Publisher
Vandana Publications
www.vandanapublications.com

International Journal of Engineering and Management Research

2026 Volume 16 Number 3 June

E-ISSN:2250-0758
P-ISSN:2394-6962

Research Article

Deep Learning

Deep Learning Techniques for Image Classification: Architectures, Advances, and Open Challenges

Mohite A[1*]

DOI:10.31033/IJEMR/16.3.2026.1929

[1*] Ashutosh Mohite, Assistant Professor, Department of Computer Science, Chouksey College of Science and Commerce, Bilaspur, Chhattisgarh, India.

Image classification has become a key task in computer vision. This shift is driven by rapid advancements in deep learning. Over the past decade, deep neural networks, especially Convolutional Neural Networks (CNNs), have greatly outperformed traditional machine learning methods. They do this by automatically learning feature representations from raw image data. More recently, new approaches like Vision Transformers (ViTs), hybrid architectures, and self-supervised learning methods have pushed the limits of classification performance across various datasets and applications. This paper offers a review of deep learning techniques for image classification, focusing on innovations in architecture, key milestones, and recent progress. We examine popular model families, including CNN-based architectures, transformer-based models, and lightweight networks suited for environments with limited resources. Additionally, we look at important training strategies, datasets, and evaluation metrics that have influenced the field. Despite significant advancements, several challenges persist. These include problems with model interpretability, stability against adversarial attacks, dependence on data, high computational needs, and the ability to generalize across different domains. This review highlights these ongoing challenges and suggests possible future research directions. We aim to give researchers and practitioners a clear understanding of the current state and future possibilities of deep learning in image classification.

Keywords: Computer Vision Image Classification, Deep Learning(DL), Convolutional Neural Networks(CNN)

Corresponding Author

Ashutosh Mohite, Assistant Professor, Department of Computer Science, Chouksey College of Science and Commerce, Bilaspur, Chhattisgarh, India.
Email: amohite306@gmail.com

How to Cite this Article

Mohite A, Deep Learning Techniques for Image Classification: Architectures, Advances, and Open Challenges. Int J Engg Mgmt Res. 2026;16(3):87-100.
Available From
https://ijemr.vandanapublications.com/index.php/j/article/view/1929

To Browse

Manuscript Received	Review Round 1	Review Round 2	Review Round 3	Accepted
2026-05-08	2026-05-23			2026-06-10

Conflict of Interest	Funding	Ethical Approval	Plagiarism X-checker	Note
None	Nil	Yes	4.72	

1. Introduction

Image classification is the process of assigning a label to an input image from a set of categories. It is one of the most basic problems in computer vision. This task acts as a foundation for many real-world applications, such as medical image analysis, autonomous driving, surveillance systems, and content-based image retrieval. Traditional methods for image classification relied on handcrafted features like Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG), along with classical machine learning classifiers. While these methods were somewhat effective, they were limited by their reliance on manual feature engineering and struggled to generalize to complex visual patterns.

The rise of deep learning has transformed image classification by allowing models to learn hierarchical representations directly from raw data [2]. The success of deep Convolutional Neural Networks (CNNs), especially after AlexNet's introduction in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), was a key moment in the field. Later architectures like VGG, GoogLeNet, and ResNet brought in deeper and more effective designs. They greatly improved classification accuracy while tackling problems like vanishing gradients and computational efficiency.

In recent years, transformer-based architectures have emerged. These were originally developed for natural language processing and have been successfully applied to vision tasks. Vision Transformers (ViTs) and their variants use self-attention mechanisms to capture long-range dependencies in images. This provides an alternative to convolution-based methods [34]. Hybrid models that combine CNNs and transformers, along with lightweight architectures like MobileNet and EfficientNet, have been suggested to balance performance and efficiency, especially for use on edge devices [29, 30]. Along with these architectural changes, improvements in training strategies—such as transfer learning, data augmentation, self-supervised learning, and large-scale pretraining—have further boosted the capabilities of deep learning models. Key datasets like ImageNet, CIFAR, and COCO have been vital in driving progress by offering standardized evaluation platforms [7].

Despite these successes, several challenges remain. Deep learning models often need large amounts of labeled data and significant computational resources, which limits their accessibility and scalability. In addition, issues related to interpretability, robustness against adversarial attacks, and fairness continue to be major obstacles for widespread use in safety-critical areas.

This paper provides a thorough review of deep learning methods for image classification. It focuses on architectural developments, recent advances, and ongoing challenges. By bringing together existing research, this work aims to offer useful insights into current trends and highlight promising areas for future exploration.

2. Related Works

The field of image classification has changed a lot with the rise of deep learning. It has moved from traditional methods that rely on handcrafted features to advanced neural architectures. This section looks at important contributions in the literature, organized by major method developments and approaches.

2.1. Traditional Machine Learning Approaches

Before deep learning became popular, image classification mainly depended on manual feature extraction methods and traditional classifiers. Techniques like Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), and Local Binary Patterns (LBP) were commonly used to capture local image details. These features were usually input into classifiers like Support Vector Machines (SVMs), k-Nearest Neighbors (k-NN), and Random Forests. While these methods had some success, they relied heavily on expert knowledge for feature design and struggled to capture high-level context. As datasets became larger and more complex, these methods found it difficult to keep up, leading to the rise of data-driven techniques.

The workflow for deep learning-based image classification starts with data collection. Here, a large and varied set of labeled images is gathered. Next, data preprocessing standardizes the inputs through resizing, normalization, and augmentation to improve robustness. The dataset is then divided into training, validation, and test sets to ensure reliable performance evaluation.

After that, an appropriate model architecture, like CNNs, Transformers, or hybrid models, is chosen based on the problem at hand. The model is trained using forward propagation, loss computation, and backpropagation, along with optimization techniques like SGD or Adam. Its performance is assessed using metrics such as accuracy, precision, recall, and F1-score. Hyperparameters are then adjusted to improve performance. Recent developments, like transfer learning, self-supervised learning, and attention mechanisms, are integrated to enhance efficiency and accuracy. Finally, the model is tested and deployed in real-world settings, such as edge devices or cloud systems, while important challenges like data scarcity, model interpretability, resistance to adversarial attacks, and computational cost continue to be vital areas for ongoing research.

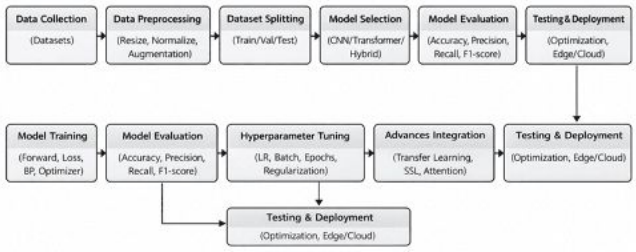

Figure 1: Block Diagram of the deep learning workflow for image classification, showing the pipeline from data acquisition and preprocessing through model development, evaluation, deployment and open research challenges

2.2. Emergence of Convolutional Neural Networks

The introduction of deep Convolutional Neural Networks (CNNs) changed the game in image classification. A turning point came with AlexNet, which showed how effective deep learning could be on large datasets like ImageNet [2]. AlexNet used multiple convolutional and fully connected layers, along with techniques like ReLU activation and dropout, to achieve remarkable performance. After this, several architectures were developed to improve accuracy and efficiency. VGG networks focused on depth by using small convolutional filters. GoogLeNet (Inception) introduced multi-scale feature extraction with parallel convolutional layers [4]. ResNet tackled the degradation problem in deep networks by adding residual connections, which allowed for the training of very deep architectures. These models greatly advanced the field and became essential in modern computer vision.

2.3 Advanced CNN Architectures and Optimization

Subsequent research focused on improving computational efficiency and model scalability. Architectures like DenseNet introduced dense connectivity patterns to improve feature reuse and gradient flow. EfficientNet proposed a method that uniformly scales network depth, width, and resolution for better performance [8].

Lightweight models, including MobileNet and ShuffleNet, were designed for use in resource-constrained environments like mobile and embedded devices [7]. These models used depthwise separable convolutions and channel shuffling techniques to lower computational costs while maintaining competitive accuracy. Besides architectural improvements, optimization techniques such as batch normalization, better weight initialization, and advanced optimizers like Adam and RMSProp were important for stabilizing training and speeding up convergence.

2.4 Transformer-Based Models in Image Classification

Inspired by their success in natural language processing, transformer architectures have recently been adapted for image classification tasks. Vision Transformers (ViTs) treat images as sequences of patches and apply self-attention mechanisms to capture global dependencies. Unlike CNNs, which rely on local receptive fields, transformers can model long-range relationships more effectively.

Variants such as DeiT (Data-efficient Image Transformers) and Swin Transformer have further improved performance by addressing data efficiency and computational complexity. Hybrid architectures that combine CNN feature extractors with transformer-based attention mechanisms have also been proposed, leveraging the strengths of both approaches [12]. Despite their strong performance, transformer-based models often require large-scale datasets and high computational resources, which remain key limitations.

2.5 Training Strategies and Data-Centric Approaches

Beyond architectural innovations, training methods have played an important role in improving image classification. Transfer learning allows models pretrained on large datasets like ImageNet to be adjusted for smaller, specific tasks.

This reduces the need for extensive labeled data. Data augmentation techniques, such as random cropping, flipping, rotation, and more advanced methods like Mixup and CutMix, have improved how well models can generalize. Moreover, self-supervised and unsupervised learning approaches have gained attention for their ability to learn useful representations from unlabeled data. Benchmark datasets, including ImageNet, CIFAR-10/100, and MNIST, have provided standard platforms for evaluating and comparing models. This has driven steady progress in the field.

2.6 Challenges Identified in Existing Literature

Despite notable progress, several challenges remain in the literature. One major concern is the lack of interpretability in deep learning models; this is often called "black-box" behavior. It limits their use in critical areas like healthcare and autonomous systems. Another issue is robustness, as deep learning models can be weak against adversarial attacks and noise changes. Additionally, the need for large labeled datasets and high computational power creates obstacles for broader use, especially in low-resource environments. Generalization across different domains is another challenge, as models trained on specific datasets often do not perform well on new or real-world data distributions. These limitations point to the need for more efficient, strong, and understandable models in future research.

3. Comparison of Deep Learning Architectures for Image Classification

The rapid progress of deep learning has greatly changed image classification. It allows models to achieve high accuracy in many applications. Over the last decade, various architectures have emerged. These include early convolutional neural networks and more complex designs like residual networks and vision transformers. They aim to tackle challenges in feature extraction, scalability, and generalization. This section provides a comparative analysis of key deep learning architectures for image classification. It highlights their structural differences, performance, and suitability for different tasks. By evaluating these models, we offer insights into their strengths and weaknesses. This will help in choosing the right architectures for specific research and real-world uses.

The evolution of convolutional neural network (CNN) architectures has moved from early deep learning breakthroughs to highly optimized, hybrid models that incorporate transformer-inspired ideas. One of the first milestones, AlexNet, brought deep CNNs with ReLU activation and dropout. It achieved a significant performance improvement despite its high computational cost (Krizhevsky et al., 2012). Next came VGG-16, which focused on simplicity with small 3×3 filters. However, it resulted in a large number of parameters (Simonyan & Zisserman, 2014). Around the same time, GoogLeNet introduced Inception modules for effective multi-scale feature extraction. This approach reduced the number of parameters while making the architecture more complex (Szegedy et al., 2015).

Subsequent architectures focused on improving training stability and using features more effectively. ResNet-50 tackled the vanishing gradient problem with residual connections, allowing for much deeper networks (He et al., 2016). DenseNet-121 improved feature sharing with dense connectivity, though it required more memory (Huang et al., 2017). Efficiency became a key goal with EfficientNet-B0. This model used compound scaling to balance network depth, width, and resolution for the best performance (Tan & Le, 2019). More recent architectures stress scalability, efficiency, and integration with transformer-like methods. EfficientNet-V2 sped up training using fused MBConv blocks and progressive learning (Tan & Le, 2021). MobileNet-V3 aimed at mobile and embedded systems with lightweight depthwise separable convolutions (Howard et al., 2019). RegNetY offered a regularized design space for scalable CNNs with strong generalization (Radosavovic et al., 2020). Modern CNN designs increasingly take ideas from transformers. ConvNeXt updated CNNs with large kernels, LayerNorm, and patch-based processing to match or exceed vision transformers (Liu et al., 2022). Its successor, ConvNeXt-V2, further enhanced representation learning by adding masked autoencoder pre-training and Global Response Normalization (Woo et al., 2023). Conv2Former replaced self-attention with convolution-based modulation for better feature extraction (Hou et al., 2022).

3.1. Hybrid Architectures

Hybrid and specialized architectures have also emerged to tackle specific challenges. MaxViT combines convolution with multi-axis attention for high performance, but this comes with increased memory usage (Tu et al., 2022). Sequencer2D integrates sequence modeling into CNNs, which improves spatial understanding (Tadmor et al., 2022). For segmentation tasks, SegNeXt offers an efficient backbone optimized for dense prediction (Guo et al., 2022). SPD-Conv boosts performance on low-resolution images by replacing pooling with space-to-depth operations (Sunkara & Luo, 2022). Recent developments also explore hybridization and edge efficiency. Swin-ConvNeXt Hybrid combines ConvNeXt blocks with Swin Transformer modules for better multi-scale representation, although this adds complexity (Madhavi et al., 2025).

Edge-Aware ResNet focuses on edge AI by incorporating grouped, shuffle, and depthwise convolutions into a ResNet backbone for better efficiency (Korkmaz, 2025). Overall, the trend in CNN architectures shows a clear move from designs that emphasize depth toward efficiency, scalability, and hybridization with transformer-based concepts, enabling top performance across many computer vision tasks. These models are shown in Table 1, and the architecture of a hybrid CNN model appears in Figure 2.

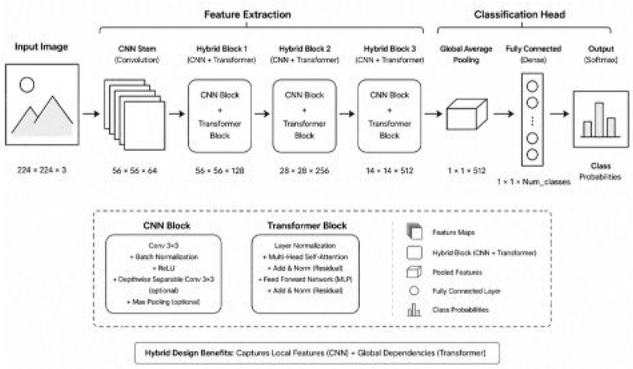

Figure 2: Hybrid CNN model for Image Classification

Table 1: Comparison of CNN-Based Architectures (with References)

Architecture	Year	Key-Idea	Depth	Parameters	Advantages	Limitations	Reference (Authors)
ConvNeXt	2022	Modernized CNN using transformer design ideas (large kernels, LayerNorm, patchify stem)	~88 layers	28M–350M	Matches or surpasses Vision Transformers on many tasks	High computational cost for large variants	Liu et al.
ConvNeXt-V2	2023	Combines CNN scaling with masked autoencoder pre-training and Global Response Normalization	100+ layers	3.7M–650M	Strong representation learning and improved feature competition	Training complexity and large models	Woo et al.
Conv2Former	2022	Transformer-style convolutional modulation replacing self-attention	60–100 layers	~27M–100M	Efficient feature extraction with large kernels	Still computationally expensive	Hou et al.
EfficientNet-V2	2022 widely used	Fused-MBConv blocks and progressive training for faster convergence	100+ layers	21M–55M	Faster training and good parameter efficiency	Architecture tuning required	Tan & Le
RegNetY	2022 widely used	Regularized design space for scalable CNN architecture	32–120 layers	15M–145M	Simple scalable design and strong generalization	Less efficient compared with newest CNNs	Radosavovic et al.
MobileNet-V3	2022 widely used	Lightweight architecture using depthwise separable convolution and NAS	~53 layers	~5.4M	Very efficient for mobile and embedded systems	Lower accuracy than large CNN models	Howard et al.
MaxViT-CNN Hybrid	2022	Combines convolution with blocked attention and multi-axis attention	50–100 layers	~30M–120M	High performance on vision tasks	Higher memory requirements	Tu et al.

Sequencer2D	2022	Sequence modeling approach integrated with convolution blocks	~90 layers	~26M	Effective spatial modeling in images	Less mature ecosystem	Tadmor et al.
SegNeXt	2022	Efficient convolutional backbone designed for segmentation tasks	50–80 layers	~13M–50M	Efficient segmentation with large kernels	Mainly optimized for segmentation tasks	Guo et al.
SwinConvNeXt (Hybrid CNN)	2025	Fusion of ConvNeXt convolution blocks with Swin transformer modules	~100+ layers	~50M+	Improved multi-scale feature extraction	Increased model complexity	Madhavi et al.
SPD-Conv CNN	2022	Replaces pooling and strided convolutions with space-to-depth convolution blocks	~50–100 layers	Depends on backbone	Better performance for low-resolution images and small objects	Not widely adopted yet	Sunkara & Luo
Edge-Aware ResNet CNN	2025	Integrates grouped, shuffle, and depthwise convolution operations into ResNet backbone	~50 layers	~25M	Improved efficiency for edge AI applications	Limited benchmarks compared to mainstream models	Korkmaz
AlexNet	2012	Deep CNN with ReLU & dropout	8	~60M	Breakthrough performance	High computation	Krizhevsky et al. (2012)
VGG-16	2014	Small (3×3) filters, deep network	16	~138M	Simple architecture	Very large size	Simonyan & Zisserman (2014)
GoogLeNet	2014	Inception modules (multi-scale)	22	~5M	Efficient computation	Complex design	Szegedy et al. (2015)
ResNet-50	2015	Residual connections	50	~25M	Solves vanishing gradient	Resource intensive	He et al. (2016)
DenseNet-121	2017	Dense connectivity	121	~8M	Feature reuse	Memory heavy	Huang et al. (2017)
EfficientNet-B0	2019	Compound scaling	82	~5.3M	High efficiency	Needs tuning	Tan & Le (2019)

3.2. Lightweight Architectures

Lightweight CNN architectures have been designed to help deep learning models run efficiently on devices with limited resources, such as mobile phones, embedded systems, and IoT platforms. One of the earliest and most notable models in this group is MobileNetV1. This model introduced depthwise separable convolutions, which significantly cut down on computation costs and model size while maintaining reasonable accuracy. Expanding on this concept, MobileNetV2 added inverted residual blocks and linear bottlenecks. This further improved efficiency, making the model ideal for embedded systems that must manage memory and power use carefully. Another important lightweight architecture is ShuffleNet. It introduced channel shuffling to improve information flow across feature channels, keeping the model very small, around 1 to 2 million parameters. This feature makes it especially useful for IoT applications with limited computational resources. In addition, EfficientNet-Lite expands the EfficientNet family by using scaling techniques tailored for edge devices. This optimization helps balance latency and performance. The layered architecture of lightweight models has shown in Figure 3.

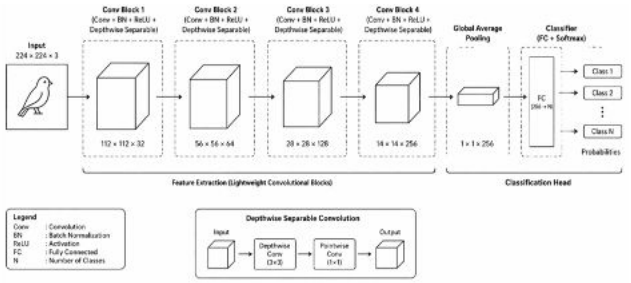

Figure 3: Architecture of Lightweight CNN based model for image classification

Overall, these lightweight architectures aim to reduce parameters and computation while keeping acceptable accuracy. This makes them perfect for real-time and deployment-focused applications like mobile-based gesture recognition, smart surveillance, and edge AI systems. You can see a comparison of these lightweight models in Table-2.

Table 2: Lightweight Architectures (with References)

Archi-tecture	Year	Key Technique	Para-meters	Advantages	Use Case	Reference (Authors)
Mobile NetV1	2017	Depthwise separable conv	~4.2M	Low computation	Mobile	Howard et al. (2017)
Mobile NetV2	2018	Inverted residuals	~3.4M	Efficient	Embedded systems	Sandler et al. (2018)
Shuffle Net	2018	Channel shuffle	~1–2M	Very lightweight	IoT	Zhang et al. (2018)
Efficient Net-Lite	2020	Scaled lightweight model	Varies	Better latency	Edge AI	Tan & Le (2020)

The evolution of lightweight convolutional neural network (CNN) architectures has greatly improved image classification in settings with limited resources. MobileNetV1, introduced in 2017 by Andrew G. Howard and others, brought in depthwise separable convolutions to cut down on computational complexity and model size. It achieved about 4.2 million parameters while still performing well for mobile applications. Then, MobileNetV2, created by Mark Sandler and others in 2018, enhanced efficiency with inverted residual blocks and linear bottlenecks. This reduced parameters to around 3.4 million, making it very effective for embedded systems. In the same year, ShuffleNet, developed by Xiangyu Zhang and colleagues, added channel shuffling and pointwise group convolutions. This approach significantly lowered computational costs and achieved a compact model size of about 1 to 2 million parameters, which is ideal for IoT devices. More recently, EfficientNet-Lite proposed by Mingxing Tan and Quoc V. Le in 2020, used compound scaling techniques to optimize network depth, width, and resolution. This led to better trade-offs between latency and performance across various parameter sizes, making it a great fit for edge AI applications. Together, these architectures show a clear trend toward reducing computational needs while keeping classification accuracy high, which allows for use in various low-power and real-time settings.

3.3. Transformer Based Architecture

Transformer-based architectures for image classification, known as Vision Transformers (ViTs), process images using a method originally developed for natural language processing. Rather than using convolutional operations like CNNs, these models break an input image into fixed-size patches and treat each patch as a token. For an image sized $H \times W \times C$, it is divided into smaller patches of size $P \times P$, resulting in a sequence of flattened vectors. Each of these vectors is then projected into a fixed-dimensional embedding space. To keep spatial information, positional encodings are added to these embeddings, and a special classification token is typically added to the sequence to represent the overall image.

The sequence of embedded patches goes through a stack of transformer encoder layers. Each encoder layer has two main parts: multi-head self-attention and a feed-forward neural network. The self-attention mechanism helps the model capture global relationships between all image patches by calculating interactions between every pair of tokens. This is a key advantage over CNNs, which focus only on local receptive fields. Multi-head attention enhances this ability by letting the model learn different types of relationships at the same time. After the attention module, a feed-forward network processes each token independently, and residual connections with layer normalization are used to stabilize training and improve performance.

After going through several encoder layers, the output for the classification token is extracted and sent into a fully connected layer (classification head), followed by a softmax function to produce class probabilities. Transformer-based models are particularly good at capturing long-range dependencies and global context, making them very effective for large-scale image classification tasks. However, they generally require larger datasets and more computational resources than lightweight CNN architectures. Despite this, their flexibility and strong performance have made them a popular choice in current computer vision research. Figure 4 shows transformer-based architecture.

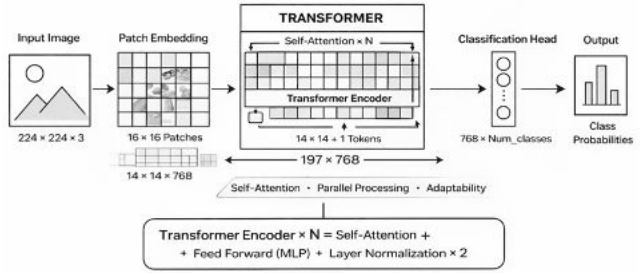

Figure 4: Transformer based architecture for image classification

The rise of transformer-based designs for image classification has created a new approach focused on understanding global connections through attention methods. The Vision Transformer (ViT), proposed by Alexey Dosovitskiy et al. in 2021, uses a patch-based attention system. It breaks images into fixed-size patches and treats them as token sequences. While it effectively captures global context, the model is computationally heavy, with around 86 million parameters, and needs large datasets for the best results. To tackle data dependency issues, the Data-efficient Image Transformer (DeiT), introduced by Hugo Touvron et al. in 2021, employs knowledge distillation strategies that allow training with smaller datasets. Despite this advancement, it still has a similar parameter count of about 86 million and remains demanding in terms of computation.

Further developments include the Swin Transformer, proposed by Ze Liu et al. in 2021. This model uses shifted window-based self-attention to lower computational complexity while keeping strong performance. With approximately 28 million parameters, it offers better scalability and hierarchical feature representation. However, its design is more complex. At the same time, Hybrid CNN-Transformer Models have appeared as a promising option. They merge convolutional layers for local feature extraction with transformer-based attention for global context. These hybrid models provide a good balance between accuracy and efficiency, though their design complexity and variation among implementations can pose challenges. Overall, these architectures show a shift toward combining global attention methods with efficient design strategies, aiming to enhance performance while dealing with computational and data challenges.

Table 3: Transformer-Based Architectures

Archi-tecture	Year	Key Idea	Para-meters	Adv.	Limi-tations	Reference (Authors)
ViT	2020	Patch-based attention	~86M	Global context	Needs large data	Dosovitskiy et al. (2021)
DeiT	2021	Data-efficient ViT	~86M	Works with less data	Still heavy	Touvron et al. (2021)
Swin Transformer	2021	Shifted window attention	~28M	Scalable	Complex	Liu et al. (2021)
Hybrid CNN-Transformer	2021+	CNN + attention	Varies	Best of both	Complex design	Various authors

3.4. CNN vs. Transformer Based Architectures

For image classification tasks, Convolutional Neural Networks (CNNs) and Transformer-based models represent two different approaches, each with its own strengths and weaknesses. CNNs work by applying convolutional filters to local areas of an image. This method allows for efficient extraction of spatial features like edges, textures, and shapes. The localized processing introduces strong biases such as translation invariance and spatial locality. These features allow CNNs to perform well even with moderately sized datasets. Consequently, architectures like AlexNet, VGG, and ResNet have historically achieved high accuracy while using relatively less computing power. This makes CNNs suitable for real-time and edge-based applications.

In contrast, Transformer-based models, especially Vision Transformers (ViTs), treat an image as a sequence of patches. They use self-attention mechanisms to capture global relationships across the entire image. This ability lets Transformers model long-range relationships more effectively than CNNs, leading to better performance on large-scale datasets. However, this strength comes with a drawback: Transformers have weaker biases, requiring significantly more training data and computational power to perform well. They also provide better scalability and can be processed in parallel, making them ideal for high-performance computing and large-scale vision tasks.

Table 4 presents a comparative analysis of Convolutional Neural Networks (CNNs) and Transformer-based models across various architectural and performance aspects. CNNs rely mainly on localized feature extraction through convolutional kernels, allowing them to efficiently capture spatial hierarchies with strong biases like translation invariance.

On the other hand, Transformers use self-attention to model global relationships in the input, resulting in a superior ability to capture long-range dependencies. While CNNs generally require moderate training data and have lower computational costs, Transformers need large datasets and more computing resources due to their attention operations.

Moreover, the receptive field in CNNs has inherent limits but can grow with increased network depth. Transformers already have a global receptive field through self-attention. When it comes to parallel processing, Transformers have significant advantages because of their non-sequential processing, which enhances scalability compared to CNNs. However, CNNs offer greater stability during training, especially on smaller datasets, making them suitable for resource-limited and edge AI applications like object detection and image segmentation. In contrast, Transformers excel in large-scale vision tasks and multimodal learning scenarios, achieving very high performance when trained on extensive datasets. Overall, this comparison shows a trade-off between efficiency and scalability. CNNs remain practical for lightweight applications, while Transformers are more dominant in high-performance, large-scale environments.

Table 4: CNN Model vs Transformer Model

Aspect	CNN	Transformer	Key References
Feature Extraction	Local	Global	Krizhevsky et al. (2012); Dosovitskiy et al. (2021)
Inductive Bias	Strong	Weak	He et al. (2016); Touvron et al. (2021)
Data Requirement	Moderate	High	Tan & Le (2019); Dosovitskiy et al. (2021)
Computation	Lower	Higher	Howard et al. (2017); Liu et al. (2021)
Performance	High	Very High (large-scale)	He et al. (2016); Liu et al. (2021)
Feature Extraction	Local	Global	Krizhevsky et al. (2012); Dosovitskiy et al. (2021)
Inductive Bias	Strong	Weak	He et al. (2016); Touvron et al. (2021)
Receptive Field	Limited (expands with depth)	Global (self-attention)	Simonyan & Zisserman (2015); Dosovitskiy et al. (2021)
Data Requirement	Moderate	Large-scale datasets	Tan & Le (2019); Khan et al. (2022)
Computational Cost	Lower	Higher	Howard et al. (2019); Vaswani et al. (2017)
Parallelization	Moderate	High	Vaswani et al. (2017); Dosovitskiy et al. (2021)
Context Modeling	Local dependencies	Long-range dependencies	Wang et al. (2018); Liu et al. (2021)
Scalability	Limited scaling	Highly scalable	Tan & Le (2019); Touvron et al. (2021)
Training Stability	Stable on small datasets	Requires large datasets	Khan et al. (2022); Dosovitskiy et al. (2021)
Typical Use	Edge AI, detection, segmentation	Large-scale vision and multimodal models	Liu et al. (2022); Dosovitskiy et al. (2021)

Overall, CNNs are favored for situations with limited data, smaller computational budgets, and applications that need efficiency. On the other hand, Transformer-based models are preferred in large-scale image classification tasks where capturing global context and reaching top accuracy are essential.

4. Results and Discussions

For a gesture recognition or classification project, ConvNeXt is an effective and practical choice as the main model. This architecture is well-suited because it updates traditional convolutional neural networks. It includes transformer-inspired design features like larger kernel sizes and better normalization techniques. These help capture fine details of hands and the broader spatial context. Such characteristics are crucial in gesture recognition, where small changes in finger positions and hand orientation affect classification accuracy. Additionally, ConvNeXt strikes a good balance between performance and computational efficiency, making it suitable for medium-sized datasets and typical research settings without needing high-end hardware.

As an alternative, EfficientNet V2 can be used when speed and efficiency are priorities. It uses fused MBConv blocks and learning strategies that cut down training time while keeping high accuracy. This is especially helpful when working with limited resources or when multiple experiments and hyperparameter tuning are necessary. While it may fall slightly behind ConvNeXt in capturing very complex spatial relationships, it still excels in most gesture recognition tasks.

For baseline comparison, using ResNet-50 is highly recommended. ResNet-50 remains a standard benchmark due to its residual learning design, which reduces vanishing gradient issues and enables stable training.

Comparing your main model against ResNet-50 shows performance improvements in a scientifically sound way. Overall, combining ConvNeXt as the primary model, EfficientNet-V2 as an efficient alternative, and ResNet-50 as a baseline creates a solid experimental setup for gesture recognition research.

The comparison of lightweight CNN architectures shows that MobileNetV2 offers the best balance between accuracy and efficiency among the models examined. While MobileNetV1 introduced depthwise separable convolutions to cut computation, it is less efficient and accurate than MobileNetV2. The latter enhances this by using inverted residual blocks and linear bottlenecks, which improve feature representation with fewer parameters and make it very suitable for mobile and embedded vision tasks. In contrast, ShuffleNet creates an even smaller model size through channel shuffling and group convolutions, ideal for very resource-limited IoT environments. However, this simplicity often leads to slightly lower accuracy. Similarly, EfficientNet-Lite improves latency and performance for edge devices through compound scaling, but it may need more careful tuning and does not always outperform MobileNetV2 in every situation. Overall, considering the trade-offs between model size, computational cost, and classification performance, MobileNetV2 is the best lightweight architecture for practical use, especially in applications like gesture recognition where both efficiency and accuracy matter.

5. Research Gap

Despite the great success of deep learning techniques in image classification, several key challenges remain. One main limitation is the reliance on large annotated datasets, which are costly and time-consuming to create. Although self-supervised and semi-supervised learning approaches have been suggested, achieving competitive performance with limited labeled data is still a problem (Dosovitskiy et al., 2021; He et al., 2020).

Another important gap is the generalization ability of models when faced with domain shifts. Models trained on benchmark datasets often do not perform well in real-world settings due to differences in lighting, background, and viewpoint. This shows the need for strong domain adaptation techniques that ensure consistent performance across varied conditions (Torralba & Efros, 2011).

In addition, deep learning models lack interpretability and transparency, making them unsuitable for critical areas like healthcare and autonomous systems. While explainable AI (XAI) methods have been developed, they often do not provide reliable and understandable explanations (Rudin, 2019).

Adversarial vulnerability is also a major gap in research. Deep neural networks are very sensitive to small, barely noticeable changes in input data, which can greatly reduce classification performance. Creating models that are naturally resistant to these adversarial attacks is still a challenge (Goodfellow et al., 2015).

Moreover, computational complexity and scalability are ongoing concerns, especially for transformer-based architectures. These models need significant computational power and memory, limiting their use in resource-limited environments like mobile and embedded systems (Tan & Le, 2019).

Another less explored area is managing class imbalance and fine-grained classification. Models often struggle to tell apart visually similar categories or classes that are underrepresented. Current techniques frequently do not generalize well across different datasets (Krizhevsky et al., 2012).

Lastly, issues related to bias, fairness, and ethics continue to exist, with models often reflecting the biases found in their training data. Tackling these issues is critical for creating trustworthy and socially responsible AI systems (Mehrabi et al., 2021).

6. Conclusion

This review examined the evolution of image classification techniques. It starts with traditional machine learning methods and moves to modern deep learning models. These include Convolutional Neural Networks (CNNs), Residual Networks (ResNet), Dense Convolutional Networks (DenseNet), EfficientNet, Vision Transformers (ViTs), and hybrid deep learning models. The review also discusses the significance of image preprocessing techniques, data augmentation strategies, transfer learning, optimization algorithms, and performance evaluation metrics that contribute to better model performance.

The comparison shows that CNN-based architectures remain effective for many image classification tasks due to their ability to learn hierarchical spatial features efficiently.

On the other hand, transformer-based models have achieved great results in large-scale image recognition by capturing long-range dependencies with self-attention mechanisms. Despite these advancements, significant research challenges still limit the widespread use of deep learning models in real-world situations. These challenges involve the need for large annotated datasets, high computational demands, limited model interpretability, susceptibility to adversarial attacks, poor generalization across different domains, and concerns about fairness, bias, and energy use.

Overall, this review offers a clear view of the current state of deep learning in image classification. It summarizes existing methods, points out their strengths and weaknesses, and identifies key research gaps. The findings in this paper can serve as a useful reference for researchers and practitioners looking to create accurate, strong, and efficient image classification systems for future intelligent applications.

7. Future Works

Although deep learning has made great strides in image classification, many opportunities for future research still exist. First, we need to create data-efficient learning methods, such as self-supervised, semi-supervised, few-shot, and zero-shot learning, to lessen our reliance on large labeled datasets. These methods could significantly cut annotation costs while still delivering strong classification results. Second, upcoming studies should work on enhancing the robustness and generalization ability of deep learning models in various environmental conditions, like changes in light, occlusion, background clutter, noise, and domain shifts.

Additionally, improving the interpretability and explainability of deep learning models is crucial, especially in critical areas like healthcare, autonomous driving, industrial inspection, and surveillance. Using explainable artificial intelligence (XAI) techniques can build user trust and support clear decision-making. Future research should also explore hybrid architectures that combine CNNs, Vision Transformers, graph neural networks, and attention mechanisms to take advantage of different feature representations.

Progress in these areas is likely to influence the next generation of intelligent visual recognition technologies and help them be adopted in healthcare, agriculture, smart cities, robotics, manufacturing, and human-computer interaction.

References

[1] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In: *Proceedings of the Neural Information Processing Systems (NeurIPS)*.

[2] Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. Available at: arXiv:1409.1556.

[3] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[5] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K.Q. (2017). Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

[7] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. Available at: arXiv:1704.04861.

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[10] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16×16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations (ICLR)*.

[11] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In: *Proceedings of the International Conference on Machine Learning (ICML)*.

[12] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In: *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105.

[14] Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. Available at: arXiv:1409.1556.

[15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.

[16] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

[17] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K.Q. (2017). Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708.

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

[19] Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. Available at: arXiv:1704.04861.

[20] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520.

[21] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6848–6856.

[22] Tan, M., & Le, Q.V. (2020). *EfficientNet-Lite: Improving efficiency of convolutional neural networks for mobile and edge devices*. Available at: Google AI Blog / arXiv (related work).

[23] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16×16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations (ICLR)*.

[24] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10347–10357.

[25] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 10012–10022.

[26] Various authors. (2021–2024). Hybrid CNN-Transformer models. In: *Proceedings of various IEEE Conference on Computer Vision and Pattern Recognition (CVPR) and IEEE International Conference on Computer Vision (ICCV)*.

[27] Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., & Adam, H. (2019). Searching for MobileNetV3. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324.

[28] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

[29] Hou, Q., et al. (2022). Conv2Former: A simple transformer-style convolutional network for visual recognition. In: *Advances in Neural Information Processing Systems (NeurIPS)*.

[30] Howard, A., et al. (2019). Searching for MobileNetV3. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[31] Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., & Shah, M. (2022). Transformers in vision: A survey. *ACM Computing Surveys*, *54*(10), 1–41.

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

[33] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11976–11986.

[34] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10012–10022.

[35] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In: *International Conference on Learning Representations (ICLR)*.

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

[37] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers and distillation through attention. In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 10347–10357.

[38] Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., & Li, Y. (2022). MaxViT: Multi-axis vision transformer. In: *European Conference on Computer Vision (ECCV)*.

[39] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

[40] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., & Xie, S. (2023). ConvNeXt V2: Co-designing and scaling ConvNets with masked autoencoders. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[41] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16×16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations (ICLR)*.

[42] Goodfellow, I.J., Shlens, J., & Szegedy, C. (2015). Explaining and harnessing adversarial examples. In: *International Conference on Learning Representations (ICLR)*.

[43] He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9729–9738.

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

[45] Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM Computing Surveys*, *54*(6), 1–35.

[46] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215.

[47] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 6105–6114.

[48] Torralba, A., & Efros, A.A. (2011). Unbiased look at dataset bias. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1521–1528.