Lecture 7: Convolutional Neural Network (CNN) – Advanced CNN Architectures

Areej Alasiry

CIS 6217 – Computer Vision for Data Representation

College of Computer Science, King Khalid University

Outline

1. LeNet – 5
2. AlexNet
3. VGGNet
4. GoogLeNet
5. ResNet

Learning Outcomes

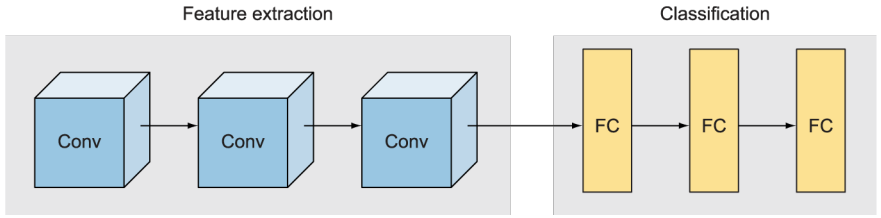
- Identify major CNN architectures and their design evolution.
- Describe the unique features and motivations behind LeNet, AlexNet, VGG, Inception, and ResNet.
- Analyze architectural trade-offs: depth, performance, and computational complexity.
- Recognize how innovations like 1×1 convolutions, inception modules, and skip connections changed modern CV models.

Overview

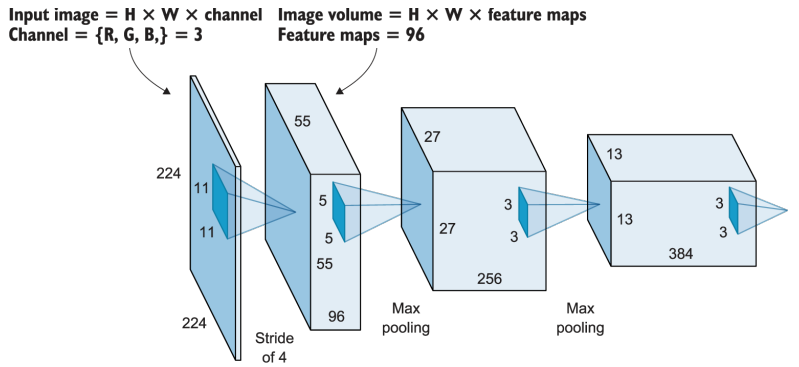
- The field of computer vision has progressed rapidly through continuous innovation in Convolutional Neural Networks (CNNs).
- Each new CNN architecture built upon the limitations of its predecessors — improving depth, accuracy, efficiency, and training stability.
- This lecture explores five landmark CNN architectures that shaped modern vision systems:
  - LeNet (1998) – The pioneer of CNNs.
  - AlexNet (2012) – The deep learning breakthrough.
  - VGGNet (2014) – Simplicity and uniform depth.
  - Inception/GoogLeNet (2015) – Multi-scale feature learning.
  - ResNet (2015) – Solving the vanishing gradient problem.

CNN Design Patterns

- Pattern 1: Feature extraction and classification



- Pattern 2: Image depth increases, and dimensions decrease



- Pattern 3: Fully Connected Layers

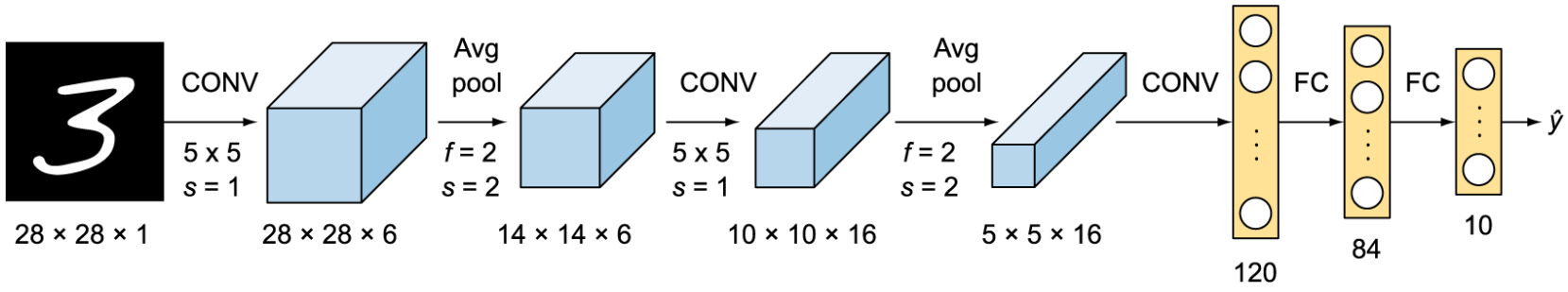
Deep Learning for Vision Systems by Mohamed Elgendy (2020)

LeNet-5

In 1998, Lecun et al. introduced a pioneering CNN called LeNet-5

LeNet Architecture

Layer	Type / Function	Output Size	Purpose
Input	32 × 32 grayscale image	32×32×1	Handwritten digit
C1	Convolution (6 × 5×5 kernels)	28×28×6	Extract local features
S2	Subsampling / Average Pooling	14×14×6	Reduce spatial dimensions



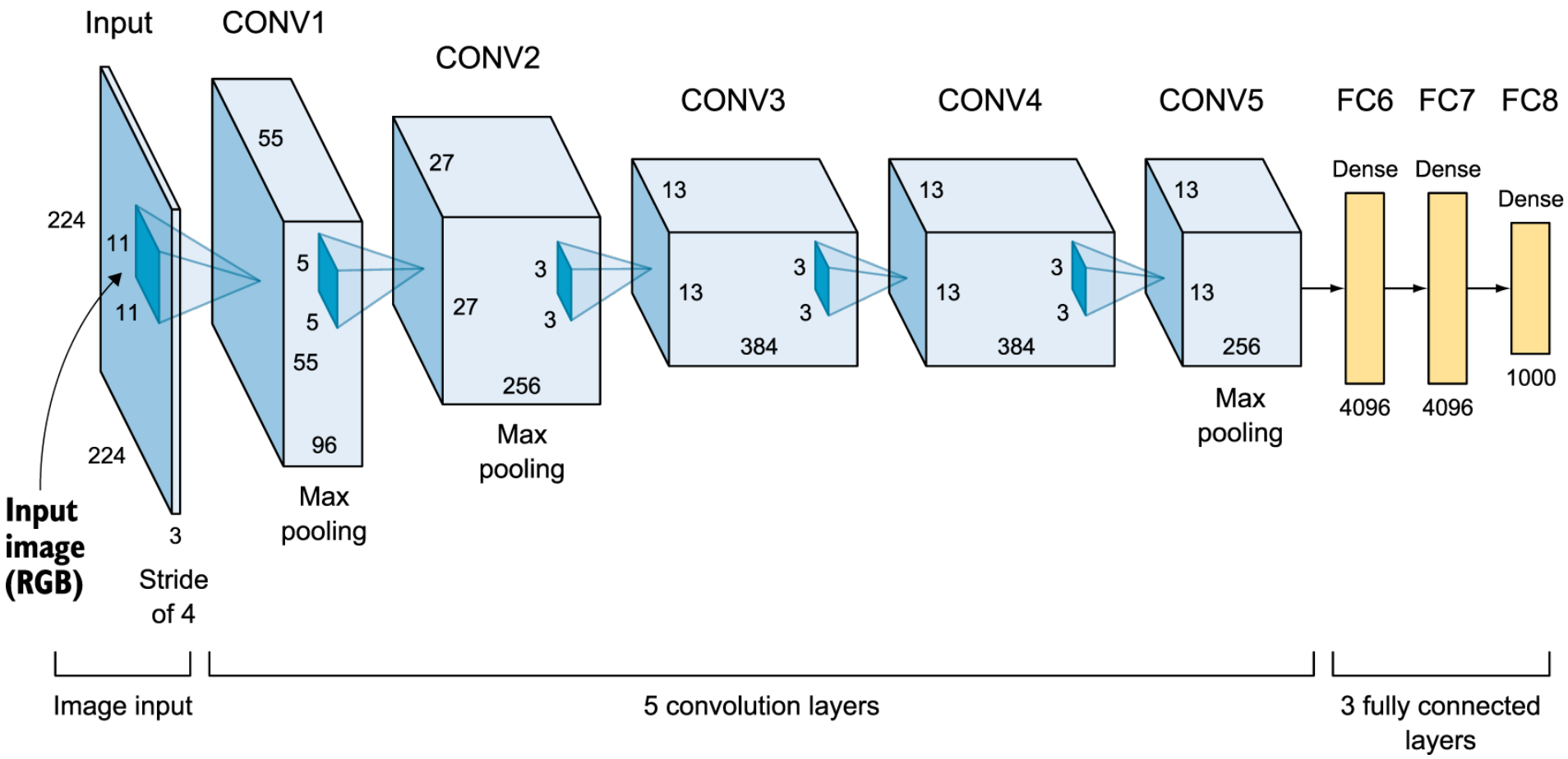
Deep Learning for Vision Systems by Mohamed Elgendy (2020)

AlexNet

The winner of the ILSVRC image classification competition in 2012

AlexNet

Deep Learning for Vision Systems by Mohamed Elgendy (2020)



AlexNet

Layer	Type / Operation	Kernel / Units	Stride / Padding	Output Volume	Remarks
Input	Image	227×227×3	–	–	RGB image input
Conv1	Convolution + ReLU	96 × 11×11	Stride 4	55×55×96	Large receptive field, aggressive stride
Pool1	Max Pooling	3×3	Stride 2	27×27×96	Downsampling
Conv2	Convolution + ReLU	256 × 5×5	Pad 2	27×27×256	Split across 2 GPUs
Pool2	Max Pooling	3×3	Stride 2	13×13×256	–
Conv3	Convolution + ReLU	384 × 3×3	Pad 1	13×13×384	Combined across GPUs
Conv4	Convolution + ReLU	384 × 3×3	Pad 1	13×13×384	–
Conv5	Convolution + ReLU	256 × 3×3	Pad 1	13×13×256	–
Pool3	Max Pooling	3×3	Stride 2	6×6×256	Final conv feature map
FC6	Fully Connected	4096	–	4096	Dropout applied
FC7	Fully Connected	4096	–	4096	Dropout applied
FC8	Fully Connected + Softmax	1000	–	1000	ImageNet classes

RELU Activation Funtion

- Used Rectified Linear Units (ReLU) instead of sigmoid or tanh.
- Formula: $f(x)$=max(0,x)
- Benefits:
  - Accelerates convergence (6× faster than tanh).
  - Reduces vanishing gradient problem.
  - Introduces sparsity in activations.

Data Augmentation

- Artificially increases dataset size to reduce overfitting.
- Techniques used:Random cropping and horizontal flipping.
- Color jittering (adjusting brightness, contrast, and saturation).
- Helps model generalize better to unseen data.

Local Response Normalization

- Encourages competition among neuron activations across feature maps.
- Mimics biological lateral inhibition — enhances prominent features, suppresses weaker ones.

- Two main types:
- Intra-channel LRN: Normalizes across neighboring neurons within the same feature map channel.
- Inter-channel LRN: Normalizes across different feature map channels at the same spatial location.

Weight Regularization

- **Two regularization techniques** were used in AlexNet to reduce overfitting and improve generalization:
  - L2 Regularization (Weight Decay)
    - Applied to all convolutional and fully connected layers.
    - Weight decay value: $\lambda = 0.0005$.
  - Dropout

  - Applied in the fully connected layers (FC6 and FC7) with a 50% dropout rate.
  - Randomly disables neurons during training to prevent co-adaptation.
  - Improves the network's ability to generalize to unseen data.

Multi-GPU Training

- Trained across two NVIDIA GTX 580 GPUs — each processed half the model (feature maps split).
- Enabled training of 60 million parameters efficiently.
- Pioneered parallel deep learning computation.

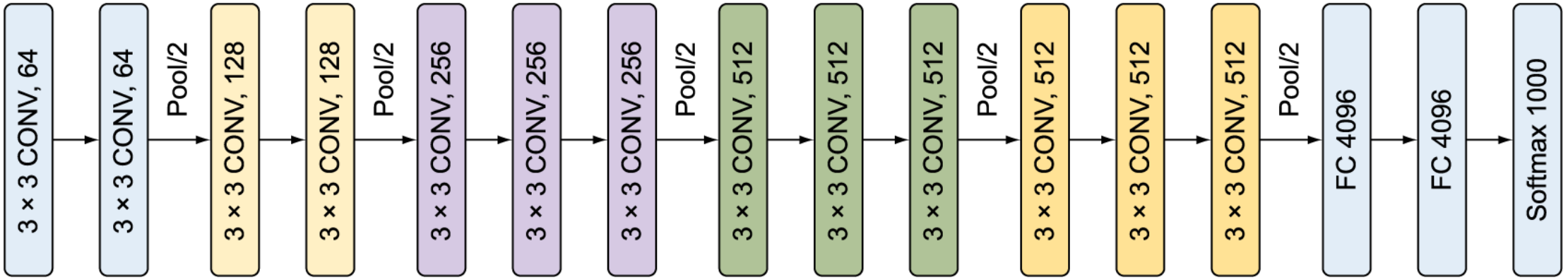
Summary of AlexNet

Technique	Purpose	Impact
ReLU	Faster, stable learning	Solved vanishing gradient issues
Data Augmentation	Improved generalization	Reduced overfitting
LRN	Enhanced feature selectivity	Sharper feature maps
Dropout	Regularization	Prevented co-adaptation
Multi-GPU	Computational scalability	Enabled deep networks

VGGNet

VGGNet was developed in 2014 by the Visual Geometry Group at Oxford University

VGGNet



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Novel Features

- The architecture is composed of a series of uniform convolutional building blocks followed by a unified pooling layer, where:
  - All convolutional layers are 3 × 3 kernel-sized filters with a strides value of 1 and a padding value of same.
  - All pooling layers have a 2 × 2 pool size and a strides value of 2.

Why smaller kernels ?

To avoid overfitting

- Two techniques
  - L2 regularization with weight decay of 5×10^{-4}. For simplicity, this is not added to the implementation that follows.
  - Dropout regularization for the first two fully connected layers, with a dropout ratio set to 0.5.

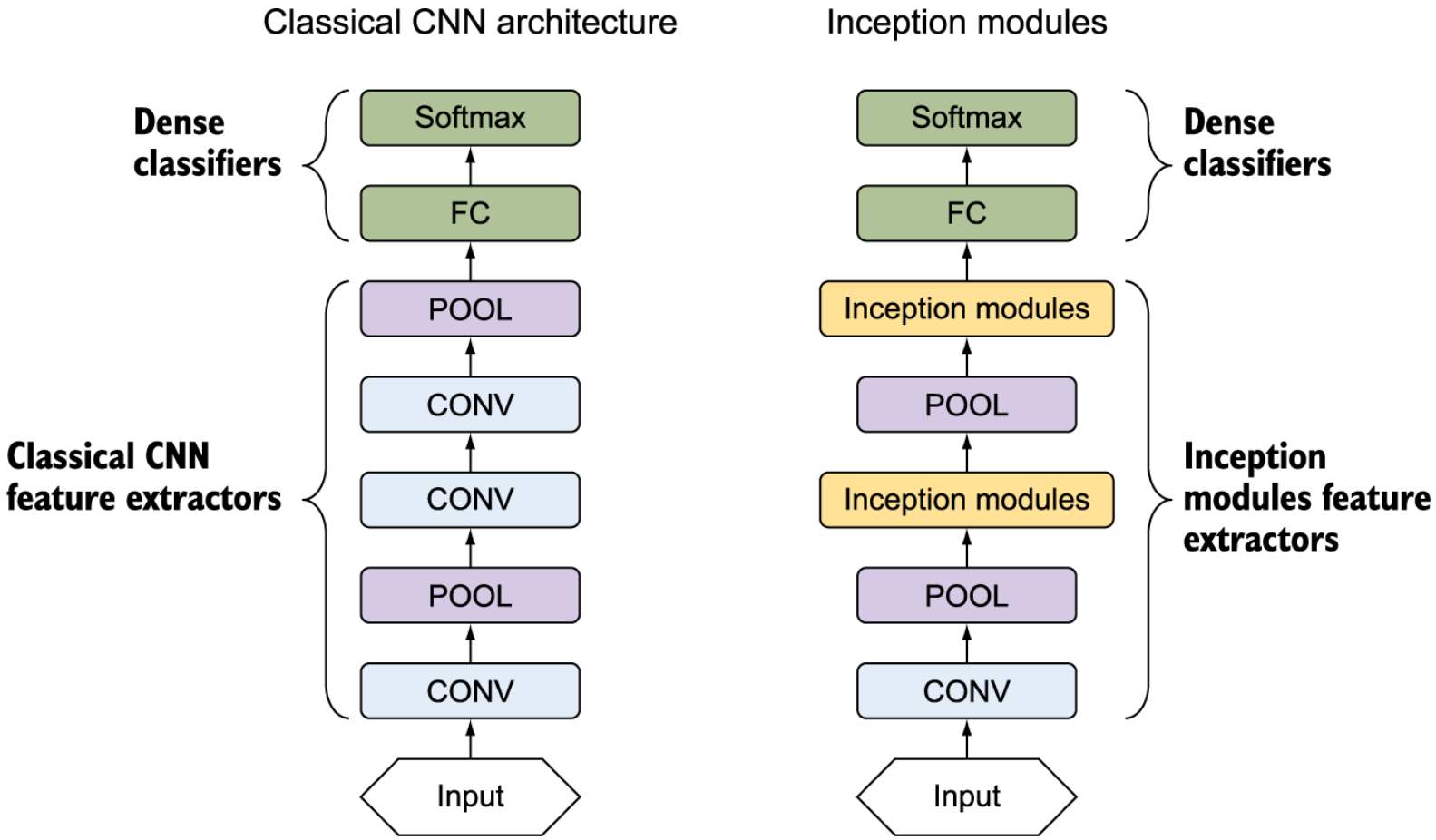
Inception and GoogLeNet

The Inception network came to the world in 2014 when a group of researchers at Google published their paper, “Going Deeper with Convolutions.”

GoogLeNet

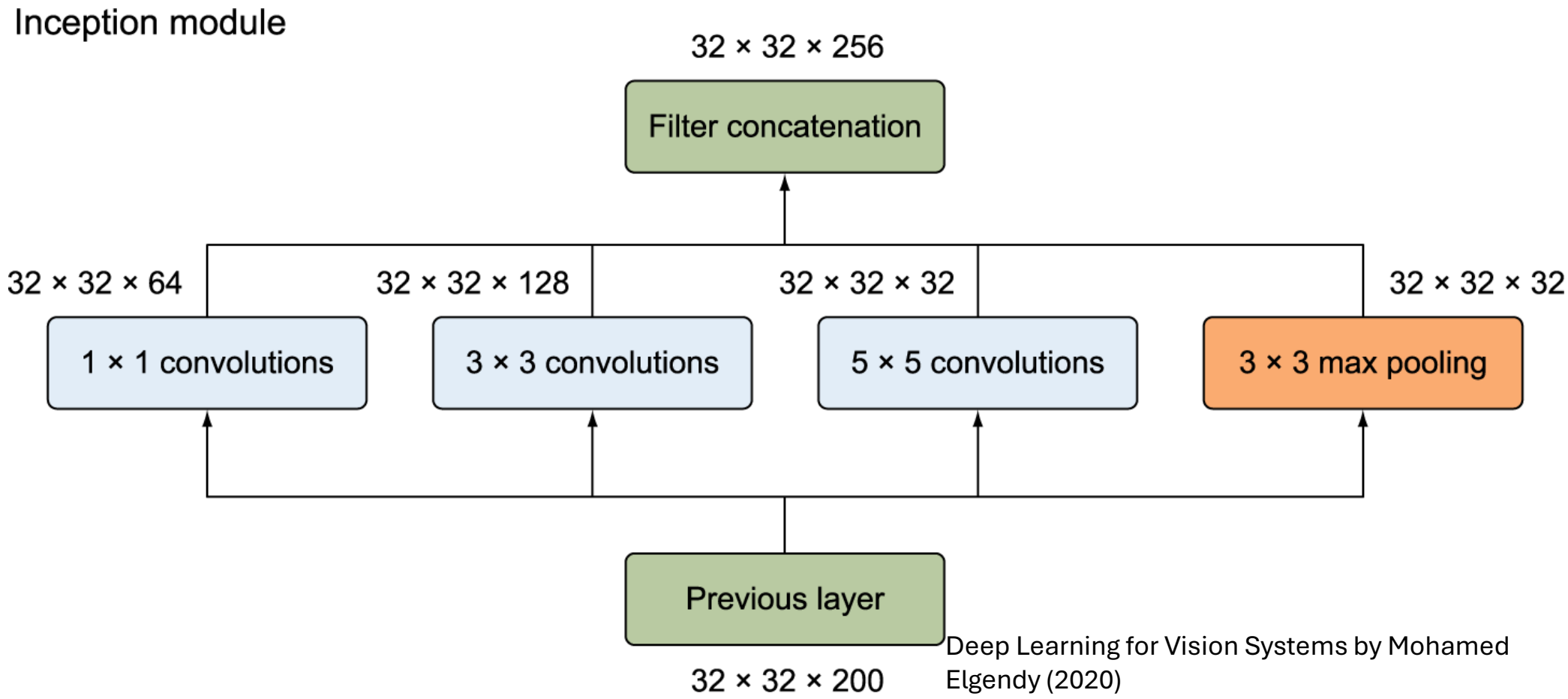
- Developed by: Szegedy et al., 2015 (Google Research)
- Winner of ILSVRC 2014 with 6.7% top-5 error rate, outperforming VGG and AlexNet.
- GoogLeNet introduced a new architectural concept — the Inception module — that allows a CNN to capture visual information at multiple scales simultaneously while keeping computational cost low.

The difference



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Inception Module

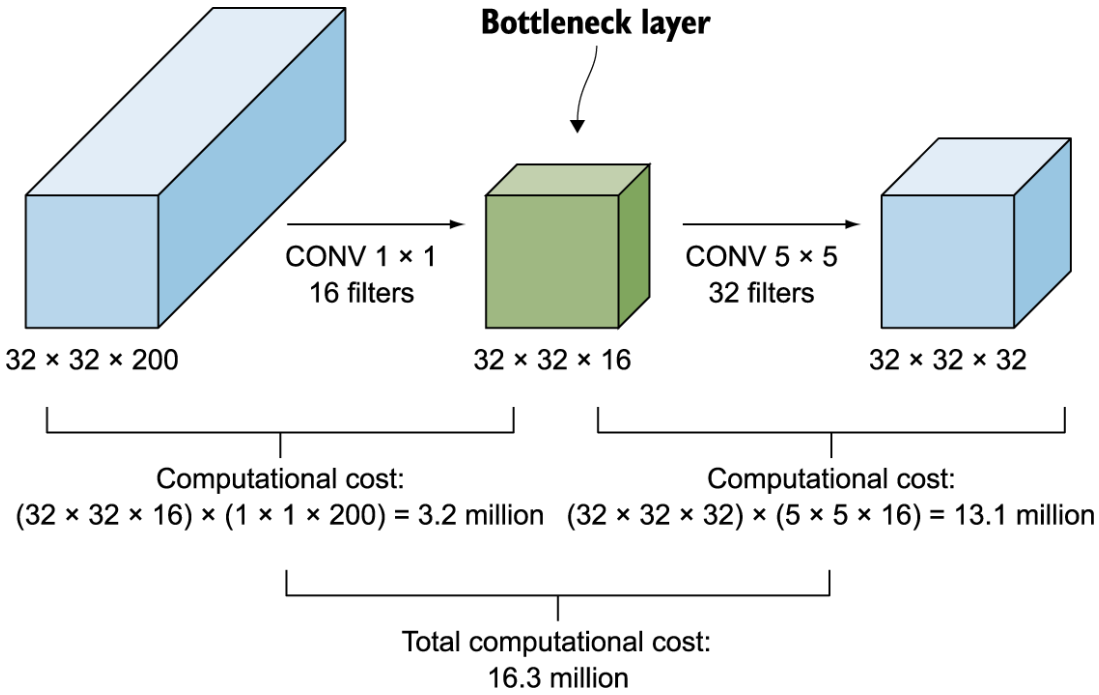


Deep Learning for Vision Systems by Mohamed Elgendy (2020)

What is the cost of Inception Module?

Dimensionality Reduction

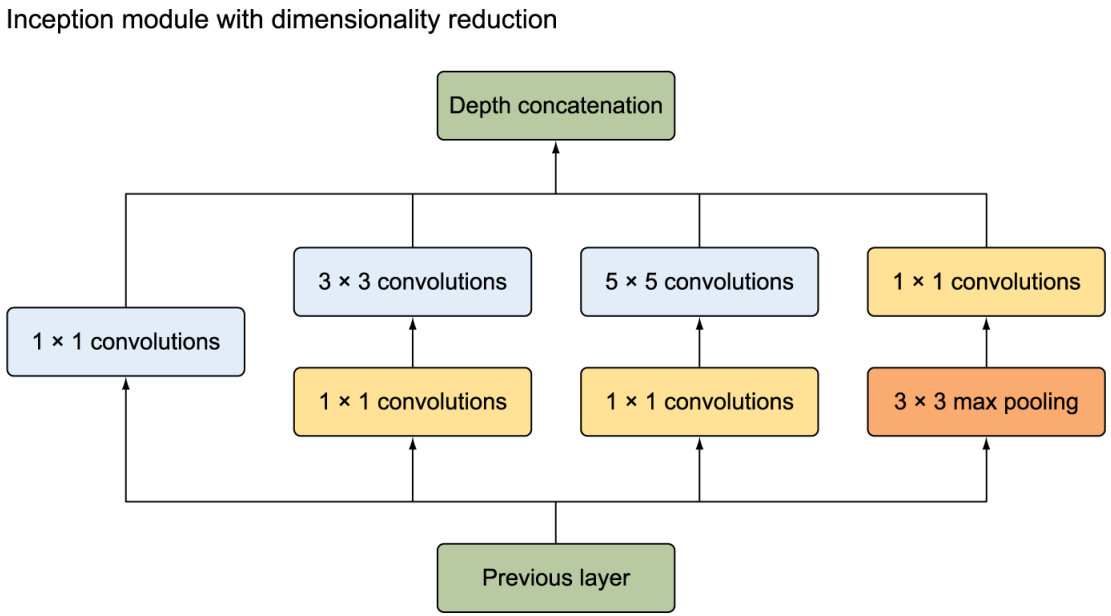
- Reduce Layer: The idea here is to add a 1 × 1 convolutional layer before the bigger kernels like the 3 × 3 and 5 × 5 convolutional layers, to reduce their depth, which in turn will reduce the number of operations.



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Dimensionality Reduction and Inception Module

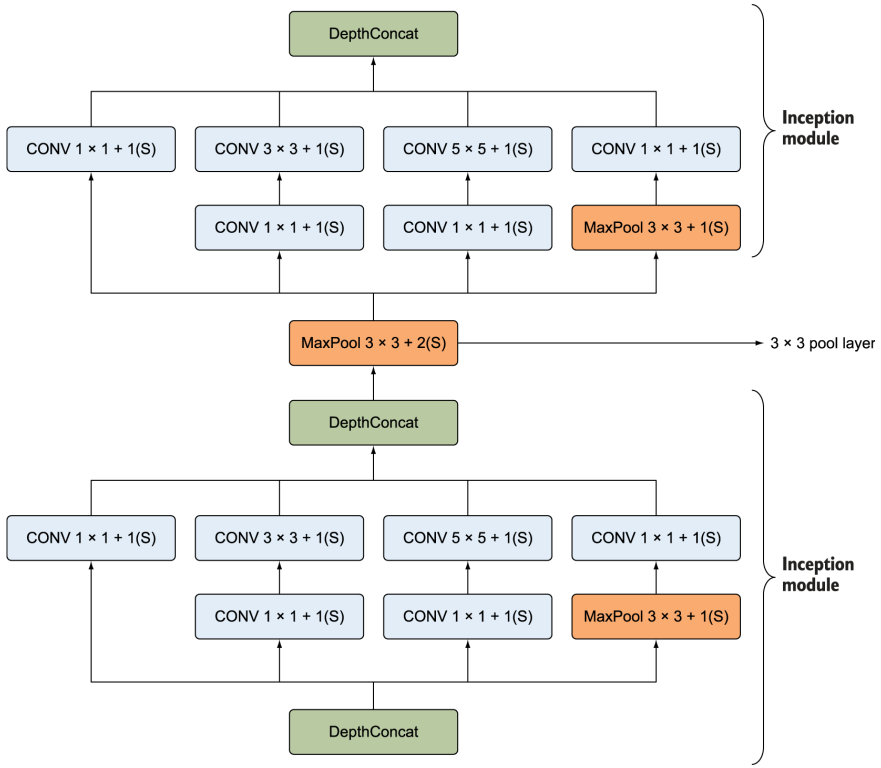
- add a 1 × 1 convolutional reduce layer before the 3 × 3 and 5 × 5 convolutional layers to reduce their computational cost.
- We will also add a 1 × 1 convolutional layer after the 3 × 3 max-pooling layer because pooling layers don't reduce the depth for their inputs.



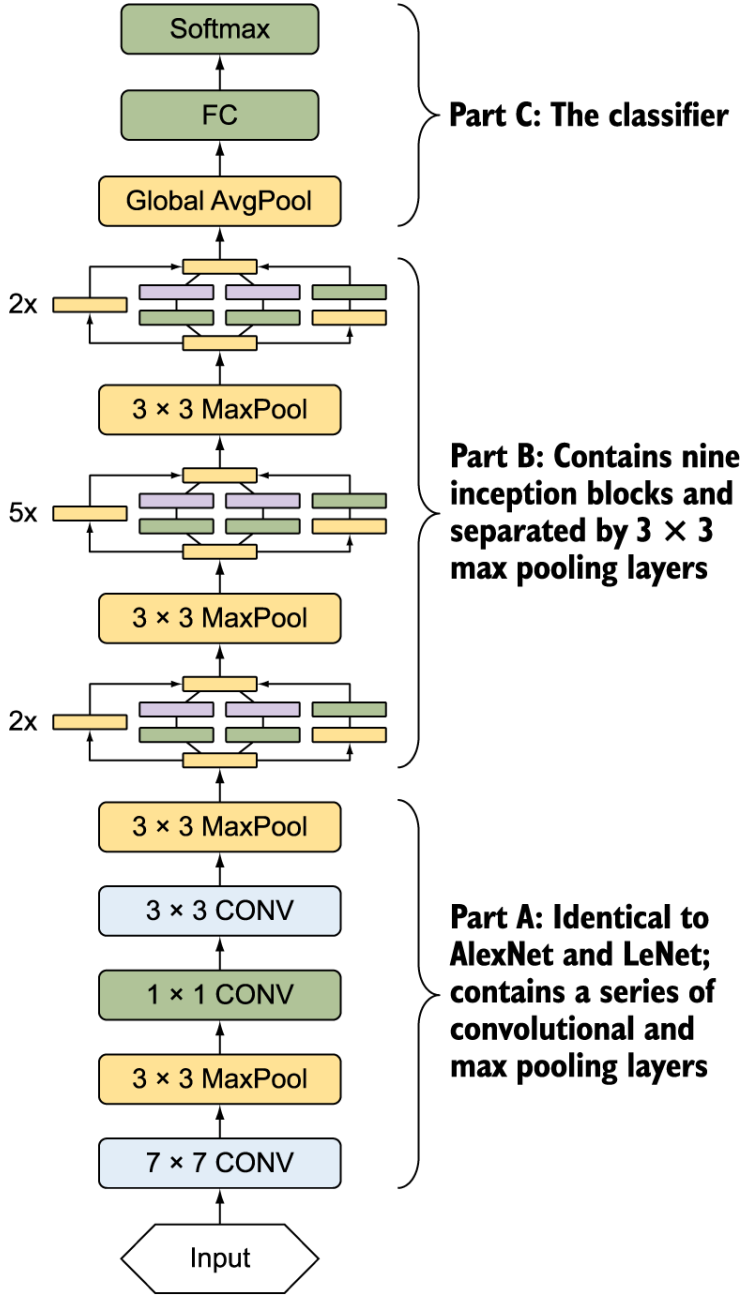
Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Recap

- Inception Architecture



- GooglLeNet



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

ResNet

The Residual Neural Network (ResNet) was developed in 2015 by a group from the Microsoft Research team

ResNet

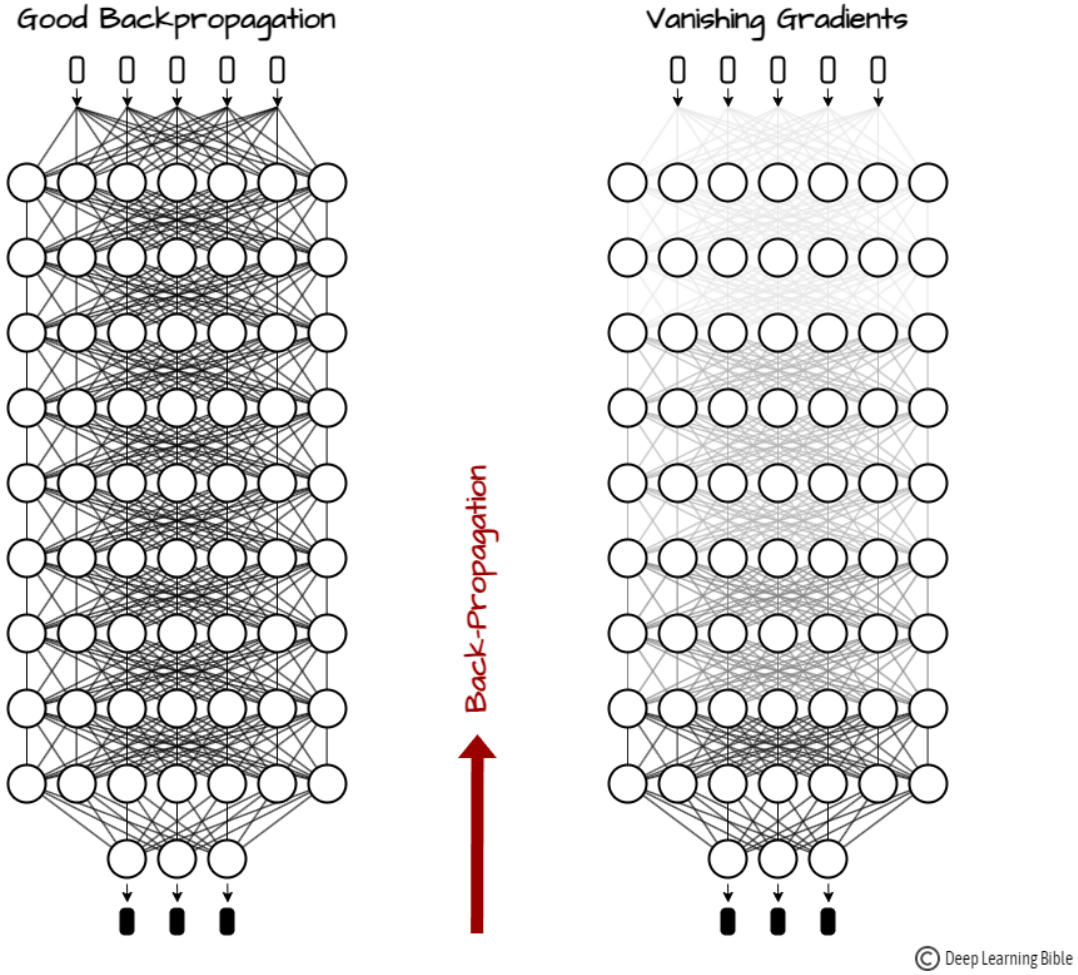
- Developed by: Kaiming He et al., Microsoft Research (2015)
- Innovation: Introduced the Residual Module with skip (shortcut) connections.
- Core Idea: Instead of learning the desired mapping directly, the network learns a residual function — the difference between the input and the desired output.

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

- This architecture allows gradients to flow directly through skip paths, solving the vanishing gradient problem that previously limited very deep networks

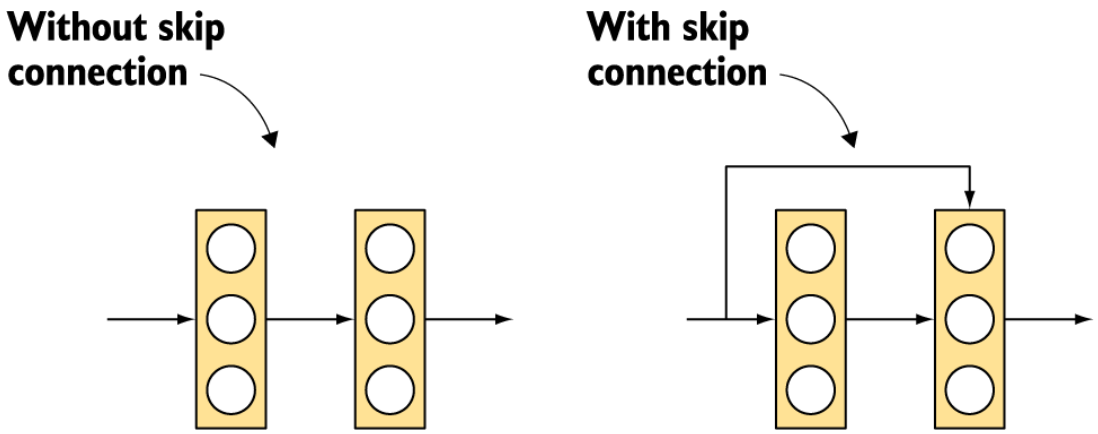
Vanishing Gradients

- The **vanishing gradient problem** occurs when gradients used to update the weights in a deep neural network become **extremely small** as they are propagated backward through the layers during **backpropagation**



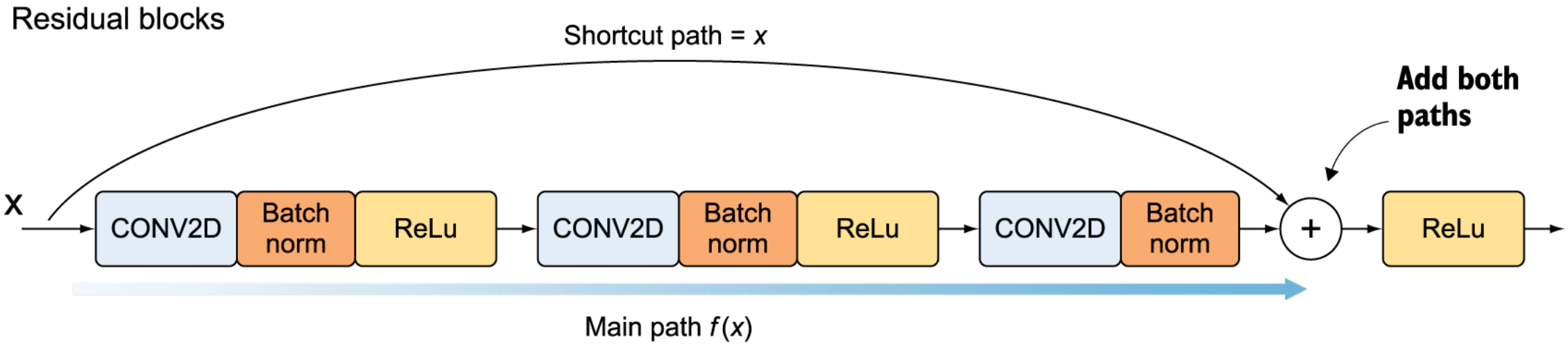
Skip Connections

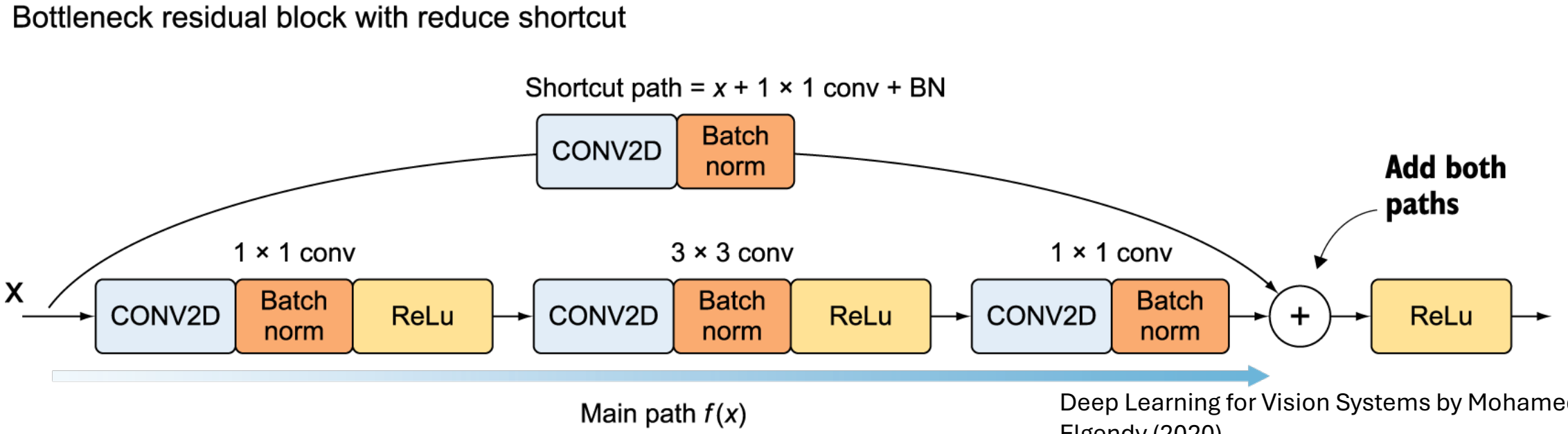
- Used to flow information from earlier layers in the network to later layers, creating an alternate shortcut path for the gradient to flow through.



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

Residual Block

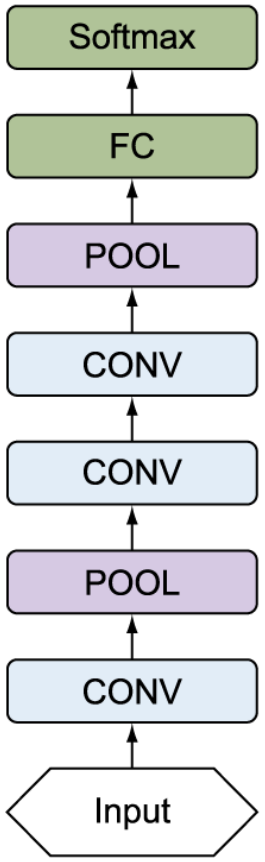




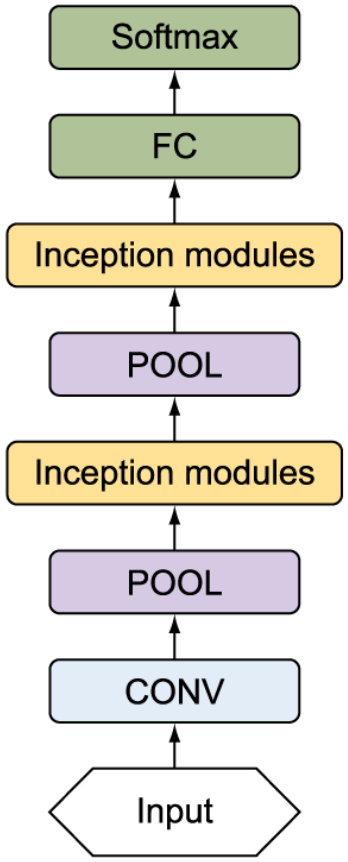
Deep Learning for Vision Systems by Mohamed Elgendy (2020)

The difference

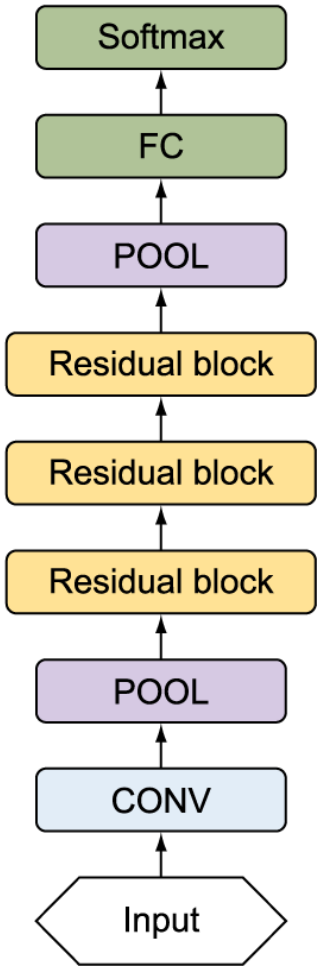
Classical CNN architecture



Inception modules



Residual blocks



Deep Learning for Vision Systems by Mohamed Elgendy (2020)

References

- Guide to CNNs for CV – Khan et al. (2018)
- Deep Learning with Python – Chollet (2018)
- Deep Learning in Computer Vision – Awad & Hassaballah (2020)

- Deep Learning for Vision Systems by Mohamed Elgendy (2020)