning and quantization techniques for efficient operation on edge devices to deliver fast and energy-efficient histopathological classification.

2. The optimized model delivers high classification performance while minimizing inference time and lowering memory usage, together with energy consumption, thus proving operation-ready for mobile and AI edge devices deployment.

3. An integration system incorporating the graphical user interface embeds the model to deliver real-time image assessments with visual feedback, which automatically creates clinical diagnostic reports.

4. The system undergoes a complete experimental assessment alongside contemporary lightweight models through testing of performance indicators, which include speed, processing volume, and size requirements for demonstrating usefulness in constrained resource frameworks.

This research paper presents an organized structure for showing both the proposed system design and evaluation methodology. The next section investigates the recent approaches in histopathological image classification through deep learning, while it analyzes deployment difficulties from practical settings. Section 3 provides detailed explanations of the methodology, including the CNN architecture design process and optimization strategies such as pruning methods and quantization techniques, alongside graphical user interface development. Section 4 presents the experimental framework and comprehensive results, evaluating the system's performance based on accuracy, along with inference latency, throughput, and energy consumption measurements. Section 5 analyzes the experimental findings by contrasting the new approach to lightweight models in edge settings and demonstrating its performance benefits. Section 6 concludes the paper with a summary that includes both contributions and proposed research paths for clinical validation, as well as domain generalization.

2. LITERATURE REVIEW

Deep learning has been established as the fundamental technology that helps automated tissue classification achieve high precision in histopathological image analysis. Weighting plays a crucial role in deep learning with Convolutional Neural Networks (CNNs), which succeed in identifying spatial arrangements and complex morphological features within histological patterns [23]. The initial designs of CNN systems

prioritized accuracy metrics above all else, which resulted in tiny models that failed to prioritize deployment-related factors like system size and performance speed, along with power requirements. Real-time and embedded hardware systems were unable to use these models due to their high computational requirements [24, 25]. Lightweight CNN architectures have recently gained popularity because of their ability to address this issue. A separable CNN architecture for breast cancer histopathology classification was introduced by Nneji et al. [26]. The use of depthwise separable convolutions enabled them to reduce both parameter count and computational demands, leading to mobile-friendly system capabilities. Despite its performance in minimal resource environments, this model did not support interactive interfaces or real-time clinical procedures. The research conducted by these authors focused on binary classification tasks, which limits their ability to extend their findings to the multi-class categorization needed in this study. The team of Hasan et al. [27] developed a multi-scale CNN, along with XAI explanation tools, for diagnosing lung and colon cancers. The implementation of multi-scale filters improved interpretability by enabling the network to detect diverse morphological features and produce heat and saliency maps. The model reached 4.5 MB, preventing its deployment on edge devices despite lacking quantization capabilities and deployment tools. This method reduces model size while maintaining diagnostic accuracy standards for deployment readiness.

The adoption of attention mechanisms, together with transfer learning methods, has shown successful results in improving both classification performance and interpretability. The research by Zubair et al. [28] developed an interpretable model by applying transfer learning with multi-channel attention modules to classify gastric cancer. The method's interpretation capabilities, combined with its performance, were robust while operating on large-scale parameter models that limited its practicality for resource-constrained systems. The authors of [29] proposed a hybrid system that integrates CNN features with transformer capabilities to extract local features and capture global context. The combination of transformer layers produces good explainability and classification outcomes, but it also introduces substantial inference latency and computational complexity, making it suitable only for server environments. Parallel to their predecessors, EfficientNet-based models have established themselves as a leading choice for achieving high performance through sensible complexity. Sengodan [30] developed a CBAM-integrated

EfficientNetV2 model for breast cancer histopathology classification, leveraging both spatial and channel attention to identify discriminative regions. This model delivered better accuracy and improved attention capabilities but required significant memory resources and lacked functional deployment capabilities for specific platforms. Lilhore et al. [31] developed a breast cancer detection system by hybridizing EfficientNet-B0 with Bi-LSTM layers, achieving 99.2% accuracy. The introduction of recurrent components introduces temporal dependencies and extends processing time, thereby reducing the feasibility of real-time execution.

The deployment of systems that operate at the edge computing level has recently become more prevalent. Ochoa-Ornelas et al. [32] developed a dual MobileNetV2 and EfficientNetB3 network using grey wolf optimization for colon and lung cancer identification tasks. The model reduced training time while improving generalization, with mobile deployments in mind. The evaluation focused on how well the model could classify data, without examining power output or latency in depth. Abioye et al. [33] studied EfficientNetV2 models on benign breast cancer datasets, where EfficientNetV2 achieved the highest accuracy level of 97%. The research omitted information about both model dimension and resource management limitations, even though these factors were not emphasized as main priorities.

Advancements in model interpretability and performance enhancements have exposed an ongoing problem in deploying real-time edge systems that integrate computer interfaces. Previous research on pathological diagnosis and visual explanation primarily targets either model accuracy or visual explanation, without covering complete system deployment from inference through the entire user pathway. This research establishes a connection by developing a quantized CNN model that achieves high accuracy while maintaining a lightweight size for real-time MATLAB GUI deployment. The system enables batch image categorization, watermark insertion, and automatic report generation, thereby increasing its usability in mobile diagnostic applications, field tests, and basic healthcare facilities. In short, the previous solutions focus either on the precision of the heavy backbones or on mobile-friendliness, lacking a full workflow that caters to the user community. We hope to fill this gap with our study, which provides a compact, quantized model and a workable interface prototype, and presents unified accuracy-latency-size-energy indicators to support informed edge deployment decisions.

3. METHODS AND MATERIALS

The proposed research establishes a comprehensive approach to building an efficient, robust deep learning system for classifying histopathological images of lung and colon diseases. The proposed system balances accuracy performance with mobile and edge device capabilities because it supports deployment in clinical medical environments. The complete system development process starts with data preparation and follows with neural network design before proceeding to model optimization and quantization, then conducts pre-trained model testing, followed by GUI deployment. The sequential stages of the proposed methodology lead to GUI-based deployment starting from image acquisition, as shown in Figure 1.

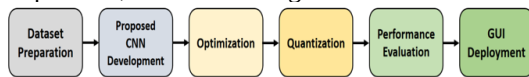


Fig. 1 Proposed Methodology Workflow.

3.1. Dataset Preparation and Preprocessing

A total of five balanced classes comprise the LC25000 database for this study, which contains histopathological images from Colon Adenocarcinoma, Normal Colon Tissue, Lung Adenocarcinoma, Lung Benign Tissue, and Lung Squamous Cell Carcinoma. Classes are distributed evenly throughout the dataset to prevent class imbalance. The research divided its image dataset using the standard 80-20 split, assigning 80% to training and 20% to testing. The partition strategy created two sets with balanced class samples. Table 1 presents an overview of the dataset distribution.

The applied preprocessing procedures improved training efficiency and model generalization. The images in both models were reconstructed to match specific dimensions: 128x128x3 pixels for the CNN and 96x96x3 pixels for the optimized version, while color conversion converted gray images to RGB for network input compatibility. Data augmentation using random transformations, including scaling, rotational techniques, and flipping, was used to obtain more training data as a strategy to prevent overfitting. To facilitate reproducibility, all baselines and the proposed models were trained and assessed using the same LC25000 split protocol and aligned

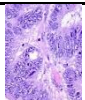
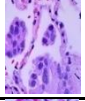
augmentation pipeline. We kept the default values of canonical input sizes because that is how they are used. To minimize the effects of input scaling on accuracy and runtime, we will report harmonized input resolutions for all models in future work.

3.2. Proposed CNN Architecture

The system's main component features an architecture specifically designed to achieve both high classification accuracy and operational efficiency. The proposed 19-layer CNN architecture consists of four blocks, each with initialization, batch normalization, ReLU activation, and max pooling for spatial downscaling. After passing through the fully connected layer, the system utilizes a softmax layer with a classification output. The model requires a standardized 128x128x3 RGB format for its input. A graphical illustration of the CNN structure and its layer connections appears in Figure 2.

The network design provides a deep feature-processing pipeline that operates at high speed in real-time applications. The construction summary of model parameters and input specifications in Table 2 presents the network architecture by listing the layers in order alongside their corresponding feature map dimensions.

Table 1: Dataset Summary and Class Distribution.

Class name	Abbreviation	Number of images	Training	Testing	Sample
Colon Adenocarcinoma	CA	5,000	4000	1000	
Normal Colon Tissue	CBT	5,000	4000	1000	
Lung Adenocarcinoma	LA	5,000	4000	1000	
Lung Benign Tissue	LBT	5,000	4000	1000	
Lung Squamous Cell Carcinoma	LSCC	5,000	4000	1000	

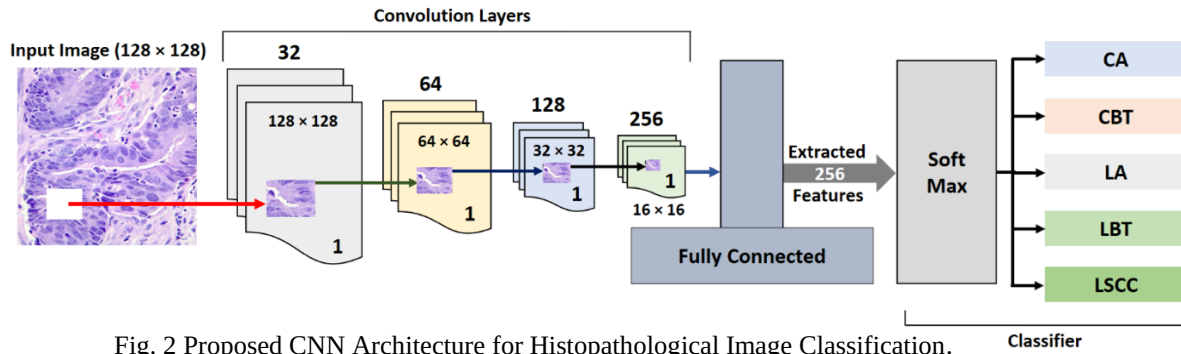


Fig. 2 Proposed CNN Architecture for Histopathological Image Classification.

3.3. Model Optimization

New architectural changes to the CNN model reduced deployment costs, memory usage, and inference latency requirements. Modernizing the model involved multiple adjustments, such as shrinking input pictures to 96×96 dimensions and splitting convolution layers in half, then implementing grouped convolutions. Reducing the size of the model became possible while the model maintained its performance qualities. Table 3 shows a side-by-side comparison of the original model architecture design with its optimized version. The optimization phase had a latency-first goal and did not compromise accuracy by halving channel widths, using grouped convolutions, and reducing image input resolution; structured pruning then took place. The changes reduce arithmetic intensity and memory traffic, the two prevailing factors in edge inference, thereby enabling effective mixed-precision deployment.

3.4. Quantization Strategy

The model needed to be more compact for mobile and edge device deployment, so a mixed-precision quantization method was used to reduce its size. The algorithm applied mixed-precision quantization, with integer 8-bit INT8 convolutional layers for speed and memory efficiency, and floating-point 16-bit FP16 batch normalization layers for intermediate-precision stability. In contrast, the fully connected output layer maintained 32-bit floating-point (FP32) precision. Figure 3 illustrates the quantized mixed precision structure.

The precision decisions established a balance that minimized the model size by more than 60% while maintaining accuracy. The quantization process and the precision distribution hierarchy throughout the model framework are shown in Table 4. The developed model is ready for deployment on mobile systems and edge devices because of its quantized format.

Table 2: The Proposed CNN Architecture.

No.	Layer Name	Type	Output Size
1	Input	Image Input Layer	$(128 \times 128 \times 3)$
2	Conv1	2-D Convolution (3×3)	$(128 \times 128 \times 32)$
3	BN1	Batch Normalization	$(128 \times 128 \times 32)$
4	Relu1	ReLU	$(128 \times 128 \times 32)$
5	Pool1	2×2 Max Pooling	$(64 \times 64 \times 32)$
6	Conv2	2-D Convolution (3×3)	$(64 \times 64 \times 64)$
7	BN2	Batch Normalization	$(64 \times 64 \times 64)$
8	Relu2	ReLU	$(64 \times 64 \times 64)$
9	Pool2	2×2 Max Pooling	$(32 \times 32 \times 64)$
10	Conv3	2-D Convolution (3×3)	$(32 \times 32 \times 128)$
11	BN3	Batch Normalization	$(32 \times 32 \times 128)$
12	Relu3	ReLU	$(32 \times 32 \times 128)$
13	Pool3	2×2 Max Pooling	$(16 \times 16 \times 128)$
14	Conv4	2-D Convolution (3×3)	$(16 \times 16 \times 256)$
15	BN4	Batch Normalization	$(16 \times 16 \times 256)$
16	Relu4	ReLU	$(16 \times 16 \times 256)$
17	Fc	Fully Connected	(1×5)
18	Softmax	Softmax Layer	(1×5)
19	Class Output	Classification Output	(1×5)

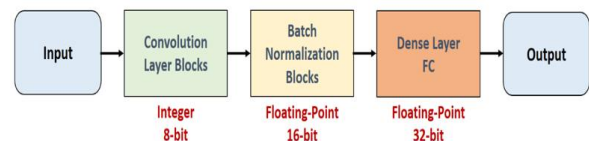


Fig. 3 Mixed-Precision Quantization Flow of the Optimized CNN Model.

This precision allocation (INT8 convolution, FP16 BN, FP32 classifier) was chosen empirically as a compromise between the stability of the activation range and the quantization-induced accuracy loss at a very low cost in terms of compute and memory.

Table 3: Architectural Comparison Between Original and Optimized CNN Models.

Feature	Original CNN	Optimized CNN
Input image size	128×128×3	96×96×3
Total learnable parameters	717K	190K
Number of layers	19	19
Convolutional layers	4	4
Filter sizes	conv1: 3×3, 32 filters	conv1: 3×3, 16 filters
	conv2: 3×3, 64 filters	conv2: 3×3, 32 filters
	conv3: 3×3, 128 filters	conv3: 3×3, 64 filters
	conv4: 3×3, 256 filters	conv4: 3×3, 128 filters
FC learnable parameters	65536 → 5 outputs	18432 → 5 outputs
Batch normalization	Used after each convolution	Used after each convolution
Max pooling	Used after 3 conv blocks	Used after 3 conv blocks
Activation function	ReLU	ReLU
Total depth	Deep (more filters, wider FC)	Lightweight (smaller filters, FC)
Parameter reduction strategy	-	Half filters & input size reduced

Table 4: Mixed-Precision Quantization Strategy for Optimized CNN Deployment.

Component	Quantization precision	Rationale
Convolutional Layers	INT8	To reduce latency, memory footprint, and improve throughput on hardware accelerators.
Batch normalization layers	FP16	To maintain training distribution, scaling precision with minimal cost.
Fully connected Layer	FP32	Ensures stable output logits and reduces quantization-induced accuracy drop.

3.5. Performance Evaluation and Benchmarking

The assessment of the proposed quantized CNN used benchmark performance testing against four widely used lightweight pre-trained models: SqueezeNet, ShuffleNet, MobileNetV2, and EfficientNet. Table 5 summarizes the specifications of pre-trained networks.

Table 5: Specifications Summary of Pre-trained Models

Model	Total layers	Connections	Depth	Learnable parameters	Input image size
SqueezeNet	69	76	18	1.2	227×227×3
ShuffleNet	172	187	50	1.4	224×224×3
MobileNetV2	154	163	53	3.4	224×224×3
EfficientNet	290	363	82	5.3	224×224×3

The evaluation used combined metrics from classification and hardware performance for evaluating the system through accuracy measures with precision, recall, and F1-score, together with inference time (latency), throughput, model size, memory usage, computational workload (GOP), processing speed (GOPS), and estimated energy consumption values. The assessment was conducted in controlled settings using an NVIDIA RTX 3060 GPU with 6GB VRAM to run models at a batch size of 32. The training process used an initial learning rate of 0.001, which decreased by 0.5 every 10 epochs. The training period lasted 30 epochs, and validation occurred every 50 iterations. The evaluation metrics with their definitions used for comparative analysis are presented in Table 6.

Table 6: Evaluation Metrics for Model Performance and Efficiency Benchmarking

Metric	Definition
Accuracy (%)	Percentage of correct predictions over all test samples.
F1-Score	Harmonic mean of precision and recall for balanced evaluation.
Latency (ms)	Time taken to process a single image during inference.
Throughput (img/s)	Number of images processed per second during inference.
Model Size	Total size of the saved model file in KB or MB.
Memory Usage (MB)	Runtime memory consumption of the model during inference.
GOP	Giga operations per inference represent the computation load.
GOPS	Giga Operations Per Second measures the model's processing speed.
Energy (J/image)	Estimated energy consumption per image processed, in joules.

Standard mathematical formulas ensure consistent performance evaluation by calculating each metric listed in Table 6. The mathematical formulas used for benchmarking classification accuracy and efficiency, as well as hardware performance, are crucial in performance evaluation [34, 35].

$$\text{Accuracy \%} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (1)$$

$$\text{Precision \%} = \frac{TP}{TP + FP} \times 100 \quad (2)$$

$$\text{Recall \%} = \frac{TP}{TP + FN} \times 100 \quad (3)$$

$$\text{F1 - Score \%} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100 \quad (4)$$

$$\text{Latency} = \frac{\text{Total Inference Time}}{\text{Number of Images}} \quad (5)$$

$$\text{Throughput} = \frac{\text{Number of Images}}{\text{Total Inference Time (Sec.)}} \quad (6)$$

$$\text{GOP} = \frac{\Sigma \text{FLOPs per layer}}{10^9} \quad (7)$$

$$\text{FLOPs} = 2 \times C_{in} \times K_H \times K_W \times H_{out} \times W_{out} \times C_{out} \quad (8)$$

$$\text{GOPS} = \frac{\text{GOP (Work Load)}}{\text{Latency (Second)}} \quad (9)$$

$$\text{Energy per Image} = \text{Power (W)} \times \text{Latency (Sec.)} \quad (10)$$

True Positives get designated as TP, while True Negatives are TN, and False Positives are FP, together with False Negatives being FN. Programs use TensorRT and PyTorch profilers to measure throughput and inference latency as part of their inference profiling workflow to determine precise runtime performance. In convolutional operations, the input channels are denoted by C_{in} , while the kernel has heights K_H and widths K_W , along with output feature map heights H_{out} and widths W_{out} , and output channels C_{out} . Software-based tools enable power consumption estimation when direct hardware measurements are not available. The approximate power usage tools measure runtime statistics, including GPU utilization rates and operation counts, and execution time information, to estimate power consumption and enable researchers to study energy efficiency across different model configurations without specialized hardware sensors [36, 37].

3.6. User Interface and System

Deployment

A MATLAB-based graphical user interface (GUI) enabled clinicians and researchers to use the final quantized CNN model by providing its functionality. Users can use this GUI to select up to 5 histopathological images and display both input and output predictions, along with a visual watermark for the prediction class. A depiction of the GUI structure is shown in Figure 4.

Histopathology images that users select appear in the upper portion of the GUI, and the lower section presents both classified outputs with predicted labels for matching images. The system uses a drop-down menu to specify the number of image classifications. Users can begin classification via "Classify Images" and save reports via "Save Report". The classification engine automatically saves images containing their predicted labels, along with their original names, in the designated output directory. Healthcare

professionals benefit from this interface because it offers an easy-to-use workflow, efficient report generation, and streamlined controls. The user-friendly interface supports real-time functionality for documentation, along with analysis capabilities that make the system effective in clinical diagnostic and research environments.

4. EXPERIMENTAL RESULTS

The experimental findings for the proposed deep learning pipeline for histopathological image classification are presented in this section. The evaluation conducts training sequences by following the methodology flow, starting with training original and optimized CNN architectures on the LC25000 dataset. The optimized CNN underwent benchmarking tests against the commonly used lightweight pre-trained MobileNetV2, ShuffleNet, EfficientNet, and SqueezeNet deep learning models after the application of model optimization and quantization techniques. A dual approach, including classification metrics and hardware performance indicators (accuracy, precision, recall, F1-score, inference latency, throughput, memory usage, model size, GOP, GOPS, and energy consumption), was used for performance evaluation. The experiments were run exclusively on an NVIDIA RTX 3060 GPU with 6GB of VRAM, using batch processing with 32 samples, and trained for up to 30 epochs. The research group integrated the developed model into a clinical and research GUI built using MATLAB to demonstrate real-time operation for use in both research laboratories and medical facilities. All tested metrics support the model's effective performance, operational effectiveness, and deployability on limited system resources.

4.1. Pre-trained Models Evaluation and Classification Results

A performance benchmark for the proposed CNN was created by reviewing lightweight pre-trained convolutional neural networks, including EfficientNet-b0, MobileNet-v2, ShuffleNet, and SqueezeNet. These models were selected for their widespread adoption as AI solutions for mobile and embedded devices. The assessment metrics evaluated accuracy rates and F1-scores, along with model dimensions and parameter counts, and screening duration. A summary of the comparative analysis appears in Table 7. Among the reviewed models, ShuffleNet provided the optimal testing accuracy (99.84%) and F1-score (99.86%) utilizing a digestible model size (3.24 MB). MobileNet-v2 demonstrated superior performance, achieving 99.80% accuracy and 99.69% F1-score with an 8.2 MB model that provided faster inference than EfficientNet-b0.

Understandably, SqueezeNet consumes 4.43 MB of memory at the cost of poor performance, achieving 96.00% accuracy and 95.8% F1-score, making it impractical for detailed medical assessments. ShuffleNet served as the pre-trained reference model because its performance-efficiency balance proved optimal for the latter benchmarking against the proposed CNN and its

improvements. The model's exceptional accuracy with low memory requirements qualifies ShuffleNet as an ideal choice for edge AI applications and as a meaningful evaluation of this research development. Figure 5 shows the confusion matrices CM for the pre-trained models of the histopathological image testing set samples.

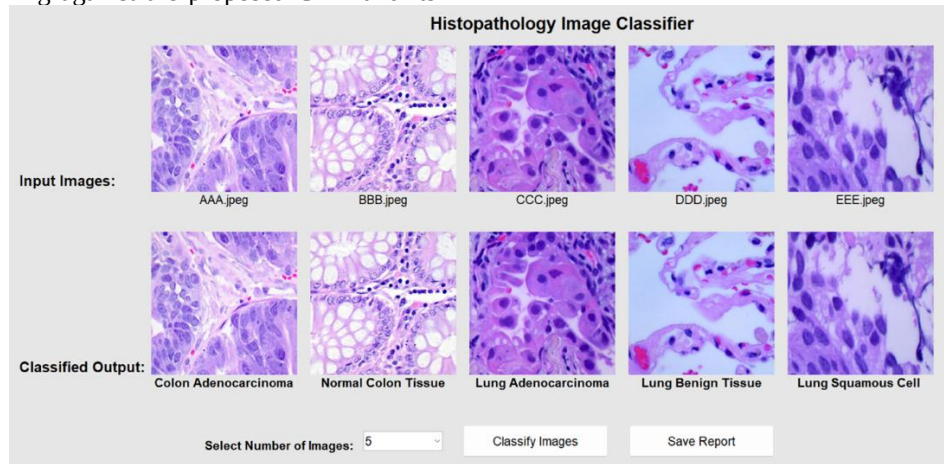


Fig. 4 MATLAB-Based GUI for Histopathology Image Classification.

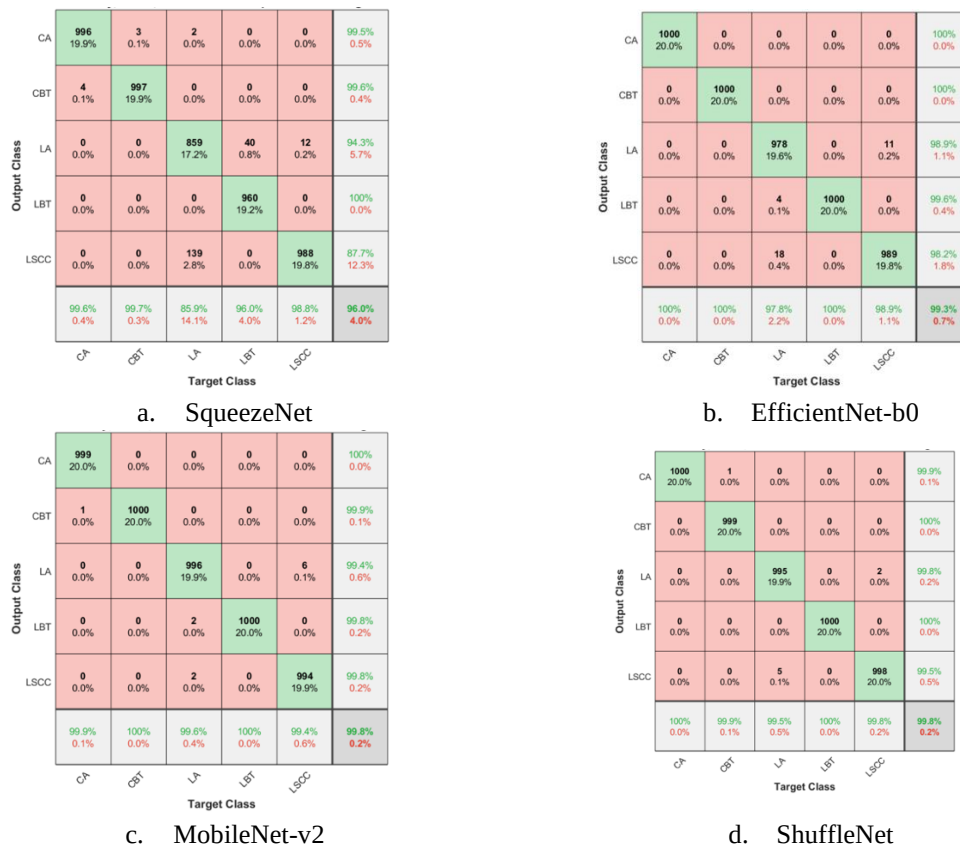


Fig. 5. CMs for Pre-trained Models on Histopathological Image Classification.

Table 7: Pre-Trained Model Performance Results

Pre-trained Model	Model size (MB)	Learnable parameters		Testing accuracy (%)	F1-score (%)
		Total	Fully connected		
EfficientNet-b0	14.5	4M	6.41K	99.34	99.35
MobileNet-v2	8.2	2.2M	6.41K	99.8	99.69
ShuffleNet	3.24	864.4K	2.73K	99.84	99.86
SqueezeNet	4.43	1.2M	5K	96.00	95.8

The performance data for five histopathological groups (Colon Adenocarcinoma (CA), Normal Colon Tissue (CBT), Lung Adenocarcinoma (LA), Lung Benign Tissue (LBT), and Lung Squamous Cell Carcinoma (LSCC)) are shown in each matrix. It shows correct and incorrect patient identifications, along with their respective percentages. The evaluation results demonstrate that ShuffleNet, along with MobileNet-v2, is a leader in medical AI system applications due to its accuracy and class consistency. This evaluation confirms that ShuffleNet is the most suitable option for pre-trained systems powering lightweight medical artificial intelligence solutions in histopathological image classification. ShuffleNet strikes the optimal trade-off between efficiency and accuracy, demonstrating precise performance in the confusion matrix and a modest model design while achieving superior classification results. The ShuffleNet architecture enables successful deployment on resource-limited devices for real-time diagnostic tasks. The baselines were measured with their standard input sizes to reflect the commonly used deployment practices for each architecture. Even though different input resolutions may affect accuracy and runtime, the same dataset split and preprocessing policy were used across all the models. Ablations to be done in the future will coordinate the input sizes across models to enable resolution-controlled comparisons.

4.2. Proposed Model Evaluation and Classification Results

The performance evaluation of the proposed CNN architecture is based on three successive designs: original, optimized, and quantized models. The research uses a quantized ShuffleNet as a benchmarking model to evaluate performance, having previously identified it as the most effective lightweight pre-trained model. Table 8 presents a detailed breakdown of all performance indicators, including accuracy, F1-score, model size, and number of parameters.

A testing accuracy of 98.86% and F1-score of 98.86% served as the baseline performance metric for the original CNN design. The model operated at 2.6 MB while using 717K learnable parameters. Despite its accuracy, the network proved challenging to implement on edge devices due to its large parameter count and model size. To address this, the authors employed structured pruning techniques and grouped convolutions to develop the optimized CNN. The optimized CNN model achieved 99.16% test accuracy, 99.21% F1-score, 99.26% precision, and 99.16% recall, as shown in Figure 6 below. The compact model size amounted to 708 KB. Architectural optimization specifically targeted at performance goals leads to simultaneous improvements in both processing speed and system performance quality.

Additionally, to reduce the computational load of operations, the CNN model employed mixed-precision quantization, applying INT8 to convolutional layers and FP16 to fully connected layers. Due to compression, the model reached 263.40 KB, representing a 62.62% decrease from the original optimized model size. Despite the compression, the model maintained a high accuracy of 99.14%, coupled with a 99.14% F1-score, as shown in Figure 6, and both precision and recall of 99.14%, demonstrating its efficiency, robustness, and potential for real-time embedded systems. The research team included a quantized ShuffleNet in their evaluation as a point of reference. The quantized proposed CNN achieved slightly better accuracy (99.88%), but its model size of 3315 KB exceeded the practical threshold for a low-memory setup compared to the proposed CNN. Despite its competitors, this model establishes the highest accuracy level in this assessment process.

Figure 6 shows the classification outcomes for four assessed models: the proposed CNN, an optimized CNN, a quantized CNN, and a quantized ShuffleNet, on five-class histopathological image data.

Table 8 : Comparative Performance of Proposed CNN Variants and Quantized ShuffleNet

Model	Model size (MB)	Learnable parameters		Testing accuracy (%)	F1-score (%)
		Total	Fully connected		
Proposed CNN	2.6	717K	65.54k	98.86	98.86
Optimized CNN	0.708	189.6K	18.43k	99.16	99.21
Quantized CNN	0.263	189.6K	18.43k	99.14	99.14
Quantized ShuffleNet	3.315	864.4K	2.73K	99.88	99.92

The matrices supplement table-reported quality measurements with comprehensive case-by-case results and pattern analysis of misclassifications. Each confusion matrix exhibits strong diagonal dominance, indicating accurate class predictions with low misinterpretation. The model variations of the proposed CNN delivered top-class precision and recall metrics after size reduction to 708 KB for the optimized version and 263 KB for the quantized variant. The obtained results match previously reported statistics for F1-

scores and testing accuracy evaluations. The quantized ShuffleNet demonstrated its reliability as a benchmark, achieving the best score combination of accuracy and F1-score at 3.315 MB. The confusion matrices indicate that the proposed CNN variants perform effectively under compression, making them suitable for medical diagnosis applications, especially for real-time histopathological image classification on limited hardware resources.

CA	995 19.9%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
CBT	5 0.1%	998 20.0%	0 0.0%	0 0.0%	0 0.0%	99.5% 0.5%
LA	0 0.0%	0 0.0%	971 19.4%	0 0.0%	21 0.4%	97.9% 2.1%
LBT	0 0.0%	0 0.0%	0 0.0%	1000 20.0%	0 0.0%	100% 0.0%
LSCC	0 0.0%	0 0.0%	29 0.6%	0 0.0%	979 19.6%	97.1% 2.9%
	99.5% 0.5%	99.8% 0.2%	97.1% 2.9%	100% 0.0%	97.9% 2.1%	98.9% 1.1%
	CA	CBT	LA	LBT	LSCC	

a. Proposed CNN

CA	996 19.9%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
CBT	4 0.1%	998 20.0%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
LA	0 0.0%	0 0.0%	962 19.6%	0 0.0%	18 0.4%	98.2% 1.8%
LBT	0 0.0%	0 0.0%	1 0.0%	1000 20.0%	0 0.0%	99.9% 0.1%
LSCC	0 0.0%	0 0.0%	17 0.3%	0 0.0%	962 19.6%	98.3% 1.7%
	99.6% 0.4%	99.8% 0.2%	98.2% 1.8%	100% 0.0%	98.2% 1.8%	99.2% 0.8%
	CA	CBT	LA	LBT	LSCC	

b. Optimized CNN

CA	996 19.9%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
CBT	4 0.1%	998 20.0%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
LA	0 0.0%	0 0.0%	962 19.6%	0 0.0%	19 0.4%	98.1% 1.9%
LBT	0 0.0%	0 0.0%	1 0.0%	1000 20.0%	0 0.0%	99.9% 0.1%
LSCC	0 0.0%	0 0.0%	17 0.3%	0 0.0%	961 19.6%	98.3% 1.7%
	99.6% 0.4%	99.8% 0.2%	98.2% 1.8%	100% 0.0%	98.1% 1.9%	99.1% 0.9%
	CA	CBT	LA	LBT	LSCC	

c. Quantized CNN

CA	1000 20.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
CBT	0 0.0%	1000 20.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
LA	0 0.0%	0 0.0%	996 19.9%	0 0.0%	2 0.0%	99.8% 0.2%
LBT	0 0.0%	0 0.0%	0 0.0%	1000 20.0%	0 0.0%	100% 0.0%
LSCC	0 0.0%	0 0.0%	4 0.1%	0 0.0%	998 20.0%	99.6% 0.4%
	100% 0.0%	100% 0.0%	99.6% 0.4%	100% 0.0%	99.8% 0.2%	99.9% 0.1%
	CA	CBT	LA	LBT	LSCC	

d. Quantized ShuffleNet

Fig. 6 CMs of Proposed CNN Variants and Quantized ShuffleNet.

4.3. Computational Efficiency and Latency

This part analyzes how computational efficiency and inference latency interact with the inference performance of five models, including ShuffleNet and quantized ShuffleNet, alongside the proposed CNN, an optimized CNN, and a quantized CNN. Real-time deployment for medical imaging systems depends heavily on these important aspects when operating on low-power edge hardware. Table 9 presents runtime performance results, including model latency (milliseconds), images processed per second, total model size, memory allocation (kilobytes), and the

total number of trainable parameters. Structural pruning coupled with grouped convolutions made the CNN-optimized model the fastest, achieving 442 images per second and a minimum latency of 2.26 milliseconds. The proposed quantized CNN version operated at 2.63 milliseconds while maintaining efficiency, with a small footprint of 263.4 KB and 0.72 MB of memory consumption, indicating suitability for low-memory devices.

The proposed CNN achieved higher accuracy at 3.59 milliseconds and a lower image processing volume of 279 images per second, thanks to its comprehensive architectural design and extensive parameter utilization. Both

ShuffleNet versions demonstrated longer inference times than the proposed models, with their original and quantized versions taking 21.36 ms and 19.62 ms, respectively, to process, although they achieved strong classification results. The combination of the optimized CNN and the quantized CNN proved superior to other models, delivering lower inference latency and higher throughput, making them compatible with embedded platforms operating at real-time speeds.

Figure 7 illustrates the computational efficiency and latency results for the tested models. Figure 7 displays the throughput and model size values for all five examined models to understand the results better. The visual display depicts trade-offs between performance and efficiency, highlighting the optimized, quantized CNNs as the optimal choice for the edge environment. The examination was conducted using an NVIDIA RTX 3060 GPU with 6GB VRAM under managed test settings. It measured all models with a batch size of 32 to achieve stable runtime execution of the hardware-constrained resources. All runtime measurements were performed in a controlled desktop-GPU environment to estimate relative behavior. Device-level measurements on embedded/edge hardware are planned for subsequent work.

Table 9: Inference Efficiency and Latency Comparison

Model	Latency (ms)	Throughput (img/s)	Learnable Parameters	Model Size (KB)	Memory Usage (MB)
ShuffleNet Original	21.363	47	837.6K	3322	3.1952
ShuffleNet Quantized	19.622	51	837.6K	3315.1	3.1952
Proposed CNN	3.5897	279	716.1K	2639.1	2.7317
Optimized CNN	2.2646	442	189.6K	704.68	0.72329
Quantized CNN	2.6333	380	189.6K	263.4	0.72329

4.4. Complexity Analysis and Energy Estimation

An analysis of computational complexity and energy efficiency for ShuffleNet, quantized ShuffleNet, the proposed CNN, the optimized CNN, and the quantized CNN models will determine their effectiveness for real-time and resource-constrained implementations. The analysis examines four measurement parameters: Giga Operations per Second (GOPS), energy consumption per image (mJ), power usage (mW), and energy requirement for 1,000 inferences (mWs). System-level performance details are presented in Table 10, which shows the

performance changes resulting from architectural and quantization-based strategies. The Proposed CNN achieved the maximum complexity of 0.39898 GOPS due to its complex design with deeper layers and extensive computation requirements. The high complexity of this system consumed 1.289 mJ per image and required 1289 mWs to perform 1K inferences, limiting its applicability in embedded or mobile environments despite its high measurement accuracy.

The optimized CNN reduced its computational complexity to 0.16745 GOPS via grouped convolutions and filter reduction, resulting in a 73% decrease in energy usage compared to the original CNN. This led to a total energy consumption rate of 0.34129 mJ/image. The quantized CNN algorithm achieved a complexity level of 0.144 GOPS without affecting its low energy consumption requirements. Different subtasks of classification performance remain unaffected, as the combination of pruning and quantization produces improved efficiency. The quantized ShuffleNet achieved superior accuracy, but it consumed high power (1.5077 mJ/image) because its architecture had limited optimization potential. In a manner similar to the standard ShuffleNet, both models exhibited the same power consumption characteristics, which, together, reduced their suitability for ultra-low-power use compared to the new CNN variants.

Figure 8 provides a graphical representation of the relationship between computational complexity and energy efficiency. Figure 8a displays GOPS measurements among all chosen networks, indicating their required computational operations during inference. The energy consumption graphs shown in Figure 8b directly display the runtime performance of each model per image.

Energy usage decreases significantly when complexity is reduced in specific ways, especially in optimized and quantized CNNs. The optimized CNN and quantized CNN achieve optimal performance, balancing complexity requirements and power needs. The targeted compression methodologies demonstrate that they reduce model size while simultaneously minimizing energy usage, making these versions appropriate for real-time diagnostic systems operating on edge computing resources. The quantized CNN achieves the smallest model size of 263 KB, with 99.14% accuracy and F1-scores, along with low-latency operation and enhanced throughput. This section highlights its advantages by utilizing 0.144 GOPS of computational capacity and 0.34129 mJ/image of energy consumption. The operation counts and runtime statistics of the profilers are used to derive energy values instead

of hardware power sensors; thus, we view them as relative comparative data rather than absolute clinical power budgets. The quantized model outperforms pre-trained models and the original CNN across core metrics, including accuracy, speed, memory consumption, and power usage, making it an outstanding choice for portable, scalable histopathological AI systems.

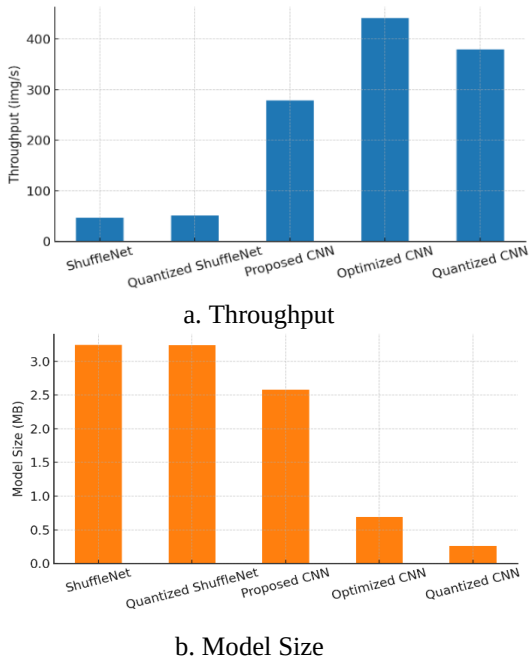


Fig. 7 Throughput and Model Size Comparison Across Models

Table 10. Computational Complexity and Energy Consumption Comparison

Model	GOPS	Energy (mJ/image)	Power (mW)	Power for 1K Inferences (mWs)
ShuffleNet Original	0.078415	1.5077	70.574	1507.7
Quantized ShuffleNet	0.085375	1.5077	76.837	1507.7
Proposed CNN	0.39898	1.289	359.08	1289
Optimized CNN	0.16745	0.34129	150.71	341.29
Quantized CNN	0.144	0.34129	129.60	341.29

a. Computational Complexity

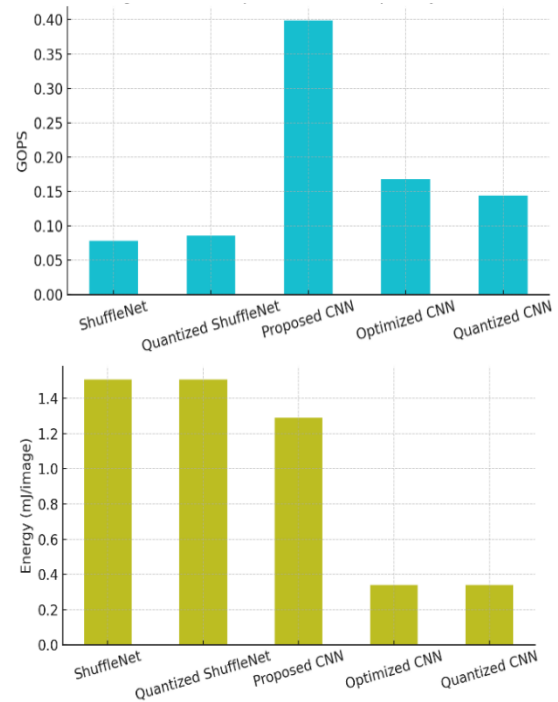


Fig. 8 Computational Load and Power Efficiency

4.5. System Integration and User Interface Demonstration

The quantized CNN model was chosen for system integration after comprehensive testing because it demonstrated exceptional performance, with accurate classification and fast operation, along with low memory requirements and minimal power consumption. A customized graphical user interface in MATLAB was developed, allowing end users to perform histopathological image classification and generate automated reports. Users benefit from the friendly GUI in clinical histopathological image classification, which enables smooth image selection, effective classification processes, clear result presentation, and documentation capabilities. The MATLAB GUI is a research prototype to demonstrate the integration of the workflow and to support wider portability and integration with clinical IT. We will have a lightweight, cross-platform front end (e.g., Python/Streamlit or a web client), but we will consider data security, auditability, and maintainability. Users can run the entire classification process through the GUI, from loading input images to producing final reports containing the results. The GUI layout appears in Figure 4, which was presented in the methods and materials section. Table 11 presents a detailed breakdown of GUI elements together with their operational functions.

The automated output report contains timestamps and patient information at the start of the document, which is followed by a sorted list of images and their predicted results. The illustration below, in Figure 9, shows an automatic output report.

Table 11. GUI Features and Functional Description

Component	Description
Input Image Panel	Displays selected histopathology images from the user's local directory.
Output Classification Panel	Shows classified results below each image, with predicted class labels displayed.
Embedded Watermark Label	The classification label is watermarked directly onto each output image.
Image Selector Dropdown	Allows the user to select the number of images (e.g., 1 to 5) for classification.
Classify Images Button	Executes the quantized CNN model on selected images and displays predictions.
Save Report Button	Generates and saves a structured report summarizing the classification results.
Report Format	Includes date/time, patient name, and file-wise classification entries.

The user interface, with its optimized, quantized model integration, enables real-time,

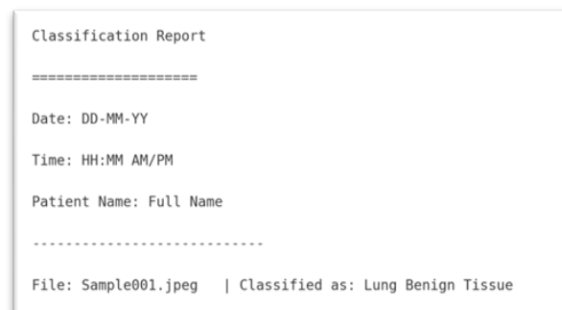


Fig. 9 Clinical Report Format Generated by GUI System

along with delayed response times. The quantized CNN achieved 99.14% accuracy with a 263 KB model size, consuming 0.34129 mJ/image and taking 2.63 ms to process. The model has ideal characteristics for low-power device applications, particularly for mobile or remote healthcare system deployments.

Healthcare professionals achieved smooth user control through a MATLAB-based custom GUI where the model was successfully integrated. The system provides functions to upload images, present classified outputs with watermarked labels, and generate reports containing diagnostic reports, timestamps, and patient information. The completed end-to-end

lightweight image classification, which is suitable for edge computing, telemedicine, and diagnostic tools in digital pathology. The small size of the model enables operation across hardware with minimal resources, with quick processing times and low power usage, ensuring uninterrupted medical practice. The system features an intuitive, user-friendly graphical user interface that enables medical staff and technicians to use it successfully despite their limited machine learning expertise. The solution stands out as a valuable diagnostic workflow improvement tool for medical facilities because it unifies sophisticated model development with practical system implementation capabilities.

5. DISCUSSION

The research delivers an efficient deep learning system framework for histopathological image analysis that satisfies the real-time needs of medical edge-based applications. Three model development phases led to the design of an original baseline CNN, an optimized version with grouped convolutions, and a quantized model. During the evaluation process, the framework demonstrated progressive improvements at each step while maintaining classification precision, especially in power usage and size reduction,

system combines fast classification technology with time-sensitive reporting functions to create a usage-ready clinical platform. Our research evaluation compared our system against modern state-of-the-art approaches with lightweight architectures or ready for deployment. A list containing vital information about external models appears in Table 12.

The model accuracy of 99.20% reported by Hasan et al. is accompanied by a large model architecture (~4.5 MB), making it unsuitable for power-efficient applications [27]. Nneji et al. developed a separable CNN for mobile deployment but achieved 97.23% accuracy, with no integration for GUI or power-efficiency optimization [26]. Zubair et al. used attention mechanisms for interpretability, while Djoumessi et al. employed transformers for interpretability [28, 29]. The high complexity of these cloud-based diagnosis models, along with their incompatibility with edge systems, reduces their practical value in real-time medical environments characterized by constrained hardware resources.

The quantized CNN presented in this paper provides a trade-off between accuracy, low latency, small model size, and simple integration, demonstrating its potential for edge-based digital pathology rather than clinical-level readiness. Our prototype, developed using MATLAB GUI,

presents an end-to-end workflow for selecting batches, performing inference, generating labeled output, and reporting, and its small footprint and quick runtime imply it can run on embedded targets. Simultaneously, we clearly identify the most relevant steps toward a reliable translation (i) cross-dataset validation outside of LC25000 to test how well the models can withstand the clinical variability, (ii) device-level benchmarking of possible edge hardware to report on-device latency, memory, and power, (iii) integration of interpretable visualization (e.g., Grad-CAM) to help gain trust with clinicians and analyze errors, and (iv) a portable and cross-platform front-end implementation needed to fit healthcare IT fully. Despite these limitations, the suggested framework is a feasible and scalable path towards histopathological image classification in resource-constrained environments, and an explicit pathway to high-quality clinical implementation is evident.

6. CONCLUSION

A research work presented a resource-efficient CNN model that included pruning and quantization features to perform histopathological image classification in real-time edge system deployments. Model optimization by resizing inputs, applying grouped convolutions, and using

filter elimination and precision quantification methods yielded a system with 99.14% precision, 2.63 ms processing speed, and 263 KB storage requirements. The model is integrated into a MATLAB-based graphical user interface (GUI) that provides physicians with practical capabilities to input images, perform classification with watermarking, and generate automatic reports. Rigorous experiments showed that the proposed system matched the state-of-the-art performance of lightweight networks such as ShuffleNet and MobileNetV2, especially in edge-based applications. The standard evaluation metrics of accuracy, F1-score, throughput, GOPS, and energy efficiency demonstrate that this model serves resource-limited real-time diagnostic settings. Overall, the proposed framework indicates that small, latency-conscious CNNs can achieve near-state-of-the-art performance with orders of magnitude smaller sizes and inference times, making them feasible for edge-based digital pathology. The work in the future will focus on cross-dataset validation, benchmarking device-level performance on representative embedded hardware, interpretability at the clinician level, and a portable front end for use in the healthcare setting.

Table 12. Comparative Analysis with Recent Lightweight Histopathological Models

Study	Model Type	Accuracy (%)	Model Size (MB)	Latency (ms)	Deployment Capability
Hasan et al., 2024 [27]	Multi-Scale CNN + XAI	99.20	~ 4.5	Not specified	Desktop/ Telemedicine
Nneji et al., 2023 [26]	Separable CNN	97.23	~ 2.1	Not specified	Suitable for Mobile Devices
Zubair et al., 2025 [28]	Attention-based CNN (TL)	98.48	High	Not specified	Cloud-Compatible
Djoumessi et al., 2025 [29]	Hybrid CNN-Transformer	~ 90	High	Not specified	Cloud-Based Interpretation
This Work	Pruned + Quantized Custom CNN	99.14	0.263	2.63	Real-time Edge GUI System

REFERENCES

- [1] J. Yao, H. Zhang, S. Shen, X. Zhu, J. Jonnagaddala, N. Hawkins, and J. Huang, "Whole slide images based cancer survival prediction using attention guided deep multiple instance learning networks," *Med. Image Anal.*, vol. 65, p. 101789, 2020. <https://doi.org/10.1016/j.media.2020.101789>.
- [2] A. Daood, A. Al-Saegh, and A. Mahmood. "Handwriting detection and recognition of Arabic numbers and characters using deep learning methods." *Journal of Engineering Science and Technology* 18, no. 3, p. 1581-1598, 2023. https://jestec.taylors.edu.my/Vol%2018%20Issue%203%20June%202023/18_3_16.pdf.
- [3] 3A. M. Hasan, R. M. Hagem, and A. I. Daood, "Face detection and recognition techniques analysis", *In 2022 International Conference on Computer Science and Software Engineering (CSASE)*, pp. 265-270. IEEE, 2022. <https://ieeexplore.ieee.org/document/9759573>.
- [4] M. Ramzi, and A. Daood. "Smart surveillance system to monitor the committed violations during the pandemic." *International Journal of Computing and Digital System, Vol:11, No:1*: 1415-1426,2021. <https://hdl.handle.net/20.500.14536/4344>.
- [5] B. M. Hasan, A. I. Daood, and W. H. Allen. "Comparison of classifiers for prediction of human actions in a smart home." *In 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pp. 287-288. IEEE, 2018. <https://doi.org/10.1109/IoTDI.2018.00043>.

- [6] A. Daood, E. Ribeiro, and M. Bush. "Pollen Recognition Using Multi-Layer Feature Decomposition." In *FLAIRS*, pp. 26-31. 2016. <https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS16/paper/view/12954>.
- [7] A. Daood, "Food Classification using Deep Learning Techniques." In *2022 2nd International Conference on Advances in Engineering Science and Technology (AEST)*, pp. 727-733. IEEE, 2022. <https://doi.org/10.1109/AEST55805.2022.10412924>.
- [8] A. Fadhil, A. Alkabajji, and A. Daood. "Resilient embedded system for classification respiratory diseases in a real time." *Biomedical Signal Processing and Control*, Vol. 90: 105876, 2024. DOI: 10.1016/j.bspc.2023.105876. <https://doi.org/10.1016/j.bspc.2023.105876>.
- [9] N. Hashimoto, S. Kusakabe, and M. Okabe, "Multi-scale domain-adversarial multiple-instance CNN for cancer subtype classification with unannotated histopathological images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. <https://doi.org/10.1109/CVPR42600.2020.00391>.
- [10] W. Ding, Q. Xu, and L. Yang, "FMDNN: A fuzzy-guided multi-granular deep neural network for histopathological image classification," *IEEE Trans. Fuzzy Syst.*, early access, 2024. <https://doi.org/10.1109/TFUZZ.2024.3410929>.
- [11] L. Qu, S. Guo, and C. Liu, "Towards label-efficient automatic diagnosis and analysis: A comprehensive survey of advanced deep learning-based weakly-supervised, semi-supervised and self-supervised techniques in histopathological image analysis," *Phys. Med. Biol.*, vol. 67, no. 20, p. 20TR01, 2022. <https://iopscience.iop.org/article/10.1088/1361-6560/ac910a>.
- [12] M. Jasim, A. Mohammed, and S. R. Awad, "Efficient gender classifier for Arabic speech using CNN with dimensional reshaping," in *Proc. 2021 7th Int. Conf. Elect., Electron. Inf. Eng. (ICEEIE)*, 2021, pp. 104–109. <https://doi.org/10.1109/ICEEIE52663.2021.9616930>.
- [13] M. A. M. Abdullah, A. A. Mohammed, and S. R. Awad, "RockDNet: Deep learning approach for lithology classification," *Appl. Sci.*, vol. 14, no. 13, p. 5511, 2024. <https://doi.org/10.3390/app14135511>.
- [14] F. S. Alghareb and S. R. Awad, "Secured encoding-based real-time power-saving approach for AIoT applications," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 1, pp. 82–90, 2025. <https://doi.org/10.22266/ijies2025.0229.51>.
- [15] J. Lin, Y. Yu, and X. Zhang, "PDBL: Improving histopathological tissue classification with plug-and-play pyramidal deep-broad learning," *IEEE Trans. Med. Imaging*, vol. 41, no. 9, pp. 2252–2262, Sep. 2022. <https://doi.org/10.1109/TMI.2022.3161787>.
- [16] I. M. B. Ahmed, R. Maalej, and M. Kherallah, "MobileNet-based model for histopathological breast cancer image classification," in *Proc. Int. Conf. Hybrid Intell. Syst.*, Springer, 2022, pp. 636–643. https://doi.org/10.1007/978-3-031-27409-1_58.
- [17] M. Ahmadi, N. Karimi, and S. Samavi, "A lightweight deep learning pipeline with DRDA-Net and MobileNet for breast cancer classification," *arXiv preprint*, arXiv:2403.11135, 2024. <https://doi.org/10.48550/arxiv.2403.11135>.
- [18] X. Zhou, Y. Wang, and Y. Jiang, "A comprehensive review for breast histopathology image analysis using classical and deep neural networks," *IEEE Access*, vol. 8, pp. 90931–90956, 2020. <https://doi.org/10.1109/ACCESS.2020.2993788>.
- [19] M. Z. Alom, Q. T. Tran, and B. A. Orr, "Learned image resizing with efficient training (LRET) facilitates improved performance of large-scale digital histopathology image classification models," *arXiv preprint*, arXiv:2401.11062, 2024. <https://arxiv.org/abs/2401.11062>.
- [20] S. R. Awad, M. A. Al Jbaar, and M. A. M. Abdullah, "Efficient and low-cost Arduino based solar tracking system," in *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 745, no. 1, p. 012016, 2020. <https://doi.org/10.1088/1757-899X/745/1/012016>.
- [21] Mohammed, A. A., Awad, S. R., Abdullah, M. A. M., Elbasi, E., and Woo, W. L. "Quantifying the impact of watermarking on deep learning accuracy in medical image classification," *IEEE Access*, vol. 12, pp. 35678–35690, 2024. <https://doi.org/10.1109/ACCESS.2024.3486757>.
- [22] S. R. Awad and F. S. Alghareb, "Encoding-based machine learning approach for health status classification and remote monitoring of cardiac patients," *Algorithms*, vol. 18, no. 2, p. 94, 2025. <https://doi.org/10.3390/a18020094>.
- [23] Y. Xu, R. Song, and X. Gu, "Edge deep learning in computer vision and medical diagnostics: A comprehensive survey," *Artif. Intell. Rev.*, vol. 58, no. 3, pp. 1–78, 2025. <https://doi.org/10.1007/s10462-024-11033-5>.
- [24] H. Mewada, "Extended deep-learning network for histopathological image-based multiclass breast cancer classification using residual features," *Symmetry*, vol. 16, no. 5, p. 507, 2024. <https://doi.org/10.3390/sym16050507>.
- [25] N. Sengodan, "EfficientNet with hybrid attention mechanisms for enhanced breast histopathology classification: A comprehensive approach," *arXiv preprint*, arXiv:2410, 2024. <https://arxiv.org/html/2410.22392v2>.
- [26] G. U. Nneji, O. P. Samuel, and A. I. Ekwueme, "Lightweight separable convolution network for breast cancer histopathological identification," *Diagnostics*, vol. 13, no. 2, p. 299, 2023. <https://doi.org/10.3390/diagnostics13020299>.
- [27] M. A. Hasan, R. Chowdhury, and M. I. Uddin, "An end-to-end lightweight multi-scale CNN for the classification of lung and colon cancer with XAI integration," *Technologies*, vol. 12, no. 4, p. 56, 2024. <https://doi.org/10.3390/technologies12040056>.
- [28] M. Zubair, K. Sadiq, and S. M. Khan., "An interpretable framework for gastric cancer classification using multi-channel attention

- mechanisms and transfer learning approach on histopathology images," *Sci. Rep.*, vol. 15, no. 1, p. 13087, 2025. <https://doi.org/10.1038/s41598-025-97256-0>.
- [29] K. Djoumessi, S. O. Mensah, and P. Berens, "A hybrid fully convolutional CNN-Transformer model for inherently interpretable medical image classification," *arXiv preprint*, arXiv:2504.08481, 2025. <https://doi.org/10.48550/arXiv.2504.08481>.
- [30] N. Sengodan, "CBAM-EfficientNetV2 for histopathology image classification using transfer learning and dual attention mechanisms," *arXiv preprint*, arXiv:2410.22392, 2024. <https://arxiv.org/html/2410.22392v4>.
- [31] U. K. Lilhore, P. A. Borole, and A. D. Chaudhari, "Hybrid convolutional neural network and Bi-LSTM model with EfficientNet-B0 for high-accuracy breast cancer detection and classification," *Sci. Rep.*, vol. 15, no. 1, p. 12082, 2025. <https://doi.org/10.1038/s41598-025-95311-4>.
- [32] R. Ochoa-Ornelas, A. Gudiño-Ochoa, and J. A. García-Rodríguez, "A hybrid deep learning and machine learning approach with mobile-EfficientNet and grey wolf optimizer for lung and colon cancer histopathology classification," *Cancers*, vol. 16, no. 22, p. 3791, 2024. <https://doi.org/10.3390/cancers16223791>.
- [33] O. A. Abioye, A. E. Ewwiekpaefe, and A. J. Awujoola, "Performance evaluation of EfficientNetV2 models on the classification of histopathological benign breast cancer images," *Sci. J. Univ. Zakho*, vol. 12, no. 2, pp. 208–214, 2024. <https://doi.org/10.25271/sjuoz.2024.12.2.1261>.
- [34] S. Dawwd, "FPGA based flexible implementation of lightweight inference on deep convolutional neural networks," *Int. Arab J. Inf. Technol.*, vol. 21, no. 3, pp. 408–417, 2024. <https://doi.org/10.34028/iajit/21/3/5>.
- [35] C. Zhu, Y. Song, and S. Liu, "Breast cancer histopathology image classification through assembling multiple compact CNNs," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, pp. 1–17, 2019. <https://doi.org/10.1186/s12911-019-0913-x>.
- [36] M. Dörrich, M. Fan, and A. M. Kist, "Impact of mixed precision techniques on training and inference efficiency of deep neural networks," *IEEE Access*, vol. 11, pp. 57627–57634, 2023. <https://doi.org/10.1109/ACCESS.2023.3284388>.
- [37] K. Datta Gupta, D. Mandal, A. Dey, and M. Pal, "A novel lightweight deep learning-based histopathological image classification model for IoMT," *Neural Process. Lett.*, vol. 55, no. 1, pp. 205–228, 2023. <https://doi.org/10.1007/s11063-021-10555-1>.