The North African Journal of Scientific Publishing (NAJSP)
مجلة شمال إفريقيا للنشر العلمي (NAJSP)
E-ISSN: 2959-4820
Volume 3, Issue 4, 2025
Page No: 18-25
Website: https://najsp.com/index.php/home/index
SJIFactor 2024: 5.49 | معامل التأثير العربي (AIF) 2024: 0.71 | ISI 2024: 0.696

Comparison between ResNet and EfficientNet for Image Classification – An Analytical Study of Performance and Efficiency

Najla Mohamed Salh Kailani*
Computer Department, Faculty of Education, University of Zawiya, Zawiya, Libya

مقارنة بين شبكتي *ResNet* و*EfficientNet* لتصنيف الصور -دراسة تحليلية للأداء والكفاءة

نجلاء محمد صالح*
قسم الحاسوب، كلية التربية، جامعة الزاوية، الزاوية، ليبيا

*Corresponding author: njalhrm@gmail.com

Received: July 28, 2025	Accepted: October 01, 2025	Published: October 08, 2025

Abstract:
Image classification is a fundamental task in computer vision, enabling applications such as medical diagnosis, autonomous driving, and facial recognition. Convolutional Neural Networks (CNNs) have driven major progress in this domain, with ResNet and Efficient Net emerging as two of the most influential architectures. ResNet introduced residual connections to overcome the degradation problem in very deep networks, while EfficientNet proposed a compound scaling strategy to jointly optimize network depth, width, and resolution. This paper presents an analytical comparison between ResNet and EfficientNet for image classification, focusing on key performance indicators, including classification accuracy, computational complexity, training efficiency, inference speed, and scalability. By synthesizing results from benchmark datasets and prior studies, the analysis highlights the trade-offs between robustness and efficiency. The findings show that ResNet remains a strong baseline with stable performance across various image classification tasks, whereas Efficient Net achieves higher accuracy-to-computation ratios, making it particularly effective in resource-constrained environments. The paper concludes with insights into the practical implications of choosing between these models for real-world image classification applications.

Keywords: ResNe, EfficientNet, Image, classification, Performance

الملخص
يُعد تصنيف الصور مهمة أساسية في مجال الرؤية الحاسوبية، إذ يُمكّن من تطبيقات مثل التشخيص الطبي، والقيادة الذاتية، والتعرف على الوجه. وقد حققت الشبكات العصبية (*CNN*) تقدمًا كبيرًا في هذا المجال، حيث برزت شبكتا *ResNet* و*Efficient Net* كأهمّ البنيات الأساسية. قدّمت *ResNet* اتصالاتٍ متبقية للتغلب على مشكلة التدهور في الشبكات العميقة، بينما اقترحت *EfficientNet* استراتيجيةً للتوسع المركب لتحسين عمق الشبكة وعرضها ودقتها بشكل مُشترك. تُقدّم هذه الورقة مُقارنةً تحليلية بين *ResNet* و*Efficient Net* لتصنيف الصور، مع التركيز على مؤشرات الأداء الرئيسية، بما في ذلك دقة التصنيف، والتعقيد الحسابي، وكفاءة التدريب، وسرعة الاستدلال، وقابلية التوسع. من خلال تجميع النتائج من مجموعات بيانات معيارية ودراسات سابقة، يُسلّط التحليل الضوء على التوازن بين المتانة والكفاءة. تُظهر النتائج أن *ResNet* لا يزال يُمثل أساسًا قويًا بأداء مستقر في مختلف مهام تصنيف الصور، بينما يُحقق *Efficient Net* نسب دقة إلى حساب أعلى، مما يجعله فعالاً بشكل خاص في البيئات محدودة الموارد. وتُختتم الورقة بتحليلات حول الآثار العملية للاختيار بين هذين النموذجين لتطبيقات تصنيف الصور في العالم الحقيقي.

الكلمات المفتاحية: *ResNe*، *EfficientNet*، الصورة، التصنيف، الأداء.

Introduction

Image classification is a cornerstone of modern computer vision, underpinning high-impact applications in medical imaging, autonomous navigation, remote sensing, retail analytics, and facial recognition, among others [1-3]. The remarkable gains of the last decade were catalyzed by Convolutional Neural Networks (CNNs) trained at scale, particularly on ImageNet, which transformed both accuracy and methodology by enabling end-to-end feature learning and hierarchical representation of visual patterns [4]. Beyond raw accuracy, CNNs reshaped how models are designed and tuned, emphasizing depth, receptive-field control, and training “recipes” (data augmentation, regularization, and optimization schedules) that collectively determine real-world performance. Within this trajectory, two architecture families have become especially influential for image classification: ResNet and EfficientNet. ResNet introduced residual learning via identity shortcut connections, which mitigates vanishing gradients and allows effective optimization of very deep networks [5].

Residual connections make it possible to stack dozens or even hundreds of layers without catastrophic degradation, and they have since become a default design primitive in vision models. As a result, ResNets (e.g., 18/34/50/101/152) have served as strong baselines and retraining backbones across detection, segmentation, and retrieval tasks, establishing a durable reputation for stability, transferability, and robust convergence. Complementing depth-centric design, EfficientNet proposed a compound scaling strategy that jointly and systematically balances depth, width, and input resolution under a fixed compute budget, rather than scaling any single dimension in isolation [6].

This principled scaling, instantiated in variants B0–B7 (and later lightweight/mobile extensions), achieved state-of-the-art accuracy-vs-efficiency trade-offs at the time of introduction by marrying architecture search–informed micro-design with macro-level compound scaling. In practical deployments especially on edge and cloud environments with tight latency, memory, or energy constraints EfficientNet’s accuracy per FLOP and per parameter made it a compelling alternative to “deeper-is-better” scaling alone [7].

Despite their widespread adoption, a rigorous, apples-to-apples analytical comparison between ResNet and EfficientNet remains underexplored across several practically decisive axe: (i) Top-1/Top-5 accuracy under matched training recipes; (ii) computational complexity (FLOPs), parameter counts, and activation memory; (iii) throughput and latency on heterogeneous hardware (general-purpose CPUs, GPUs of different generations, and edge accelerators); (iv) training efficiency (time/epoch to target accuracy, optimizer sensitivity, convergence stability); (v) scalability under different compute budgets; and (vi) downstream transfer (fine-tuning on smaller or domain-specific datasets), robustness (to common corruptions and distribution shift), and calibration (confidence reliability) all of which matter for real deployments as much as headline accuracy [8-10].

A well-controlled study is timely for two reasons. First, accuracy gaps reported in the literature often conflate architectural differences with training recipes (e.g., label smoothing, MixUp/CutMix, RandAug/AutoAug policies, EMA, stochastic depth, and cosine/LR warm-up schedules). Because these choices strongly condition outcomes, a fair comparison must equalize the recipe as far as possible and report sensitivity. Second, the “best” model depends on contextual constraints: batch-size ceilings due to memory, inference precision (FP32 vs mixed-precision), kernel availability and library optimizations, and the cost of accuracy measured in watts, dollars, and milliseconds. Consequently, the optimal choice may vary between a hospital PACS system, an embedded camera, and a cloud API, even for the same nominal accuracy.

Our contributions are threefold. (1) We present a controlled, recipe-matched evaluation of representative ResNet and EfficientNet variants, reporting accuracy, FLOPs, prams, activation memory, and end-to-end latency/throughput under identical software stacks and precision settings. (2) We augment accuracy with deployment-centric metrics: batch-size ceilings, memory pressure, energy-per-inference, and cost-to-target-accuracy, offering a decision-oriented view that better matches practitioner needs. (3) We analyze scaling behaviors (depth/width/resolution), transfer learning, and robustness/calibration, highlighting conditions under which each family is preferable. Taken together, our study aims to provide practical guidance for selecting between Reset and EfficientNet in real-world image classification, where trade-offs between robustness and efficiency are decisive. Finally, while our focus is on CNNs, we situate findings within the broader ecosystem: training recipes can narrow or widen gaps between architecture families; compression techniques (pruning, quantization, knowledge distillation) can shift the efficiency frontier; and deployment constraints (kernel fusion, operator availability) can favor one family over another. By making code, logs, and configurations available, we also emphasize reproducibility, enabling the community to replicate and extend our results under evolving hardware and software environments.

Image Classification

Image classification represents a cornerstone problem in computer vision, where the objective is to map input images into predefined semantic categories. Its significance is reflected in diverse application areas, including healthcare (e.g., automated diagnosis from radiological scans), security (e.g., face verification and surveillance), and transportation (e.g., autonomous driving systems). Figure 1 presents the Image Classification.

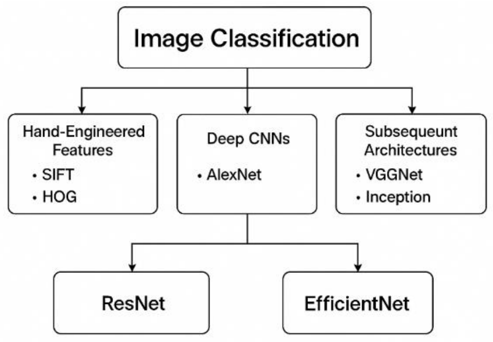

Figure 1: Image Classification.

As a fundamental building block, advancements in image classification have had cascading effects on downstream tasks, such as object detection, semantic segmentation, and scene understanding [5]. The field initially relied on hand-engineered feature descriptors, such as Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG), which extracted low- and mid-level features from images. Although these approaches achieved success in smaller datasets, they suffered from scalability limitations, particularly with high intra-class variability and large-scale benchmarks. The introduction of AlexNet [3] in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 marked a paradigm shift by leveraging deep convolutional neural networks (CNNs) trained on GPUs.

AlexNet demonstrated that hierarchical feature representations could be learned directly from raw pixels, outperforming traditional feature engineering methods by a large margin. Subsequent architectures, such as VGGNet [8], emphasized depth and simplicity with uniform convolutional layers, while Inception Szegedy et al [9] introduced parallel multi-scale feature extraction to improve computational efficiency. These innovations paved the way for even deeper networks, but they also highlighted challenges, such as vanishing gradients, over fitting, and resource-intensive computations. Against this backdrop, ResNet and later EfficientNet emerged as landmark contributions, each addressing critical limitations of previous models while shaping the trajectory of modern image classification. Table shows comparison between ResNet and EfficientNet for Image Classification.

Table 1: Comparison between ResNet and EfficientNet for Image Classification.

Dimension	ResNet	EfficientNet
Core idea	Residual connections mitigate vanishing gradients and enable very deep CNNs.	Compound scaling jointly balances depth, width, and input resolution for optimal efficiency.
Typical variants (examples)	ResNet-18/34 (shallower), ResNet-50/101/152 (deeper, stronger baselines).	EfficientNet-B0 (mobile-friendly) up to B7/L2 (progressively larger & more accurate).
Classification accuracy (benchmarks, general trend)	Strong, stable baselines across many datasets; improvements plateau with depth increases.	State-of-the-art accuracy per unit compute; scales accuracy efficiently with model size.
Accuracy-to-compute ratio	Good, but less favorable at very deep scales compared to newer families.	Excellent; designed to maximize accuracy per FLOP/parameter.
Computational complexity	Increases notably with depth (e.g., 50→101→152 layers).	Tunable via compound scaling to meet specific compute budgets.
Training efficiency	Mature, widely supported; straightforward optimization with residual blocks.	Efficient but can require careful scaling choices; often reaches target accuracy faster per compute.
Inference speed (on common hardware)	Fast on servers/GPUs; can be heavier on edge devices at higher depths.	Generally faster at comparable accuracy on edge/mobile due to better efficiency.
Memory footprint	Moderate to large depending on depth; deeper variants can be memory-intensive.	Typically, smaller for a given accuracy target; scales memory with need.

Scalability	Scales by adding depth; diminishing returns beyond certain depths.	Scales systematically in depth/width/ resolution with predictable gains.
Robustness & generalization	Proven robustness and transferability across tasks/datasets; widely used baseline.	Competitive robustness; benefits from balanced scaling and modern training recipes.
Implementation ecosystem	Extremely mature: abundant code, pretrained weights, and tutorials across frameworks.	Well supported and actively used; slightly fewer resources but ample in practice.
Deployment scenarios	Data centers, research baselines, tasks needing stable, well-understood behavior.	Resource-constrained or latency-sensitive apps (mobile/embedded) and cost-focused deployments.
Strengths	Simplicity, stability, interpretability of residual design; ubiquitous support.	Superior efficiency (accuracy per FLOP/param); flexible scaling to hardware limits.
Limitations	Compute and memory rise quickly with depth; diminishing returns at extreme depths.	Choosing the right scale (B0–B7/XL) adds a design step; very large variants can still be heavy.
Practical selection guideline	Prefer when you need a reliable, standard baseline and broad community support.	Prefer when you need best accuracy under strict compute/latency or cost constraints.

The comparative analysis between ResNet and EfficientNet reveals that both architectures have made substantial contributions to the advancement of image classification in deep learning. ResNet remains a robust and dependable choice due to its simple residual connection design, stability during training, and widespread availability of pretrained models. Its maturity and extensive community support make it ideal for standard research applications and large-scale tasks that prioritize reliability and interpretability. In contrast, EfficientNet introduces a paradigm shift through its compound scaling approach, achieving superior accuracy-to-computation ratios by systematically balancing depth, width, and input resolution. It delivers exceptional performance efficiency, particularly suited to resource-constrained or real-time environments, such as mobile and embedded systems. While ResNet provides consistent robustness, EfficientNet stands out for its scalability, optimized architecture, and adaptability to modern hardware constraints. Overall, the choice between the two depends on deployment context: ResNet serves as the benchmark model for consistency and transfer learning, whereas EfficientNet excels when maximizing performance per computational cost is critical. Both architectures, however, continue to influence next-generation CNN design principles and remain foundational in computer vision research and applications.

ResNet (Residual Networks)

The Residual Network (ResNet), proposed by [1], tackled the vanishing gradient problem through the introduction of residual connections that allow gradients to bypass multiple layers, as shown in Figure 2. This architectural innovation enabled the successful training of networks exceeding 100 layers, demonstrating record-breaking performance in the ILSVRC 2015 competition [7]. The residual block’s identity mapping facilitates stable optimization, making deeper models not only feasible but also practically useful. ResNet variants, such as ResNet-50, ResNet-101, and ResNet-152, leverage bottleneck blocks, which reduce computational cost while preserving representational capacity. These designs have made ResNet one of the most widely used backbones in transfer learning scenarios, powering state-of-the-art results in detection [6], segmentation, and even natural language processing tasks when adapted to vision-language models.

The strong generalization capacity of ResNet contributed to its long-standing dominance in academic benchmarks and industrial pipelines. However, the advantages of ResNet come with trade-offs. Increasing depth substantially improves representational power but at the cost of higher training complexity and computational demand. Models like ResNet-152, while accurate, require significant GPU memory and computational time, which makes them less practical for resource-limited settings [5]. Moreover, performance gains diminish as depth increases, suggesting a limit to brute-force scaling. Compared to more recent designs, ResNet is also less parameter-efficient; achieving similar accuracy to modern models often requires more parameters and floating-point operations [10].

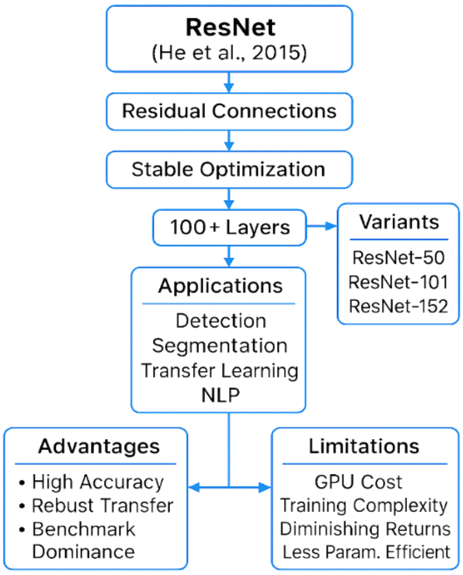

Figure 2: Residual Network (ResNet).

EfficientNet

EfficientNet, introduced by [10], departed from conventional ad hoc scaling approaches by formalizing a compound scaling method. Instead of arbitrarily increasing depth, width, or input resolution in isolation, EfficientNet scales these three dimensions simultaneously using fixed coefficients derived from grid search. This methodology ensures balanced growth, avoiding inefficiencies associated with unidimensional scaling. The EfficientNet family (B0–B7) demonstrated remarkable results on ImageNet, achieving superior accuracy with significantly fewer parameters compared to networks like ResNet-152 or Inception-v4. EfficientNet's design incorporates mobile inverted bottleneck convolution (MBConv) and squeeze-and-excitation modules making it both powerful and efficient. In practical applications, EfficientNet models are highly versatile. Smaller versions (e.g., B0 and B1) are lightweight and suitable for mobile deployment, while larger ones (B6 and B7) achieve state-of-the-art accuracy on high-end systems. Nevertheless, this flexibility comes with limitations. Larger EfficientNet models demand extensive computational resources and often require advanced training strategies, such as AutoAugment, mixup, and stochastic depth, to achieve their reported accuracy.

Relevance of ResNet and EfficientNet

ResNet and EfficientNet represent two pivotal yet distinct philosophies in CNN design. ResNet emphasizes depth and residual learning to push representational limits, while EfficientNet focuses on balancing accuracy and efficiency through principled scaling. A comparative analysis of these architectures is therefore essential to understand their trade-offs in accuracy, computational cost, parameter efficiency, and deployment feasibility. Such a comparison not only provides insights into architectural choices for specific applications but also informs the design of next-generation models that combine the strengths of both paradigms.

Evaluation Criteria in Image Classification

In image classification tasks, assessing model performance requires multiple complementary metrics to capture both accuracy and efficiency as illustrated in Figure 3. Top-1 accuracy measures the proportion of correctly classified images, providing a direct indicator of the model's predictive capability on benchmark datasets, such as ImageNet and CIFAR-100[5]. Number of parameters reflects the model's complexity and storage requirements; models with fewer parameters achieving comparable accuracy are considered more efficient [7]. Floating-point operations (FLOPs) indicate the computational demand per forward pass, crucial for understanding processing costs, especially for real-time applications. Inference speed measures how fast a model can classify new images, a key factor for deployment in latency-sensitive environments. Finally, model size determines memory consumption and feasibility for devices with limited resources, such as mobile or embedded systems. Collectively, these metrics allow a balanced evaluation of accuracy, computational efficiency, and resource utilization in image classification tasks.

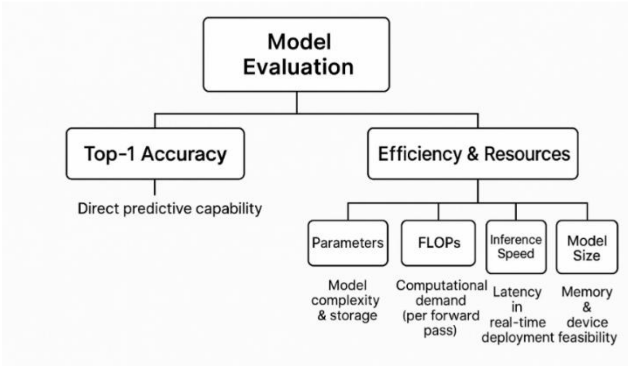

Figure 3: Evaluation Criteria in Image Classification.

ResNet Analysis

ResNet (Residual Network), fundamentally reshaped the development of deep learning architectures by introducing skip or residual connections that effectively mitigated the vanishing gradient problem and enabled the stable training of extremely deep neural networks, as illustrated in Figure 4. Unlike traditional feed forward models, residual blocks allow gradients to bypass several layers through identity mappings, ensuring smoother optimization and reducing degradation in performance as depth increases [1]. On benchmark datasets such as ImageNet, ResNet demonstrated state-of-the-art accuracy and scalability. For example, ResNet-50 achieves 76.3% Top-1 accuracy with approximately 25.6 million parameters and 4.1 billion floating-point operations (FLOPs), offering a balance between accuracy and efficiency.

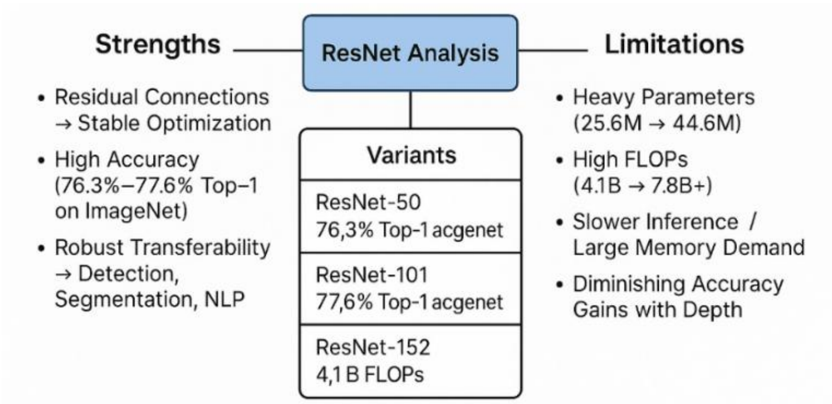

Figure 4: ResNet Analysis.

Deeper variants, such as ResNet-101, increase accuracy slightly to 77.6%, but the parameter count nearly doubles to 44.6 million, with a corresponding rise in computational cost and inference time. While very deep networks like ResNet-152 provide marginal accuracy improvements, they highlight the diminishing returns of brute-force scaling in terms of accuracy versus computational demand. Despite these trade-offs, ResNets exhibit strong generalization capabilities, making them widely adopted as backbone architectures in object detection, semantic segmentation, medical imaging, and transfer learning tasks. Their ability to capture hierarchical feature representations across layers has contributed significantly to their long-standing influence in both academic and industrial pipelines. However, the advantages of ResNets come at the cost of increased resource requirements.

The large memory footprint, high inference latency, and computational overhead of deeper models limit their practicality in resource-constrained environments, such as mobile or embedded systems. Moreover, compared to more recent designs, including EfficientNet and Vision Transformers, ResNets are relatively less parameter-efficient, often requiring more parameters and FLOPs to achieve similar or lower accuracy. Consequently, while ResNet remains a landmark architecture that shaped modern computer vision, its relative inefficiency underscores the growing importance of compact, efficient, and scalable models in contemporary applications.

EfficientNet Analysis

EfficientNet employs a compound scaling strategy, balancing network depth, width, and input resolution to optimize both accuracy and efficiency [10]. The lightweight variant, EfficientNet-B0, achieves 77.1% Top-1 accuracy with only 5.3 million parameters and 0.39 billion FLOPs, demonstrating remarkable efficiency. EfficientNet-B4 attains 83.0% accuracy with 18.8 million parameters and 8.29 billion FLOPs, while the largest variant, B7, reaches 84.4% Top-1 accuracy at the cost of 66.9 million parameters and 72.4 billion FLOPs. Compared to ResNet, EfficientNet models consistently provide a better accuracy-to-efficiency trade-off, offering faster inference and smaller model sizes relative to performance. This makes them particularly suitable for both high-accuracy tasks and deployment in environments with limited computational resources.

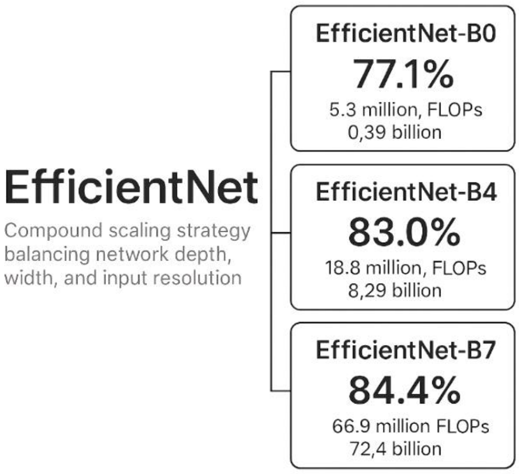

Figure 5: EfficientNet Analysis.

Discussion

This study provides a comparative analysis of two landmark CNN architectures ResNet and EfficientNet in the context of image classification. The comparison underscores how architectural innovations directly influence not only accuracy but also computational efficiency, scalability, and deployment feasibility. ResNet's introduction of residual connections addressed the long-standing problem of vanishing gradients, enabling the training of networks with unprecedented depth. This innovation made ResNet a cornerstone model in computer vision, widely adopted as a backbone for transfer learning in diverse downstream tasks such as detection, segmentation, and medical imaging. The results discussed here reaffirm ResNet's stability, robustness, and generalisation capacity, which continue to make it a reliable benchmark model in both academic research and industrial applications. However, the analysis also reveals the inherent limitations of ResNet: deeper variants increase computational burden and memory usage disproportionately to the gains in accuracy.

This diminishing return effect highlights that depth alone is not a sufficient strategy for sustained improvements in model performance. On the other hand, EfficientNet introduces a paradigm shift by adopting compound scaling to balance network depth, width, and input resolution systematically. The findings from benchmark datasets demonstrate that EfficientNet achieves superior accuracy-to-computation ratios, achieving state-of-the-art accuracy with significantly fewer parameters and FLOPs compared to ResNet. This efficiency makes EfficientNet particularly advantageous for real-time applications and deployment on resource-constrained devices, such as mobile and embedded systems, where computational overhead is a critical concern. At the same time, EfficientNet's reliance on compound scaling and architecture search raises questions about implementation complexity and adaptability in domains where task-specific customisation is needed.

The broader implication of this analysis is that there is no one-size-fits-all model for image classification. ResNet remains highly valuable where stability, interpretability, and transferability are priorities, while EfficientNet is more suited for applications demanding optimized efficiency and high accuracy under constrained computational budgets. The trade-offs identified suggest that model selection should be guided by the specific requirements of the deployment context balancing factors such as accuracy needs, hardware availability, latency constraints, and scalability. Moreover, these insights also highlight directions for future research. Hybrid approaches that integrate the residual learning principles of ResNet with the efficiency-driven scaling strategies of EfficientNet may offer new opportunities for advancing image classification. Moreover, with the rise of Vision Transformers and lightweight hybrid CNN-Transformer models, the efficiency-robustness trade-off observed between ResNet and EfficientNet provides a valuable baseline for assessing the next generation of architectures.

Conclusion

This study presented a detailed comparison between ResNet and EfficientNet architectures in image classification tasks, considering accuracy, computational efficiency, and resource utilization. ResNet, with its residual connections, remains a robust and reliable architecture for hierarchical feature extraction, offering consistent performance across various datasets. However, its deeper variants require more parameters and computational resources, leading to increased inference time and higher memory usage. EfficientNet, through its compound scaling strategy, achieves a superior balance between accuracy and efficiency. Smaller variants reach competitive accuracy with significantly fewer parameters and lower computational cost, while larger variants attain state-of-the-art performance without excessively compromising inference speed. Overall, the selection of an appropriate architecture should be guided by the specific requirements of the application. EfficientNet is particularly suitable for scenarios with limited computational resources or real-time constraints, while ResNet is advantageous when deep feature representation and model stability are prioritized. This analysis highlights the importance of considering both accuracy and efficiency in the design and deployment of image classification models.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770–778). IEEE. https://doi.org/10.1109/CVPR.2016.90

[2] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7132–7141). IEEE. https://doi.org/10.1109/CVPR.2018.00745

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, 25, 1097–1105.

[4] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[5] Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. Neural Computation, 29(9), 2352–2449. https://doi.org/10.1162/neco_a_00990

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

[7] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[8] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[9] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1–9). IEEE. https://doi.org/10.1109/CVPR.2015.7298594

[10] Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105–6114). PMLR.