Lightweight Model Design

- Objective: Minimize latency and memory usage for resource-constrained devices.
- Strategies:
  - Depthwise separable convolutions.
  - Reduced parameter count and FLOPs (floating-point operations).
  - Model tuning using hyperparameters:
    - Width Multiplier (α): Adjusts layer width.
    - Resolution Multiplier (ρ): Scales input image resolution.

MobileNet Overview

- MobileNet is a lightweight convolutional neural network designed for mobile and embedded vision applications.
- Developed by Google to balance performance and computational efficiency.
- Utilizes depth-wise separable convolutions to reduce computation.

Standard Convolution

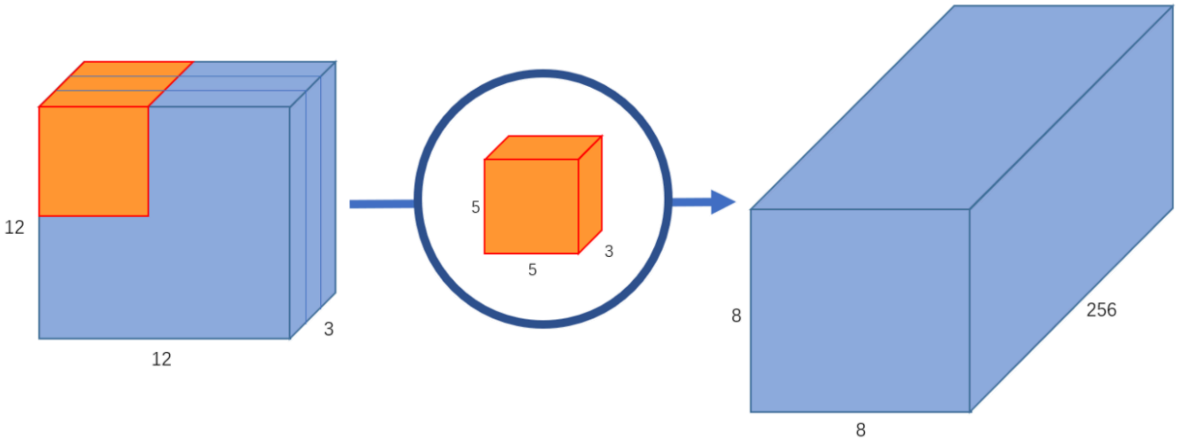

Figure 1: Standard convolution with 8x8x256 output

Steps involved:

- Applying a 3D kernel across spatial dimensions.
- Combining depth channels for feature extraction.

Equation:

$$y(h', w', c') = \sum_{c=1}^{C} \sum_{m=1}^{M} \sum_{n=1}^{N} x(h' + m, w' + n, c) \cdot k_{c'}(m, n, c)$$

where:

- x: Input feature map.
- k_i: convolution kernel.
- y: Output feature map.

Metrics

- Input: $H \cdot W \cdot C$
- Output: $H' \cdot W' \cdot C'$
- Parameters: $M \cdot N \cdot C \cdot C'$
- Operations: $(H' \cdot W' \cdot C') \cdot (M \cdot N \cdot C)$

Depth-wise Separable Convolution

Idea

Break a standard convolution into two parts:

1. Depth-wise Convolution: Apply a single filter per input channel.
2. Point-wise Convolution: Use a 1x1 convolution to combine features across channels.

Advantages:

- Reduces computation and parameters.
- Speeds up inference.

Equation

- Depth-wise Convolution

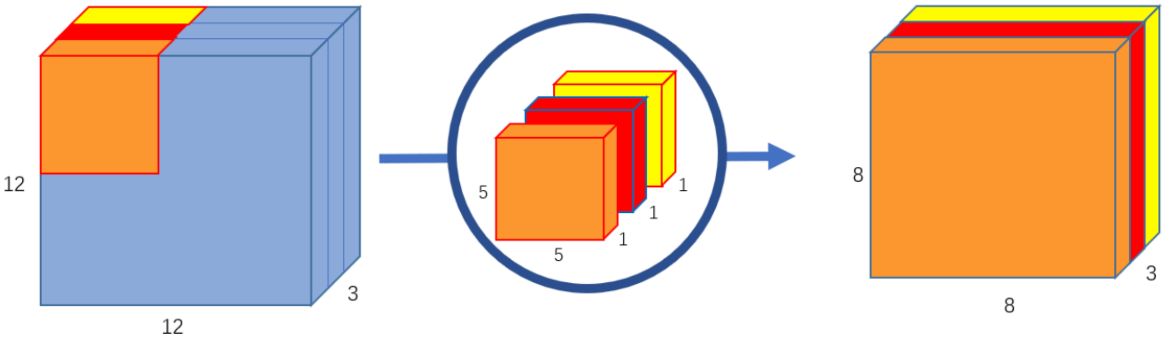

Figure 2: Depth-wise convolution, use 3 kernels to transform a 12x12x3 image to a 8x8x3 image

$$y(h', w', c) = \sum_{m=1}^{M} \sum_{n=1}^{N} x(h' + m, w' + n, c) \cdot k_c(m, n, 1)$$

where:

  - x: Input feature map.
  - k_i: convolution kernel.
  - y: Output feature map.

- Metrics
  - Input: $H \cdot W \cdot C$
  - Output: $H' \cdot W' \cdot C$
  - Parameters: $M \cdot N \cdot C$
  - Operations: $(H' \cdot W' \cdot C) \cdot (M \cdot N)$
- Point-wise Convolution Equation:

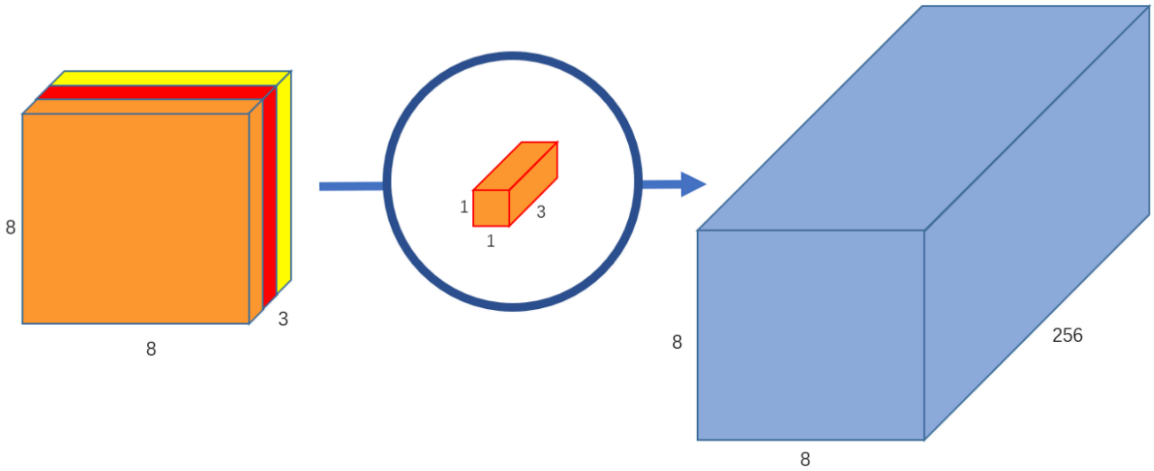

Figure 3: Point-wise convolution with 256 kernels, outputting an image with 256 channels

$$y(h', w', c') = \sum_{c=1}^{C} y(h', w', c) \cdot k_{c'}(1, 1, c)$$

where:

  - x: Input feature map.
  - k_i: convolution kernel.
  - y: Output feature map.

- Metrics
  - Input: $H' \cdot W' \cdot C$
  - Output: $H' \cdot W' \cdot C'$
  - Parameters: $C \cdot C'$
  - Operations: $(H' \cdot W' \cdot C') \cdot C$

Combined Metrics:

- Input: $H \cdot W \cdot C$
- Output: $H' \cdot W' \cdot C'$
- Parameters: $M \cdot N \cdot C + C \cdot C'$
- Operations: $(H' \cdot W' \cdot C) \cdot (M \cdot N + C')$

Compare to Standard Convolution

Parameters

$$\begin{aligned}\frac{\text{Params (Depthwise Separable)}}{\text{Params (Standard)}} &= \frac{M \cdot N \cdot C + C \cdot C'}{M \cdot N \cdot C \cdot C'} \\ &= \frac{1}{C'} + \frac{1}{M \cdot N}\end{aligned}$$

Operations

$$\begin{aligned}\frac{\text{Ops (Depthwise Separable)}}{\text{Ops (Standard)}} &= \frac{(H' \cdot W' \cdot C) \cdot (M \cdot N + C')}{(H' \cdot W' \cdot C') \cdot (M \cdot N \cdot C)} \\ &= \frac{1}{C'} + \frac{1}{M \cdot N}\end{aligned}$$

Convolution significantly reduces both parameter count and operation count, approximately by a factor of:

$$\text{Reduction Factor } = \frac{1}{C'} + \frac{1}{M \cdot N}$$

MobileNet V1

- Introduced depth-wise separable convolutions to replace standard convolutions.
- Achieved significant reductions in:
  - Model size (parameters).
  - Computation (FLOPs).

MobileNet V2

- Improvement over V1 with better accuracy and efficiency.

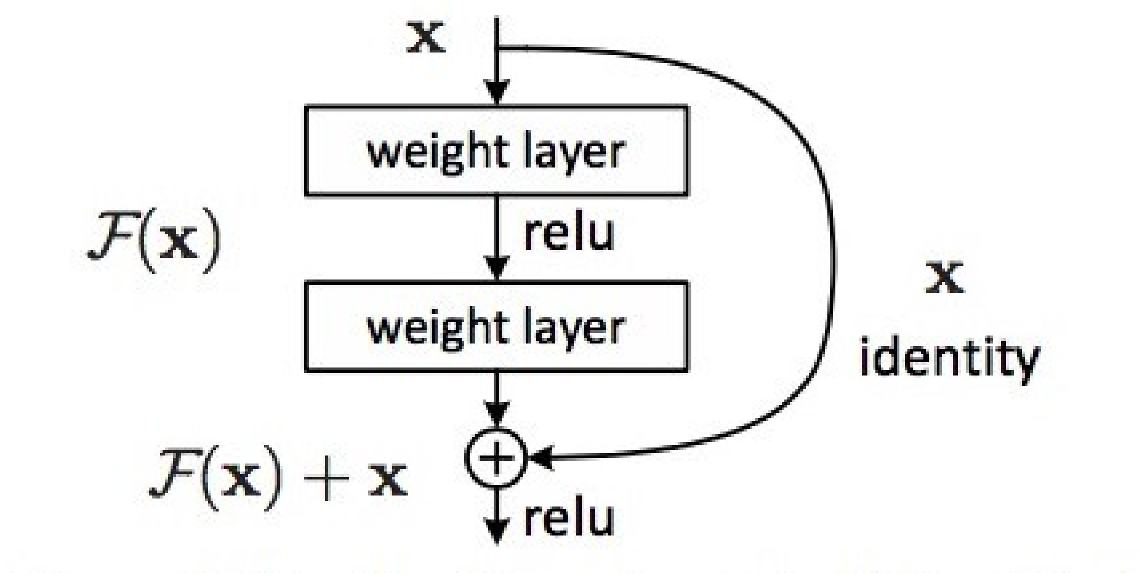
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 4: Residual block

- Introduced the Inverted Residual Block:
  - Expands input with point-wise convolution.
  - Applies depth-wise convolution.
  - Uses point-wise linear projection to compress the output.
- Linear Bottlenecks prevent information loss from non-linearities.

Comparison:

- MobileNet V2 achieves higher accuracy with fewer parameters compared to V1.

Applications of MobileNet

- Real-time object detection on mobile devices.
- Image classification for embedded systems.
- Facial recognition in AR/VR systems.
- Autonomous driving and robotics.

Conclusion

- MobileNet is a pioneering approach to designing lightweight neural networks.
- Depth-wise separable convolutions enable efficient computation.
- MobileNet V2 builds on V1 with improved architecture for better accuracy.