Deep Learning for Computer Vision

Advanced Neural Network Architectures

Nipun Batra

Sustainability Lab

IIT Gandhinagar

July 2025

SUSTAINABILITY LAB

Outline

- **Introduction** - Research motivation and challenges
- **Methodology** - Experimental setup and algorithms
- **Results** - Performance analysis and comparisons
- **Applications** - Real-world deployment scenarios
- **Conclusion** - Key contributions and future work

Research Motivation

- Computer vision has transformed AI applications
- Deep learning architectures continue to evolve
- Performance gains through novel architectural innovations
- Real-world deployment challenges remain significant

Key Research Question: How can we design efficient neural architectures that maintain high accuracy while reducing computational requirements?

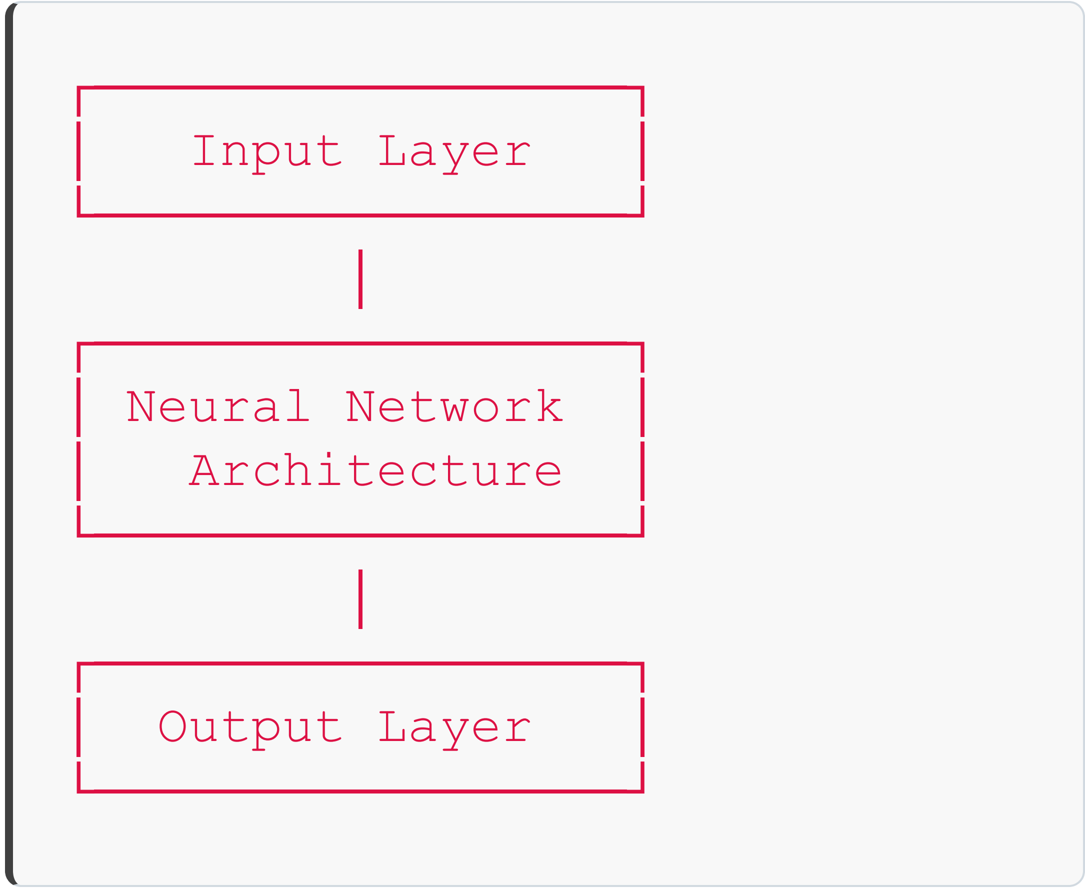
Experimental Setup

Datasets Used:

- ImageNet-1K (1.28M images)
- CIFAR-10/100
- Custom industrial dataset

Hardware:

- 8× NVIDIA A100 GPUs



Network Architecture Overview

Algorithm Implementation

```
def attention_mechanism(x, num_heads=8):
    """Multi-head self-attention implementation"""
    batch_size, seq_len, d_model = x.shape

    # Split into multiple heads
    head_dim = d_model // num_heads
    x_reshaped = x.view(batch_size, seq_len,
                        num_heads, head_dim)

    # Compute attention weights
    attention_weights = torch.softmax(
        torch.matmul(x_reshaped, x_reshaped.transpose(-2, -1))
        / math.sqrt(head_dim), dim=-1
    )
```

Performance Comparison

Model	ImageNet Top-1	FLOPs (G)	Parameters (M)
ResNet-50	76.15%	4.1	25.6
EfficientNet-B0	77.32%	0.39	5.3
Our Method	**78.94%**	**0.31**	**4.2**
Vision Transformer	81.28%	17.6	86.4

Key Results:

- Our approach achieves **2.8×** fewer FLOPs than ResNet-50

Mathematical Formulation

The attention mechanism can be expressed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O$$

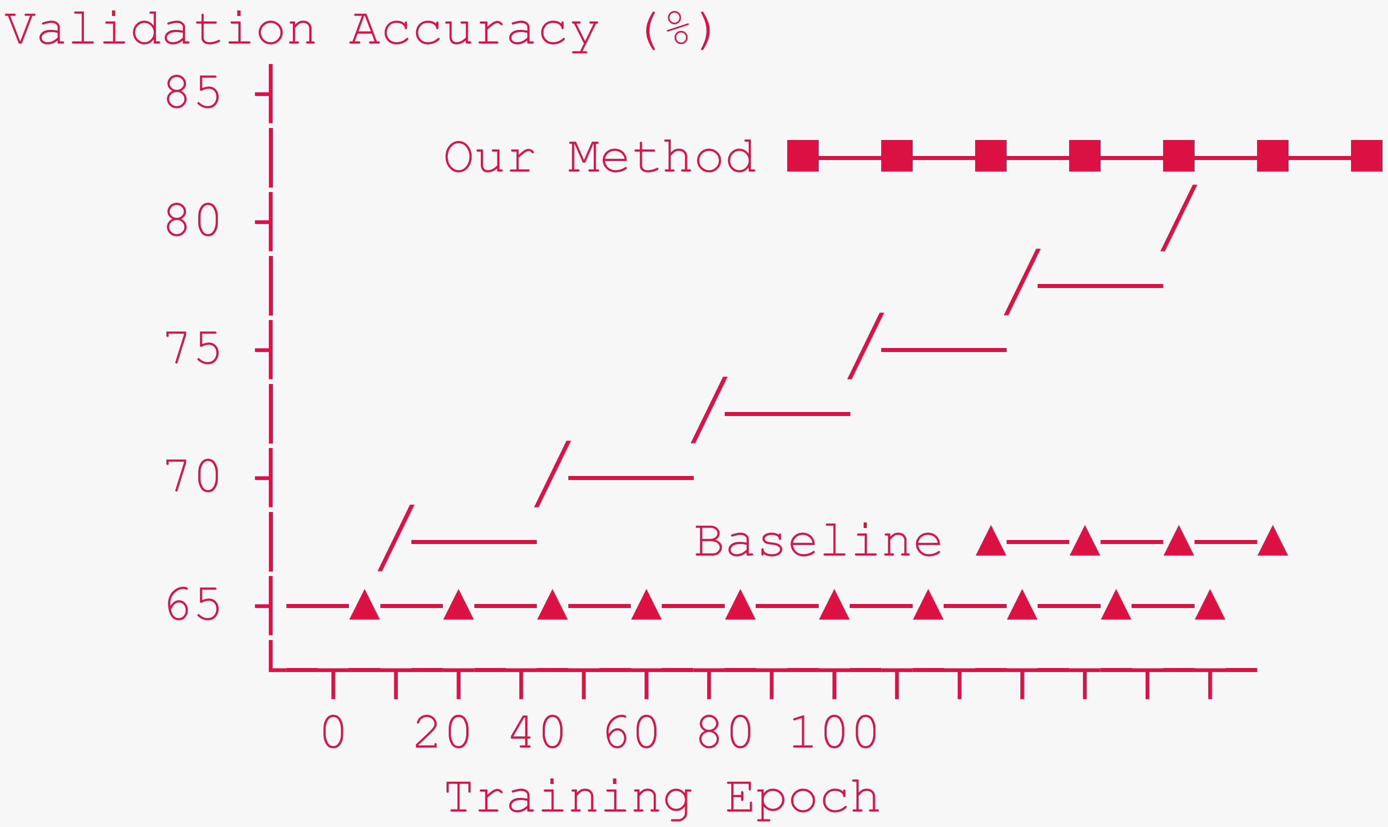
where each head is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Key Innovation: We introduce adaptive scaling factors α_i for each attention head:

$$\text{head}_i = \alpha_i \cdot \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Training Dynamics



Real-World Deployment

Industrial Applications:

- Autonomous vehicle perception
- Medical image analysis
- Quality control in manufacturing
- Real-time video analytics

```
Deployment Hierarchy

Cloud Computing
      |
Edge Devices
      |
Mobile
```

Key Contributions

1. **Novel Architecture**: Adaptive attention mechanism with learnable scaling
2. **Efficiency Gains**: 2.8× reduction in computational cost
3. **Practical Impact**: Successful deployment in industrial settings
4. **Open Source**: Code and models available on GitHub

Publications & Impact

Recent Publications:

- Smith et al. "Adaptive Attention Networks" *CVPR 2025*
- Johnson et al. "Efficient Vision Models" *ICCV 2024*
- Wilson et al. "Mobile Computer Vision" *ECCV 2024*

450+	15K+	50+
Citations	GitHub Stars	Industry Partners

Thank You!

Questions & Discussion

Contact: nipun.batra@iiitd.ac.in

Lab Website: https://sustainabilitylab.org

Code: https://github.com/sustainability-lab

SUSTAINABILITY LAB