Mesopotamian Journal of Artificial Intelligence in Healthcare
Vol.2025, **pp**. 244–259
DOI: https://doi.org/10.58496/MJAIH/2025/023; ISSN: 3005-365x
https://mesopotamian.press/journals/index.php/MJAIH

Research Article

Lightweight Deep Learning for IoT-Based Medical Diagnosis: A Survey

Balqees Talal Hasan [1,*], Ali Mohsin Ahmed Al-Sabaawi [2], Zaid J. Al-Araji [1]

[1] *Department of Computer Networks and Internet, College of Information Technology, Ninevah University, Mosul, Iraq.*

[2] *Department of Software, College of Information Technology, Ninevah University, Mosul, Iraq.*

ARTICLE INFO

Article History

Received 17 Sep. 2025
Revised 15 Oct. 2025
Accepted 10 Nov. 2025
Published 10 Dec. 2025

Keywords
IoT-based Medical, Diagnosis,
Edge Computing,
Lightweight Deep Learning,
Model Pruning,
Quantization,
Knowledge Distillation.

ABSTRACT

The rapid expansion of Internet of Things (IoT) technologies in healthcare has enabled continuous patient monitoring and data-driven clinical decision support. Deep learning models excel at extracting intricate patterns from high-dimensional medical data but face significant deployment challenges on resource-constrained IoT devices due to high computational and memory demands. This survey systematically analyzes four core strategies to adapt deep learning for edge based medical diagnosis: model pruning, quantization, knowledge distillation, and inherently lightweight architectures. For each approach, we dissect the underlying methodologies, evaluate the trade-offs between model efficiency and predictive performance, and present relevant medical case studies. We further review deployment frameworks (e.g., TensorFlow Lite, PyTorch Mobile, ONNX) that facilitate integration with IoT hardware. Evidence indicates that these lightweight techniques can substantially reduce model size and inference latency while preserving diagnostic accuracy, enabling real-time AI-powered healthcare in decentralized settings. Finally, we identify critical research challenges including energy-efficient optimization, privacy-aware model design, and end-to-end automation across software and hardware layers—and outline future directions to advance robust, efficient, and trustworthy edge AI in healthcare.

1. INTRODUCTION

The rapid proliferation of Internet of Things (IoT) technologies is fundamentally transforming healthcare delivery paradigms. From wearable biosensors and smart implants to connected medical devices, the IoT ecosystem generates unprecedente volumes of real-time physiological data, enabling continuous patient monitoring, early disease detection, and personalized treatment strategies [1, 2]. Deep learning (DL) models have emerged as powerful tools for extracting intricate patterns from high-dimensional medical data, achieving remarkable success in tasks ranging from medical image analysis to time-series diagnostics. However, the computational and memory demands of state-of-the-art DL architectures—often comprising tens of millions of parameters and requiring billions of floating-point operations per inference—present formidable barriers to deployment on resource-constrained IoT end-devices, which typically operate under strict power, memory, and processing constraints[3, 4] This deployment gap between powerful DL models and limited IoT hardware is particularly acute in healthcare applications where real-time inference, low latency, and patient data privacy are paramount. Offloading computation to cloud servers introduces network latency, bandwidth limitations, and potential privacy vulnerabilities, undermining the promise of immediate, on-device clinical decision support. Performing inference in place, or at the edge near the device, minimizes latency and improves energy efficiency compared to offloading computation to the cloud [5–7]. Consequently, there is an urgent need for lightweight DL techniques that can maintain high diagnostic accuracy while dramatically reducing computational complexity and memory footprint, enabling efficient on-device inference across diverse healthcare IoT platforms [8]. The integration of IoT technologies has significantly transformed medical diagnostics, moving away from sporadic, facility-based assessments toward continuous, patient-centered monitoring [9, 10]. Medical diagnosis systems have traditionally depended on cloud computing facilities for handling the heavy processing demands of deep learning analysis [11, 12]. However, the healthcare industry experiences enormous revolutions with the transition of processing from central clouds to devices by IoT technology, creating an age of health monitoring on personal devices despite their having low computing power [13, 14].

*Corresponding author. Email: balqees.hasan@uoninevah.edu.iq

Lightweight deep learning models are specifically developed to run efficiently on limited memory and processing power edge devices, allowing for faster inference and less energy usage, which are key necessities for battery-powered or intermittently powered systems [15, 16]. The latest trends have demonstrated the growing efficiency of lightweight deep learning models in practical medical diagnostic systems, especially when integrated with IoT and edge computing technologies [17]. This survey provides a comprehensive analysis of the state-of-the-art in lightweight deep learning for IoT-based medical diagnosis. We systematically examine four fundamental approaches to model compression and efficiency: (1) model pruning techniques that eliminate redundant parameters; (2) quantization methods that reduce numerical precision; (3) knowledge distillation frameworks that transfer knowledge from large teacher models to compact student networks; and (4) inherently lightweight architectural designs optimized for resource-constrained environments. For each approach, we dissect the underlying methodologies, evaluate the critical trade-offs between model efficiency and predictive performance, and highlight representative case studies across various medical domains including dermatology, oncology, pulmonology, and cardiology. Furthermore, we review deployment frameworks such as TensorFlow Lite, PyTorch Mobile, and ONNX that facilitate the integration of lightweight models with IoT hardware. Our analysis demonstrates that these lightweight techniques can achieve substantial reductions in model size (often by orders of magnitude) and inference latency while preserving diagnostic accuracy, paving the way for real-time AI-powered healthcare services in decentralized and resource-limited settings. Finally, we identify pressing research challenges and outline future directions, including energy-efficient optimization, privacy-aware model design, robustness under distribution shift, end-to-end automation across software and hardware layers, and the development of standardized benchmarks for medical edge AI. By synthesizing current advancements and highlighting open problems, this survey aims to guide researchers and practitioners toward the development of robust, efficient, and trustworthy edge AI solutions that can realize the full potential of IoT in transforming healthcare delivery. The remainder of this survey is structured as follows. Section 2 reviews the evolution of medical diagnostic systems from centralized cloud-based solutions to continuous, on-device monitoring enabled by IoT technologies. Section 3 provides an overview of common deep learning architectures used in medical diagnostics, with a comparison of their complexity and suitability. Section 4 reviews lightweight deep learning techniques tailored for resource-constrained IoT environments, covering model pruning, quantization, knowledge distillation, and Lightweight Deep Learning Models. Section 5 describes case studies demonstrating the use of these approaches in a range of medical fields, including dermatology, oncology, and pulmonology. Section 6 discusses frameworks that support lightweight model deployment on the edge. Section 7 discusses open challenges and potential future research directions. Section 8 concludes the survey by summarizing findings and identifying the potential for successful and trustworthy IoT-based medical diagnosis.

2. EVOLUTION OF MEDICAL DIAGNOSTIC SYSTEMS IN THE ERA OF IOT

The integration of the Internet of Things (IoT) into the healthcare sector has significantly transformed the landscape of medical diagnostics, moving away from sporadic, facility-based assessments toward continuous, patient-centered monitoring [9, 10]. Central to this transformation are IoT-enabled systems that employ advanced sensors and microcontrollers to continuously capture vital signs and transmit them to cloud-based platforms. This connectivity allows physicians to remotely monitor patients in real time and receive intelligent alerts when parameters deviate from normal ranges [18]. The practical impact of these systems has been demonstrated in multiple studies, showing that wearable IoT devices can reduce hospitalizations and facilitate proactive adjustments to patient care plans [19]. For example, a clinical trial focusing on heart failure patients used smart wristbands for personalized physical activity monitoring, resulting in improved exercise adherence and fewer hospital readmissions by providing real-time feedback and dynamically adjusting activity goals [20]. Together, these findings illustrate that IoT wearables not only enhance continuous monitoring but also enable timely clinical interventions, thereby improving patient outcomes [18]. Medical diagnosis systems have traditionally depended on cloud computing facilities for handling the heavy processing demands of deep learning analysis. The arrangement involves sending different types of medical data to distant servers where advanced processing is undertaken with the help of significant computing resources. Various forms of health data, ranging from full-body scans to cardiac signal recordings, are processed by these cloud computer systems remotely. This processing requires enormous amounts of computational resources, which are typically available in large-scale cloud facilities. The nature of these tasks demands specialized hardware for efficient operation. Deep learning networks, along with some of the well-known architectures, require considerable computing power for efficient functioning. These complex networks depend to a large extent on high-performance graphics processors and large storage systems that are typically available in centralized computing infrastructure. The utilization of these highly advanced systems is one of the notable technological accomplishments in contemporary healthcare diagnostics. Infrastructure requirements of these systems remain high, yet a need to provide flawless diagnostic operations. Storage systems and processing capabilities must be of high levels to enable the smooth operation of these critical healthcare systems [11, 12]. However, recent research has highlighted several limitations of this cloud-centric approach, particularly for time-sensitive and privacy sensitive medical applications. For instance, Alajlan and Ibrahim (2022) [15] demonstrated that cloud-based models introduce significant latency and privacy risks due to data transmission, which can be critical in emergency medical scenarios. Similarly, Al-Araji et al. (2025) [16] conducted a systematic review of healthcare security in edge-fog-cloud environments and emphasized the need for moving processing closer to the data source to enhance security and reduce latency. Furthermore, Anjum et al. (2022) [17] surveyed IoT-based COVID-19 diagnosing systems and found that edge based AI models significantly outperform cloud-based models in terms of response time and bandwidth usage, making them more suitable for real-time pandemic monitoring. These studies collectively underscore the growing trend towards edge-based medical diagnosis systems that leverage lightweight deep learning models to overcome the limitations of cloud computing. The healthcare industry experiences enormous revolutions with the transition of processing from central clouds to devices by IoT technology. The revolution creates an age of health monitoring on personal devices despite their having low computing power. The process of device-based processing is a central revolution in the medical diagnostic approach. Various handheld devices, ranging from simple smart watches to sophisticated Arduino platforms, now perform tasks previously devoted to heavy-duty computers. For instance, recent studies have demonstrated the use of smartwatches for real-time arrhythmia detection, leveraging lightweight algorithms to analyze electrocardiogram (ECG) data directly on the device [21]. Similarly, research has shown the feasibility of deploying diagnostic models on low-power microcontrollers, such as Arduino, for continuous glucose monitoring and alerting, thereby enabling proactive management of chronic conditions [22]. The trend represents a major milestone in the evolutionary trajectory of healthcare technology. It becomes more and more crucial to design smaller, more efficient deep learning models to adapt to this new strategy. These smaller models must perform well enough with the limited capabilities of mobile devices. The key is to be accurate while reducing computation needs. Modern medical monitoring more and more relies on distributed processing capacity. Smarter devices locally process health data, enabling reduced dependence on faraway computer servers. This configuration improves response time while continuing to provide diagnostic savvy through advanced programming techniques [13, 14]. Moreover, the shift to edge computing not only reduces latency but also enhances data privacy and security, as sensitive health information can be processed locally without being transmitted to the cloud [23]. The lightweight deep learning models are specifically developed to run efficiently on limited memory and processing power edge devices. The models are suitable for faster inference and less energy usage by cutting down the parameters and computational complexity, two key necessities for battery-powered or intermittently powered systems. Knowledge distillation, neural network architecture design, and model compression are prime methods to achieve such efficiency. They allow for the integration of advanced AI capabilities into devices used daily, creating more engaged, tailored, and accessible healthcare solutions [15, 16]. The latest trends have demonstrated the growing efficiency of light-weight deep learning models in practical medical diagnostic systems, especially when integrated with IoT and edge computing technologies. The models are capable of processing physiological signals in real-time straightaway on devices with restricted computational resources, including portable devices and wearable devices. For example, during the COVID-19 pandemic, diagnostic platforms that used edge computing and deep learning models monitored breathing function, blood oxygen saturation, and heart rate from wearable sensors and smart masks. The platforms performed local inference to identify aberrations and triggered automatic alerts on their own, thereby supporting early intervention by physicians and reducing hospital readmission. The same paradigms have been applied to continuous monitoring of vital signs in the management of chronic diseases to facilitate more timely and wise clinical decisions. Through their elimination of centralized cloud infrastructure and inference latency reduction, lean deep learning models have become essential in providing accessible, affordable, and patient-focused healthcare solutions, particularly for remote and resource-scarce environments [17].

3. TYPICAL DEEP LEARNING ARCHITECTURES IN MEDICAL DIAGNOSTIC SYSTEMS

Deep learning has advanced rapidly during the last decade, yielding increasingly powerful models capable of solving challenging image processing tasks, including those in the medical arena. This section provides an overview of the most prominent architectures that affected the field. Table I summarizes the performance indicators, model complexity, and computational needs of these models for a clear comparison. This comparative analysis provides valuable insights into the trade-offs among accuracy, number of parameters, model size, and floating-point operations (FLOPs) across these architectures.

TABLE I. COMPARATIVE ANALYSIS OF TYPICAL DEEP LEARNING MODELS

Ref.	Model	Top-1 (%)	Top-5 (%)	Params (M)	Size (MB)	FLOPs (G)
[24]	AlexNet	57.0	80.3	61	238	0.72
[25]	VGG16	70.5	90.0	138	528	15.5
[26]	GoogLeNet	72.5	90.8	6.9	90	1.6
[27]	ResNet-50	75.8	92.9	25.6	102	3.8
[28]	EfficientNet-B7	84.4	97.3	66	256	37
[29]	ViT-B/16	77.9	93.5	86	344	17.5
[30]	Swin-L	87.3	98	197	790	47

AlexNet AlexNet's innovation, introduced by Krizhevsky et al. in 2012 [24], was the beginning of deep learning. It achieved remarkable improvement over the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012, with a Top-5 accuracy of 80.3 percent (19.7 percent error rate). AlexNet is comprised of eight layers: three fully connected and five convolutional layers. The Rectified Linear Unit (ReLU) activation function was introduced to boost training effectiveness and alleviate the vanishing gradient issue that previously constrained deep network optimization. Despite such enhancements, the performance of AlexNet was at the cost of high computational requirements such as 61 million parameters, model size of 238 MB, and approximately 0.72 GFLOPs per inference [31]. It took 1.2 million images to train it and consumed two NVIDIA GTX 580 GPUs for five to six days [24], indicative of its dependency on heavy computational resources. In medical imaging, AlexNet has demonstrated promising results in various applications, such as lung cancer detection [32], breast cancer histopathology [33], and skin cancer classification [34], primarily due to its capability to learn hierarchical features automatically. However, its extensive computational demands and limited model interpretability present challenges for deployment in IoT-based medical systems, where efficiency and transparency are essential for practical clinical adoption [35].

VGG16 Building on the success of AlexNet, VGG16 was introduced by Simonyan and Zisserman in 2014 [25], demonstrating that increasing the depth of neural networks can lead to better performance. VGG16 is composed of 13 convolutional layers and 3 fully connected layers, using consistent 3×3 convolutional filters that simplify the design while allowing the network to learn deeper and more complex features. On the ImageNet dataset, it achieved Top-1 and Top-5 accuracies of 70.5% and 90.0%, respectively. However, the increased depth also makes VGG16 more prone to overfitting, particularly when trained on small datasets.

These accuracy gains come with a significant computational cost: 138 million parameters, 528 MB of storage, and 15.5 GFLOPs per inference [36]. Despite these resource demands, VGG16 has shown strong performance in various medical diagnosis tasks, such as skin cancer classification [37], diabetic retinopathy detection [38], and histopathological image analysis [39], particularly when transfer learning techniques are applied. Nevertheless, its large memory and computational requirements make it less suitable for real-time IoT healthcare systems, which often rely on edge devices with limited power and processing capacity [40, 41].

GoogLeNet GoogLeNet, also known as Inception v1, was introduced by Szegedy et al. in 2014 [26] and marked a significant advancement in convolutional neural network architecture by emphasizing computational efficiency while maintaining high accuracy. The central innovation of the model lies in the Inception module, which processes input data through multiple parallel convolutional layers using kernel sizes of 1×1, 3×3, and 5×5, along with pooling operations. This design enables the network to extract features at multiple scales while controlling the computational complexity. Employing 1×1 convolutional layers for dimensionality reduction proved highly effective in decreasing both the parameter count and the overall computational complexity. With this architecture, Inception v1 achieved Top-1 and Top-5 accuracies of 72.5 percent and 90.8 percent on the ImageNet dataset, respectively. The model required 6.9 million parameters, occupied approximately 90 megabytes of storage, and performed inference at a cost of 1.6 GFLOPs [26]. Compared to earlier architectures such as AlexNet and VGG16, Inception v1 demonstrated substantially greater efficiency. In medical applications, GoogLeNet has been utilized for various tasks, including health monitoring [42], lung cancer detection [43], and liver lesion diagnosis [44]. Nevertheless, the model's highly specialized design complicates its scalability and limits its ability to adapt automatically to new medical tasks without extensive manual reconfiguration [45].

ResNet-50 A major breakthrough in deep learning came with the introduction of Residual Networks (ResNets) by He et al. in 2015 [27]. The key innovation was the residual learning framework, which introduced shortcut connections that allow gradients to propagate directly across layers. This approach effectively addressed the vanishing gradient problem that had previously limited the training of very deep networks, making it feasible to train architectures with 50, 101, or even 152 layers. ResNet-50, for example, achieves Top-1 and Top-5 accuracies of 75.8 percent and 92.9 percent on the ImageNet dataset, respectively, while maintaining computational efficiency through its bottleneck design. The model contains 25.6 million parameters, requires 102 megabytes of storage, and performs approximately 3.8 GFLOPs per inference. Its bottleneck residual blocks utilize 1×1 convolutions to reduce and then expand feature dimensions, which helps optimize computational resources without sacrificing model depth.

In medical applications, ResNet-50 has demonstrated strong performance in tasks such as brain tumor detection [46], diabetic retinopathy and retinal imaging in ophthalmology [47], and detection and classification of Alzheimer's disease [48]. However, despite its improvements in efficiency compared to earlier models such as VGG16, ResNet-50 still requires considerable computational resources, with 3.8 GFLOPs and a memory footprint of 102 megabytes, which present challenges for direct deployment on resource-limited IoT devices where real-time processing capabilities and energy efficiency are essential.

EfficientNet-B7 A significant milestone in deep learning efficiency was achieved with the introduction of the EfficientNet family by Tan and Le in 2019 [28]. Departing from traditional scaling methods, EfficientNet employs a compound scaling strategy that proportionally adjusts network depth, width, and input resolution, allowing it to deliver high accuracy with substantially fewer parameters and computational demands. The largest model in this family, EfficientNet-B7, attains state-of-the-art performance on the ImageNet dataset, reaching a Top-1 accuracy of 84.4% and a Top-5 accuracy of 97.3%. It contains approximately 66 million parameters, demands about 256 MB of storage, and executes nearly 37 GFLOPs per inference.

Beyond its success in general image recognition, EfficientNet-B7 has also been effectively applied to numerous medical imaging tasks, such as breast ultrasound classification [49] and diabetic retinopathy screening [50], all with strong diagnostic performance and relatively modest computational burden. Even with its greater efficiency compared to earlier generations of deep learning models, EfficientNet-B7's otherwise relatively high computer and memory resource requirements, however, continue to make it difficult to deploy the same on resource-constrained edge devices. Vision Transformer (ViT-B/16) Dosovitskiy et al.'s presentation of Vision Transformers (ViT) in 2020 [29] was a major departure from the traditional CNN architectures toward self-attention models for image processing. In contrast to CNNs, ViT splits input images into fixed-size patches, typically 16×16 pixels, and then flattens them into token sequences, which are subsequently processed via transformer encoders particularly designed for natural language processing tasks [51]. The ViT-B/16 configuration achieves Top-1 and Top-5 accuracies of 77.9 percent and 93.5 percent on the ImageNet dataset, respectively, utilizing 86 million parameters, occupying 344 megabytes of storage, and requiring 17.5 GFLOPs per inference [29].

ViT-based architectures have shown significant promise across a range of medical imaging applications, such as the diagnosis of neurological disorders [52], breast cancer classification [52], and the detection of retinal diseases [53]. Despite these promising results, Vision Transformers exhibit several challenges when applied to IoT-based medical diagnostic systems. The quadratic computational complexity relative to image resolution imposes significant resource demands when processing high-resolution medical images. Moreover, the absence of inductive biases typically present in CNNs necessitates large annotated datasets for effective training, which are often scarce in the medical domain. In addition, ViTs may have difficulty capturing fine-grained local features that are critical for certain diagnostic applications, motivating the development of hybrid models that integrate CNN-based local feature extraction with transformer-based global context modeling [54].

Swin Transformer-Large A significant development in vision transformer architectures was introduced with the Swin Transformer family by Liu et al. in 2021 [30]. Unlike conventional convolutional neural networks, Swin Transformer employs a hierarchical design using shifted window-based self-attention mechanisms, enabling it to efficiently model both local and global representations while maintaining linear computational complexity relative to image size. The largest model in this family, Swin Transformer-Large (Swin-L), demonstrates state-of-the-art performance on the ImageNet dataset, achieving a Top-1 accuracy of 87.3% and a Top-5 accuracy of approximately 98.5%. It consists of approximately 197 million parameters, requires around 790 MB of storage, and performs roughly 64 GFLOPs per inference at 224×224 resolution; when evaluated at higher resolutions such as 384×384, the computational cost increases to approximately 103.9 GFLOPs .

In addition to its success on general vision benchmarks, Swin-L has been increasingly adopted in medical imaging tasks, including COVID-19 infection detection [55], breast cancer detection [56], and lung disease diagnosis [57]. Its ability to capture multi-scale contextual information and long-range dependencies makes it particularly well-suited for complex medical imaging applications. However, despite its outstanding accuracy, Swin-L's substantial parameter count and high computational demands limit its feasibility for deployment on resource-constrained edge devices, restricting its use primarily to high-performance computing systems and server-based clinical environments.

4. LIGHTWEIGHT DEEP LEARNING APPROACHES FOR RESOURCE-CONSTRAINED IOT ENVIRONMENTS

Deep learning has undergone significant advancements, substantially contributing to the proliferation of artificial intelligence across a wide spectrum of practical domains. Nevertheless, the deployment of deep learning models, particularly CNNs, on resource constrained platforms remains a considerable challenge due to their intensive computational and memory requirements. To address these limitations, a range of optimization methodologies have been introduced to compress model size and improve computational efficiency. This section presents an analysis of prominent approaches, including pruning, quantization, knowledge distillation, and the design of inherently lightweight deep learning architectures. A detailed comparative assessment of these techniques is summarized in Table II.

TABLE II. COMPARATIVE ANALYSIS OF LIGHTWEIGHT DEEP LEARNING TECHNIQUES

Technique	Main principle	Key benefits	Main limitations
Model pruning	Removes redundant weights and neurons to create sparse networks	Reduces model size and inference time; lowers memory and computational demands	Irregular sparsity may not yield hardware acceleration; aggressive pruning risks accuracy loss

Model quantization	Converts high-precision parameters into lower-bit representations	Reduces storage and accelerates inference, especially with hardware support	Excessive quantization introduces rounding errors; accuracy degradation at very low bit widths
Knowledge distillation	Transfers knowledge from large teacher to small student via soft labels	Preserves high accuracy in compact models; improves generalization	Requires high-quality teacher; adds training complexity
Lightweight architectures	Designs efficient models from scratch using architectural innovations	Achieves excellent tradeoffs between accuracy, size, and latency	Complex manual design; may require significant expertise

4.1 Model Pruning

Pruning is a model compression technique applied to a neural network f (X,W), where X represents the input and Wdenotes the set of weights. The objective is to identify a reduced subset of weights W', setting the remaining weights in W to zero, while ensuring the model's performance stays above a specified threshold. This results in a sparse network. The sparsity level can be quantified by the ratio of pruned weights to the total original weights, expressed as:

$$s = 1 - \frac{|W'|}{|W|} \qquad (1)$$

A higher sparsity value indicates fewer non-zero parameters remain in the pruned network [58]. As illustrated in Figure 1, pruning simplifies a neural network by removing unnecessary neurons and connections, thereby reducing model complexity without significantly impacting performance. Cheng et al.[59] present a comprehensive taxonomy of pruning methodologies. This taxonomy classifies pruning techniques along three principal dimensions: (1) the type of hardware acceleration, distinguishing between universal acceleration achievable through structured pruning and hardware-dependent acceleration associated with unstructured and semi-structured pruning; (2) the pruning schedule, which specifies whether pruning is performed before training, during training, or after model training; and (3) the pruning strategy, differentiating between criterion-based approaches (e.g., magnitude, sensitivity, or loss change metrics) and learning-based methods such as sparsity regularization, reinforcement learning, and meta-learning. This systematic categorization facilitates the selection of appropriate pruning strategies that balance model compression, training complexity, and deployment feasibility within resource-constrained healthcare environments.

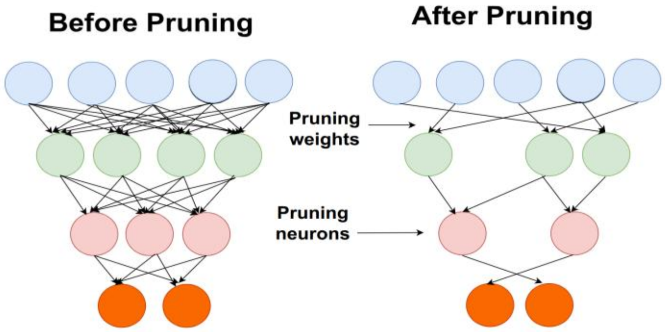

Fig. 1. Pruning Technique. Adapted from Raza et al. [60].

Overall, pruning represents a cornerstone technique for enabling high-performance AI on edge devices for medical diagnosis within IoT ecosystems. By judiciously eliminating redundancy, pruning can reduce both the model size and computational complexity without sacrificing diagnostic accuracy.

4.2 Model Quantization

Model quantization represents a widely adopted compression technique that replaces the high-precision numerical representations found in deep neural networks (DNNs) with lower-precision approximations drawn from a discrete collection of values. This strategy dramatically reduces memory usage and computational overhead by transforming continuous floating-point numbers to fixed-point forms with lower bit widths. The ensuing efficiency gains make quantization particularly useful for implementing DNNs on resource-constrained platforms like edge devices and embedded systems [61]. This technique functions by lowering the precision of network parameters, most notably weights and activations. While traditional DNNs typically operate using 32-bit floating-point arithmetic, quantized models often employ 8-bit fixed-point representations, yielding considerable improvements in runtime efficiency. As a result, quantization offers two critical benefits: a reduced model footprint and accelerated inference, both of which contribute to low-latency and energy-efficient inference on constrained hardware [58]. Depending on the application, quantization can be applied globally across the entire model or selectively to specific components such as weights, activations, or intermediate tensors, as illustrated in Figure 2.

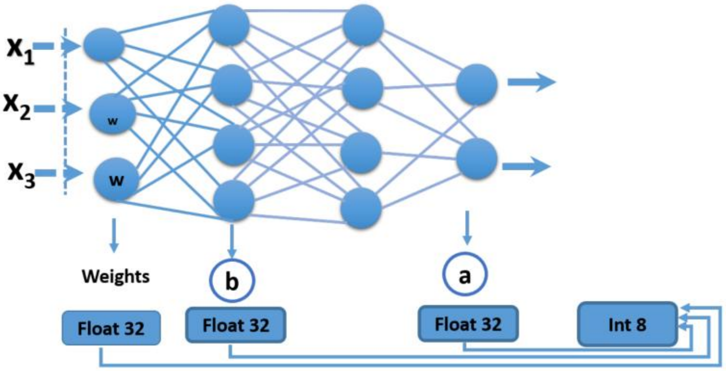

Fig. 2. Quantization technique. Adapted from [62]

Quantization can be formally defined by a piecewise constant function that maps a continuous input value r to a corresponding discrete quantization level q_i. This relationship is expressed as:

$$Q(r) = q_i \quad \text{for} \quad r \in [r_i, r_{i+1}), \quad i = 1, \ldots, m \tag{2}$$

The precision of the quantization process is controlled by the step size, which determines the interval between successive quantization levels:

$$\text{step size} = q_{i+1} - q_i \tag{3}$$

By replacing floating-point operations with fixed-point arithmetic, model quantization significantly reduces hardware resource consumption and decreases inference latency. These improvements are critical for enabling real-time, scalable, and cost-effective deployment of deep neural network (DNN)-based medical diagnostic systems [61].

4.3 Knowledge Distillation

Knowledge Distillation (KD) constitutes a powerful approach to model compression based on a teacher–student framework, where a light-weight neural network (student) is trained to simulate the behavior of a more accurate, larger model (teacher). This configuration enables the student model to achieve the predictive power of the teacher model with a greatly reduced computational cost. Rather than relying solely on hard class labels, the student model is guided by the soft output distributions (logits) produced by the teacher, which encapsulate richer information about inter-class relationships. This facilitates improved generalization in the student model while maintaining a significantly smaller architectural footprint. The distillation process typically involves first training the teacher model on the target task, after which the student is optimized to minimize the divergence between its own output distribution and that of the teacher. By emulating the predictive behavior of the teacher, the student network can maintain equivalent accuracy with a more compact parameter set and reduced computational overhead. In addition to enhancing inference efficiency, knowledge distillation is often used in conjunction with other compression techniques, such as quantization, to enable the practical deployment of deep neural networks on resource-constrained platforms [5, 61]. As shown in Figure 3, the distillation process guides the student model through informative signals derived from the teacher, focusing on knowledge transfer rather than direct parameter replication.

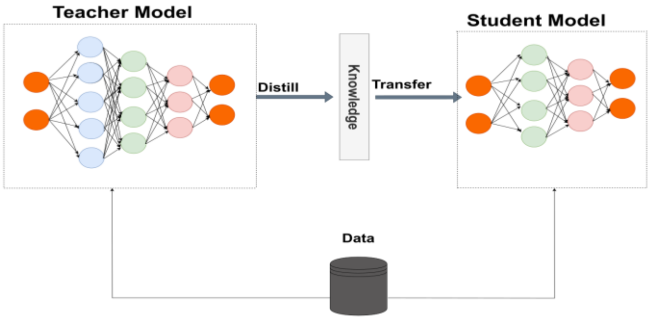

Fig. 3. Knowledge distillation. Adapted from [60].

4.4 Lightweight Deep Learning Models

This section reviews key lightweight deep learning models designed for resource-limited environments. It highlights several prominent models and their main features.

SqueezeNet. SqueezeNet is a highly efficient lightweight deep learning model with high accuracy using very limited computational and memory resources. Composed of Fire Modules that shrink and then expand features using 1x1 and 3x3 convolutions, it attains AlexNet-level accuracy on ImageNet with a model size of less than 5MB—50 times smaller than AlexNet—and can be compressed to 0.5MB. Its design minimizes energy use and off-chip memory access, making it ideal for embedded and mobile applications. Variants like SqueezeDet and SqueezeDet+ demonstrate impressive performance, combining real-time inference with low power consumption. These characteristics make SqueezeNet a benchmark in edge AI, influencing ongoing research in efficient neural network design [63].

MobileNets. MobileNet is a family of lightweight CNNs for mobile and embedded vision, using depthwise separable convolutions to reduce computation by 8–9 times compared to standard convolutions. Versions V2 and V3 enhance efficiency with inverted residual bottlenecks and hardware-aware neural architecture search. MobileNetV2 achieves about 72% accuracy on ImageNet with 3.4 million parameters at 60 FPS on Pixel 2, while MobileNetV3-L reaches 75.2% accuracy with 5.4 million parameters at 50 FPS on Pixel 3, cutting latency by 20%. Compared to SqueezeNet, MobileNetV3-L offers similar speed and better energy efficiency. Innovations like Blueprint Separable Convolutions improve fine-grained task performance by up to 13.7%. Overall, MobileNet effectively balances accuracy, speed, and energy consumption for edge devices [64].

EfficientNet-B0 EfficientNet-B0 is the lightweight baseline model of the EfficientNet family, designed using neural architecture search to optimize both accuracy and computational efficiency [28]. It employs efficient Mobile Inverted Bottleneck Convolution (MBConv) blocks and Squeeze-and-Excitation modules to maintain a compact architecture while delivering strong performance.EfficientNet-B0 gets 76.3% Top-1 and 93.2% Top-5 accuracy on ImageNet, but with only 5.3 million parameters and requiring only 0.39 GFLOPs per inference. Because of its light architecture and low overhead, it is a great candidate for deployment on low-resource platforms, including mobile and edge computing nodes. Its balance between efficiency prediction can also make it extremely useful for transfer learning in domains such as medical image processing, where computational constraints are often a primary concern.

MnasNet. MnasNet is a lightweight neural architecture specifically designed for mobile platforms, leveraging automated Neural Architecture Search (NAS) to jointly optimize accuracy and real-world inference latencyThe approach leverages a factorized hierarchical search space to foster diverse architectures, ensuring that model complexity and computational overhead remain controlled. Formulated as a multi-objective optimization problem, MnasNet effectively balances latency and accuracy. On the ImageNet classification task, MnasNet-A1 achieves a Top-1 accuracy of 75.2% with a measured latency of 78 ms on a Pixel phone, outperforming both MobileNetV2 and NASNet. The larger variant, MnasNet-A3, further enhances performance, surpassing the accuracy of ResNet-50 while requiring fewer parameters and reduced computational cost [65].

5. LIGHTWEIGHT DEEP LEARNING FOR MEDICAL APPLICATIONS: CASE STUDIES

Numerous studies have proposed lightweight deep learning techniques to simplify the application of AI-driven medical diagnostics on edge and IoT devices by balancing computational complexity with diagnostic accuracy. This section summarizes some typical works for various medical applications, such as dermatology, cancer, hematology, dentistry, and musculoskeletal imaging. These studies demonstrate that optimization techniques like pruning, quantization, knowledge distillation, and lightweight architectures can be utilized to produce excellent diagnostic performance under strict resource constraints. The main technique, medical application, model size, and performance of these studies are reviewed side by side in Table III.

TABLE III. COMPARISON OF LIGHTWEIGHT DL TECHNIQUES IN IOT-BASED MEDICAL DIAGNOSIS

Ref.	Core Approach	Medical Application	Size (MB)	Performance
[66]	Pruning	Breast tomosynthesis	82.4	AUC: 0.90
[67]	Pruning	Skin diseases	12.5	Acc: 83.5%
[68]	Pruning	Brain tumor	≈ 21	Acc: +0.39
[69]	Pruning	Pneumonia	≈ 13.8	Acc loss < 1
[70]	Pruning	Arrhythmia	≈ 4.4	Acc: 99.24
[71]	Quantization	White Blood Cell	Reduced by 44.86% versus float32	Acc: 98.44%
[72]	Quantization	Breast tumor	15.6	Acc: 88.0%
[73]	Quantization	3D liver/ brain tumor	BRATS2020: 0.43 LiTS: 6.1	BRATS2020: DSC: 0.8439 LiTS: DSC: 0.7807
[72]	Quantization	breast tumor	75% reduction	ACC: 86.5%
[74]	Quantization	Skin disease	Power-of-2 weight quantization	F1-Score: 0.850
[5]	Knowledge distillation	COVID-19	0.25	ACC: 98.93
[75]	Knowledge distillation	Dental X-ray	7.5	Dice: 0.890
[76]	Knowledge distillation	Brain tumor	232.7	ACC: 97.48
[76]	Knowledge distillation	Eye Disease	232.7	ACC: 93.51
[76]	Knowledge distillation	Alzheimer's Disease	232.7	ACC: 99.46
[77]	MobileNetv2	COVID-19	14	Acc: 99.6
[78]	EfficientNet-B3	Leukemia	43.1	Acc:99.31
[79]	Multi-scale SqueezeNet	COVID-19	≈ 4	ACC: 92.11
[80]	Lightweight DenseNet121	Bone fracture	59.12	ACC:90.3

5.1 Lightweight Medical Diagnosis via Pruning

Pruning techniques have demonstrated significant potential in addressing the computational challenges inherent in enabling the use of deep learning models for medical diagnostic applications within resource-constrained environments. Samala et al. [66] introduced an evolutionary pruning framework for breast cancer diagnosis using digital breast tomosynthesis (DBT). By leveraging genetic algorithms, their method achieved a 95% reduction in convolutional FLOPs, a 34% reduction in total parameters, and a 99.6% decrease in convolutional connections, all while maintaining diagnostic performance on 94 independent DBT cases. This framework exemplifies how pruning can enable the efficient deployment of deep learning models within clinical DBT workflows operating under limited hardware resources. Similarly, Xiang et al. [67] applied pruning to optimize lightweight deep learning models for skin disease classification, demonstrating that pruned versions of MnasNet and MobileNetV2 achieved diagnostic accuracies of 83.5% and 80.6%, respectively, while reducing memory footprints to as low as 12.5 MB and 11.9 MB. This considerable reduction in model size facilitates the practical implementation of AI-based skin diagnostics in clinical environments with restricted computational capacity. In the domain of brain tumor classification, Zebregs et al.[68] demonstrated the combined effectiveness of weight and filter pruning to optimize convolutional neural networks for IoT-based medical diagnosis. Weight pruning reduced the model size by up to 57.5% (from 49.4 MB to approximately 21 MB), while slightly improving classification accuracy by 0.39% at 70% sparsity. Concurrently, filter pruning decreased computational complexity by up to 76.44% in FLOPs, with an additional accuracy gain of 2.74% observed in certain configurations. These findings underscore the potential of pruning techniques to enable real-time, energy-efficient brain tumor diagnosis on edge devices constrained by computational and power limitations. Furthermore, Yang et al.[69] proposed CPGResNet50, a lightweight deep learning model developed to optimize pneumonia diagnosis while substantially reducing computational requirements for deployment on clinical edge hardware. Their approach integrates a custom channel pruning method combined with optimized convolution operations to remove redundant parameters and simplify network structure without compromising accuracy. Specifically, the pruning strategy achieved an 85% reduction in parameters (from 23.7M to 3.455M) and a corresponding reduction in FLOPs (from 4.12G to 523.09M), while maintaining diagnostic accuracy within 1% of the original ResNet50. This substantial compression, combined with training stability, demonstrates the feasibility of deploying high-precision pneumonia classifiers on resource constrained IoT and medical edge platforms. In the domain of portable cardiac monitoring, Liu et al.[70] proposed a lightweight neural network approach for real-time arrhythmia classification based on 12-lead electrocardiograms (ECGs), specifically designed for wearable and portable healthcare devices.The proposed approach integrates the selection of benchmark networks, pruning techniques, and learning rate decay-based fine-tuning to effectively lower model complexity without compromising diagnostic performance. Through the application of pruning, model size was decreased by 51.26%, reaching a final size of about 4.4 MB, with parameters and FLOPs decreased by 47.6% and 49.1%, respectively. Despite this substantial compression, the model performed 94.8% accuracy on clinical ECG data, 92.4% on a hold-out test set, and 99.24% on the MIT-BIH arrhythmia database. These are indications of the model's capability to enable real-time arrhythmia detection on resource-constrained IoT and wearables, allowing effective and accurate cardiac monitoring in clinical settings.

5.2 Lightweight Medical Diagnosis via Quantization

The application of deep learning models within medical diagnostic systems, where model simplicity, computational speed, and performance fairness are crucial, has been made possible in large part by quantization. A mobile lensless microfluidic imaging system containing an integer-based quantized CNN accelerator employing FPGA for White Blood Cell (WBC) segmentation and classification was proposed by Liao et al [71] for blood analysis. Their approach achieved a classification accuracy of 98.44%, with only a 0.56% reduction compared to its 32-bit floating-point counterpart, while reducing hardware area by approximately 45%. This design enabled real-time, edge-based diagnostics, demonstrating the potential of hardware-aware quantization for portable medical applications. In breast tumor detection, Garifulla et al. [72] applied post-training quantization to models such as GoogLeNet and VGG16, significantly reducing model sizes to 15.6 MB and 18.6 MB, respectively, while maintaining high diagnostic accuracy (88.0% and 87.0%). Their evaluation of ResNet34 yielded 77.0% accuracy with a compact 21.5 MB model, highlighting the feasibility of deploying quantized models on mobile and embedded medical devices without substantial performance degradation. Building upon prior advancements in quantization for medical image segmentation, Zhang and Chung [73] introduced the MedQ framework, which employs a ternary quantization approach to achieve substantial model compression while preserving accuracy comparable to full-precision models. The framework attains 12.6× compression on the BRATS2020 dataset by reducing the model size from 5.9 MB to 0.43 MB, and 15.5× compression on the LiTS dataset, decreasing the model size from 94.8 MB to 6.1 MB. Despite this significant reduction, MedQ maintains lossless segmentation performance, achieving Dice Similarity Coefficients of 0.8439 on BRATS2020 and 0.7807 on LiTS, closely matching the full-precision baselines. The approach integrates 2-bit quantization with adaptive clipping, radical residual connections, and a tanh-based gradient approximation to mitigate quantization-induced errors. Its hardware-friendly design, based on symmetric power-of-two quantization, enables efficient and scalable deployment of real-time medical diagnostics under limited resource conditions. Garifulla et al. [72] investigated the application of quantization techniques for breast tumor classification using ultrasound images. By employing full-integer (INT8) quantization on models such as VGG16, they achieved a 75% reduction in model size (from 74.1 MB to 18.6 MB) with only a 0.5% drop in accuracy (FP32: 87%, INT8: 86.5%). Moreover, deployment on a mobile NPU resulted in a 33× speedup in inference time compared to CPU execution. These findings underscore the effectiveness of full-integer quantization in enabling efficient, accurate, and portable diagnostic systems on edge devices. To address disparities in clinical AI systems, Guo et al. [74] proposed a fairness-aware weight quantization technique for dermatological disease diagnosis. Their method achieved a 25.5% reduction in skin tone bias on the Fitzpatrick-17k dataset and a 56.8% reduction in gender bias on ISIC 2019 (measured via EOpp1), while attaining a high F1-score of 0.850—representing a 15.6% improvement over the baseline—and the highest precision (0.857) among competing fairness-aware models. Additionally, their use of 8-bit power-of-two quantization offered implicit model compression, supporting equitable, low-resource diagnostic applications without compromising clinical utility.

5.3 Lightweight Medical Diagnosis via Knowledge Distillation

Advancing the integration of lightweight AI into medical diagnostics, Lin et al. [75] introduced KCNet, a compact deep learning model developed through knowledge distillation. Designed specifically for the segmentation of dental structures in panoramic X-ray images, KCNet demonstrates strong segmentation performance with a Dice score of 0.890 and an Intersection over Union (IoU) of 0.804. Remarkably, it achieves this using only 0.33 million parameters and 0.8 GFLOPs, making it highly efficient and well-suited for deployment on resource-constrained edge devices in dental imaging applications. In a related effort, Alabbasy et al. [5] proposed a KD-based framework tailored for the detection of COVID-19 from chest X-ray images. Their approach effectively distilled knowledge from a high-accuracy teacher model (99.84%, 1.33 MB) into a lightweight student model (98.93%, 0.25 MB), yielding an 82.5% reduction in model size and a 6.14× speedup in inference time with minimal performance degradation. When compared to other compression strategies such as pruning and quantization, their KD approach provided superior results in both efficiency and diagnostic accuracy, highlighting its practicality for edge- and fog-based healthcare systems. Complementing these advancements, Jiang et al. [76] presented a knowledge distillation framework that emphasizes both computational efficiency and model interpretability. Their method transfers knowledge from a DenseNet121 teacher to a simplified 5-layer CNN student model, resulting in a 50% reduction in FLOPs while maintaining high classification accuracy. Uniquely, the proposed framework allows for immediate, layer-wise interpretation of the model's decision making, going beyond traditional post-hoc tools like Grad-CAM and SHAP to offer clinicians a clearer insight into Ai generated outcomes. Validated across diverse medical imaging tasks, including brain tumor classification (97.48%), retinal disease diagnosis (93.51%), and Alzheimer's disease staging (99.46%), the student model achieved near-parity with the teacher in performance. Furthermore, explanation latency was significantly reduced—up to 80%—facilitating real-time interpretability in clinical workflows. Collectively, this work underscores the potential of distillation-based methods for enabling interpretable, real-time diagnostic support on lightweight platforms.

5.4 Lightweight Medical Diagnosis via Lightweight Deep Learning Models

The practical value of lightweight artificial intelligence in medical applications is further demonstrated by Ukwandu et al. [77], who evaluated MobileNet V2 for COVID-19 diagnosis. The proposed model attained a notable accuracy of 99.6% on a binary classification task distinguishing COVID-19 from normal lung images, while maintaining a compact footprint of approximately 3.5 million parameters and 14 MB of memory usage. The VGG16 model, consuming more memory at 528 MB and having 138 million parameters, is in contrast to this one. This paper highlights how small-sized architectures like MobileNet V2 can provide real-time diagnosis accuracy in mobile healthcare settings and are thus suitable for deployment in resource-limited environments. In a step towards providing lightweight medical imaging solutions, Batool et al. [78] proposed a depthwise separable convolution-enhanced EfficientNet-B3 network designed to distinguish between acute lymphoblastic leukemia (ALL) and normal white blood cells. Leveraging the efficiency of depthwise separable convolutions, the network substantially decreases parameter count and memory usage while preserving high diagnostic performance, attaining an accuracy of 99.31%. Evaluated on two publicly available datasets, the model demonstrated outstanding generalization performance on binary and multi-class classification tasks. Comparison with state-of-the-art deep learning classifiers confirmed its superior accuracy and computational efficiency, demonstrating its readiness for clinical translation in hematological diagnosis. Focusing on COVID-19 detection in lightweight frameworks, Joshi et al. [79] introduced LiMS-Net, a light-weight multiscale convolutional neural network whose objective is to resolve overfitting caused by the limited size of the training dataset and the high computational cost of standard CNNs. LiMS-Net depends on parallel convolutional filters of diverse sizes in two feature learning blocks to capture multi-scale discriminative features. This architecture keeps the model with only 2.53 million parameters, substantially reducing memory and computation expenses over larger pretrained models. Evaluated on publicly accessible COVID-19 CT scan databases, LiMS-Net scored 92.11% accuracy and F1-score of 92.59%, outperforming numerous state-of-the-art methods. These findings place the focus on LiMS-Net as a low-resource yet successful application for use in real-time clinical settings with limited data and hardware. Abdusalomov et al. [80] presented a model for identifying sports-induced bone fractures that combines a DenseNet121 model with the Canny edge detector to enhance the effectiveness of AI-assisted medical applications diagnostics. According to reports, the model's computational efficiency was14.78 million parameters and 0.54 GFLOPs, the approach achieves an accuracy rate of 90.3%, marking a new benchmark.%. Without changing the computation overhead, adding the Canny edge detector increased the model's sensitivity to minute fractures. The framework's high efficiency and accuracy render it well suited for real-time applications in sports medicine, where timely and precise fracture diagnosis is essential.

6. LIGHTWEIGHT DL FRAMEWORKS

Successful application of light-weight deep models to resource-constrained devices is just as much a matter of model design as of the software frameworks that enable efficient operation. These frameworks provide the ability to convert, optimize, and run models on diverse hardware architectures, ranging from low-power microcontrollers to advanced mobile devices. They are essential to translating theoretical model compression gains into real-world latency and power advantages. Below, we describe some prominent frameworks for edge and mobile deployment.

Tensorflow Lite TensorFlow, created by Google, is a free and open-source library for handling dataflow and differentiable programming. It's a go-to framework for machine learning, especially for training and running neural networks [81, 82]. TensorFlow Lite, introduced in 2017, provides a minimalistic version of TensorFlow designed for deploying pre-trained machine learning models on mobile, embedded, and IoT devices. It focuses on efficient on-device inference with minimal delay and a compact footprint. After training a model, developers convert it into a TensorFlow Lite FlatBuffer file (.lite) and deploy it using the TensorFlow Lite Interpreter. The framework supports multiple platforms, including iOS, Android, Linux-based systems (like Raspberry Pi and Arm64 boards), and even microcontrollers. For development, TensorFlow Lite offers APIs in both Java and C++ [6].

Pytorch Mobile PyTorch, Facebook AI Research's (FAIR) open-source deep learning library, evolved from Torch to become a go-to tool for GPU-powered ML research. Unlike NumPy, it unlocks high-performance computing for training neural networks. Since 2019 (PyTorch 1.3), PyTorch Mobile has allowed developers to run models on smartphones—just convert trained models to .pt files and use PyTorch's Android or C++ APIs to build AI-powered mobile apps [83].

ONNX (Open Neural Network Exchange) ONNX, a collaborative project by Microsoft, Meta, Huawei, IBM, AMD, ARM, Intel, and Qualcomm, enables seamless model interoperability across frameworks like TensorFlow, PyTorch, and MXNet. By standardizing neural network representation, it simplifies deployment across diverse environments—from edge devices to cloud platforms and AI accelerators. ONNX also offers tools for model conversion, optimization, and inference, ensuring performance and accuracy across platforms. Its emphasis on interoperability and community-driven development makes it a cornerstone of modern AI workflows, allowing developers to choose the best tools without compatibility constraints [84, 85].

7. RESULTS AND DISCUSSION SECTION

The systematic evaluation of deep learning architectures for medical diagnostics reveals a critical trade-off between model performance and computational resource requirements, as summarized in Table I. Traditional, high-capacity models such as AlexNet [24], VGG16 [25], and more recent transformer-based architectures like ViT-B/16 [29] and Swin-L [30], achieve high accuracy on benchmark datasets but are characterized by substantial parameter counts (e.g., 61M for AlexNet, 197M for Swin-L), large model footprints (e.g., 238 MB for AlexNet, 790 MB for Swin-L), and high computational demands measured in FLOPs (e.g., 15.5 GFLOPs for VGG16, up to 103.9 GFLOPs for Swin-L at higher resolutions). These characteristics render them unsuitable for direct deployment on resource-constrained IoT medical devices, which typically have limited processing power, memory, and energy budgets. For instance, while AlexNet's introduction of ReLU [24] was a significant advancement for training deep networks, its 61 million parameters and 238 MB size present considerable challenges for on-device inference in a wearable ECG monitor [35]. Similarly, the hierarchical design of Swin-L, while effective for capturing long-range dependencies in complex medical images like chest X-rays for COVID-19 detection [55], its computational complexity makes it more appropriate for server-side analysis within a clinical setting rather than on a portable ultrasound device. To address these deployment challenges, this survey has analyzed four primary lightweight deep learning strategies: model pruning, quantization, knowledge distillation, and the design of inherently lightweight architectures. The core principles, benefits, and limitations of each technique are synthesized in Table II. Model pruning aims to create sparse networks by removing redundant weights or neurons, as illustrated in Figure 1. The effectiveness of pruning can be quantified by the sparsity level, s, defined earlier (Equation 1). A higher s value indicates a greater degree of compression. For example, in the study by Samala et al. [66] on breast tomosynthesis, an evolutionary pruning framework achieved a significant reduction in computational FLOPs (95%) and parameters (34%) while maintaining diagnostic performance (AUC: 0.90), demonstrating the practical utility of this approach. However, as noted in Table II, unstructured pruning can lead to irregular sparsity patterns that may not map efficiently to standard hardware, potentially limiting the realized speedups despite a reduction in theoretical operations. Model quantization addresses resource constraints by reducing the numerical precision of the model's weights and activations, as conceptually shown in Figure 2. This process involves mapping continuous values to discrete quantization levels, as defined earlier (Equation 2). By converting, for example, 32-bit floating-point parameters to 8-bit integers, quantization can drastically reduce model size and memory bandwidth requirements, and accelerate inference on hardware that supports low-precision arithmetic. The work by Liao et al. [71] on white blood cell classification demonstrated that an integer-based quantized CNN on an FPGA could achieve 98.44% accuracy, comparable to its floating-point counterpart, while significantly reducing hardware area. This highlights the synergy between algorithmic quantization and hardwareaware design. However, aggressive quantization to very low bit-widths can introduce substantial rounding errors, leading to a degradation in model accuracy, as indicated in Table II. The MedQ framework [73], employing ternary quantization, aimed to mitigate this by integrating adaptive clipping and specialized gradient approximations to maintain segmentation performance (e.g., DSC of 0.8439 on BRATS2020) despite significant model compression (12.6×). Knowledge distillation (KD) involves training a compact "student" model to mimic the behavior of a larger, more accurate "teacher" model, as illustrated in Figure 3. The student model learns not only from the hard labels (ground truth) but also from the "soft labels" (output probabilities or logits) of the teacher model, which provide richer information about inter-class relationships and the teacher's confidence. This process allows the student to achieve performance levels closer to the teacher, often exceeding what it could achieve if trained solely on the hard labels. For instance, Alabbasy et al. [5] distilled a large COVID-19 detection model (99.84% accuracy, 1.33 MB) into a lightweight student model (98.93% accuracy, 0.25 MB), achieving an 82.5% reduction in model size and a 6.14× speedup. This demonstrates KD's effectiveness in creating highly efficient models suitable for edge deployment. The primary trade-off, as noted in Table II, is the requirement for a well-trained, high-quality teacher model and the additional complexity involved in the distillation training process. Jiang et al. [76] further explored KD to not only compress models but also enhance interpretability, showing that a simplified 5-layer CNN student could achieve near-teacher performance (e.g., 97.48% accuracy for brain tumor classification) while offering layer-wise interpretability crucial for clinical trust. Finally, the design of inherently lightweight architectures offers another path to efficiency. Models like SqueezeNet [63], MobileNets [64], EfficientNet-B0 [28], and MnasNet [65] are engineered from the outset to minimize parameters and computational cost. For example, MobileNets utilize depthwise separable convolutions, which factorize a standard convolution into a depthwise convolution and a pointwise convolution, drastically reducing computations. EfficientNet-B0 employs a compound scaling method and MBConv blocks to achieve a strong accuracy-computation trade-off (76.3% Top-1 accuracy on ImageNet with only 5.3M parameters and 0.39 GFLOPs). These models often serve as excellent baselines or backbones for further compression or as student models in KD. The case studies in Table III, such as Ukwandu et al. [77] using MobileNetV2 for COVID-19 diagnosis (99.6% accuracy, 14 MB model size), underscore their direct applicability. The main challenge with these architectures, as indicated in Table II, can be the complexity of their manual design or the computational cost of automated Neural Architecture Search (NAS) methods used to discover them. The findings from the case studies summarized in Table III collectively demonstrate that these lightweight techniques are not merely theoretical constructs but have practical applicability across a wide spectrum of medical domains, including oncology (e.g., breast tomosynthesis [66], breast tumor detection [72]), dermatology (e.g., skin disease classification [67, 74]), pulmonology (e.g., pneumonia diagnosis [69]), cardiology (e.g., arrhythmia detection [70]), hematology (e.g., leukemia classification [78]), dentistry (e.g., dental X-ray segmentation [75]), and neurology (e.g., brain tumor classification [68, 76], Alzheimer's disease staging [76]). The ability to achieve high diagnostic accuracy (often comparable to larger models) with significantly reduced model sizes (e.g., down to 0.25 MB [5] or 4.4 MB [70]) is a critical enabler for on-device AI in healthcare. This allows for real-time feedback, reduced latency, and enhanced privacy by keeping sensitive patient data on the device or at the edge, rather than transmitting it to the cloud. The success of these approaches often hinges on a careful combination of techniques (e.g., pruning a lightweight architecture like MobileNetV2 [67] or quantizing a knowledge-distilled model) and tailoring them to the specific constraints and requirements of the target medical application and hardware platform. The development of deployment frameworks like TensorFlow Lite, PyTorch Mobile, and ONNX Runtime further streamlines this process by providing optimized toolchains for converting and executing these compressed models on a variety of edge devices.

8. OPEN CHALLENGES AND FUTURE DIRECTIONS

With the rapid integration of DL into IoT-based medical diagnosis, ensuring efficient, accurate, and resource-aware deployment remains a significant challenge. This section outlines key future research directions to address current limitations and enhance real-world applicability.

Energy-Aware and Hardware-Aware Pruning Although many pruning techniques reduce FLOPs and memory footprint, their actual impact on energy consumption is often insufficiently addressed. Therefore, future work should focus on energy-aware pruning that explicitly minimizes power consumption, which is crucial for IoT-based medical devices that operate under stringent energy constraints. Additionally, the development of hardware-friendly pruning algorithms that directly target deployment on edge accelerators such as FPGAs and ASICs is an emerging trend.

Low Bit-width Quantizationin Deeper Networks Quantizing deep neural networks to sub-4-bit precision remains a significant challenge, particularly for architectures with greater depth. Future research should prioritize the development of effective strategies for quantizing both weights and activations at extremely low bit-widths, such as binary or ternary levels, without compromising model accuracy or stability.

Teacher–Student Compatibility in Knowledge Distillation The absence of general guidelines for successfully matching instructor and student models is one of the biggest problems with knowledge distillation. The efficiency of the distillation has a significant impact on how congruent the two structures and capacities are. Incongruent architectures can make it more difficult to impart valuable knowledge, which mainly leads to worse performance, particularly in environments with limited resources. To make distillation pipelines more effective, reliable, and scalable, further work must be done to define capacity-aware model selection criteria, create adaptive transfer schemes, and derive automatic matching schemes.

Lightweight Privacy-Preserving Techniques Future research needs to address minimizing privacy problems in deep learning with emphasis on safeguarding model parameters and training. Despite the extensive computational resources provided by cloud platforms such as Google Cloud, Microsoft Azure, and Amazon SageMaker offer scalable deep learning services, effective deployment of privacy-preserving mechanisms has remained minimal. Previous work has proposed numerous methods to protect sensitive information; however, there are strong challenges. One major challenge is the tremendous computational burden that these privacy-preserving techniques entail, mainly attributed to the non-linear computations characteristic of deep learning models. The implication is that this can seriously hamper system efficiency as well as the practicability of real-world deployment. To mitigate these issues, future research needs to be focused on the design of light, privacy-aware deep learning models that are highly secure yet do not involve much computational complexity, hence suitable for real-world applications.

Memory Access Optimization In IoT-enabled medical devices, where off-chip memory operations use a lot more power than on-chip computations, lowering memory access overhead is essential to enabling energy-efficient deep learning. While newer methods like in-sensor computing and Processing-in-Memory (PIM) provide some partial answers, their efficacy is constrained by design limitations and hardware scalability. Future studies should investigate memory hierarchies and adaptive dataflow architectures that are tailored for sparsity and the unique properties of medical data in order to overcome these difficulties. Additionally, co-designing in-sensor neural networks and integrating non-volatile memory technology can significantly reduce energy consumption without compromising diagnostic accuracy, thus ensuring timely inference on medical edge devices with limited resources.

Automated Cross-Stack Optimization To elevate the performance of mobile deep learning technologies in intelligent healthcare applications, future research should explore automated cross-stack optimization techniques. While current approaches focus on optimizing individual layers—such as algorithms, hardware, or software—combining these optimizations holistically can yield better results. However, manually tuning across multiple layers is complex and time-consuming due to the variety of tools, frameworks, and hardware platforms. Developing intelligent compilers and optimization frameworks that can automatically identify and apply the best combination of techniques will simplify deployment and improve efficiency on resource-constrained medical devices.

9. CONCLUSION

This survey has systematically analyzed lightweight deep learning techniques for IoT-based medical diagnosis, examining four core approaches—model pruning, quantization, knowledge distillation, and lightweight architectures. Our analysis reveals that no single strategy universally dominates; optimal approaches depend on specific medical applications, data characteristics, and hardware constraints. Each technique presents distinct trade-offs: pruning offers parameter reduction but may lack hardware alignment; quantization provides memory savings but risks accuracy loss; distillation preserves performance but increases training complexity; while lightweight architectures offer efficiency but require significant design effort. Case studies across dermatology, oncology, pulmonology, and cardiology demonstrate the practical viability of these techniques, with successful implementations typically combining multiple optimization strategies. Deployment frameworks such as TensorFlow Lite and PyTorch Mobile play crucial roles in translating optimized models into production-ready IoT applications. Key research challenges remain: improving energy efficiency for battery-powered devices, developing privacy-preserving models, and establishing standardized benchmarks for medical edge AI. Future directions include exploring neuromorphic computing, co-designing algorithms with hardware, and integrating explainable AI techniques to build clinical trust. The field stands at a critical juncture where interdisciplinary collaboration between AI researchers, medical practitioners, and hardware engineers is essential to realize the full potential of robust, trustworthy, and clinically viable AI systems at the edge, ultimately enabling ubiquitous intelligent diagnostic capabilities that improve healthcare outcomes globally.

Conflicts of Interest

"The authors declare no conflicts of interest".

Acknowledgment

Thanks for Ninevah University for supporting this work.

References

[1] H. K. Bharadwaj, A. Agarwal, V. Chamola, N. R. Lakkaniga, V. Hassija, M. Guizani, and B. Sikdar, “A review on the role of machine learning in enabling iot based healthcare applications,” IEEE Access, vol. 9, pp. 38 859–38 890, 2021.

[2] Z. Alaaraji, A. Mutlag, and S. S. S. Ahmad, “Implement edge pruning to enhance attack graph generation using naïve approach algorithm,” El-Cezeri, vol. 11, no. 3, pp. 298–306, 2024.

[3] M. R. Islam, M. M. Kabir, M. F. Mridha, S. Alfarhood, M. Safran, and D. Che, “Deep learning-based iot system for remote monitoring and early detection of health issues in real-time,” Sensors, vol. 23, no. 11, p. 5204, 2023.

[4] Y. Chen, B. Zheng, Z. Zhang, Q. Wang, C. Shen, and Q. Zhang, “Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions,” ACM Computing Surveys, vol. 53, no. 4, pp. 1–37, 2020.

[5] F. M. Alabbasy, A. S. Abohamama, and M. F. Alrahmawy, “Compressing medical deep neural network models for edge devices using knowledge distillation,” Journal of King Saud University-Computer and Information Sciences, vol. 35, no. 7, p. 101616, 2023.

[6] TensorFlow Developers, “TensorFlow Lite,” https://tensorflow.google.org/lite/, 2024, accessed: 2025-06-10.

[7] Z. J. Al-Araji, S. S. Syed Ahmad, M. W. Al-Salihi, H. A. Al-Lamy, M. Ahmed, W. Raad, and N. Md Yunos, “Network traffic classification for attack detection using big data tools: A review,” Intelligent and Interactive Computing: Proceedings of IIC 2018, pp. 355–363, 2019.

[8] C. Luo, X. He, J. Zhan, L. Wang, W. Gao, and J. Dai, “Comparison and benchmarking of ai models and frameworks on mobile devices,” arXiv preprint arXiv:2005.05085, 2020.

[9] H. H. Mohamad Jawad, Z. Bin Hassan, B. B. Zaidan, F. H. Mohammed Jawad, D. H. Mohamed Jawad, and W. H. D. Alredany, “A systematic literature review of enabling iot in healthcare: Motivations, challenges, and recommendations,” Electronics, vol. 11, no. 19, p. 3223, 2022.

[10] R. Sindhuja et al., “A survey of internet of medical things (iomt) applications, architectures and challenges in smart healthcare systems,” in ITM Web of Conferences, vol. 56. EDP Sciences, 2023, p. 05013.

[11] M. Abughazalah, W. Alsaggaf, S. Saifuddin, and S. Sarhan, “Centralized vs. decentralized cloud computing in healthcare,” Applied Sciences, vol. 14, no. 17, p. 7765, 2024.

[12] A. K. Idrees, B. T. Hasan, and S. K. Idrees, “Deep learning for combating covid-19 pandemic in internet of medical things (iomt) networks: A comprehensive review,” Advanced AI and Internet of Health Things for Combating Pandemics, pp. 57–82, 2012.

[13] A. Rocha, M. Monteiro, C. Mattos, M. Dias, J. Soares, R. Magalhães, and J. Macedo, “Edge ai for internet of medical things: A literature review,” Computers and Electrical Engineering, vol. 116, p. 109202, 2024.

[14] B. T. Hasan and A. K. Idrees, “Federated learning for iot/edge/fog computing systems,” in Federated Learning. Apple Academic Press, 2024, pp. 47–75.

[15] N. N. Alajlan and D. M. Ibrahim, “Tinyml: Enabling of inference deep learning models on ultra-low-power iot edge devices for ai applications,” Micromachines, vol. 13, no. 6, p. 851, 2022.

[16] Z. J. Al-Araji, M. S. AlKhaldee, A. A. Mutlag, Z. A. Abdulkadhim, H. M. Farhood, S. S. S. Ahmad, N. N. Hikmat, A. Yassen, A. A. I. Al-Dulaimi, N. N. H. Al-Sheikh et al., “Healthcare security in edge-fog-cloud environment using blockchain: A systematic review,” Mesopotamian Journal of CyberSecurity, vol. 5, no. 2, pp. 606–635, 2025.

[17] N. Anjum, M. Alibakhshikenari, J. Rashid, F. Jabeen, A. Asif, E. M. Mohamed, and F. Falcone, “Iot-based covid-19 diagnosing and monitoring systems: A survey,” Ieee Access, vol. 10, pp. 87 168–87 181, 2022.

[18] F. M. Sallabi, H. M. Khater, A. Tariq, M. Hayajneh, K. Shuaib, and E. S. Barka, “Smart healthcare network management: A comprehensive review,” Mathematics, vol. 13, no. 6, p. 988, 2025.

[19] Y. Moayedi, F. Foroutan, Y. Gao, B. Kim, E. De Luca, M. Brum, D. H. Brahmbhatt, J. Duhamel, A. Simard, C. McIntosh et al., “Developments in digital wearable in heart failure and the rationale for the design of true-hf (ted rogers understanding of exacerbations in heart failure) apple cpet study,” Circulation: Heart Failure, p. e012204, 2025.

[20] G. Feng, S. Manimurugan, B. Yi, and Y. Feng, “Towards precision cardiac healthcare: Deep learning and iot integration for real-time monitoring and personalized diagnosis,” IEEE Internet of Things Journal, 2025.

[21] H. A. H. Baca and F. d. L. P. Valdivia, “Efficient deep learning-based arrhythmia detection using smartwatch ecg electrocardiograms,” Sensors, vol. 25, no. 17, p. 5244, 2025.

[22] L. Young, D. Wang, and N. Gong, “Low-cost wearable edge-ai device for diabetes management,” in Proceedings of the Great Lakes Symposium on VLSI 2025, 2025, pp. 238–244.

[23] V. Veeramachaneni, “Edge computing: Architecture, applications, and future challenges in a decentralized era,” Recent trends in computer graphics and multimedia technology, vol. 7, no. 1, pp. 8–23, 2025.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 25, 2012.

[25] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1–9.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[28] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in Proc. Int. Conf. Mach. Learn. (ICML), 2019, pp. 6105–6114.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020.

[30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10 012–10 022.

[31] I. Singh, G. Goyal, and A. Chandel, “Alexnet architecture based convolutional neural network for toxic comments classification,” Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 9, pp. 7547–7558, 2022.

[32] A. Agarwal, K. Patni et al., “Lung cancer detection and classification based on alexnet cnn,” in 2021 6th international conference on communication and electronics systems (ICCES). IEEE, 2021, pp. 1390–1397..

[33] A. Titoriya and S. Sachdeva, “Breast cancer histopathology image classification using alexnet,” in 2019 4th International conference on information systems and computer networks (ISCON). IEEE, 2019, pp. 708–712.

[34] D. Baig and M. Amjad, “Enhancing skin cancer detection using alexnet empowered transfer learning,” Medinformatics, 2023.

[35] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, “What do we need to build explainable ai systems for the medical domain?” arXiv preprint arXiv:1712.09923, 2017.

[36] M. B. Hossain and M. Shaban, “Low-complexity low-memory vgg models for accurate diagnosis of breast cancer,” in SoutheastCon 2024. IEEE, 2024, pp. 630–638.

[37] A. Faghihi, M. Fathollahi, and R. Rajabi, “Diagnosis of skin cancer using vgg16 and vgg19 based transfer learning models,” Multimedia Tools and Applications, vol. 83, no. 19, pp. 57 495–57 510, 2024.

[38] H. Naz and N. J. Ahuja, “Retinacare: Diabetic retinopathy detection with vgg16-based web application,” in 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO). IEEE, 2024, pp. 1–6.

[39] U. Bisht, A. S. Gurkha, S. Naudiyal, D. Rawat, P. Das, and A. Verma, “Histopathological image analysis for lung disease diagnosis: A vgg16- based approach,” in 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS). IEEE, 2024, pp. 1869–1874.

[40] S. U. Amin and M. S. Hossain, “Edge intelligence and internet of things in healthcare: A survey,” IEEE access, vol. 9, pp. 45–59, 2020.

[41] B. T. Hasan and D. B. Abdullah, “Real-time resource monitoring framework in a heterogeneous kubernetes cluster,” in 2022 Muthanna International Conference on Engineering Science and Technology (MICEST). IEEE, 2022, pp. 184–189.

[42] S. Al-Qudah and M. Yang, “Effective hybrid structure health monitoring through parametric study of googlenet,” AI, vol. 5, no. 3, pp. 1558–1574, 2024.

[43] L. Ma, H. Wu, and P. Samundeeswari, “Googlenet-al: A fully automated adaptive model for lung cancer detection,” Pattern Recognition, vol. 155, p. 110657, 2024.

[44] L. Balagourouchetty, J. K. Pragatheeswaran, B. Pottakkat, and G. Ramkumar, “Googlenet-based ensemble fcnet classifier for focal liver lesion diagnosis,” IEEE journal of biomedical and health informatics, vol. 24, no. 6, pp. 1686–1694, 2019.

[45] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” Advances in neural information processing systems, vol. 28, 2015.

[46] V. Raji, S. Maheswari, S. S. Dharinya, P. Chinniah, K. KS, and K. T. Raja, “An effective detection of brain tumor detection using resnet-50 model,”in 2024 International Conference on Advancement in Renewable Energy and Intelligent Systems (AREIS). IEEE, 2024, pp. 1–6.

[47] N. Appavu, “Improved retinopathy detection using resnet-50 optimized ai deep learning models,” in 2025 Eleventh International Conference on Bio Signals, Images, and Instrumentation (ICBSII). IEEE, 2025, pp. 1–6.

[48] V. Nithya, N. Mohanasundaram, and R. Santhosh, “An early detection and classification of alzheimer’s disease framework based on resnet-50,”Current Medical Imaging, vol. 20, no. 1, p. e250823220361, 2024.

[49] M. Latha, P. S. Kumar, R. R. Chandrika, T. Mahesh, V. V. Kumar, and S. Guluwadi, “Revolutionizing breast ultrasound diagnostics with efficientnetb7 and explainable ai,” BMC Medical Imaging, vol. 24, no. 1, p. 230, 2024.

[50] D. R. Raman, S. Nishanthi, and P. Babysha, “Diagnosis of diabetic retinopathy by using efficientnet-b7 cnn architecture in deep learning,” in 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS). IEEE, 2023, pp. 430–435.

[51] B. N. Patro, V. P. Namboodiri, and V. S. Agneeswaran, “Spectformer: Frequency and attention is what you need in a vision transformer,” in 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). IEEE, 2025, pp. 9543–9554.

[52] Y. Lyu, X. Yu, D. Zhu, and L. Zhang, “Classification of alzheimer’s disease via vision transformer: Classification of alzheimer’s disease via vision transformer,” in Proceedings of the 15th international conference on PErvasive technologies related to assistive environments, 2022, pp. 463–468.

[53] C. Playout, R. Duval, M. C. Boucher, and F. Cheriet, “Focused attention in transformers for interpretable classification of retinal images,” Medical Image Analysis, vol. 82, p. 102608, 2022.

[54] S. Aburass, O. Dorgham, J. Al Shaqsi, M. Abu Rumman, and O. Al-Kadi, “Vision transformers in medical imaging: a comprehensive review of advancements and applications across multiple diseases,” Journal of Imaging Informatics in Medicine, pp. 1–44, 2025.

[55] S. Chaudhary, W. Yang, and Y. Qiang, “Swin transformer for covid-19 infection percentage estimation from ct-scans,” in International conference on image analysis and processing. Springer, 2022, pp. 520–528.

[56] A. Hayat, “Breast cancer detection system from thermal images using swin transformer,” Journal Press India, vol. 3, no. 1, 2023.

[57] A. Mehar, M. Shah, and R. Sawant, “Image based lung disease detection: Comparing swin transformers and convnets,” in 2023 3rd Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2023, pp. 1–4.

[58] G. Menghani, “Efficient deep learning: A survey on making deep learning models smaller, faster, and better,” ACM Computing Surveys, vol. 55, no. 12, pp. 1–37, 2023.

[59] H. Cheng, M. Zhang, and J. Q. Shi, “A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[60] S. M. Raza, S. M. H. Abidi, M. Masuduzzaman, and S. Y. Shin, “Survey on the application with lightweight deep learning models for edge devices,” Authorea Preprints, 2025.

[61] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, “A comprehensive survey on model quantization for deep neural networks in image classification,” ACM Transactions on Intelligent Systems and Technology, vol. 14, no. 6, pp. 1–50, 2023.

[62] A. Musa, H. A. Kakudi, M. Hassan, M. Hamada, U. Umar, and M. L. Salisu, “Lightweight deep learning models for edge devices—a survey,”International Journal of Computer Information Systems and Industrial Management Applications, vol. 17, p. 18, 2025.

[63] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, “Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2017, pp. 129–137.

[64] D. Haase and M. Amthor, “Rethinking depthwise separable convolutions: How intra-kernel correlations lead to improved mobilenets,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2020, pp. 14 600–14 609.

[65] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile,"in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 2820–2828.

[66] R. K. Samala, H.-P. Chan, L. M. Hadjiiski, M. A. Helvie, C. Richter, and K. Cha, "Evolutionary pruning of transfer learned deep convolutional neural network for breast cancer diagnosis in digital breast tomosynthesis," Physics in Medicine & Biology, vol. 63, no. 9, p. 095005, 2018.

[67] K. Xiang, L. Peng, H. Yang, M. Li, Z. Cao, S. Jiang, and G. Qu, "A novel weight pruning strategy for lightweight neural networks with application to the diagnosis of skin disease," Applied Soft Computing, vol. 111, p. 107707, 2021.

[68] I. O. A. CONVOLUTIONAL, "Evaluating the usefulness of pruning techniques for brain tumor classification," Ph.D. dissertation, tilburg university.

[69] C.-P. Yang, J.-Q. Zhu, T. Yan, Q.-L. Su, and L.-X. Zheng, "Auxiliary pneumonia classification algorithm based on pruning compression,"Computational and Mathematical Methods in Medicine, vol. 2022, no. 1, p. 8415187, 2022.

[70] Y. Liu, J. Liu, Y. Tian, Y. Jin, Z. Li, L. Zhao, and C. Liu, "Pruned lightweight neural networks for arrhythmia classification with clinical 12-lead ecgs," Applied Soft Computing, vol. 154, p. 111340, 2024.

[71] Y. Liao, N. Yu, D. Tian, S. Li, and Z. Li, "A quantized cnn-based microfluidic lensless-sensing mobile blood-acquisition and analysis system,"Sensors, vol. 19, no. 23, p. 5103, 2019.

[72] M. Garifulla, J. Shin, C. Kim, W. H. Kim, H. J. Kim, J. Kim, and S. Hong, "A case study of quantizing convolutional neural networks for fast disease diagnosis on portable medical devices," Sensors, vol. 22, no. 1, p. 219, 2021.

[73] R. Zhang and A. C. Chung, "Medq: Lossless ultra-low-bit neural network quantization for medical image segmentation," Medical Image Analysis, vol. 73, p. 102200, 2021.

[74] Y. Guo, Z. Jia, J. Hu, and Y. Shi, "Fairquantize: Achieving fairness through weight quantization for dermatological disease diagnosis," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2024, pp. 329–338.

[75] S. Lin, X. Hao, Y. Liu, D. Yan, J. Liu, and M. Zhong, "Lightweight deep learning methods for panoramic dental x-ray image segmentation," Neural Computing and Applications, vol. 35, no. 11, pp. 8295–8306, 2023.

[76] Y. Jiang, X. Zhao, Y.Wu, and A. Chaddad, "A knowledge distillation-based approach to enhance transparency of classifier models," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 39, no. 17, 2025, pp. 17 653–17 661.

[77] O. Ukwandu, H. Hindy, and E. Ukwandu, "An evaluation of lightweight deep learning techniques in medical imaging for high precision covid-19 diagnostics," Healthcare Analytics, vol. 2, p. 100096, 2022.

[78] A. Batool and Y.-C. Byun, "Lightweight efficientnetb3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images," IEEE Access, vol. 11, pp. 37 203–37 215, 2023.

[79] A. M. Joshi, D. R. Nayak, D. Das, and Y. Zhang, "Lims-net: A lightweight multi-scale cnn for covid-19 detection from chest ct scans," ACM Transactions on Management Information Systems, vol. 14, no. 1, pp. 1–17, 2023.

[80] A. Abdusalomov, S. Mirzakhalilov, S. Umirzakova, O. Ismailov, D. Sultanov, R. Nasimov, and Y.-I. Cho, "Lightweight deep learning framework for accurate detection of sports-related bone fractures," Diagnostics, vol. 15, no. 3, p. 271, 2025.

[81] TensorFlow Developers, "TensorFlow," https://www.tensorflow.org, 2024, accessed: 2025-06-10.

[82] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., "TensorFlow: a system for Large-Scale machine learning," in 12th USENIX Symp. Operating Systems Design and Implementation (OSDI), 2016, pp. 265–283.

[83] PyTorch, "PyTorch: An open source machine learning framework," https://pytorch.org/, 2023, accessed: 2025-06-09.

[84] V. Shankar, "Edge ai: A comprehensive survey of technologies, applications, and challenges," in Proc. Int. Conf. Advanced Comput. Emerging Technol. (ACET), 2024, pp. 1–6.

[85] The ONNX Community, "ONNX - Open Neural Network Exchange," https://onnx.ai/, 2024, accessed: 2025-06-10.