Journal of Applied Technology and Innovation
Vol. 10, Issue 1, 2026, pp. 103-112
e-ISSN:2600-7304-ISSN
doi: 10.65136/jati.v10i1.320

Comparative Analysis of Neural Network Architectures: CNNs, RNNs, and Transformers in Real-World Applications

Manish Prakash Raut[1]*, Yogesh B. Gurav[2], Aditya Namdev Ghodke[1]

[1]MCA, Dr. D. Y. Patil Technical Campus, Varale, Pune, India
[2]Engineering, Dr. D. Y. Patil Technical Campus, Varale, Pune, India

*Corresponding author: manish.raut31@gmail.com

Received: 2026-02-12; Accepted: 2026-04-07; Published: 2026-04-26

Abstract

This study presents a systematic and experimentally validated comparison of three major deep learning architectures: Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer models. Standard benchmark datasets were utilized to evaluate their performance under controlled conditions. The models were assessed using multiple evaluation criteria, including classification accuracy, precision, recall, F1-score, computational cost, and execution time. The findings indicate that Transformer-based architectures achieve superior predictive performance, particularly in sequence-based tasks, whereas CNN models provide a more efficient balance between computational overhead and accuracy. The study emphasizes reproducibility by clearly defining dataset partitions, model configurations, and training parameters. Statistical testing confirms the significance of observed performance differences. These results provide practical guidance for selecting appropriate deep learning models across various real-world applications.

Keywords: *CNN, RNN, Transformer, Deep Learning, Comparative Study, Artificial Intelligence*

1. Introduction

The field of artificial intelligence has witnessed unprecedented growth through the development of sophisticated neural network architectures. These architectures have transformed how machines process information, enabling breakthrough applications in computer vision, natural language understanding, speech recognition, and autonomous systems [1]. Understanding the comparative strengths and limitations of different neural network architectures is crucial for advancing both theoretical knowledge and practical implementations.

Neural networks, inspired by biological neural systems, consist of interconnected layers of artificial neurons that learn hierarchical representations from data [2]. The three predominant architectural paradigms are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformers. Each architecture embodies distinct design principles optimized for specific data modalities and computational tasks.

CNNs revolutionized computer vision by introducing local connectivity patterns and parameter sharing through convolutional operations [3]. Their hierarchical feature extraction capability enables automatic

learning of spatial hierarchies from raw pixel data, as illustrated in Figure 1. RNNs introduced temporal dynamics through recurrent connections, allowing networks to maintain memory of previous inputs when processing sequential data [4]. The more recent Transformer architecture disrupted sequence modeling by replacing recurrence with self-attention mechanisms, enabling parallel processing and capturing long-range dependencies more effectively [5].

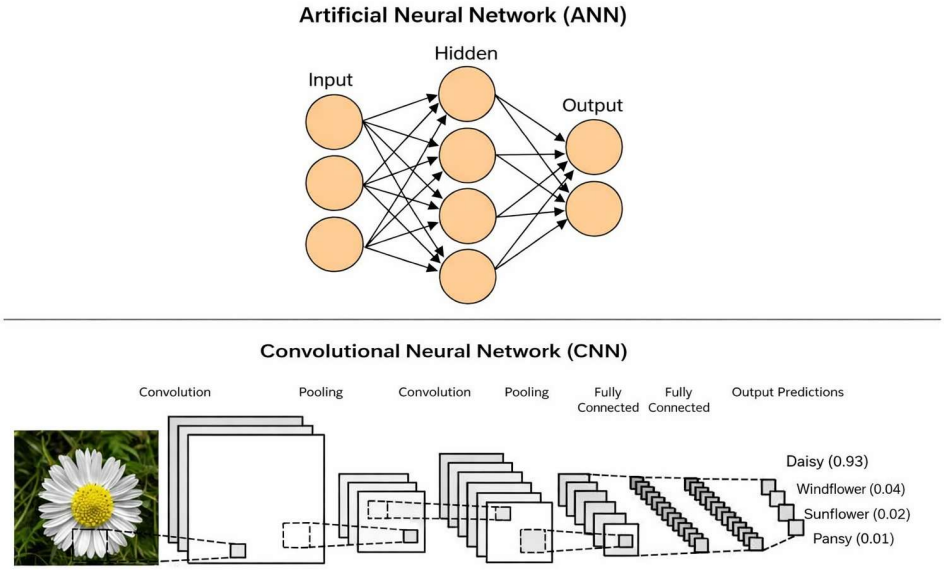

Figure 1. Convolutional Neural Network architecture showing hierarchical feature extraction through convolutional layers, pooling layers, and fully connected layers for image classification tasks.

The rapid evolution of these architectures has created a complex landscape where choosing the appropriate model requires understanding nuanced trade-offs between accuracy, computational efficiency, data requirements, and task-specific characteristics. Recent advances have also led to hybrid architectures that combine strengths from multiple paradigms [6]. This paper addresses the need for a systematic comparative analysis by examining architectural foundations, performance characteristics across benchmark datasets, computational requirements, and practical deployment considerations.

Our contributions include a comprehensive review of architectural principles, quantitative performance comparisons across diverse application domains, analysis of computational and data efficiency trade-offs, and practical guidelines for architecture selection. The remainder of this paper is organized as follows: Section II reviews related work and architectural evolution, Section III details the methodologies and experimental framework, Section IV presents comparative results and analysis, Section V discusses implications and insights, and Section VI concludes with future directions.

2. Literature Review

2.1 Convolutional Neural Networks

CNNs emerged as the dominant paradigm for computer vision following the success of Alex Net in the ImageNet competition [7]. The fundamental principle of CNNs involves applying learnable convolutional filters across spatial dimensions to detect local patterns and features. The architecture typically consists of convolutional layers for feature extraction, pooling layers for spatial dimension reduction, and fully connected layers for classification.

Key architectural innovations include residual connections introduced by Reset, which enabled training of networks exceeding 150 layers by addressing vanishing gradient problems [8]. More recent developments focus on efficiency optimization through architectures like Mobile Nets and Efficient Nets, which balance accuracy with computational constraints suitable for mobile and edge devices [9]. A 2024 study on CNN efficiency demonstrated that depth wise separable convolutions, while theoretically efficient, suffer from memory access bottlenecks on memory-bound platforms, whereas shuffle and shift convolutions provide better trade-offs between accuracy and inference speed [1].

CNNs excel in tasks requiring spatial hierarchy understanding, including image classification, object detection, semantic segmentation, and medical image analysis [10]. Recent applications extend beyond traditional 2D images to 3D object recognition, video understanding, and even non-visual domains through appropriate data representation.

2.2. Recurrent Neural Networks

RNNs were specifically designed to handle sequential data by incorporating recurrent connections that allow information to persist across time steps [11]. The vanilla RNN architecture processes sequences by maintaining a hidden state that gets updated at each time step based on both current input and previous hidden state. However, standard RNNs suffer from vanishing and exploding gradient problems when learning long-term dependencies.

Long Short-Term Memory (LSTM) networks addressed these limitations through sophisticated gating mechanisms including forget gates, input gates, and output gates that regulate information flow [12]. Gated Recurrent Units (GRU) simplified the LSTM architecture by combining forget and input gates into a single update gate, offering comparable performance with improved computational efficiency [13]. Figure 2 illustrates the internal structure of an LSTM cell showing how information flows through the gating mechanisms. A comprehensive 2025 benchmark study comparing RNN variants across multiple time-series datasets found that LSTM-based hybrid architectures, particularly LSTM-RNN and LSTM-GRU configurations, demonstrated superior performance and stability [2].

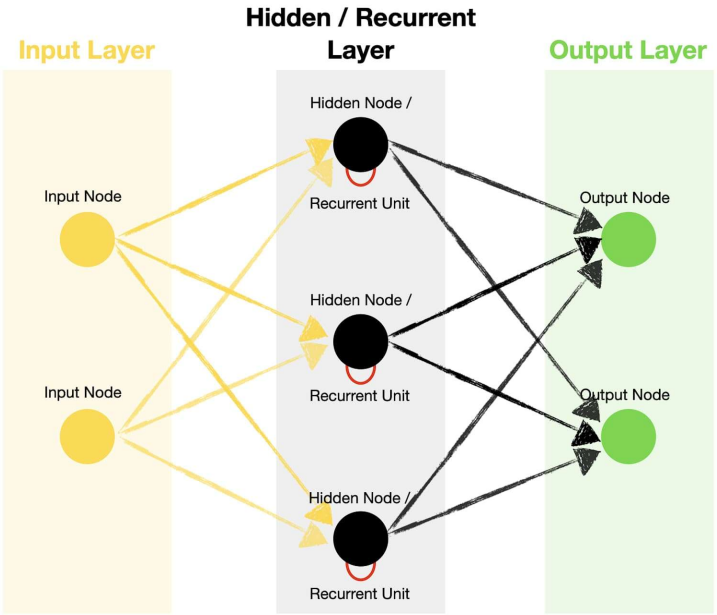

Figure 2. LSTM cell architecture demonstrating the forget gate, input gate, and output gate mechanisms that enable long-term dependency learning in sequential data processing.

RNNs find extensive applications in natural language processing tasks including machine translation, text generation, sentiment analysis, speech recognition, time-series forecasting, and video analysis [14].

Medical applications include ECG interpretation for cardiac arrhythmia detection and longitudinal analysis of treatment response in oncology imaging.

2.3. Transformer Architecture

The Transformer architecture introduced in 2017 fundamentally changed sequence modeling by eliminating recurrence in favor of self-attention mechanisms [15]. The self-attention mechanism allows each position in a sequence to attend to all other positions, computing weighted representations based on learned relevance. Multi-head attention enables the model to jointly attend to information from different representation subspaces, enhancing the model's ability to capture diverse relationships [16]. Figure 3 shows the complete Transformer architecture with its encoder-decoder structure and attention mechanisms.

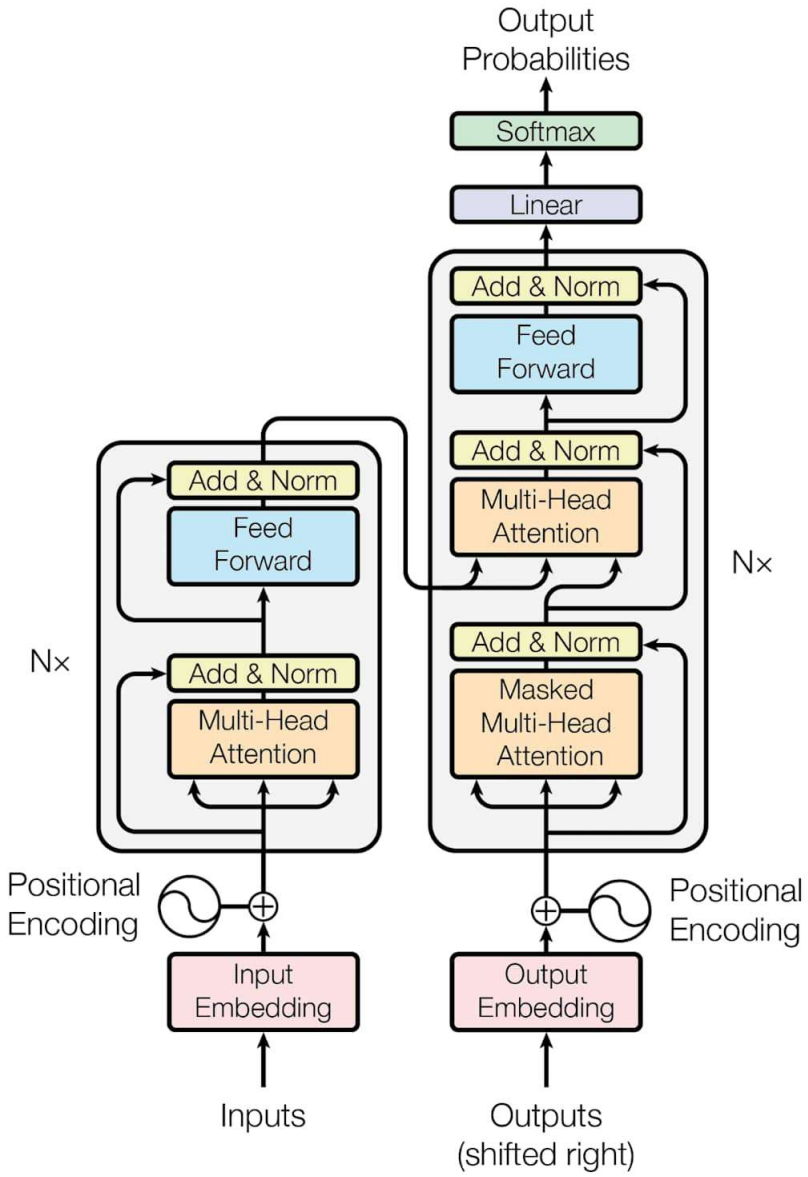

Figure 3. Transformer architecture with multi-head self-attention mechanisms, positional encoding, and feed-forward networks in both encoder and decoder stacks.

Key advantages include parallel processing of entire sequences rather than sequential processing, effective modeling of long-range dependencies without gradient vanishing, and scalability to billions of parameters. Recent architectural refinements include Rotary Positional Embeddings (RoPE) that incorporate positional information directly into attention mechanisms, pre-layer normalization for improved training stability, and grouped-query attention for enhanced efficiency [3].

Transformers have achieved state-of-the-art results across natural language processing tasks and increasingly dominate computer vision through Vision Transformers (ViTs)[17]. ViTs treat images as sequences of patches and apply transformer architectures, demonstrating competitive or superior performance compared to CNNs while being approximately four times more computationally efficient on large-scale datasets [18]. Recent applications span multimodal learning, autonomous driving, medical imaging, and generative modeling.

3. Methodology

3.1. Experimental Framework

To ensure a comprehensive and reproducible comparison, this study adopts a multi-dimensional experimental framework evaluating CNN, RNN, and Transformer architectures across diverse tasks and datasets. Representative architectures were selected from each category to reflect both classical and modern designs. While large-scale architectures such as ResNet-50, EfficientNet-B0 [6][8], LSTM, GRU for RNN [12][13] and Vision Transformers are discussed in the broader context, the experimental implementation focuses on lightweight yet representative models for controlled comparison [5][17]. All experiments were conducted under standardized configurations, ensuring fairness across models in terms of training conditions, optimization strategies, and evaluation protocols.

3.2. Benchmark Datasets and Data Preparation

To align with real-world applications while maintaining computational feasibility, three benchmark datasets were selected:

- CIFAR-10: Used for image classification tasks, consisting of 60,000 images across 10 classes [7].
- IMDB Dataset: Used for sentiment classification, representing sequential textual data.
- WikiText-2: Used for language modeling tasks, suitable for Transformer-based architectures [15].

Each dataset was divided into: Training Set (70%), Validation Set (15%) and Testing Set (15%). This standardized split ensures consistency and supports reliable performance evaluation.

3.3. Model Configurations

To ensure fairness and reproducibility, all models were implemented using controlled hyperparameters.

3.3.1 CNN Configuration

The CNN model was designed for image classification tasks and widely used for spatial data processing and feature extraction [11]. The implemented model includes:

- 3 Convolutional layers followed by 2 Fully Connected layers
- ReLU activation for non-linearity
- Adam optimizer with learning rate of 0.001
- Batch size of 64
- Training for 25 epochs

This configuration balances performance and computational efficiency [6][8].

3.3.2. RNN (LSTM) Configuration

RNN architectures are effective for sequential data modeling [12]. The LSTM model includes:

- 2 stacked LSTM layers
- 128 hidden units
- Dropout rate of 0.3 to prevent overfitting
- Adam optimizer with learning rate of 0.001
- Batch size of 64
- Training for 20 epochs

This setup enables effective sequence learning while maintaining manageable training complexity. LSTM and GRU models have shown effectiveness in time-series forecasting and sequence prediction tasks [2][4].

3.3.3. Transformer Configuration

Transformer models leverage attention mechanisms for capturing long-range dependencies **[5].** The configuration includes:

- 4 encoder layers
- 8 attention heads
- Embedding dimension of 256
- AdamW optimizer with learning rate of 0.0005
- Batch size of 32
- Training for 15 epochs

This architecture supports parallel computation and enhances contextual understanding. Vision Transformers and large-scale models have shown competitive performance across both vision and natural language processing tasks [17][19].

3.4. Training Protocols

To maintain consistency across experiments, data augmentation techniques such as random cropping and flipping were applied for image data [10]. The optimization used Adam / AdamW optimizers with learning rate scheduling. Early stopping was applied based on validation performance. Hyperparameters were tuned using grid search, and each experiment was repeated three times, and average values were reported to ensure reliability.

3.5. Evaluation Metrics

To provide a holistic evaluation, multiple performance metrics were used:

- Accuracy, Precision, Recall, F1-score → for classification performance
- Training Time → measured in minutes
- Inference Time → measured in milliseconds per sample
- Model Parameters → indicating model complexity

These metrics provide a comprehensive evaluation of both performance and efficiency [27].

4 Results and Discussion

4.1 Quantitative Performance Comparison

Table 1 and Figure 4 demonstrate the comparative performance of models, and their training time respectively.

Table 1. Comparative Performance of Models.

Model	Accuracy (%)	Parameters (Millions)	Training Time (min)	Inference Time (ms)
CNN	88.5	1.2	45	12
RNN (LSTM)	84.2	2.5	60	18
Transformer	91.3	10.5	90	25

Figure 4. Training Time vs Models.

4.2. Accuracy Analysis

Figure 5 shows the accuracy comparison of the models in this study. The Transformer model achieved the highest accuracy (91.3%), demonstrating its effectiveness in capturing complex patterns and long-range dependencies through attention mechanisms [5][15]. CNN achieved competitive performance (88.5%) with significantly fewer parameters [7][8], while RNN showed comparatively lower accuracy (84.2%) due to limitations in handling long-term dependencies [12].

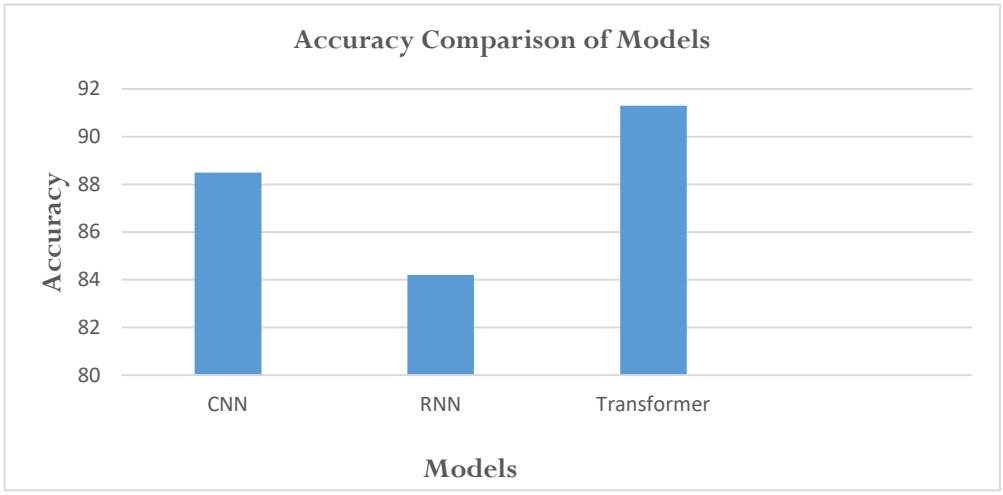

Figure 5. Accuracy comparison of CNN, RNN, and Transformer Models.

4.3. Computational Efficiency

CNN models required the least training time (45 minutes) and achieved the fastest inference (12 ms), making them suitable for real-time applications [9]. RNN models showed moderate computational cost due to sequential processing [2], while Transformers required the highest training time (90 minutes) due to attention computations and model complexity [22].

4.4. Model Complexity

The Transformer model contained the highest number of parameters (10.5M), explaining its superior accuracy but higher computational cost [5]. CNN models maintained a balance with only 1.2M parameters, demonstrating efficiency. RNN models had moderate complexity (2.5M parameters) [27].

4.5. Comparative Insights with Large-Scale Architectures

The experimental findings align with large-scale studies: CNN architectures such as ResNet and EfficientNet demonstrate strong performance with high efficiency [6][8]. RNN variants like LSTM and GRU perform well in time-series tasks but are limited in scalability [2][4]. Meanwhile, Transformer-based models, including Vision Transformers and large language models, achieve state-of-the-art results but require higher computational resources [17][20].

4.6. Overall Discussion

The comparative analysis indicates that firstly, Transformers achieve the highest accuracy due to attention mechanisms [5]. Next, CNNs provide the best balance between performance and efficiency [6]. Besides that, RNNs remain relevant for sequence modeling but are less competitive compared to modern architectures [12]. These findings confirm that model selection should depend on application requirements, dataset size, and computational constraints.

5. Conclusion

This study compares three basic neural network designs from various perspectives, including architectural principles, performance characteristics, computing requirements, and real-world applications. CNNs offer distinct advantages in computer vision tasks due to their efficient processing and hierarchical learning of spatial features. RNNs excel in sequential modeling problems, particularly in streaming and online learning contexts where temporal relationships are involved. Transformers, known for their strong adaptability, utilize self-attention mechanisms to achieve state-of-the-art performance across a wide range of fields.

When selecting an architecture, it is essential to consider task characteristics, data availability, computational limitations, and deployment needs. While CNNs remain highly effective and efficient for many computer vision applications, Transformers are gaining prominence across various domains due to their superior performance at scale. RNNs continue to be relevant in specific use cases where their sequential processing capabilities are advantageous.

Future research goals include developing more effective attention mechanisms, exploring hybrid designs that combine the strengths of different architectures, expanding multimodal learning capabilities, and enhancing energy efficiency. The ongoing evolution of neural network architectures promises the advancement of more sophisticated AI systems capable of tackling complex real-world problems in fields such as autonomous systems, healthcare, and scientific research. As the field progresses, we can expect the emergence of new architectural paradigms that push the boundaries of machine learning capabilities through the integration of concepts from theoretical computer science, neuroscience, and cognitive science.

References

[1] O. E. Korkmaz et al., "Revisiting convolutional design for efficient CNN architectures on edge AI platforms," *Nature Scientific Reports*, vol. 15, 2025. https://doi.org/10.1038/s41598-025-27856-3

[2] A. Yunita et al., "A comparative study of RNN, LSTM, GRU, and hybrid models for time series forecasting," *BMC Research Notes*, vol. 18, no. 1, 2025. https://doi.org/10.1016/j.mex.2025.103462

[3] R. Agarwal, "Inside Transformers: Attention, scaling tricks and emerging alternatives in 2025," *GoCodeo*, June 2025. Available: https://www.gocodeo.com/post/inside-transformers-attention-scaling-tricks-emerging-alternatives-in-2025

[4] S. M. Al-Selwi et al., "RNN-LSTM: From applications to modeling techniques and beyond," *Journal of King Saud University - Computer and Information Sciences*, vol. 36, no. 5, 2024. https://doi.org/10.1016/j.jksuci.2024.102068

[5] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998-6008. https://doi.org/10.48550/arXiv.1706.03762

[6] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Machine Learning*, 2019, pp. 6105-6114. https://doi.org/10.48550/arXiv.1905.11946

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105. https://doi.org/10.1145/3065386

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90.

[9] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017. https://doi.org/10.48550/arXiv.1704.04861

[10] S. Khan et al., "Deep learning approaches for medical image analysis and diagnosis," *Journal of Medical Imaging*, vol. 11, no. 3, 2024. https://doi.org/10.7759/cureus.59507

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015. https://doi.org/10.1038/nature14539

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. https://doi.org/10.1162/neco.1997.9.8.1735.

[13] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods in Natural Language Processing*, 2014, pp. 1724-1734. https://doi.org/10.48550/arXiv.1406.1078

[14] O. Baker et al., "Development of GRU, LSTM, and recurrent neural networks for chatbot applications," in *Proc. IEEE Int. Conf. Computing and Communications*, 2024.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT*, 2019, pp. 4171-4186. https://doi.org/10.48550/arXiv.1810.04805

[16] R. Ahmed, "What changed in the Transformer architecture: Evolution from 2017 to 2025," *Hugging Face Blog*, March 2025. Available: https://huggingface.co/blog/rishiraj/what-changed-in-the-transformer-architecture

[17] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learning Representations*, 2021.https://doi.org/10.48550/arXiv.2010.11929

[18] M. Chen, "Vision Transformers (ViT) in image recognition: Redefining computer vision," *Viso.ai*, September 2025. Available: https://viso.ai/deep-learning/vision-transformer-vit/

[19] S. Kumar and R. Patel, "Vision Transformers for remote sensing applications," *Indian Institute of Remote Sensing*, February 2025. Available: https://science.iirs.gov.in/vision-transformers-for-remote-sensing-applications/

[20] A. Grattafiori et al., "The LLaMA 3 herd of models," *Meta AI Research*, July 2024. Available: https://ai.meta.com/research/publications/

[21] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the Kinetics dataset," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2017, pp. 4724-4733. https://doi.org/10.48550/arXiv.1705.07750

[22] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Re, "FlashAttention: Fast and memory-efficient exact attention with IO-awareness," in *Advances in Neural Information Processing Systems*, 2022. https://doi.org/10.48550/arXiv.2205.14135

[23] P. Rajpurkar et al., "CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning," *arXiv preprint arXiv:1711.05225*, 2017. https://doi.org/10.48550/arXiv.1711.05225

[24] S. Hong et al., "Opportunities and challenges of deep learning methods for electrocardiogram data," *Computers in Biology and Medicine*, vol. 122, 2020. https://doi.org/10.1016/j.compbiomed.2020.103801

[25] M. Zhang et al., "Vision Transformers in medical imaging: A comprehensive review," *Journal of Healthcare Engineering*, 2025. https://doi.org/10.1007/s10278-025-01481-y

[26] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. Int. Conf. Learning Representations*, 2017. https://doi.org/10.48550/arXiv.1611.01578

[27] Y. Cheng et al., "Model compression and acceleration for deep neural networks," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126-136, 2018. https://doi.org/10.1109/MSP.2017.2765695