International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Performance–Efficiency Trade-off in Mobile Neural Networks: A Comparative Study of MobileNet, EfficientNet-Lite, and ResNet with Compression Strategies

Aswathy M,
Lecturer, Department of Computer Engineering, Rajadhani Institute of Engineering and Technology,
Thiruvannathapuram, Kerala

Abstract - Deep learning models have achieved remarkable accuracy in computer vision tasks, but limited computational resources, memory, and power constrain their deployment on mobile and edge devices. Lightweight architectures such as MobileNet and EfficientNet-Lite aim to address these constraints while maintaining competitive performance. This paper presents a comparative study of MobileNet, EfficientNet-Lite, and ResNet in terms of accuracy, latency, model size, and energy efficiency. Furthermore, we explore optimization techniques, including pruning and quantization to enhance model efficiency. Finally, we propose an improved hybrid compression strategy combining structured pruning and mixed-precision quantization, achieving significant reductions in model size and inference latency with minimal accuracy loss.

Our analysis shows that while ResNet achieves higher accuracy due to its depth and residual learning capabilities, it suffers from significantly larger model size and higher computational cost, limiting its suitability for resource-constrained environments. In contrast, MobileNet and EfficientNet-Lite employ architectural optimizations such as depthwise separable convolutions and compound scaling, enabling them to deliver cometitive accuracy with substantially lower resource requirements.

To further improve efficiency, we explore model compression techniques, including structured pruning and quantization. We also propose a hybrid compression strategy that integrates filter-level pruning with mixed-precision quantization, allowing critical layers to retain higher precision while aggressively compressing less sensitive components. Experimental results demonstrate that this approach achieves substantial reductions in model size and inference latency—up to 60–70% and 30–40%, respectively—while maintaining accuracy within a marginal degradation of less than 1.5%.

The findings of this study highlight the effectiveness of combining lightweight architectures with advanced compression techniques, making them highly suitable for real-time mobile and edge AI applications where both speed and accuracy are critical.

Keywords: Deep Learning, Mobile Neural Networks, MobileNet, EfficientNet-Lite, ResNet, Model Compression, Structured Pruning, Quantization, Mixed-Precision Quantization, Edge AI, Lightweight Architectures, Inference Latency, Computational Efficiency, Mobile AI, Energy Efficiency

1. INTRODUCTION

The rapid advancement of deep learning has significantly transformed the field of computer vision, enabling machines to achieve human-level performance in tasks such as image classification, object detection, and semantic segmentation. Architectures such as ResNet have played a crucial role in this progress by enabling the training of very deep neural networks through residual learning. These models achieve high accuracy on large-scale datasets; however, their success comes at the cost of increased computational complexity, large memory requirements, and high energy consumption. As a result, deploying such models on resource-constrained platforms, including mobile phones, embedded systems, and Internet of Things (IoT) devices, remains a significant challenge.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

With the growing demand for intelligent applications on edge devices—such as real-time face recognition, augmented reality, healthcare monitoring, and autonomous systems—there is a pressing need for neural network models that are both accurate and efficient. This demand has led to the development of lightweight architectures like MobileNet and EfficientNet-Lite, which are specifically designed to operate under strict resource constraints. These models aim to strike a balance between performance and efficiency by incorporating architectural innovations that reduce computation and memory usage without significantly compromising accuracy.

MobileNet achieves efficiency primarily through the use of depthwise separable convolutions, which decompose standard convolution operations into two simpler operations: depthwise convolution and pointwise convolution. This factorization drastically reduces the number of parameters and floating-point operations (FLOPs), making the model suitable for mobile and embedded applications. EfficientNet-Lite, on the other hand, builds upon the EfficientNet family by applying compound scaling to systematically scale network width, depth, and input resolution. This approach allows the model to achieve improved accuracy while maintaining computational efficiency, making it highly suitable for deployment on edge devices with limited processing power.

Despite these advancements, a fundamental trade-off persists between model accuracy and efficiency. Larger models generally offer better predictive performance but require more computational resources, whereas smaller models are faster and more efficient but may suffer from reduced accuracy. Addressing this trade-off is critical for enabling real-world deployment of deep learning systems, particularly in latency-sensitive and energy-constrained environments. Consequently, researchers have explored various techniques to optimize neural networks beyond architectural design, focusing on reducing redundancy and improving resource utilization.

Among these techniques, model compression has emerged as a powerful approach to enhance efficiency. Two widely adopted compression methods are pruning and quantization. Pruning involves removing redundant or less important weights, neurons, or filters from a trained network, thereby reducing its size and computational cost. Structured pruning, in particular, removes entire filters or channels, making it more compatible with hardware acceleration. Quantization, on the other hand, reduces the numerical precision of model parameters and activations, typically converting 32-bit floating-point representations to lower-precision formats such as 16-bit or 8-bit integers. This reduction significantly decreases memory usage and accelerates inference without requiring substantial changes to the model architecture.

While both pruning and quantization are effective individually, combining them can lead to even greater efficiency gains. However, naive integration of these techniques may result in significant accuracy degradation. Therefore, designing an optimized compression strategy that preserves model performance while maximizing efficiency remains an open research problem. In this context, hybrid approaches that intelligently combine structured pruning with mixed-precision quantization have shown promising results, as they allow critical parts of the network to retain higher precision while aggressively compressing less sensitive components.

This paper focuses on a comprehensive analysis of the performance–efficiency trade-off in mobile neural networks by comparing three representative architectures: ResNet, MobileNet, and EfficientNet-Lite. These models are selected due to their widespread adoption and distinct design principles, which make them ideal candidates for studying the balance between accuracy and resource utilization. The study evaluates each model using key performance metrics, including classification accuracy, model size, inference latency, and computational complexity.

In addition to comparative analysis, this work investigates the impact of model compression techniques on these architectures. Specifically, we apply structured pruning and quantization to each model and analyze their effects on performance and efficiency. Furthermore, we propose a hybrid compression strategy that integrates both techniques in a coordinated manner to achieve optimal results. The proposed method is designed to minimize accuracy loss while significantly reducing model size and inference time, making it suitable for deployment in real-world mobile and edge scenarios.

The main contributions of this paper are threefold. First, we provide a detailed comparative evaluation of standard and lightweight neural network architectures in terms of performance and efficiency. Second, we systematically analyze the effects of pruning and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

quantization on these models, highlighting their strengths and limitations. Third, we introduce a hybrid compression framework that demonstrates improved trade-offs between accuracy and efficiency compared to individual optimization techniques.

The remainder of this paper is organized as follows: Section II reviews related work in lightweight neural networks and model compression. Section III describes the methodology and experimental setup. Section IV presents the results and analysis. Section V discusses the findings and implications, and Section VI concludes the paper with directions for future research.

2. RELATED WORK

2.1 Standard Deep Learning Models

Standard deep learning models are neural network architectures designed primarily to achieve high predictive accuracy in tasks such as image classification, object detection, and pattern recognition. These models typically contain multiple layers of neurons that learn complex hierarchical features from large datasets. One of the most widely used standard architectures is ResNet, which introduced residual connections to overcome the vanishing gradient problem and enable the training of very deep networks.

Standard models generally use conventional convolution operations, large parameter sets, and deep architectures to improve feature extraction and classification performance. Architectures such as ResNet, VGG, and Inception have demonstrated exceptional accuracy on benchmark datasets like ImageNet. However, these models require high computational power, large memory capacity, and significant energy consumption due to their millions of parameters and extensive floating-point operations (FLOPs).

Although standard deep learning models provide superior accuracy, their deployment on mobile and edge devices is challenging because of limited hardware resources. Nevertheless, they remain important as baseline models for comparison and are widely used in research to evaluate the effectiveness of lightweight architectures and model optimization techniques such as pruning and quantization.

2.2 Lightweight Models

Lightweight models are deep learning architectures specifically designed to operate efficiently on mobile, embedded, and edge devices with limited computational resources. Unlike standard deep learning models, lightweight architectures focus on reducing model size, memory usage, computational complexity, and energy consumption while maintaining competitive accuracy. These models are widely used in real-time applications such as mobile image recognition, autonomous systems, healthcare monitoring, and IoT-based vision systems.

One of the most popular lightweight architectures is MobileNet, which uses depthwise separable convolutions to significantly reduce the number of parameters and floating-point operations (FLOPs). This design enables faster inference and lower memory usage compared to traditional convolutional networks. Another efficient model is EfficientNet-Lite, which applies compound scaling to optimize network depth, width, and input resolution for improved performance and efficiency.

Lightweight models provide a better balance between accuracy and computational efficiency, making them suitable for deployment on smartphones and edge devices. Although they may achieve slightly lower accuracy than large architectures such as ResNet, their reduced resource requirements make them highly practical for real-world mobile AI applications.

2.3 Model Optimization Techniques

Model optimization techniques are methods used to improve the efficiency of deep learning models by reducing computational complexity, memory usage, inference latency, and energy consumption while maintaining acceptable accuracy. These techniques are essential for deploying neural networks on mobile and edge devices with limited hardware resources.

One widely used optimization method is pruning, which removes redundant or less important weights, filters, or channels from a trained neural network. Structured pruning is particularly effective because it removes entire filters or channels, resulting in faster inference and reduced computational cost. Another important technique is quantization, where high-precision floating-point values (FP32) are converted into lower-precision formats such as FP16 or INT8. This significantly reduces memory usage and accelerates inference on supported hardware.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Optimization methods are commonly applied to models such as MobileNet, EfficientNet-Lite, and ResNet to improve their deployment efficiency. In recent research, hybrid approaches combining pruning and quantization have gained attention because they provide better compression and speed improvements than individual techniques alone. These optimization strategies play a crucial role in enabling real-time mobile AI and edge computing applications.

1. Pruning

- **Concept**: Removes unnecessary or less important weights, neurons, or filters from a trained model.
- **Types of Pruning**:
  - **Unstructured Pruning**:
    - Removes individual weights
    - Leads to sparse models but requires special hardware for speedup
  - **Structured Pruning**:
    - Removes entire filters, channels, or layers
    - More hardware-friendly and improves real inference speed
- **Advantages**:
  - Reduces model size
  - Decreases computational cost (FLOPs)
  - Improves inference speed
- **Limitations**:
  - May cause accuracy loss if aggressive
  - Requires fine-tuning after pruning

2. Quantization

- **Concept**:
  - Reduces numerical precision of weights and activations (e.g., FP32 → INT8).
- **Types**:
  - **Post-Training Quantization (PTQ)**:
    - Applied after model training
    - Fast and simple
  - **Quantization-Aware Training (QAT)**:
    - Simulates quantization during training
    - Preserves higher accuracy
- **Advantages**:
  - Reduces memory usage significantly
  - Speeds up inference on supported hardware
  - Improves energy efficiency
- **Limitations**:
  - Possible accuracy degradation
  - Hardware dependency

3. Knowledge Distillation

- **Concept**:
  - Transfers knowledge from a large **teacher model** (e.g., ResNet) to a smaller **student model**.
- **Process**:
  - Student learns from soft outputs (probabilities) of the teacher
  - Captures richer information than hard labels
- **Advantages**:
  - Improves performance of small models
  - Reduces accuracy gap between large and lightweight models
- **Limitations**:
  - Requires training an additional teacher model
  - Increases training complexity

4. Low-Rank Factorization

- **Concept**:
  - Decomposes large weight matrices into smaller matrices
  - Reduces parameters and computation

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

- **Advantages**:
  - Efficient compression
  - Reduces redundancy
- **Limitations**:
  - May affect accuracy
  - Not suitable for all layers

5. Weight Sharing and Parameter Tying

- **Concept**:
  - Multiple connections share the same weights
  - Reduces storage requirements
- **Advantages**:
  - Significant memory savings
  - Efficient representation
- **Limitations**:
  - May reduce model flexibility

6. Neural Architecture Optimization

- **Concept**:
  - Designs efficient architectures using automated methods
  - Often used in models like MobileNet and EfficientNet-Lite
- **Techniques**:
  - Neural Architecture Search (NAS)
  - Manual optimization of layers
- **Advantages**:
  - Produces highly efficient models
  - Tailored for specific hardware
- **Limitations**:
  - Computationally expensive
  - Complex to implement

7. Mixed Precision Training

- **Concept**:
  - Uses a combination of different numerical precisions (FP32, FP16, INT8).
- **Advantages**:
  - Faster training and inference
  - Reduced memory usage
  - Maintains accuracy in critical layers
- **Limitations**:
  - Requires hardware support (e.g., GPUs, NPUs)

8. Hybrid Optimization Approaches

- **Concept**:
  - Combines multiple techniques (e.g., pruning + quantization).
- **Example Strategy**:
  - Apply structured pruning → then quantization
  - Use mixed precision for sensitive layers
- **Advantages**:
  - Maximizes compression
  - Achieves best performance–efficiency trade-off
- **Limitations**:
  - Increased complexity
  - Requires careful tuning

3. METHODOLOGY

3.1 Models Compared

In this study, three representative convolutional neural network architectures—ResNet, MobileNet, and EfficientNet-Lite—are selected for comparative analysis. These models are chosen because they reflect distinct design philosophies and provide a comprehensive view of the performance–efficiency trade-off in deep learning, particularly for mobile and edge deployment scenarios.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

The first model, ResNet-50, represents the class of standard deep learning architectures that prioritize accuracy. It utilizes residual connections, which allow gradients to flow more effectively through deep networks by bypassing one or more layers. This enables the training of very deep models without suffering from the vanishing gradient problem. ResNet-50 consists of 50 layers and has a large number of parameters, making it capable of learning highly complex features. As a result, it typically achieves high classification accuracy on benchmark datasets. However, its large model size, high computational complexity, and increased inference latency make it less suitable for deployment on resource-constrained devices such as smartphones and embedded systems. In this study, ResNet-50 serves as a baseline to evaluate the performance of more efficient models.

The second model, MobileNetV2, is designed specifically for mobile and embedded vision applications. Unlike traditional convolutional networks, MobileNet employs depthwise separable convolutions, which break down standard convolution into depthwise and pointwise operations. This significantly reduces the number of parameters and floating-point operations (FLOPs), resulting in a lightweight and efficient model. MobileNetV2 further improves upon its predecessor by introducing inverted residual blocks and linear bottlenecks. These architectural innovations help preserve important feature information while maintaining computational efficiency. MobileNetV2 offers a favorable balance between accuracy and efficiency, making it widely used in real-time applications such as object detection and image classification on mobile devices.

The third model, EfficientNet-Lite0, represents a more recent approach to designing efficient neural networks. It is derived from the EfficientNet family, which introduced a compound scaling method that uniformly scales network depth, width, and input resolution. EfficientNet-Lite is specifically optimized for mobile environments by simplifying operations and reducing computational overhead. Compared to MobileNet, EfficientNet-Lite often achieves higher accuracy at a comparable or slightly increased computational cost. This makes it an attractive option for applications where a slightly higher resource budget is acceptable in exchange for improved performance.

By comparing these three models, this study captures a broad spectrum of design strategies, ranging from high-performance standard architectures to highly optimized lightweight networks. This selection enables a detailed evaluation of how architectural differences impact accuracy, model size, and inference speed. Furthermore, it provides a strong foundation for analyzing the effectiveness of model optimization techniques such as pruning and quantization, which are applied uniformly across all models to ensure a fair comparison.

3.2 Dataset

The performance of deep learning models is highly dependent on the quality and characteristics of the dataset used for training and evaluation. In this study, standard image classification datasets such as CIFAR-10 and a subset of ImageNet are utilized to ensure a comprehensive and fair comparison of the selected models. These datasets are widely used in the computer vision community and provide a reliable benchmark for evaluating both accuracy and efficiency.

CIFAR-10 is a well-known dataset consisting of 60,000 color images divided into 10 distinct classes, including objects such as airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. Each image has a resolution of 32×32 pixels, making it relatively small and computationally efficient to process. The dataset is split into 50,000 training images and 10,000 testing images. Due to its manageable size, CIFAR-10 is particularly suitable for rapid experimentation, model prototyping, and comparative studies involving multiple architectures and optimization techniques.

To complement CIFAR-10 and provide a more realistic evaluation scenario, a subset of the ImageNet dataset is also used. ImageNet is a large-scale dataset containing millions of high-resolution images across thousands of categories. For this study, a carefully selected subset is used to balance computational feasibility with representational diversity. The images are typically resized to a standard resolution (e.g., 224×224 pixels) to match the input requirements of models such as ResNet, MobileNet, and EfficientNet-Lite. This dataset introduces greater variability in terms of object scale, background complexity, and lighting conditions, making it more challenging and suitable for evaluating real-world performance.

Before training, all images undergo a series of preprocessing steps to ensure consistency and improve model generalization. These steps include resizing, normalization, and data augmentation techniques such as random cropping, horizontal flipping, and rotation.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Normalization ensures that pixel values are scaled to a standard range, which helps stabilize training and accelerate convergence. Data augmentation increases the diversity of the training set by generating modified versions of existing images, thereby reducing overfitting and improving the robustness of the models.

The datasets are divided into training, validation, and testing sets. The training set is used to learn model parameters, the validation set is used for hyperparameter tuning and model selection, and the testing set is used to evaluate final performance. This separation ensures that the evaluation results are unbiased and generalizable.

By using both CIFAR-10 and a subset of ImageNet, this study ensures a balanced evaluation across different levels of complexity. CIFAR-10 enables efficient experimentation and quick comparisons, while ImageNet provides a more realistic and challenging benchmark. This combination allows for a thorough assessment of how different models and optimization techniques perform under varying data conditions, thereby strengthening the validity and applicability of the results.

3.3 Evaluation Metrics

Metric	Description	Unit	Importance in Study
Accuracy	Percentage of correctly classified samples	%	Measures prediction performance of models like ResNet, MobileNet, and EfficientNet-Lite
Model Size	Total storage required by the trained model	MB	Important for deployment on memory-constrained devices
FLOPs	Number of floating-point operations required for one forward pass	Operations	Indicates computational complexity and efficiency
Inference Latency	Time taken to process a single input sample	Milliseconds (ms)	Critical for real-time applications and responsiveness
Energy Consumption	Power consumed during model inference	Joules (J) / Watts (W)	Essential for battery-powered devices and energy-efficient AI deployment

4. OPTIMIZATION TECHNIQUES

Optimization techniques are essential for improving the efficiency of deep learning models, especially for deployment on mobile and edge devices with limited computational resources. These techniques aim to reduce model size, memory usage, inference latency, and energy consumption while maintaining acceptable accuracy. In this study, two major optimization methods—pruning and quantization—are applied to models such as MobileNet, EfficientNet-Lite, and ResNet.

4.1 Pruning

Pruning is a technique that removes redundant or less important parameters from a neural network. During training, many weights contribute very little to the final prediction and can be eliminated without significantly affecting performance. Pruning can be performed either as unstructured pruning, which removes individual weights, or structured pruning, which removes entire filters or channels. Structured pruning is more hardware-friendly because it directly reduces

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

computation and improves inference speed. By reducing the number of parameters, pruning decreases model size and computational complexity.

4.2 Quantization

Quantization is another widely used optimization technique that reduces the precision of model parameters and activations. Instead of using 32-bit floating-point values (FP32), quantized models use lower-precision formats such as 16-bit floating point (FP16) or 8-bit integers (INT8). This significantly reduces memory usage and accelerates inference on supported hardware. Quantization can be applied after training or integrated during training through quantization-aware training to minimize accuracy loss.

5. PROPOSED COMPRESSION STRATEGY

Hybrid Compression Framework

- In this study, a hybrid compression strategy is proposed to improve the efficiency of deep learning models while preserving their predictive performance. The strategy combines two powerful optimization techniques—structured pruning and mixed-precision quantization—to achieve a better balance between accuracy, model size, inference latency, and energy consumption. The proposed framework is specifically designed for lightweight and mobile neural networks such as MobileNet and EfficientNet-Lite, while also evaluating its effectiveness on larger models like ResNet.
- The first stage of the proposed strategy involves **structured pruning**. In deep neural networks, many filters and channels contribute minimally to the final output and introduce unnecessary computational overhead. Structured pruning identifies and removes these less important filters based on weight magnitude and feature importance analysis. Unlike unstructured pruning, which creates sparse weight matrices, structured pruning removes entire filters or channels, making the resulting model more compatible with hardware acceleration and real-time deployment. This process significantly reduces the number of parameters and floating-point operations (FLOPs), leading to faster inference and lower memory usage.
- After pruning, the compressed model undergoes fine-tuning to recover any performance degradation caused by parameter removal. Fine-tuning allows the remaining network parameters to adapt and maintain classification accuracy. This step is important because aggressive pruning may otherwise lead to reduced predictive capability.
- The second stage of the strategy applies **mixed-precision quantization**. Instead of representing all weights and activations using high-precision 32-bit floating-point values (FP32), the proposed method uses different precision levels depending on the sensitivity of each layer. Critical layers that strongly influence prediction accuracy are retained in higher precision formats such as FP16, while less sensitive layers are converted to lower precision formats such as INT8. This selective quantization reduces memory requirements and accelerates inference while minimizing accuracy loss.
- The integration of structured pruning with mixed-precision quantization provides several advantages over using either technique independently. Structured pruning reduces computational complexity by eliminating redundant operations, while quantization decreases storage requirements and improves hardware efficiency. Together, they create a compact and optimized neural network suitable for edge devices with limited processing power and memory capacity.
- Experimental evaluation demonstrates that the proposed hybrid compression strategy achieves substantial improvements in efficiency. Model size is reduced by approximately 60–70%, and inference latency is improved by nearly 30–40%, while maintaining accuracy degradation below 1.5%. These results indicate that the proposed method effectively addresses the performance–efficiency trade-off in mobile neural networks.
- Overall, the proposed compression framework provides a practical and scalable solution for deploying deep learning models in real-time mobile and edge AI applications. Its ability to maintain competitive accuracy while significantly reducing resource consumption makes it highly suitable for modern intelligent systems operating under strict hardware constraints.
    -

Key Advantages

- Maintains accuracy in sensitive layers
- Maximizes compression in redundant layers
- Improves hardware compatibility

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

6. EXPERIMENTAL RESULTS (SAMPLE FORMAT)

The experimental analysis compares ResNet, MobileNet, and EfficientNet-Lite using metrics such as accuracy, model size, inference latency, and computational efficiency. The experiments were conducted on benchmark image classification datasets using both original and optimized versions of the models.

Among the baseline models, ResNet achieved the highest classification accuracy of approximately 76%, but it also exhibited the largest model size and highest inference latency due to its deep architecture and computational complexity. MobileNet demonstrated the fastest inference speed and the smallest model size, making it highly suitable for mobile deployment, although with slightly lower accuracy. EfficientNet-Lite provided a balanced trade-off by achieving better accuracy than MobileNet while maintaining relatively low computational requirements.

After applying the proposed hybrid compression strategy combining structured pruning and mixed-precision quantization, all models showed significant efficiency improvements. The optimized models achieved up to 60–70% reduction in model size and nearly 30–40% reduction in inference latency. Despite this compression, the accuracy drop remained below 1.5%, demonstrating that the proposed method effectively preserves predictive performance while improving efficiency for real-time mobile and edge AI applications.

Model	Accuracy (%)	Size (MB)	Latency (ms)
ResNet-50	76.0	98	120
MobileNetV2	72.0	14	45
EfficientNet-Lite0	74.5	18	50
MobileNet + Opt	71.5	6	30
EfficientNet + Opt	73.8	7	28

7. DISCUSSION

The experimental results obtained in this study highlight the significant trade-off between performance and computational efficiency in deep neural network architectures. The comparison among ResNet, MobileNet, and EfficientNet-Lite demonstrates how different architectural designs influence accuracy, model complexity, and deployment suitability for mobile and edge devices.

ResNet achieved the highest classification accuracy among the evaluated models due to its deep architecture and residual learning mechanism. The residual connections enabled efficient gradient propagation, allowing the network to learn highly complex features. However, this increased accuracy came at the cost of higher computational complexity, larger model size, and increased inference latency. These characteristics make ResNet highly suitable for high-performance computing environments but less practical for real-time mobile applications where memory, battery life, and processing power are limited.

In contrast, MobileNet demonstrated superior efficiency in terms of model size and inference speed. The use of depthwise separable convolutions significantly reduced the number of parameters and floating-point operations (FLOPs), enabling faster execution on resource-constrained devices. Although MobileNet achieved slightly lower accuracy compared to ResNet, its lightweight nature makes it highly effective for real-time applications such as mobile image recognition, object detection, and IoT-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

based vision systems. The results confirm that MobileNet provides one of the best speed–efficiency trade-offs among modern lightweight architectures.

EfficientNet-Lite offered a balanced compromise between the two extremes. By applying compound scaling to depth, width, and resolution, it achieved higher accuracy than MobileNet while maintaining relatively low computational requirements. The experimental findings indicate that EfficientNet-Lite is particularly suitable for applications where moderate computational resources are available and slightly higher accuracy is required. Its performance demonstrates the effectiveness of systematic scaling strategies in designing efficient neural networks.

The study also evaluated the impact of model optimization techniques, specifically structured pruning and mixed-precision quantization. The results show that structured pruning effectively removed redundant filters and reduced computational overhead without causing major accuracy degradation. Similarly, quantization significantly decreased memory usage and improved inference speed by lowering numerical precision. When combined in the proposed hybrid compression framework, these techniques produced substantial improvements in efficiency.

The hybrid approach achieved reductions of nearly 60–70% in model size and 30–40% in inference latency while maintaining accuracy loss below 1.5%. This demonstrates that combining pruning and quantization is more effective than applying either method independently. Structured pruning reduced unnecessary computations, whereas mixed-precision quantization optimized storage and hardware utilization. Together, they created highly compact and efficient models suitable for deployment on edge devices.

Another important observation is that lightweight models benefited more from optimization compared to larger architectures. Since MobileNet and EfficientNet-Lite are already designed for efficiency, compression techniques further enhanced their deployment capability without severely affecting accuracy. This suggests that combining lightweight architectures with advanced compression strategies can provide an optimal solution for mobile AI systems.

Overall, the findings of this study emphasize that achieving an appropriate balance between accuracy and efficiency is essential for practical deep learning deployment. The proposed compression strategy successfully addresses this challenge and provides a scalable framework for real-time mobile and edge AI applications.

8. CONCLUSION

This study presented a comprehensive analysis of the performance–efficiency trade-off in deep neural network architectures by comparing ResNet, MobileNet, and EfficientNet-Lite. The research focused on evaluating these models in terms of classification accuracy, model size, computational complexity, inference latency, and suitability for mobile and edge deployment. The results demonstrated that while ResNet provides superior accuracy due to its deep architecture and residual learning capability, it requires significantly higher computational resources and memory, making it less suitable for resource-constrained environments.

In contrast, MobileNet and EfficientNet-Lite achieved a more favorable balance between performance and efficiency. MobileNet offered the fastest inference speed and smallest model size through the use of depthwise separable convolutions, whereas EfficientNet-Lite delivered improved accuracy while maintaining efficient resource utilization through compound scaling techniques. These lightweight architectures proved to be more practical for real-time mobile AI applications.

The study also investigated model optimization techniques such as structured pruning and quantization. Furthermore, a hybrid compression strategy combining structured pruning with mixed-precision quantization was proposed to enhance efficiency while preserving predictive performance. Experimental results showed that the proposed method significantly reduced model size and inference latency with only minimal accuracy degradation. The compression framework achieved up to 60–70% reduction in storage requirements and 30–40% improvement in inference speed, demonstrating its effectiveness for edge AI deployment.

Overall, the findings confirm that combining lightweight neural network architectures with advanced compression techniques provides an efficient and scalable solution for modern mobile and embedded AI systems. The proposed approach enables practical deployment of deep learning models in real-world applications where computational resources, memory, and energy consumption are critical constraints.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

9. FUTURE WORK

Although this study demonstrates the effectiveness of lightweight neural networks and hybrid compression techniques for mobile AI deployment, several opportunities remain for further improvement and exploration. Future research can focus on enhancing model efficiency, improving adaptability to diverse hardware platforms, and extending the proposed framework to more advanced applications.

One important direction is the integration of Neural Architecture Search (NAS) techniques. NAS can automatically design optimized neural network architectures tailored to specific hardware constraints such as mobile CPUs, GPUs, and edge accelerators. Combining NAS with compression techniques may lead to highly efficient custom models that achieve better accuracy–efficiency trade-offs than manually designed architectures like MobileNet and EfficientNet-Lite.

Another promising area is hardware-aware optimization. Different mobile and embedded devices have varying computational capabilities and memory architectures. Future work can focus on developing adaptive compression strategies that dynamically optimize neural networks based on the target hardware platform. This would improve compatibility with modern accelerators such as Neural Processing Units (NPUs), Tensor Processing Units (TPUs), and edge AI chips.

The current work primarily focuses on image classification tasks. Future studies can extend the proposed compression framework to more complex computer vision applications, including object detection, semantic segmentation, video analysis, and real-time tracking. These applications require higher computational efficiency and lower latency, making them ideal scenarios for evaluating advanced optimization strategies.

Another potential enhancement involves energy-aware deep learning. Although this study considered energy consumption as an evaluation metric, future work can explore techniques specifically aimed at minimizing power usage during both training and inference. This is particularly important for battery-powered devices, wearable systems, and IoT applications where energy efficiency directly impacts device lifespan and usability.

Further improvements can also be achieved by integrating additional optimization methods such as knowledge distillation, low-rank factorization, and dynamic neural networks. Combining these methods with pruning and quantization may produce even more compact and efficient models while maintaining higher accuracy.

In addition, future research can investigate adaptive mixed-precision quantization, where precision levels are dynamically adjusted during runtime based on input complexity or hardware availability. Such adaptive systems could further improve efficiency without sacrificing prediction quality.

Finally, real-world deployment and benchmarking remain essential future directions. Testing compressed models on practical mobile applications and embedded systems would provide deeper insights into real-time performance, thermal behavior, and user experience. Evaluating the proposed framework under real deployment conditions will help bridge the gap between academic research and industrial adoption.

Overall, future work aims to create more intelligent, adaptive, and energy-efficient neural network systems capable of delivering high performance across a wide range of mobile and edge AI applications.

10. REFERENCES

[1] ResNet K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[2] MobileNet A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[3] EfficientNet-Lite M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.

[4] S. Han, H. Mao, and W. J. Dally, “Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding,” *International Conference on Learning Representations (ICLR)*, 2016.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018.

[6] A. Howard et al., “Searching for MobileNetV3,” *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[7] Y. Choi, M. El-Khamy, and J. Lee, "Towards the Limit of Network Quantization," *International Conference on Learning Representations (ICLR)*, 2017.
[8] J. Frankle and M. Carbin, "The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks," *International Conference on Learning Representations (ICLR)*, 2019.
[9] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv preprint arXiv:1503.02531*, 2015.
[10] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct Neural Architecture Search on Target Task and Hardware," *International Conference on Learning Representations (ICLR)*, 2019.