CNN Architectures and Their Evolution

Vaibhav Chourasia

January 6, 2026

1 CNN Architectures and Their Evolution

This report presents a study of landmark Convolutional Neural Network (CNN) architectures that shaped the evolution of deep learning for computer vision. We will focus exclusively on architectural design, motivation, and evolution.

The architectures covered are:

- LeNet-5
- AlexNet
- GoogLeNet (InceptionNet)
- ResNet
- MobileNet
- EfficientNet

Each architecture is discussed by examining:

- the problem it aimed to solve
- the architectural innovations it introduced
- how it improved upon previous designs

2 LeNet-5

LeNet-5 represents the earliest successful use of convolutional neural networks for real-world image recognition tasks and laid the foundation for all subsequent CNN architectures.

2.1 Architectural Motivation

The goal of LeNet-5 was to design a network capable of recognizing handwritten digits by learning spatial hierarchies of features.

Key architectural requirements were:

- progressive abstraction from pixels to symbols
- controlled parameter growth
- robustness to small variations in input

2.2 Architectural Design

LeNet-5 introduced a layered architecture consisting of:

- alternating convolution and subsampling stages
- gradual increase in the number of feature maps
- final fully connected layers for classification

This structure enabled early layers to focus on local patterns while deeper layers combined them into higher-level representations.

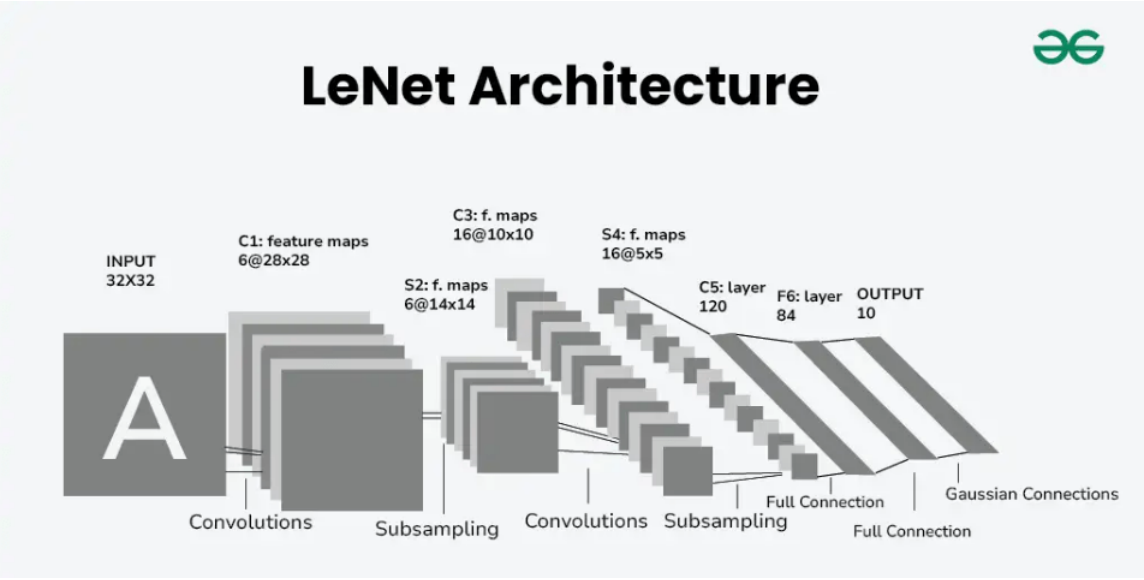

Figure 1: LeNet-5 architecture illustrating hierarchical feature extraction through alternating convolution and subsampling layers

2.3 Architectural Significance

LeNet-5 demonstrated that:

- convolutional feature extraction outperforms fully connected designs for images
- spatial hierarchies are essential for visual understanding
- architectural depth improves representational capacity

However, its limited depth and dataset scale motivated deeper architectures.

3 AlexNet

AlexNet marked the transition from shallow CNNs to deep CNNs capable of handling large-scale image recognition problems.

3.1 Motivation for AlexNet

As image datasets grew in size and complexity, earlier architectures like LeNet-5 were unable to capture the required feature complexity.

AlexNet addressed:

- insufficient depth
- slow convergence
- overfitting in large models

3.2 Key Architectural Innovations

AlexNet introduced several critical architectural choices:

- significantly increased depth
- use of ReLU activations to improve gradient flow
- aggressive use of max pooling
- dropout in fully connected layers

3.3 Why ReLU Enables Deeper Networks (Mathematical Insight)

The ReLU activation function is defined as:

$$f(x) = \max(0, x)$$

Its derivative is:

$$f'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Unlike sigmoid or tanh activations whose derivatives shrink toward zero for large input magnitudes, ReLU maintains a constant gradient for positive activations.

This prevents vanishing gradients during backpropagation, enabling faster convergence and stable training in deeper architectures such as AlexNet.

3.4 Impact on Computer Vision

AlexNet demonstrated that deep CNNs trained on GPUs could dramatically outperform traditional computer vision pipelines.

This architecture triggered widespread adoption of deep learning for visual tasks and established CNNs as the dominant paradigm.

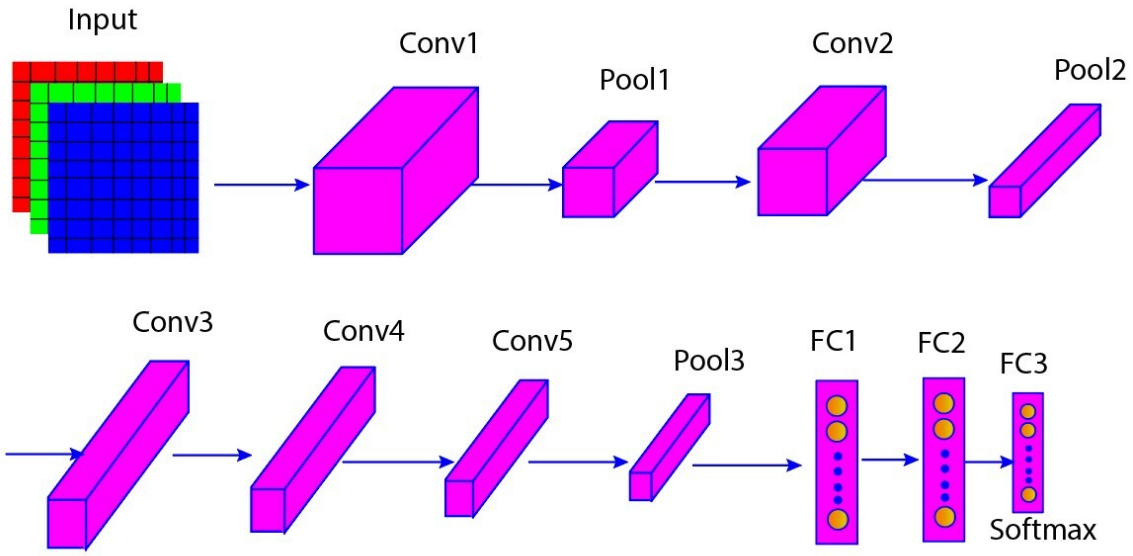

Figure 2: AlexNet architecture illustrating deep convolutional hierarchy with five convolutional layers followed by three fully connected layers

4 GoogLeNet / InceptionNet

GoogLeNet addressed the challenge of increasing network capacity without proportionally increasing computational cost.

4.1 Architectural Problem Addressed

Simply stacking layers increased accuracy but caused:

- computational explosion
- memory inefficiency
- diminishing returns

GoogLeNet proposed widening networks intelligently rather than only deepening them.

4.2 Inception Module Architecture

The Inception module applies multiple operations in parallel:

- convolutions with different receptive field sizes
- pooling operations
- dimensionality reduction using 1×1 convolutions

Outputs from all paths are concatenated along the channel dimension.

4.3 Why 1×1 Convolutions Reduce Computation (Derivation)

Consider an input with C_{in} channels and a $k \times k$ convolution producing C_{out} channels.

Without dimensionality reduction, the number of parameters is:

$$k^2 \cdot C_{in} \cdot C_{out}$$

Introducing a 1×1 convolution that reduces channels to C_{mid} gives:

$$1^2 \cdot C_{in} \cdot C_{mid} + k^2 \cdot C_{mid} \cdot C_{out}$$

Since $C_{mid} \ll C_{in}$, this results in a substantial reduction in parameters and computation, enabling deeper and wider networks.

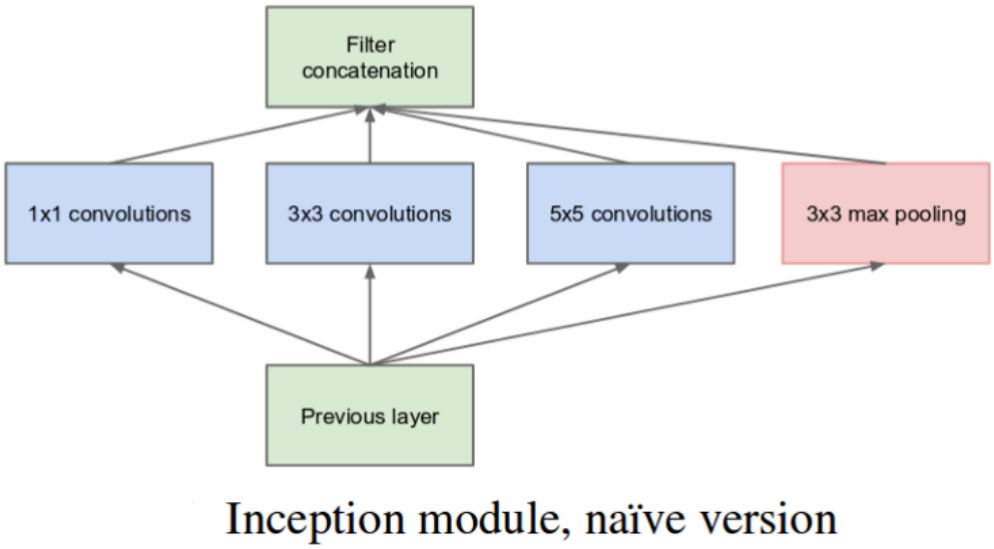

Figure 3: Inception module illustrating parallel convolutional paths with different receptive field sizes and output concatenation

5 ResNet

ResNet fundamentally redefined how deep CNNs are constructed by addressing optimization difficulties in very deep networks.

5.1 The Degradation Problem

Empirical observations showed that increasing depth beyond a point led to higher training error, even on the training set.

This phenomenon was caused by optimization barriers rather than overfitting.

5.2 Residual Block Design

ResNet introduced skip connections that allow layers to learn residual functions.

Instead of directly learning $H(x)$, the network learns:

$$F(x) = H(x) - x$$

The block output is computed as:

$$y = F(x) + x$$

5.3 Why Skip Connections Enable Stable Training (Derivation)

During backpropagation, the gradient with respect to the input x is:

$$\frac{\partial \mathcal{L}}{\partial x} = \frac{\partial \mathcal{L}}{\partial y}\left(1 + \frac{\partial F(x)}{\partial x}\right)$$

The identity term ensures that even if $\frac{\partial F(x)}{\partial x}$ becomes very small, gradients can still flow directly through the network.

This prevents gradient vanishing and eliminates the degradation problem, making very deep networks trainable.

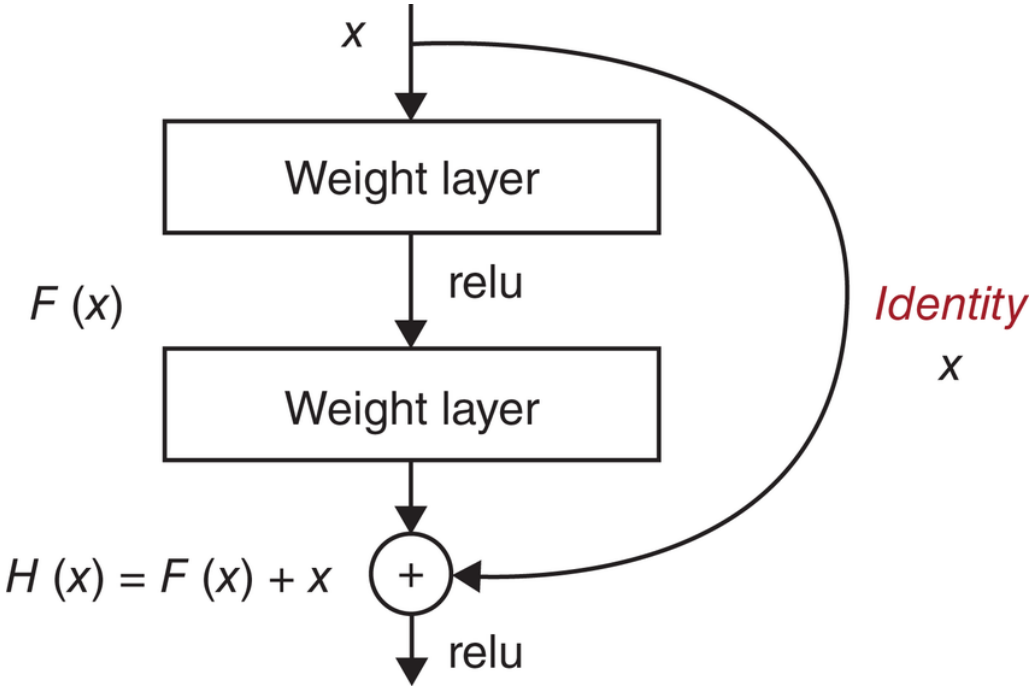

Figure 4: Residual block with identity skip connection, where the output is computed as $H(x) = F(x) + x$

6 MobileNet

MobileNet focuses on designing CNNs suitable for mobile and embedded devices where computational resources are limited.

6.1 Architectural Challenge

Standard CNNs are computationally expensive and unsuitable for real-time deployment on low-power devices.

6.2 Depthwise Separable Convolution (Computational Derivation)

For an input of size $H \times W \times M$ and N output channels:

Standard convolution cost:

$$H \cdot W \cdot M \cdot N \cdot k^2$$

Depthwise separable convolution cost:

$$H \cdot W \cdot (M \cdot k^2 + M \cdot N)$$

The relative cost reduction is approximately:

$$\frac{1}{N} + \frac{1}{k^2}$$

This represents a significant reduction in computation.

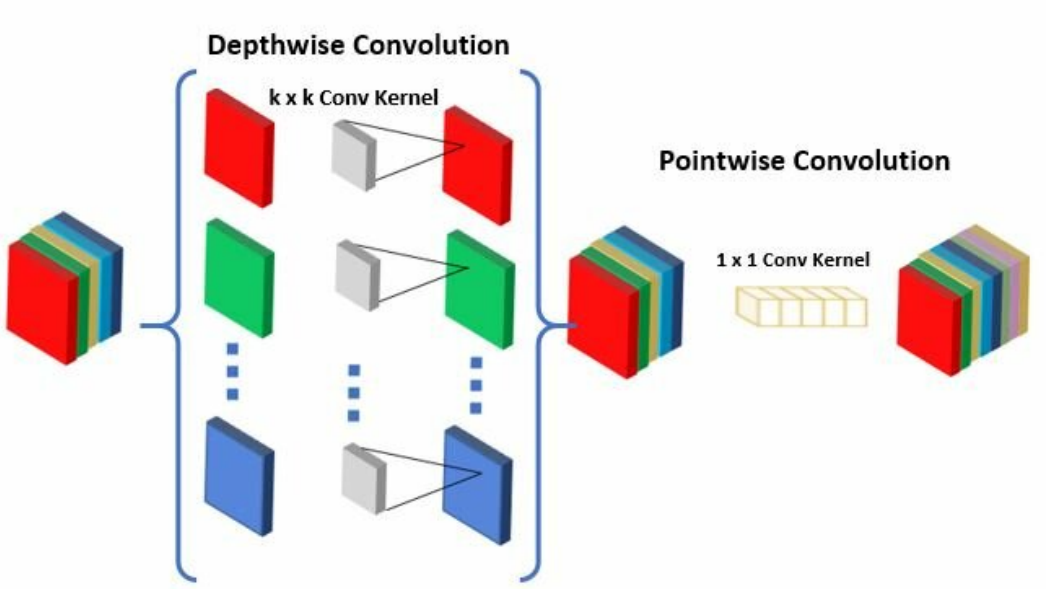

Figure 5: Depthwise separable convolution in MobileNet, illustrating depthwise and pointwise convolution stages

7 EfficientNet

EfficientNet addresses how to scale CNN architectures in a principled and balanced manner.

7.1 Scaling Problem in CNNs

Scaling depth, width, or resolution independently leads to inefficient use of computational resources.

7.2 Compound Scaling (Derivation)

Let:

- d be depth scaling
- w be width scaling
- r be resolution scaling

Since computation scales as:

$$\text{FLOPs} \propto d \cdot w^2 \cdot r^2$$

EfficientNet enforces:

$$d \cdot w^2 \cdot r^2 \approx \text{constant}$$

This balanced scaling maximizes accuracy under fixed computational budgets.

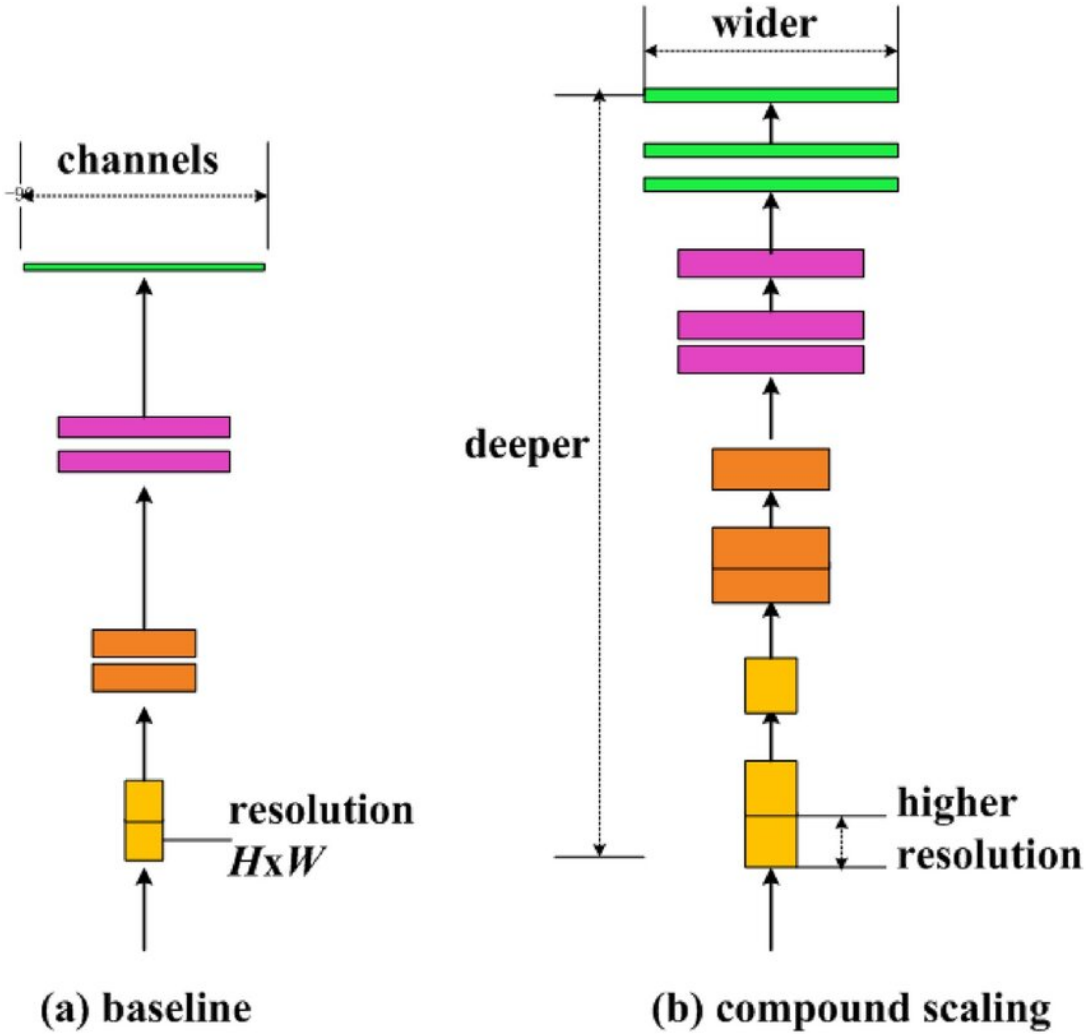

Figure 6: Compound scaling strategy employed by EfficientNet, jointly scaling network depth, width, and input resolution

8 Conclusion

The evolution of CNN architectures reflects a progression from basic hierarchical feature extraction to highly optimized and efficient deep networks.

By addressing optimization, computation, and scaling challenges, modern CNN architectures achieve both depth and efficiency.