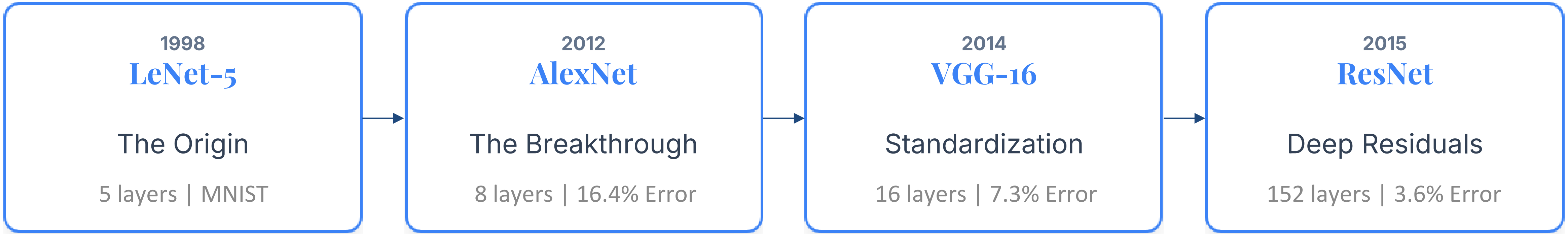
Evolution of CNNs

From LeNet-5 to ResNet: A Historical and Architectural Comparison

By Kebabist

Timeline of Innovation



1998
LeNet-5
The Origin
5 layers | MNIST

2012
AlexNet
The Breakthrough
8 layers | 16.4% Error

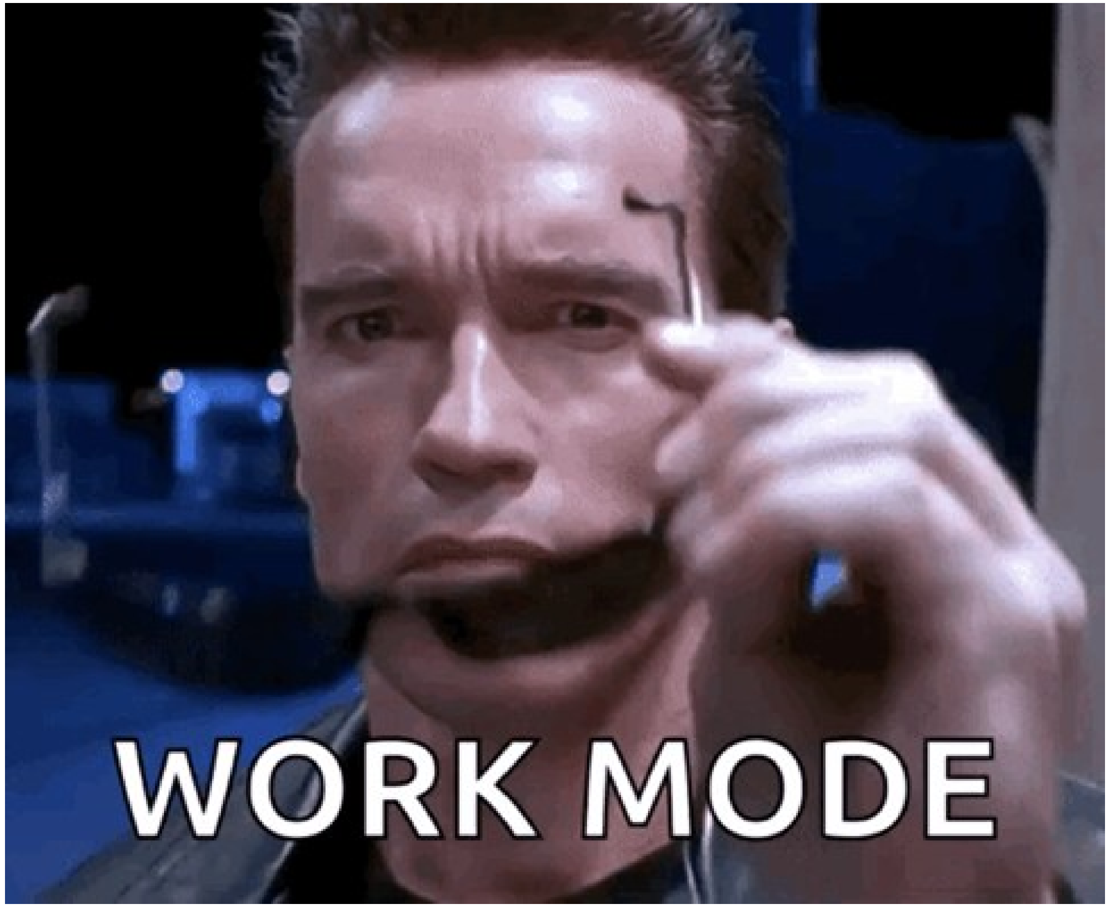
2014
VGG-16
Standardization
16 layers | 7.3% Error

2015
ResNet
Deep Residuals
152 layers | 3.6% Error

LeNet-5 (1998): The Foundation

The Grandfather of CNNs:

- **Author:** Yann LeCun et al.
- **Goal:** Handwritten digit recognition (MNIST) for banking/post (zip codes).
- **Architecture:** The first to successfully deploy the "Convolution → Pooling" hierarchy.
- **Limitations:** Used Sigmoid/Tanh activations (slower training) and lacked compute power for high-res images.



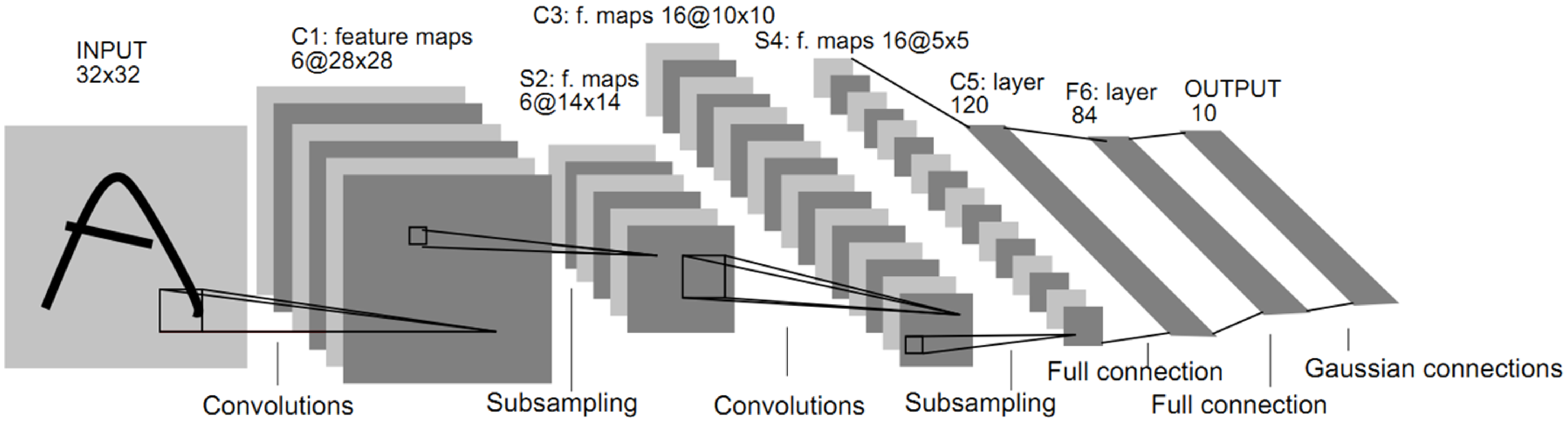
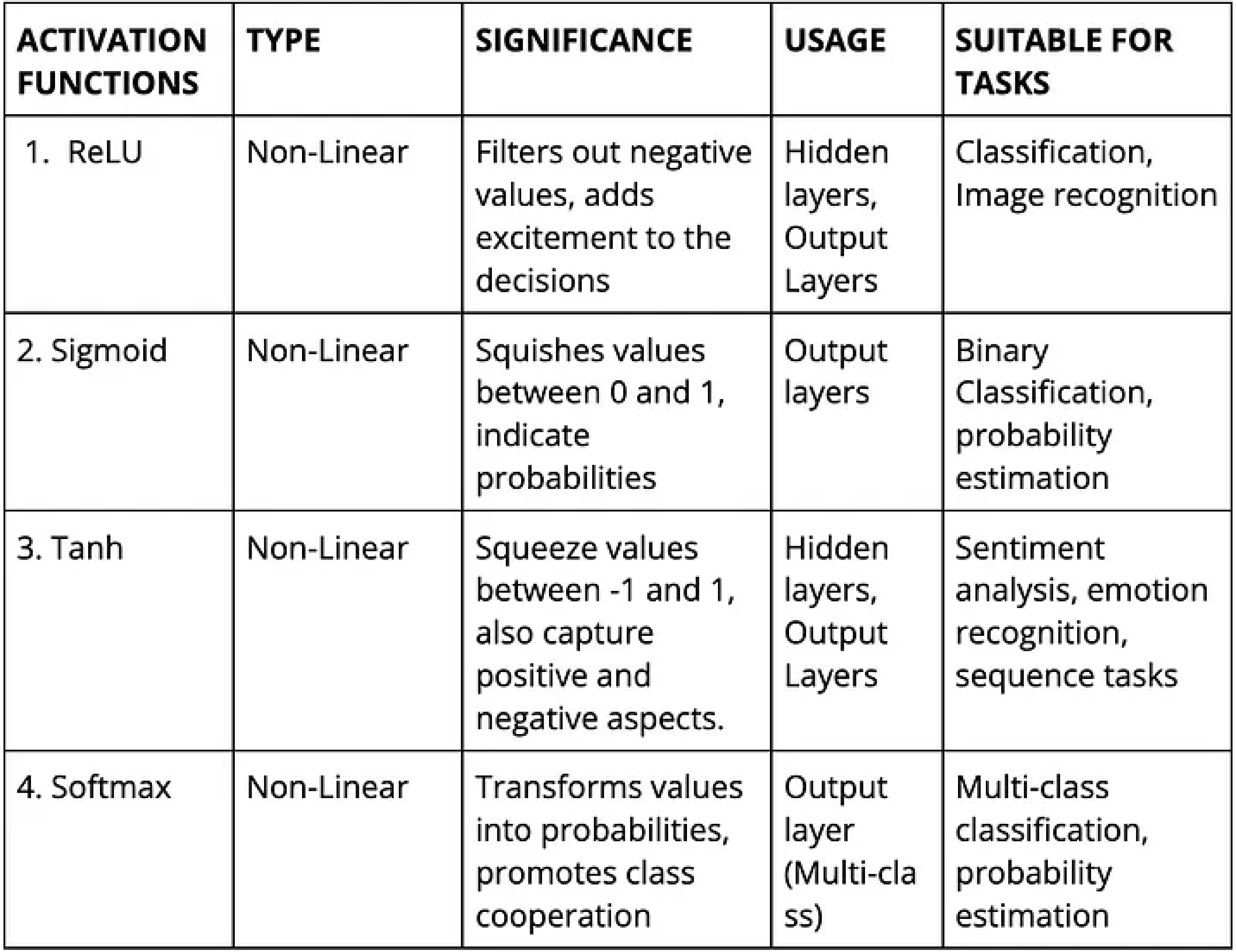


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average Pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average Pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	FC	-	84	-	-	tanh
Output	FC	-	10	-	-	softmax



ACTIVATION FUNCTIONS	TYPE	SIGNIFICANCE	USAGE	SUITABLE FOR TASKS
1. ReLU	Non-Linear	Filters out negative values, adds excitement to the decisions	Hidden layers, Output Layers	Classification, Image recognition
2. Sigmoid	Non-Linear	Squishes values between 0 and 1, indicate probabilities	Output layers	Binary Classification, probability estimation
3. Tanh	Non-Linear	Squeeze values between -1 and 1, also capture positive and negative aspects.	Hidden layers, Output Layers	Sentiment analysis, emotion recognition, sequence tasks
4. Softmax	Non-Linear	Transforms values into probabilities, promotes class cooperation	Output layer (Multi-class)	Multi-class classification, probability estimation

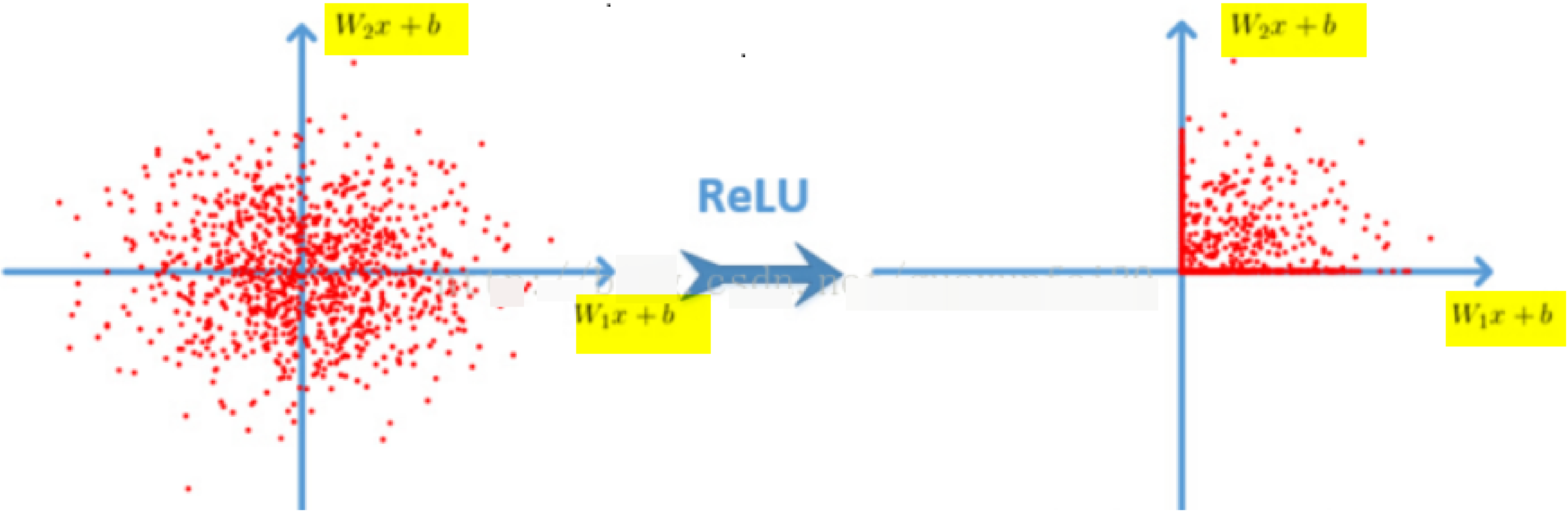
AlexNet (2012)

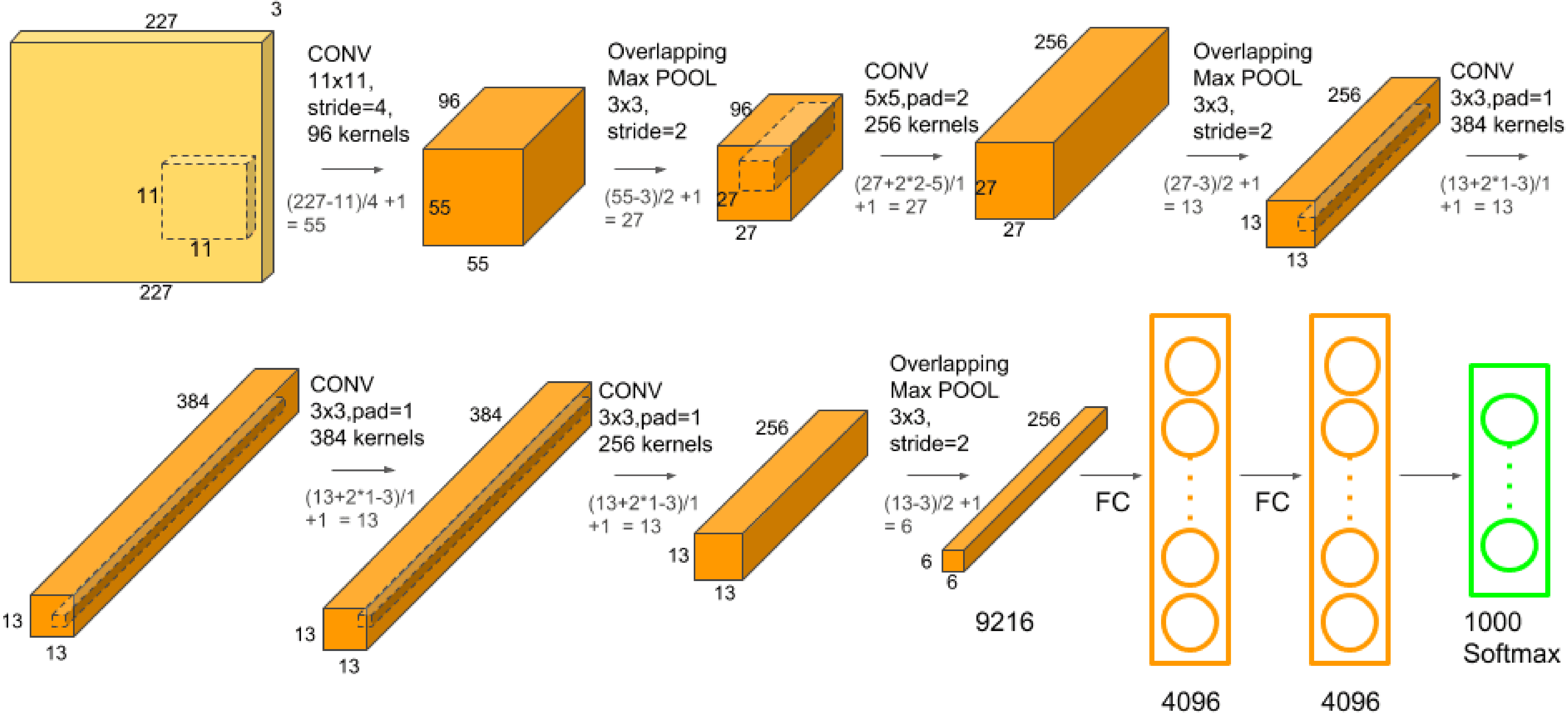
The architecture that sparked the modern Deep Learning boom.

AlexNet: Scaling Up

Key Innovations over LeNet:

- **Scale:** Deepened the network to 8 layers(5conv + 3FC) to handle ImageNet (1000 classes).
- **ReLU:** Replaced Sigmoid with ReLU to solve vanishing gradients and accelerating training in deeper networks.
- **Dropout:** Randomly deactivated neurons to stop overfitting.
- **Overlapping Pooling:** creating intersecting coverage areas that reduce error rates and make the model slightly harder to overfit.
- **Hardware:** Trained on GPUs, enabling massive parallelization.


$W_2x + b$
$W_1x + b$
ReLU
$W_2x + b$
$W_1x + b$


227
3
227
11
11
CONV
11x11,
stride=4,
96 kernels
(227-11)/4 +1
= 55
96
55
55
Overlapping
Max POOL
3x3,
stride=2
(55-3)/2 +1
= 27
96
27
27
CONV
5x5,pad=2
256 kernels
(27+2*2-5)/1
+1 = 27
256
27
27
Overlapping
Max POOL
3x3,
stride=2
(27-3)/2 +1
= 13
256
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
384 kernels
(13+2*1-3)/1
+1 = 13
384
13
13
CONV
3x3,pad=1
256 kernels
(13+2*1-3)/1
+1 = 13
256
13
13
Overlapping
Max POOL
3x3,
stride=2
(13-3)/2 +1
= 6
256
6
6
9216
FC
4096
FC
4096
1000
Softmax

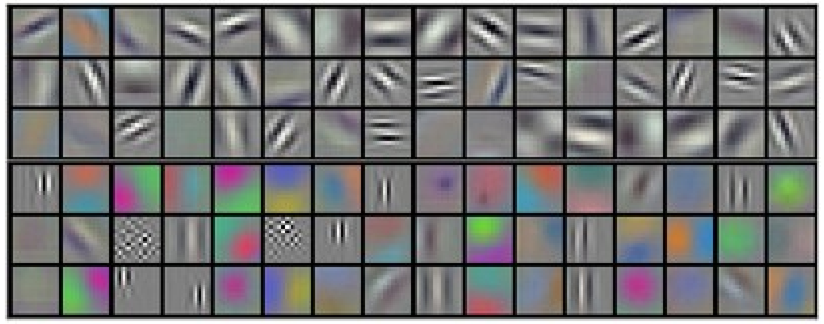
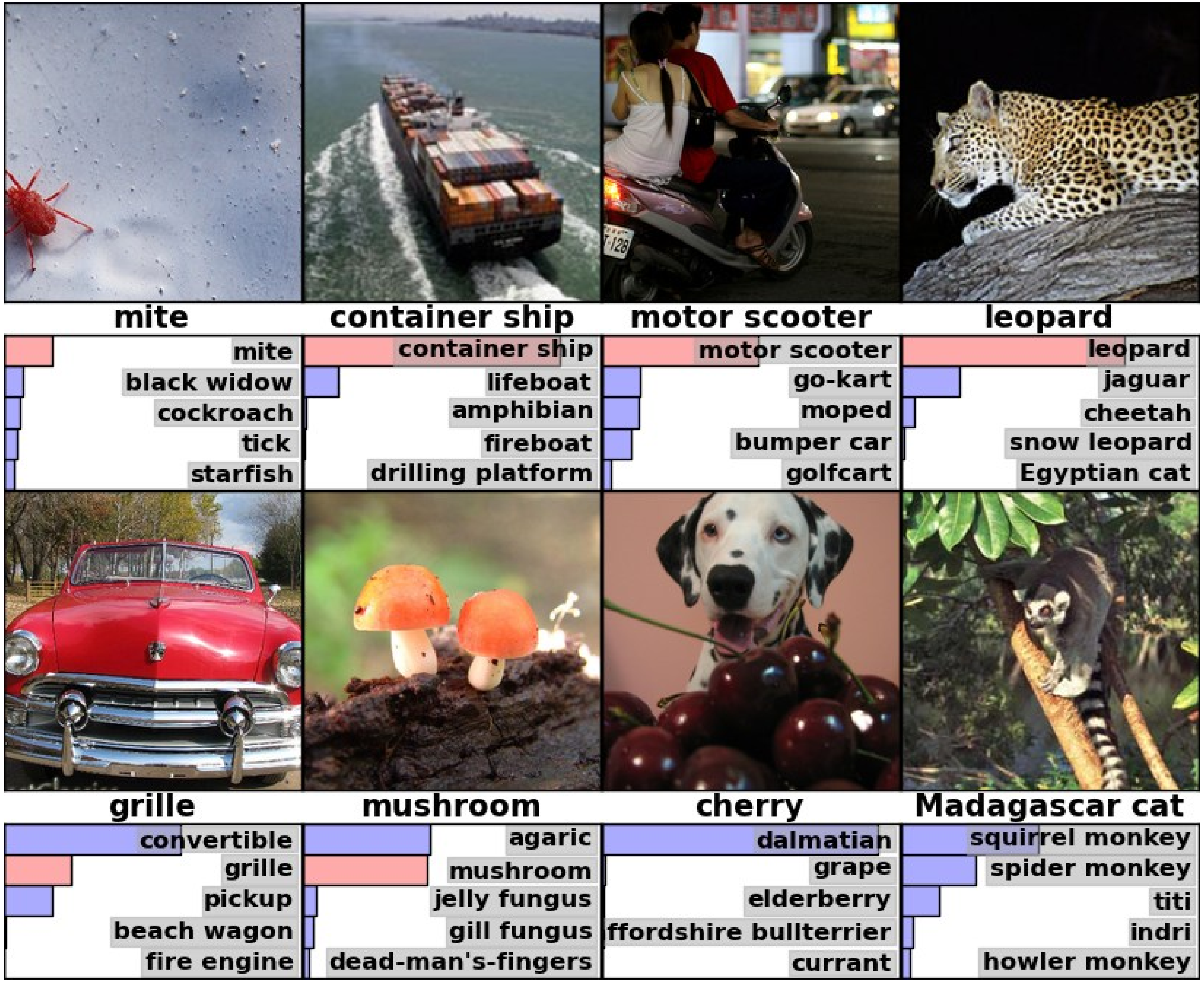
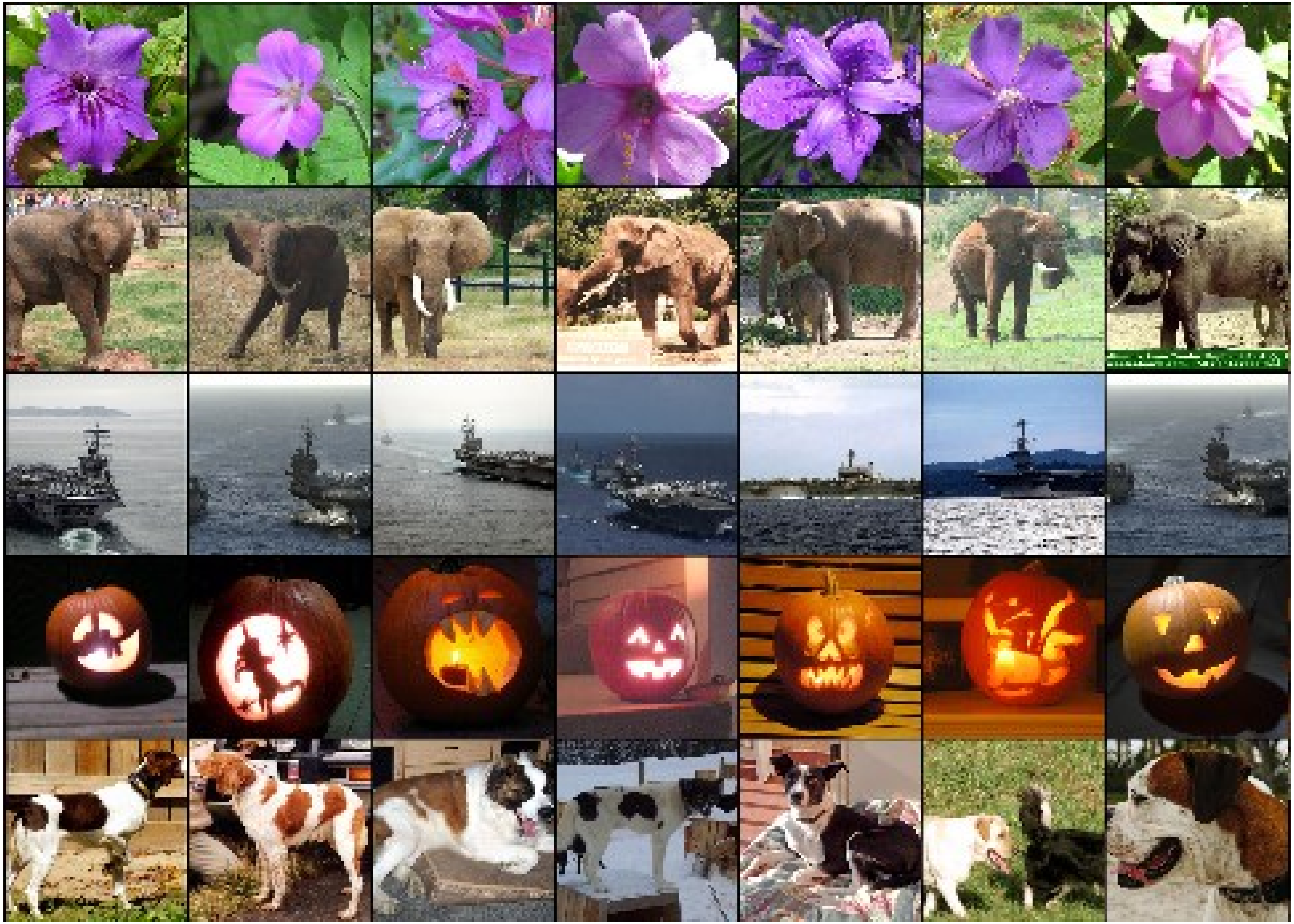


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The top 48 kernels were learned on GPU 1 while the bottom 48 kernels were learned on GPU 2. See Section 6.1 for details.





VGGNet (2014)

The philosophy of "Simplicity and Depth".

VGG: The Power of Small Filters

Small 3x3 Convolutions

VGG discarded the large filters (11x11) used in AlexNet. Stacking two 3x3 layers creates the same receptive field as a 5x5 but with fewer parameters and more non-linearity.

Uniform Blocks

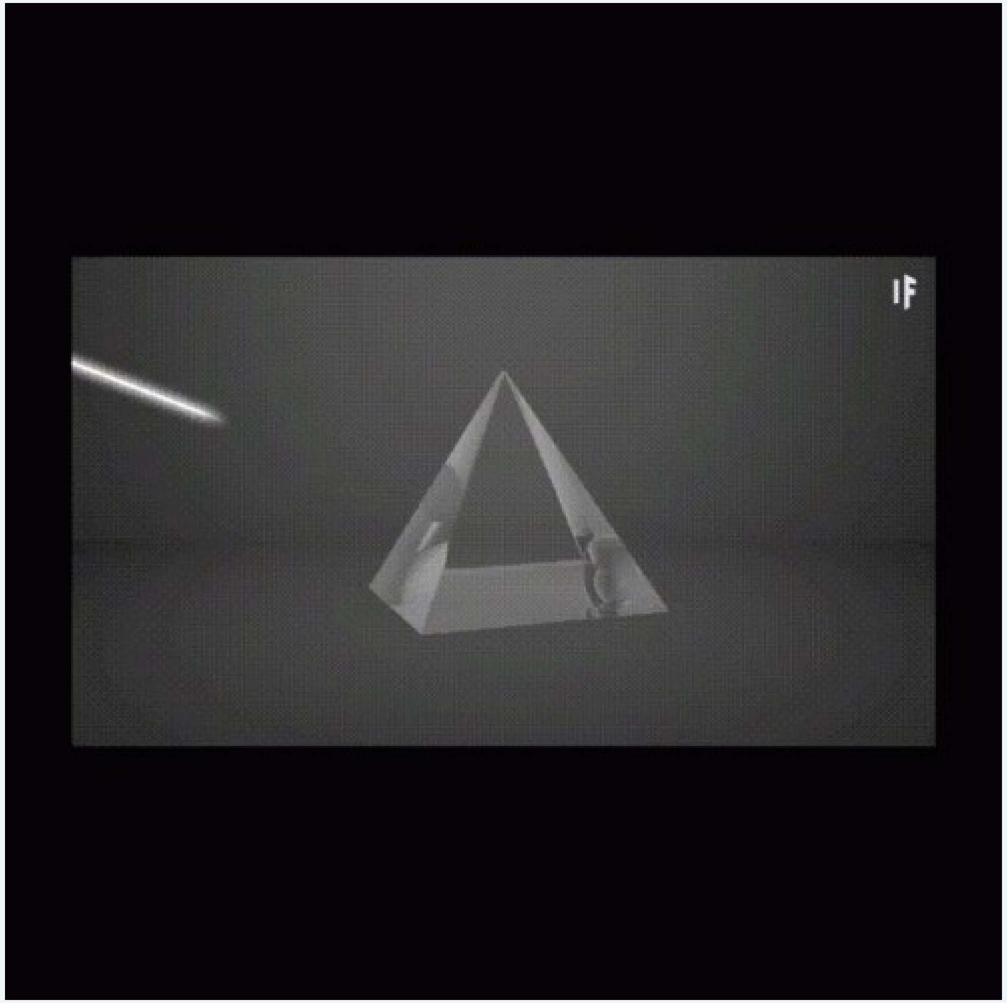
Its uniform structure (Conv-Conv-Pool) made it incredibly modular and easy to understand, becoming the standard feature extractor for years.

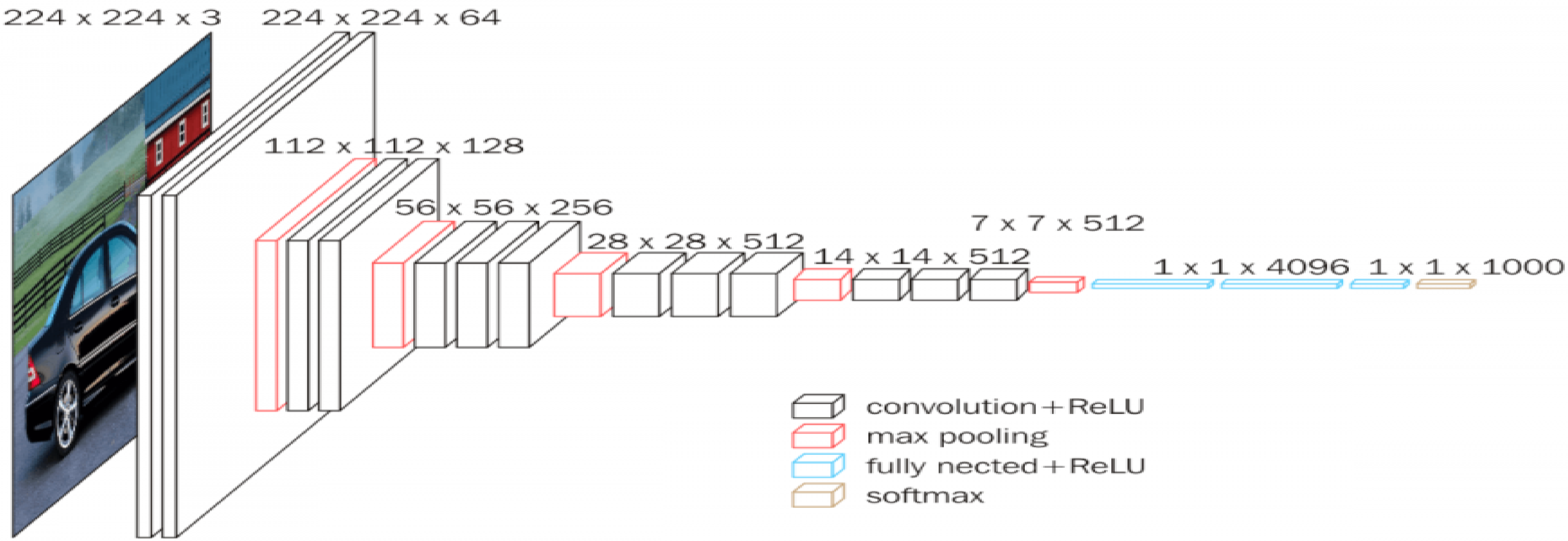
Drawback: Despite its simplicity, VGG is computationally expensive and has a massive number of parameters (~138 million), mostly due to its large fully connected layers

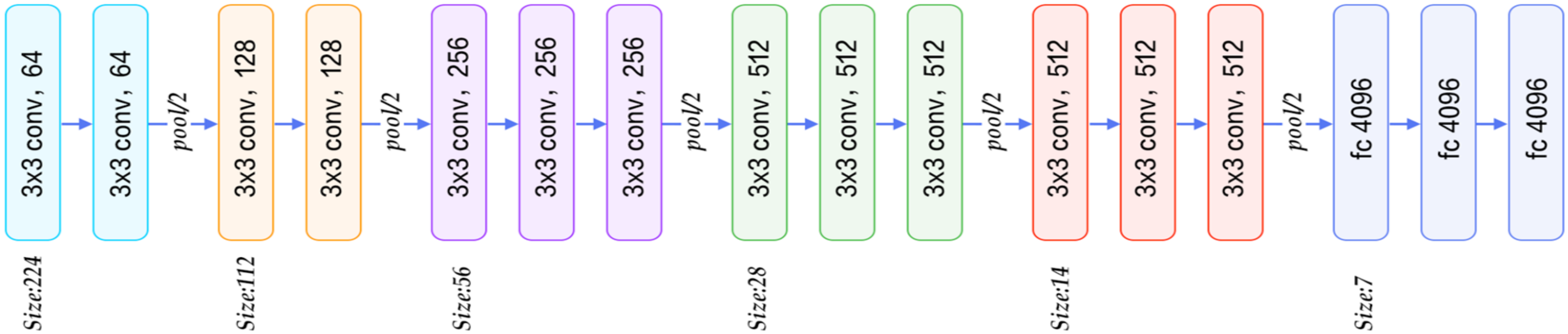
VGG Structure

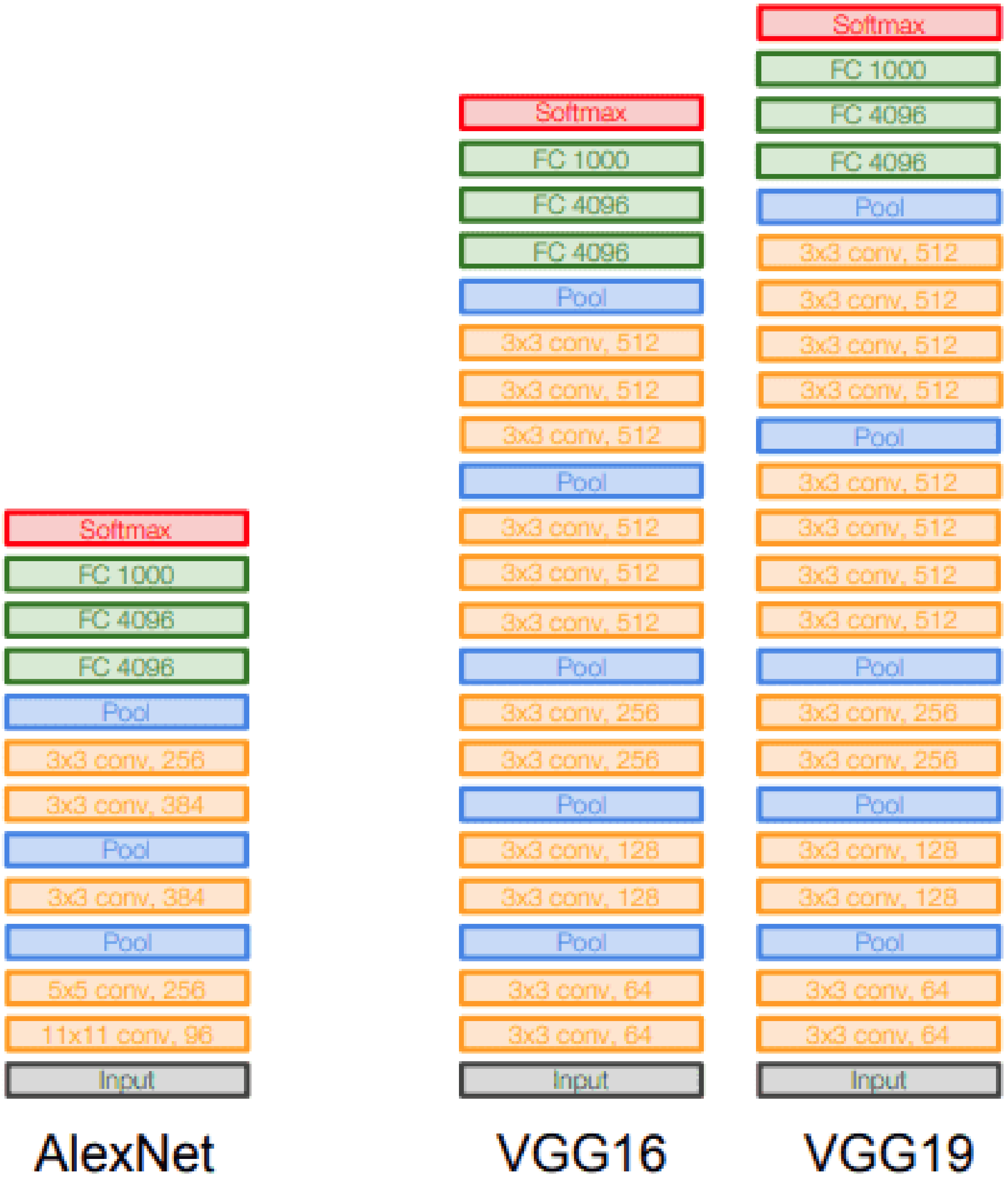
Deep & Narrow: Using multiple convolution layers with smaller convolution kernels instead of a larger convolution layer with convolution kernels can reduce parameters on the one hand, and the author believes that it is equivalent to more non-linear mapping, which increases the Fit expression ability.

Drawback: The final dense layers are massive, leading to a parameter count of ~138 Million. This makes VGG slow to train and heavy to deploy compared to modern standards.




224 x 224 x 3
224 x 224 x 64
112 x 112 x 128
56 x 56 x 256
28 x 28 x 512
14 x 14 x 512
7 x 7 x 512
1 x 1 x 4096
1 x 1 x 1000
convolution+ReLU
max pooling
fully nected+ReLU
softmax


3x3 conv, 64
3x3 conv, 64
pool/2
3x3 conv, 128
3x3 conv, 128
pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
fc 4096
fc 4096
fc 4096
Size:224
Size:112
Size:56
Size:28
Size:14
Size:7



Softmax
FC 1000
FC 4096
FC 4096
Pool
3x3 conv, 256
3x3 conv, 384
Pool
3x3 conv, 384
Pool
5x5 conv, 256
11x11 conv, 96
Input

AlexNet

Softmax
FC 1000
FC 4096
FC 4096
Pool
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Pool
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Pool
3x3 conv, 256
3x3 conv, 256
Pool
3x3 conv, 128
3x3 conv, 128
Pool
3x3 conv, 64
3x3 conv, 64
Input

VGG16

Softmax
FC 1000
FC 4096
FC 4096
Pool
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Pool
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
Pool
3x3 conv, 256
3x3 conv, 256
Pool
3x3 conv, 128
3x3 conv, 128
Pool
3x3 conv, 64
3x3 conv, 64
Input

VGG19

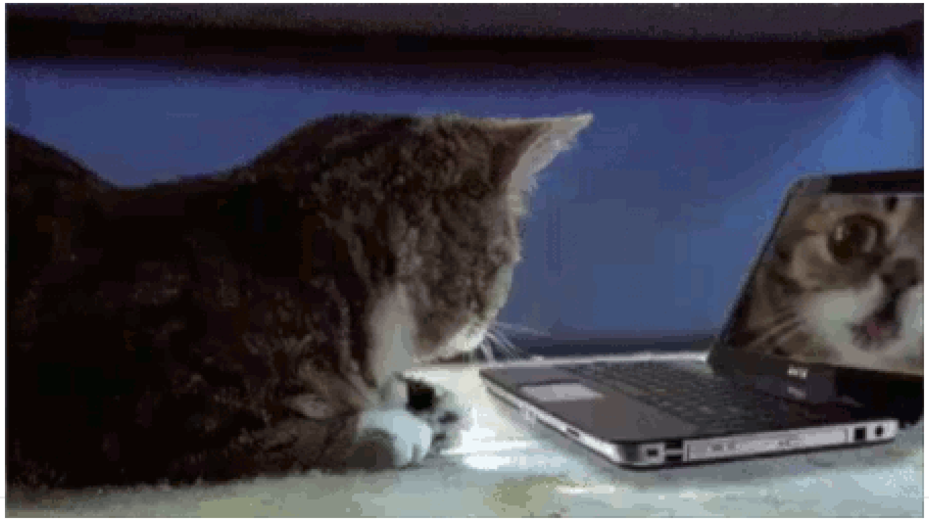
GoogLeNet / Inception (2014)

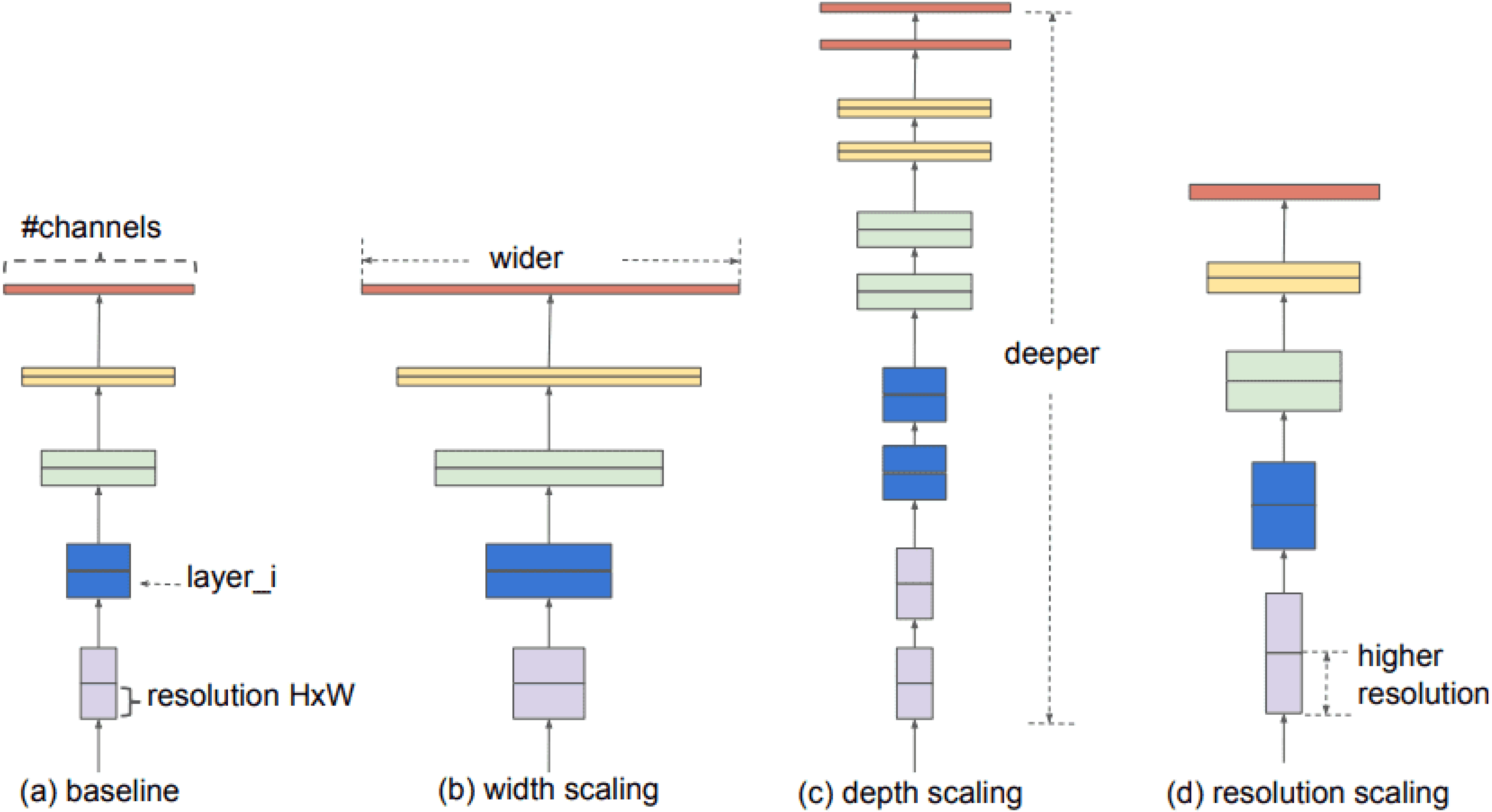
Going Deeper with Efficiency.

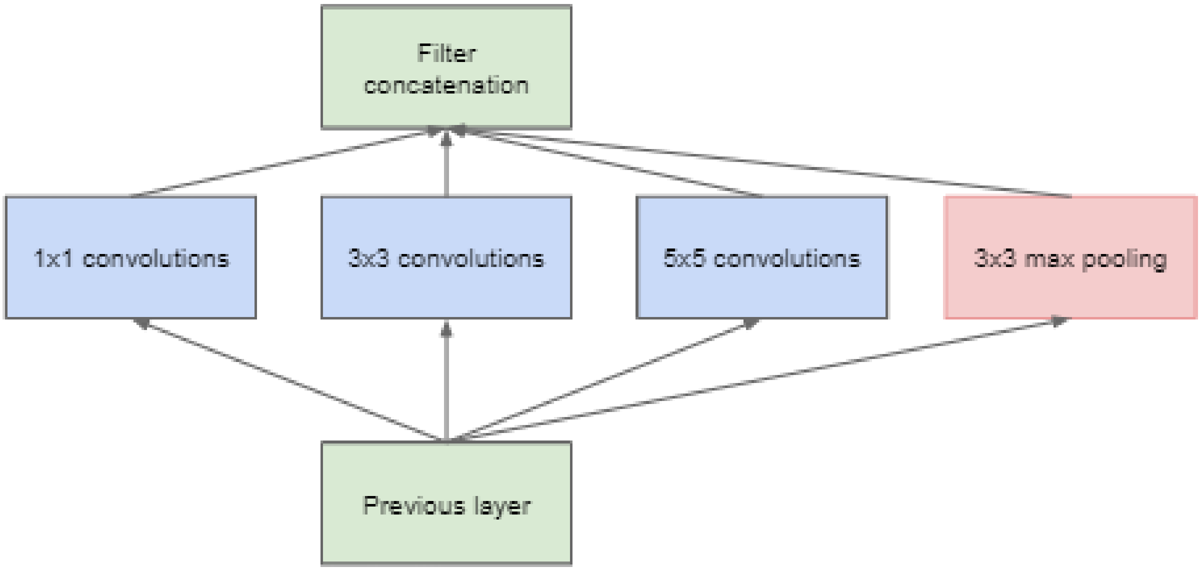
The Inception Module

Wider, Not Just Deeper:

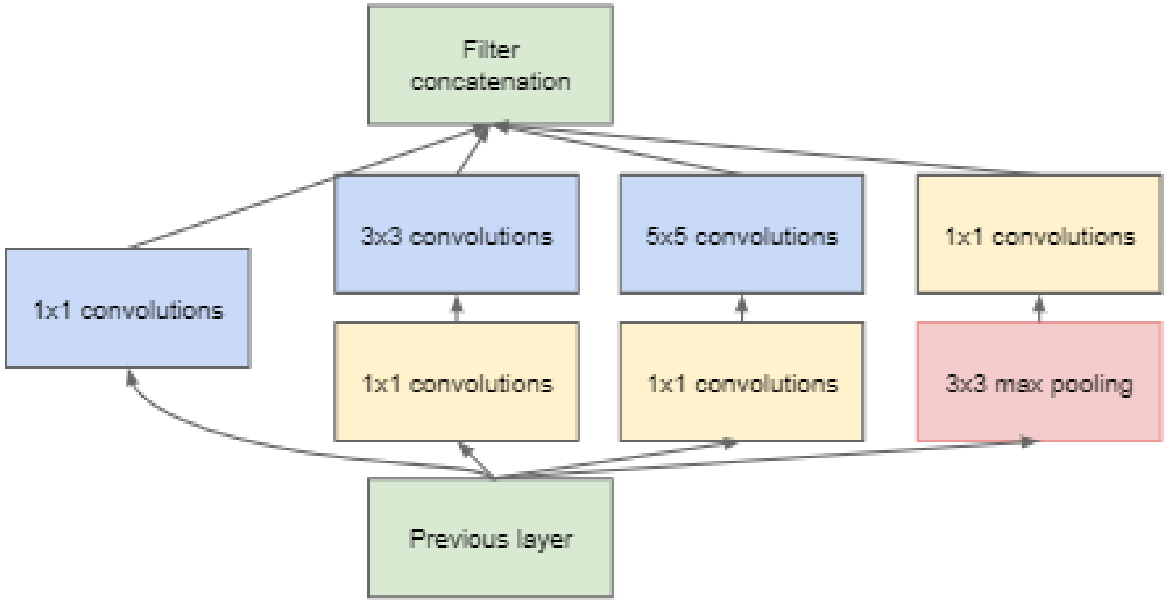
- Instead of choosing a filter size (3x3 or 5x5), Inception uses **all of them** in parallel to capture details at multiple scales.
- **1x1 Convolutions:** Used as "bottlenecks" to reduce dimensions before expensive operations, drastically cutting computation.
- The network learns which filter size is best for extracting features at each layer.




#channels
layer_i
resolution HxW
wider
deeper
higher
resolution
(a) baseline
(b) width scaling
(c) depth scaling
(d) resolution scaling



(a) Inception module, naïve version



(b) Inception module with dimension reductions

In general, a larger kernel is preferred for information that resides globally, and a smaller kernel is preferred for information that is distributed locally.

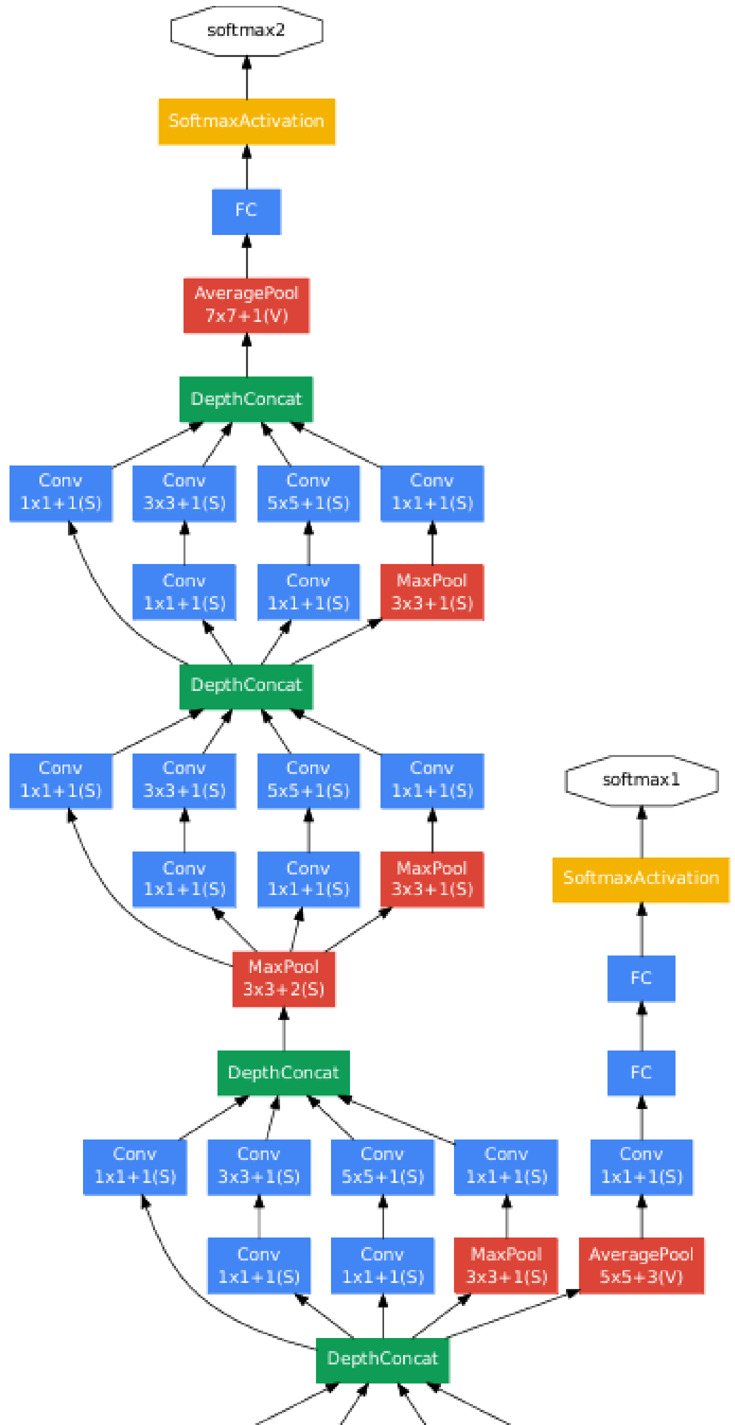
GoogLeNet Architecture

22 Layers, Yet Efficient:

GoogLeNet was much deeper than VGG but had **12x fewer parameters**.

Global Average Pooling: It removed the heavy fully connected layers at the end, replacing them with a simple average operation.

Auxiliary Classifiers: Side branches (seen in diagram) injected gradients during training to prevent them from vanishing.



ResNet (2015)

Conquering the Vanishing Gradient Problem.

The Challenge of Depth

> “ When deeper networks start converging, a degradation problem has been exposed... accuracy gets saturated and then degrades rapidly.
>
> — K. He et al., "Deep Residual Learning"

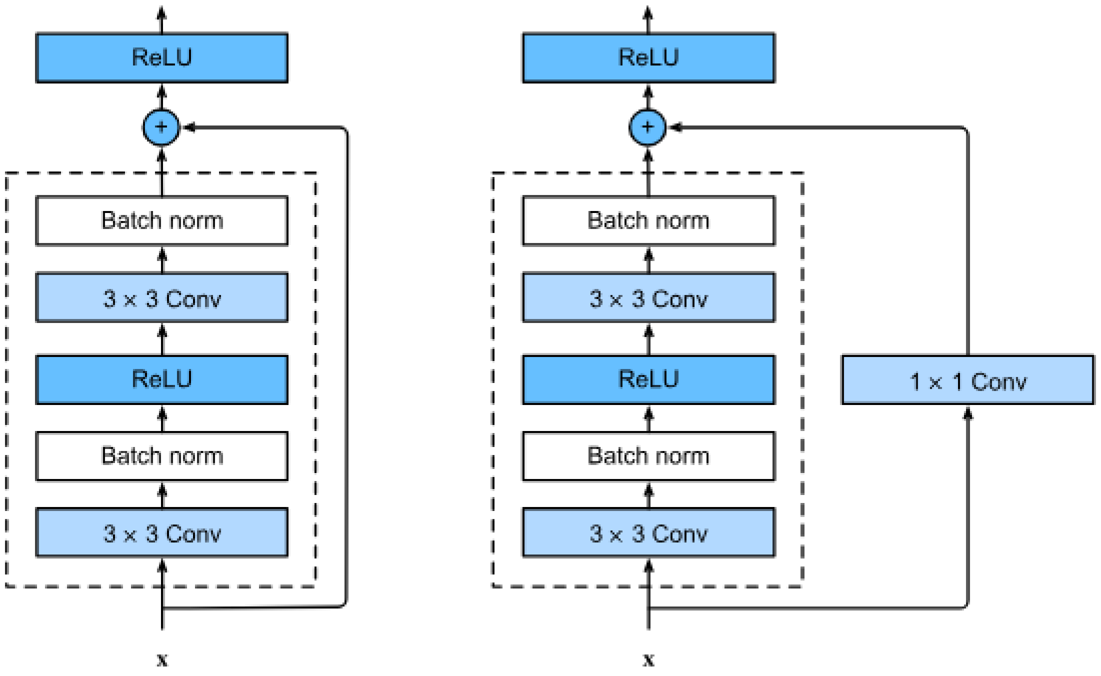
The Residual Solution

Skip Connections

ResNet introduces an "identity shortcut" connection that bypasses one or more layers.

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

This allows gradients to flow through the network unimpeded during backpropagation, enabling the training of networks with 100+ layers without degradation.



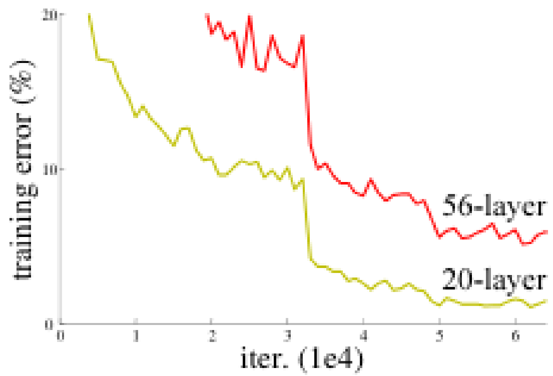
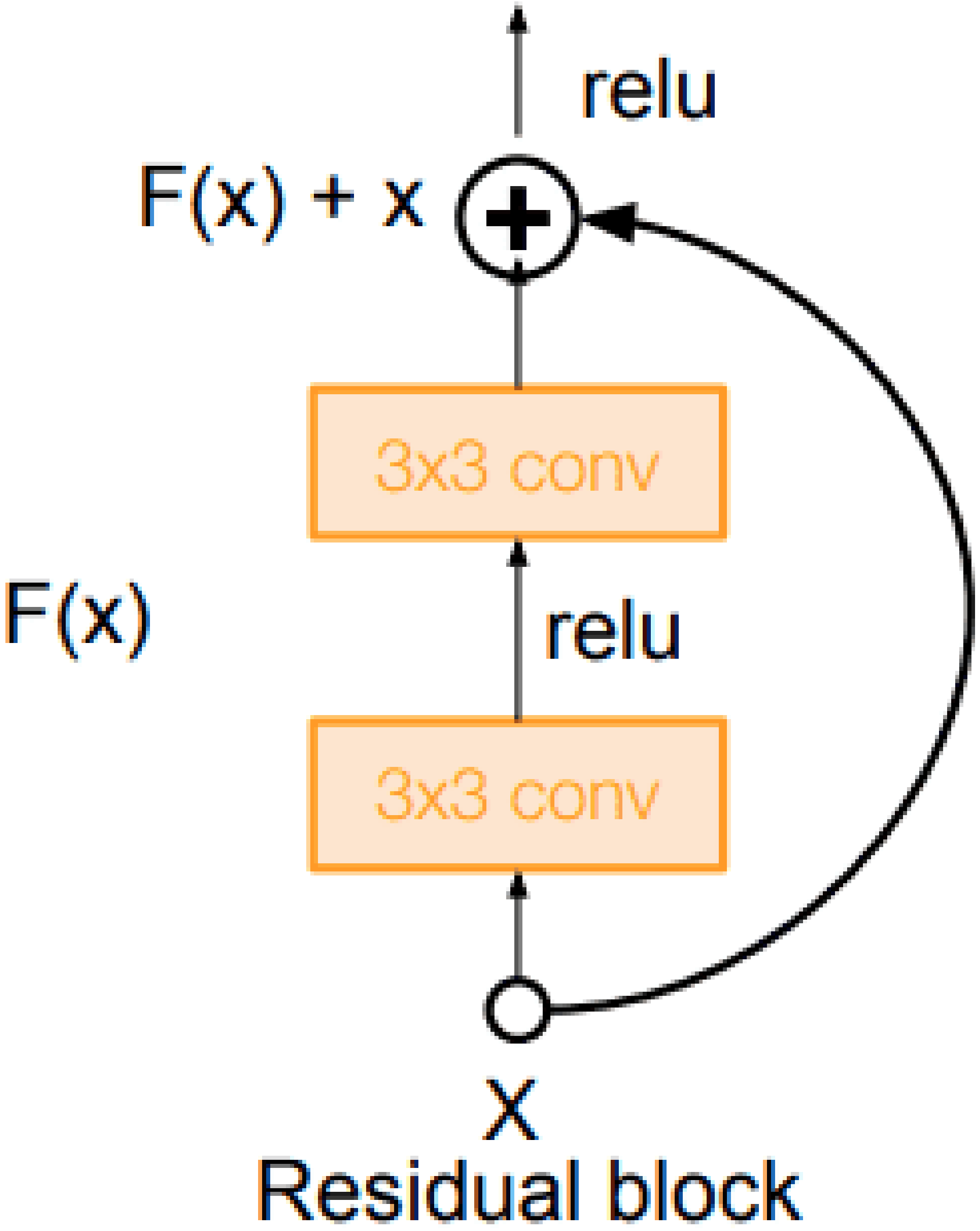


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.



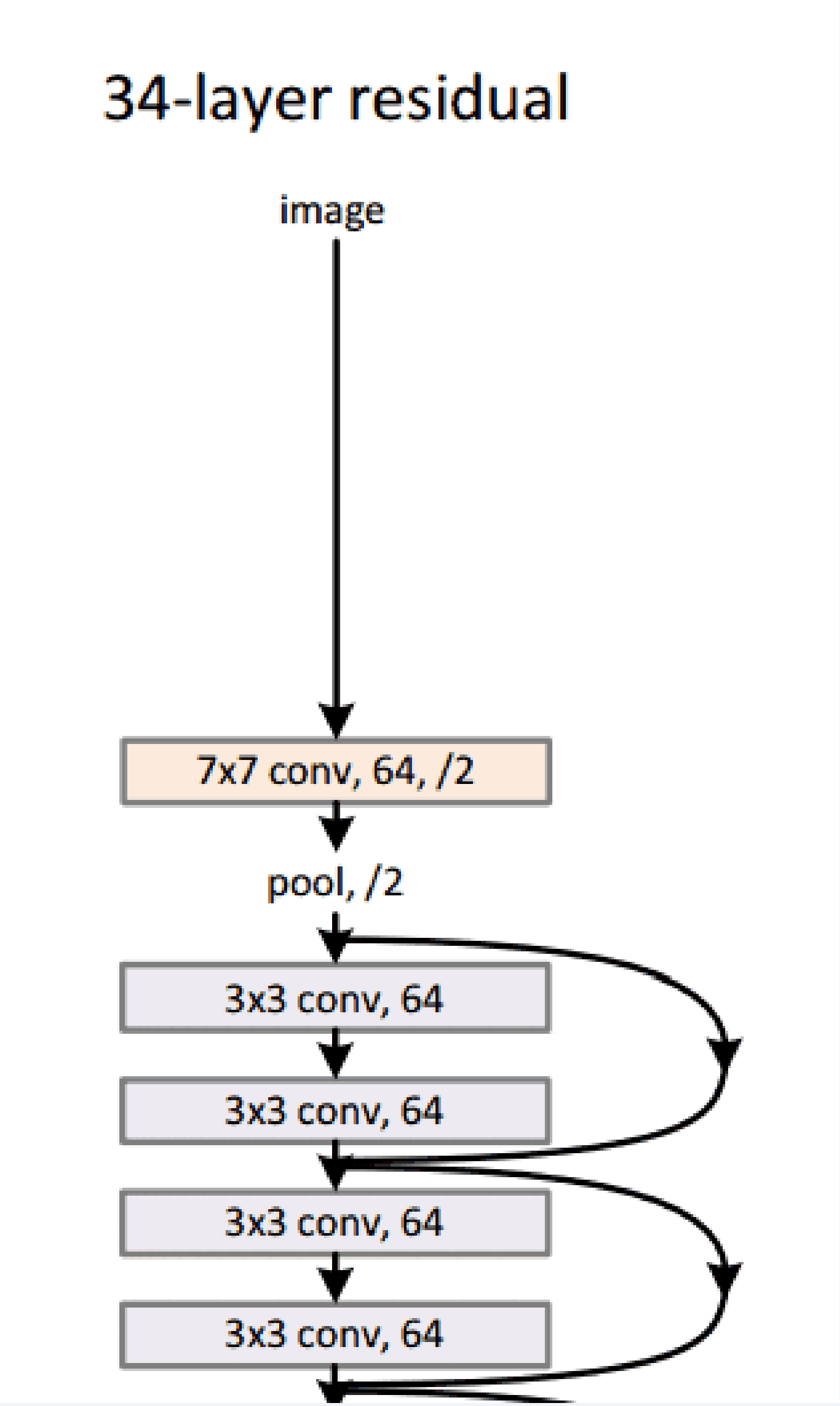
ResNet Architecture

Ultra-Deep Networks:

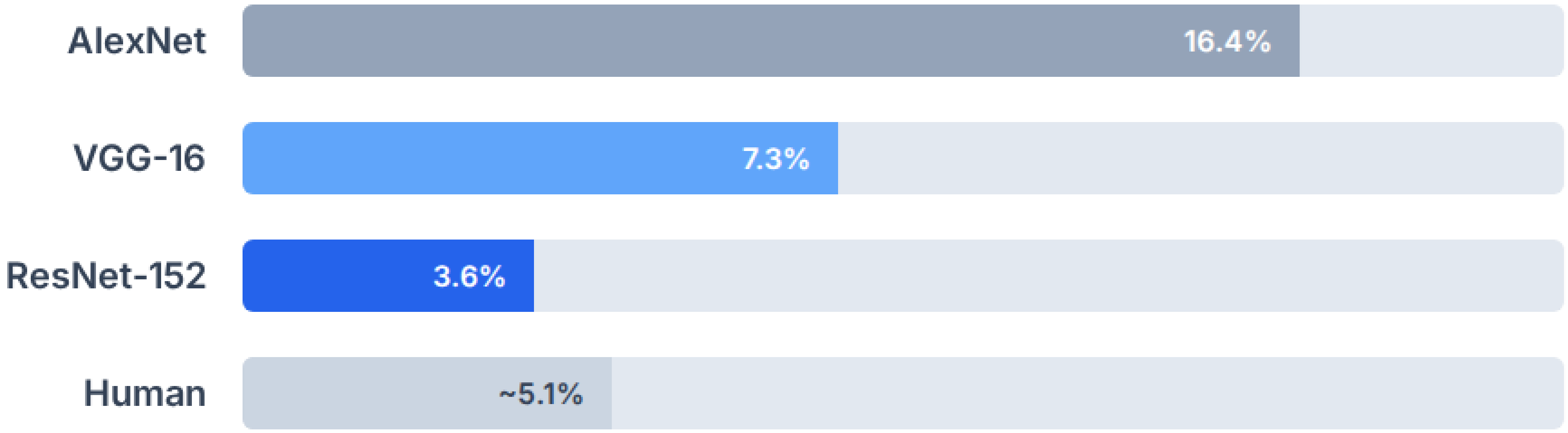
By using Residual blocks, ResNet-152 achieved superhuman accuracy.

Crucially, it replaced the heavy fully connected layers of VGG with "Global Average Pooling."

This drastic reduction in parameters means ResNet is simultaneously much deeper and much lighter than VGG.

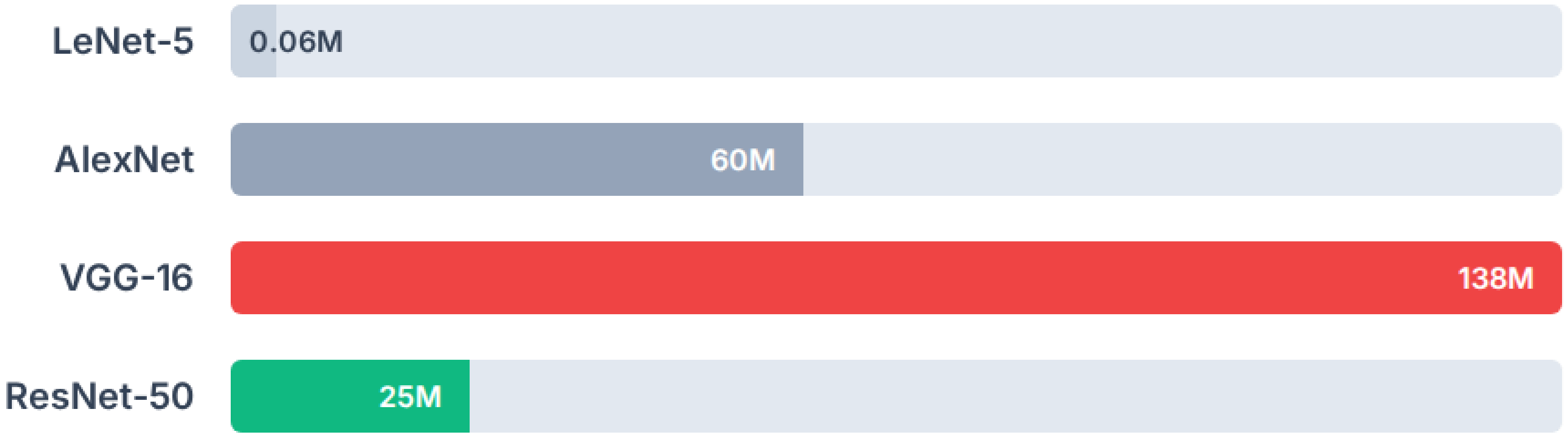


ImageNet Performance (Top-5 Error)



LeNet is excluded here as it predates ImageNet. ResNet was the first to beat human performance benchmarks.

Model Size (Parameters)



LeNet was tiny by modern standards. VGG is massive due to FC layers. ResNet is highly efficient despite its depth.

Summary Comparison

LeNet

The Origin. 1998. 5 Layers.

AlexNet

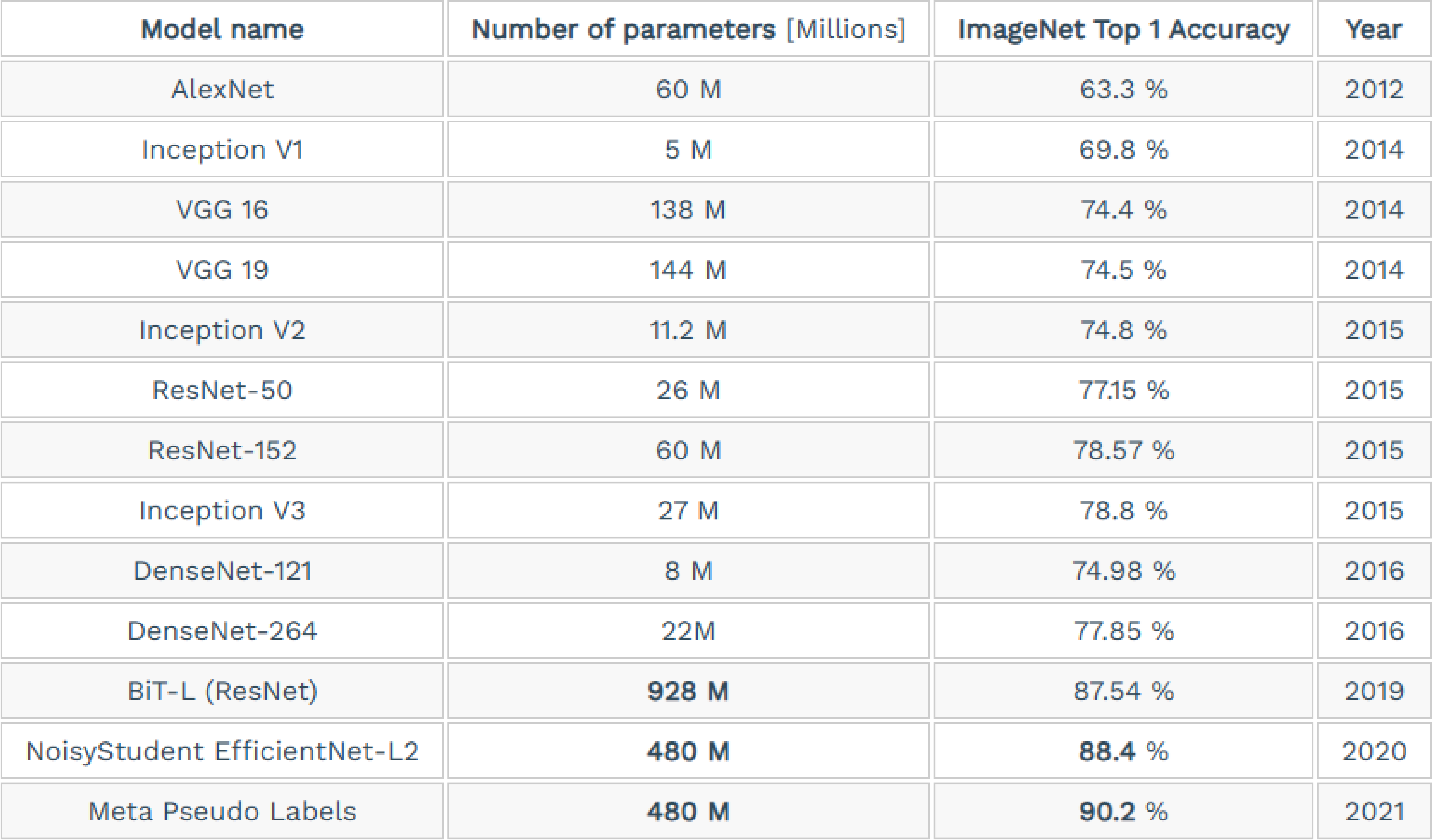
The Spark. 2012. 8 Layers.

VGGNet

Standard. 2014. 16 Layers.

ResNet

Deep. 2015. 152 Layers.



Model name	**Number of parameters** [Millions]	**ImageNet Top 1 Accuracy**	**Year**
AlexNet	60 M	63.3 %	2012
Inception V1	5 M	69.8 %	2014
VGG 16	138 M	74.4 %	2014
VGG 19	144 M	74.5 %	2014
Inception V2	11.2 M	74.8 %	2015
ResNet-50	26 M	77.15 %	2015
ResNet-152	60 M	78.57 %	2015
Inception V3	27 M	78.8 %	2015
DenseNet-121	8 M	74.98 %	2016
DenseNet-264	22M	77.85 %	2016
BiT-L (ResNet)	**928 M**	87.54 %	2019
NoisyStudent EfficientNet-L2	**480 M**	**88.4** %	2020
Meta Pseudo Labels	**480 M**	**90.2** %	2021

Questions?

Thank you for your attention.

Hasta la vista, baby.