CNN Architectures

Stephen Baek

Right off the shelf…

- AlexNet (2012)
- VGG (2014)
- GoogLeNet (or Inception) (2014)
- ResNet (2015)
- Inception-v4 (2016)
- DenseNet (2016)
- ResNext (2016)
- MobileNet (2017)
- NASNet (2018)
- …

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- ImageNet (First appearance as a poster at CVPR 2009)
  - Over 14 million images with hand-annotated labels (crowdsourced via M-turk)
  - Over 20 thousand categories
  - 1M+ images with bounding boxes.

- ImageNet Challenge
  - Since 2010
  - Uses a trimmed set of thousand non-overlapping classes.
  - Humans error is about 5%*
  - Playground of computer vision researchers.
  - Marked the start of the current AI boom.

Russakovsky et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211-252.

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- Criteria: Top-5 accuracy, Top-1 accuracy
- Say we had only four classes, namely "cat", "dog", "alpaca", and "camel". Now let's assume that the classifier returns something like this:

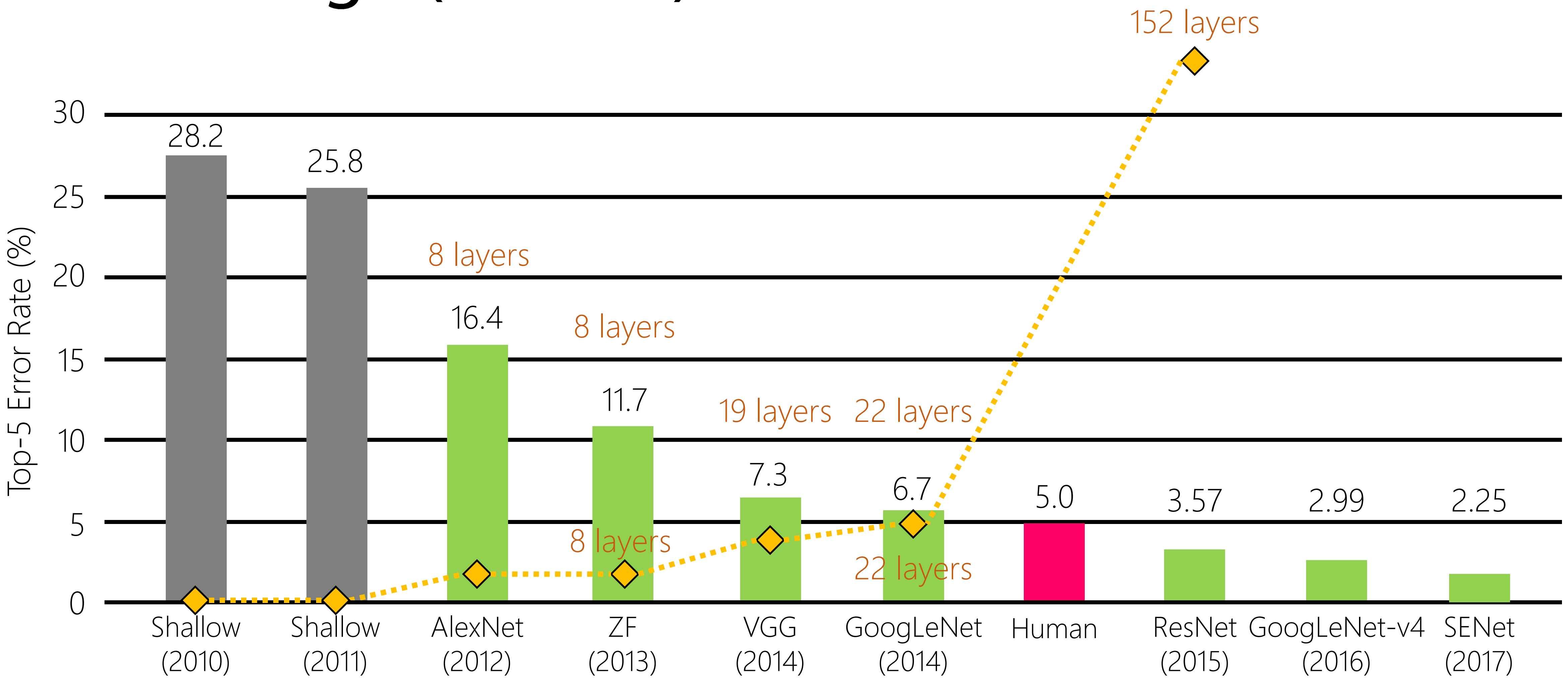
Your ML Model → [0.1, 0.5, 0.3, 0.1]

Top-1 class = {dog}
∴ Top-1 Accuracy = wrong

Top-2 class = {dog, alpaca}
∴ Top-2 Accuracy = correct

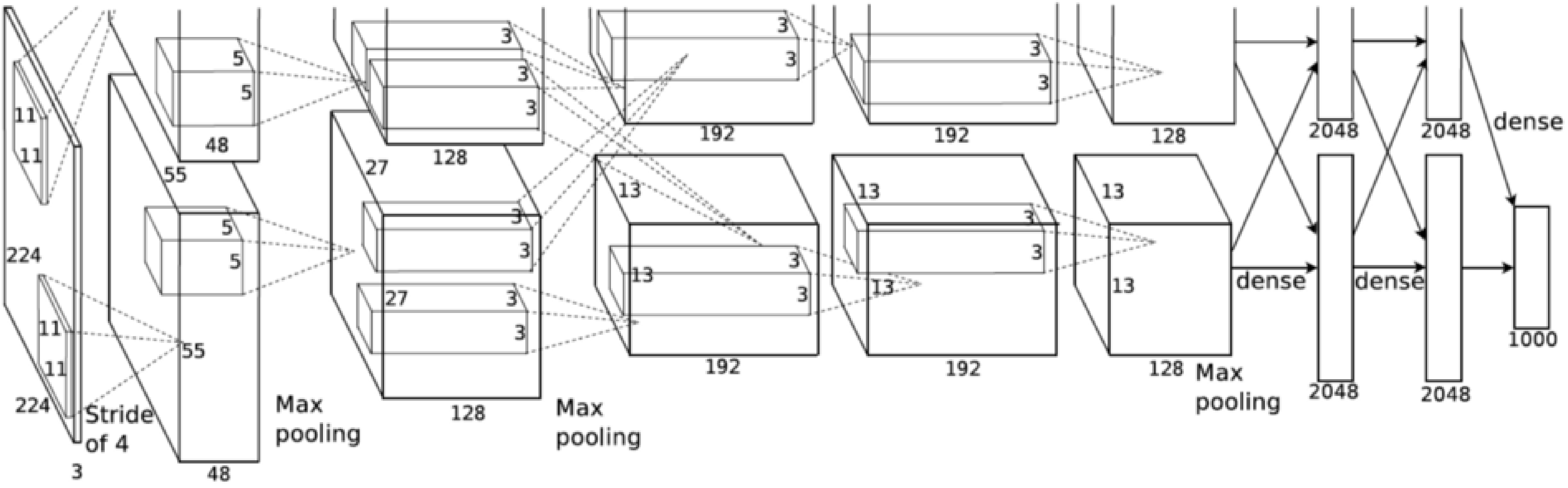
Q. Top-k accuracy for k number of classes?

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

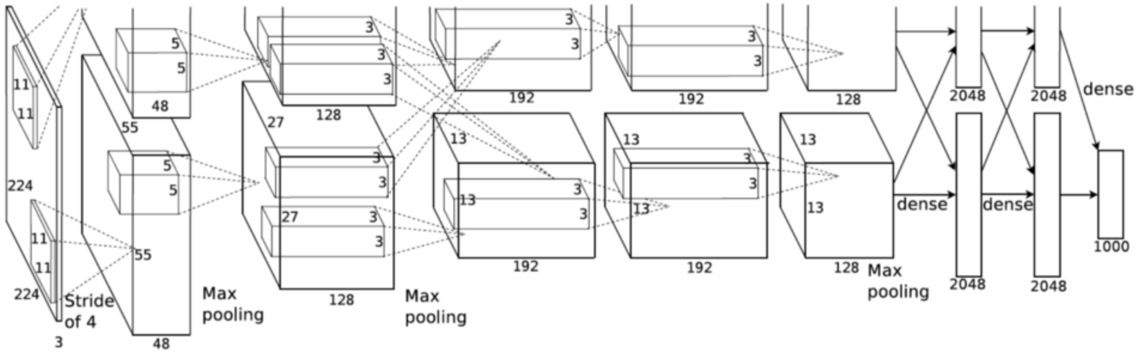


AlexNet

- Krizhevsky, Sutskever, & Hinton. (2012).



AlexNet



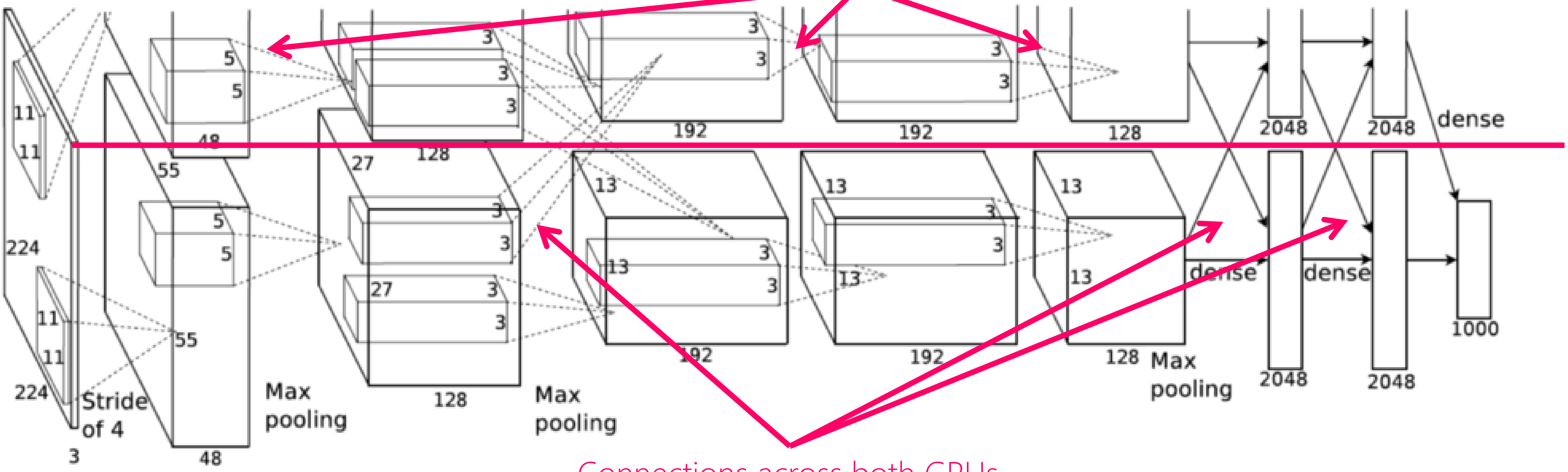
- Krizhevsky, Sutskever, & Hinton. (2012).
- 62.3M Parameters
- 1.1 billion computations in a forward pass
- Batch size 128
- SGD with Momentum 0.9
- Learning rate 0.01. The rate reduces by 10 manually when the validation accuracy plateaus
- Trained on GTX 580 GPU (3 GB memory) → 2x GPUs (5~6 days)
- 7 CNN ensemble (Accuracy 18.2% → 15.4%)

Conv11-96, stride 4
MaxPool3, stride 2
Norm
Conv5-256, stride 1
MaxPool3, stride 2
Norm
Conv3-384, stride 1
Conv3-384, stride 1
Conv3-256, stride 1
MaxPool3, stride 2
Dropout(0.5)
FC4096
Dropout(0.5)
FC4096
FC1000

AlexNet

- Krizhevsky, Sutskever, & Hinton. (2012).

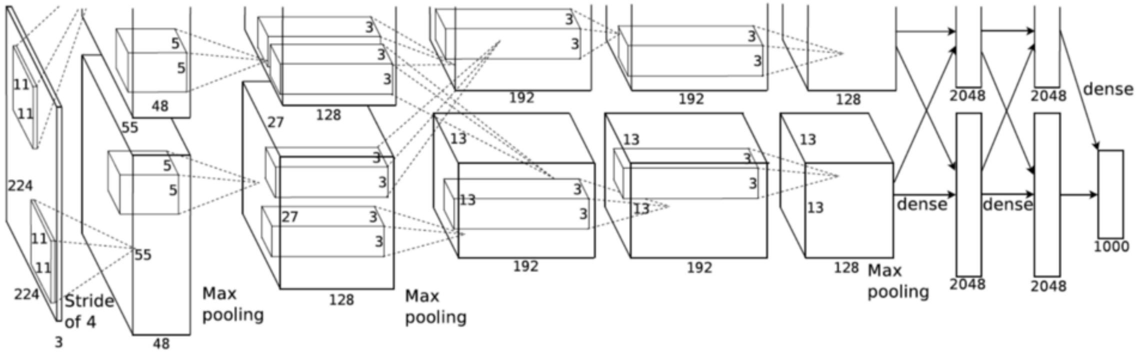
Connections only within individual GPU



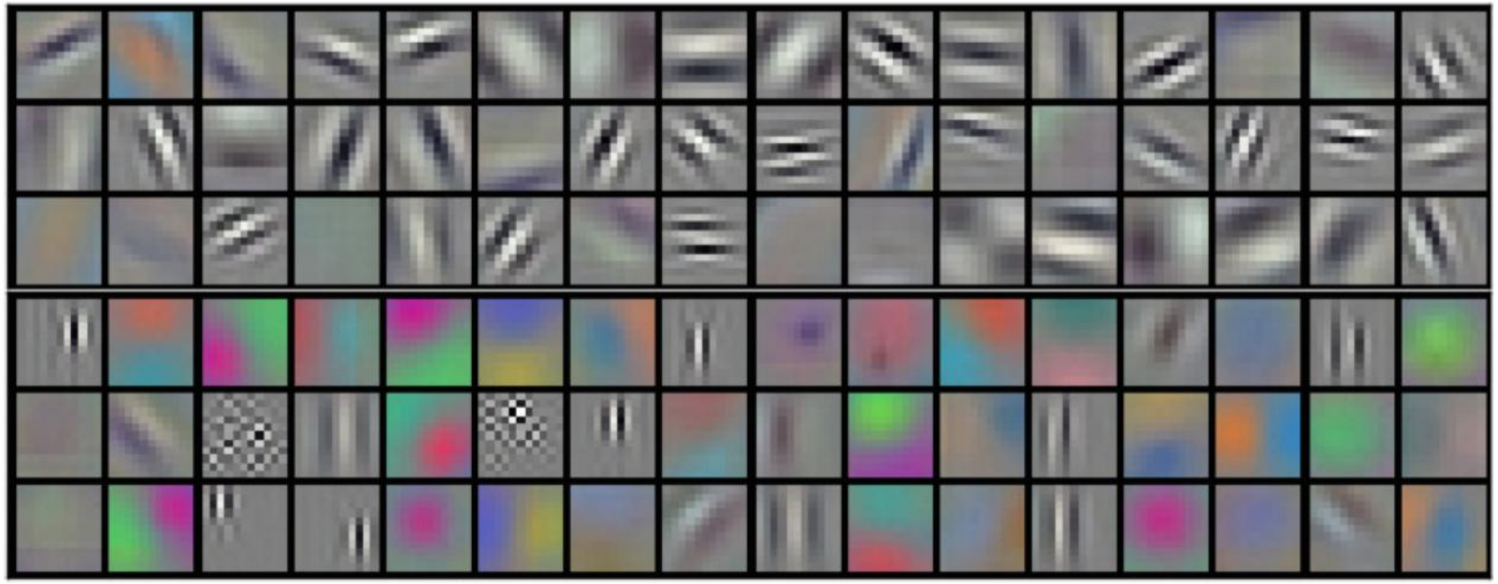
Connections across both GPUs

Dual GPU training → error rate decreases by around 1.7% (Top-1) and 1.2% (Top-5) compared with the one trained with one GPU and half neurons.

AlexNet

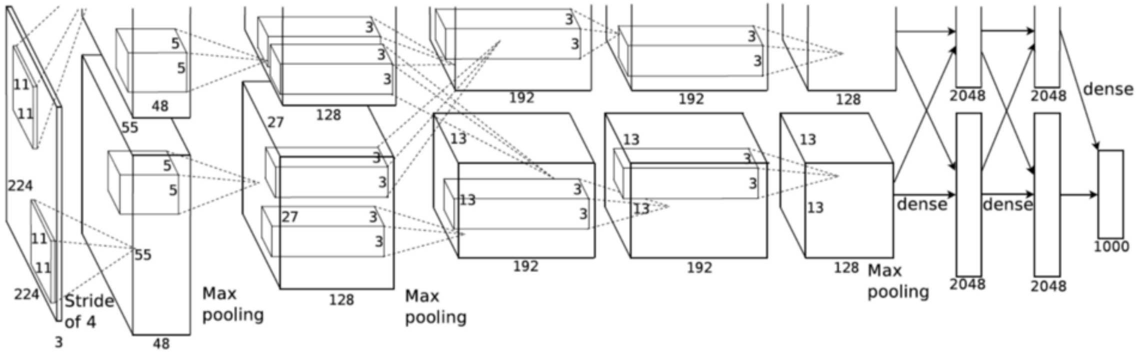


- Krizhevsky, Sutskever, & Hinton. (2012).
- 96 kernels
  - top 48 kernels on GPU 1: color-agnostic
  - Bottom 48 kernels on GPU 2: color-specific



Conv11-96, stride 4
MaxPool3, stride 2
Norm
Conv5-256, stride 1
MaxPool3, stride 2
Norm
Conv3-384, stride 1
Conv3-384, stride 1
Conv3-256, stride 1
MaxPool3, stride 2
Dropout(0.5)
FC4096
Dropout(0.5)
FC4096
FC1000

AlexNet



- Krizhevsky, Sutskever, & Hinton. (2012).

- Local Response Normalization

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^\beta$$

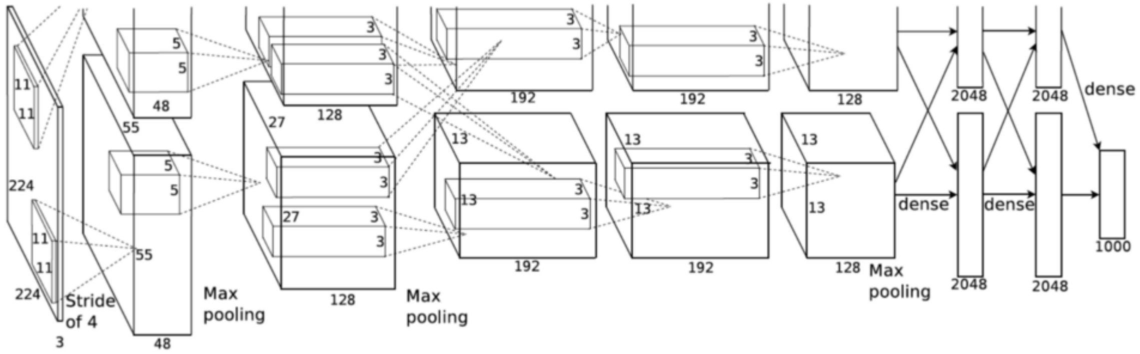
Response-normalized activity

Activity of a neuron computed by applying kernel I at position (x,y) and then applying the ReLU nonlinearity

Response normalization reduces error rates by 1.4% (Top-1) & 1.2% (Top-5)

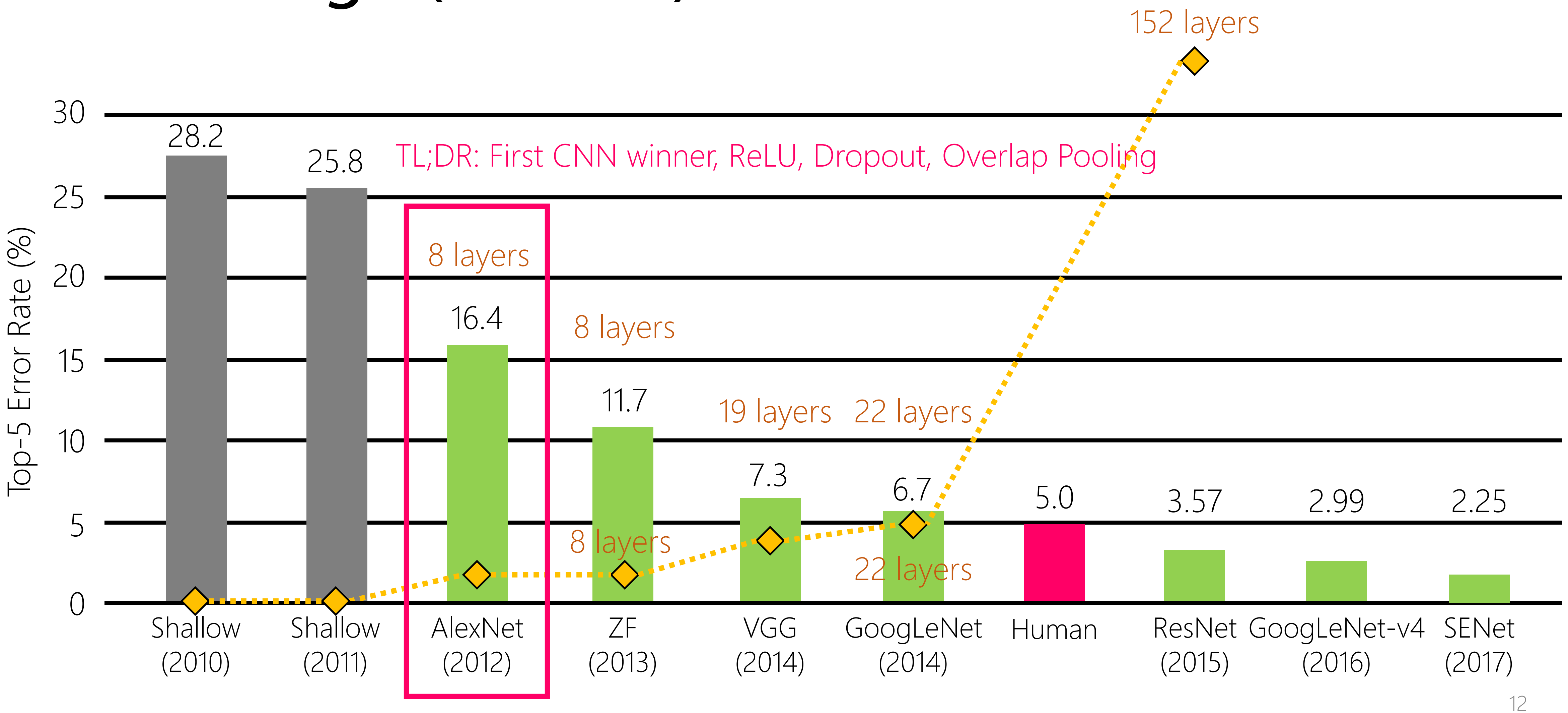
Conv11-96, stride 4
MaxPool3, stride 2
Norm
Conv5-256, stride 1
MaxPool3, stride 2
Norm
Conv3-384, stride 1
Conv3-384, stride 1
Conv3-256, stride 1
MaxPool3, stride 2
Dropout(0.5)
FC4096
Dropout(0.5)
FC4096
FC1000

AlexNet

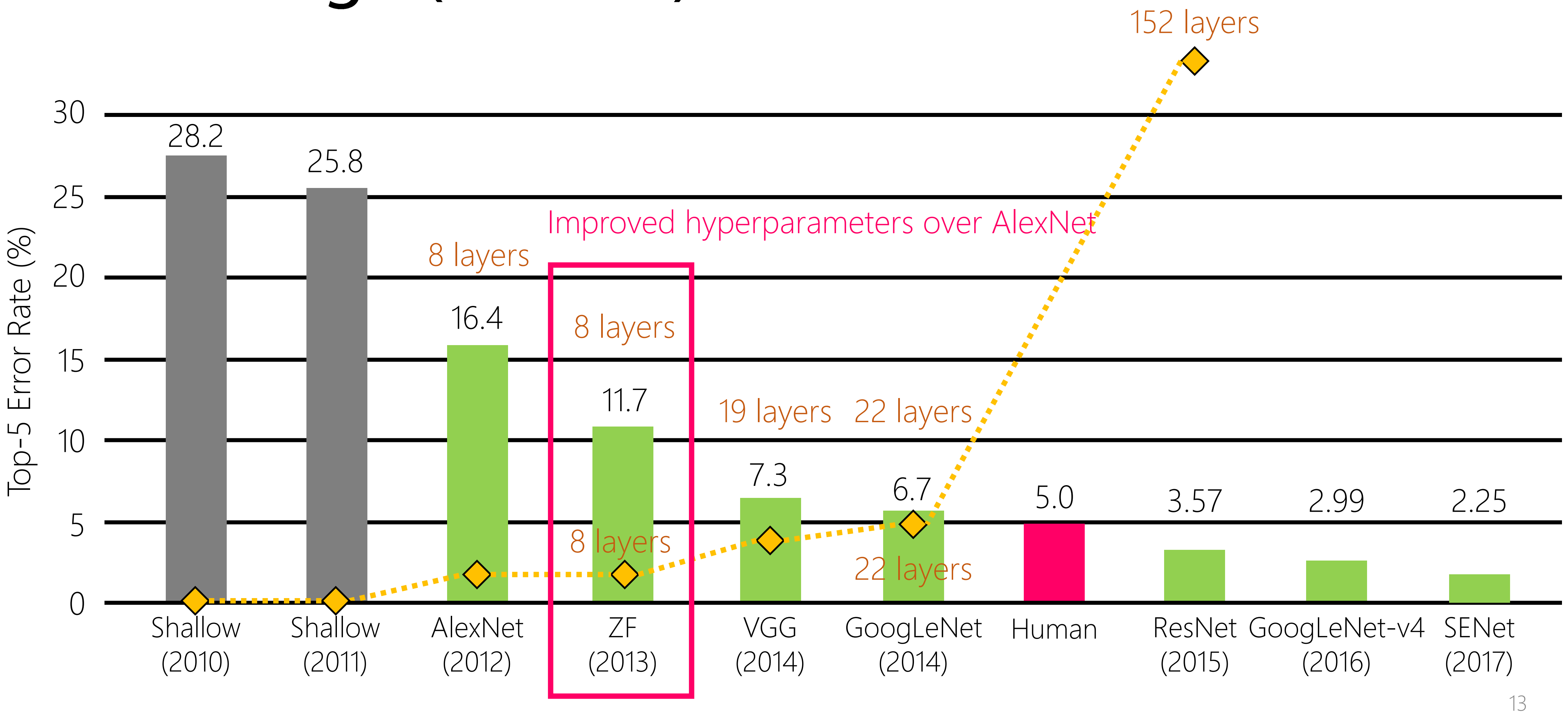


- Krizhevsky, Sutskever, & Hinton. (2012).
- Famously won the LSVRC 2012 by a large margin. (First CNN-based winner)
- First use of ReLU, instead of tanh or sigmoid.
- Dropout has been used to deal with overfitting.
- Data augmentation
- Overlap pooling → error rate decreases around 0.4% (Top-1) & 0.3% (Top-5)
- Used Norm layers (different from batch norm) → not popular anymore (turned out not very useful)

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

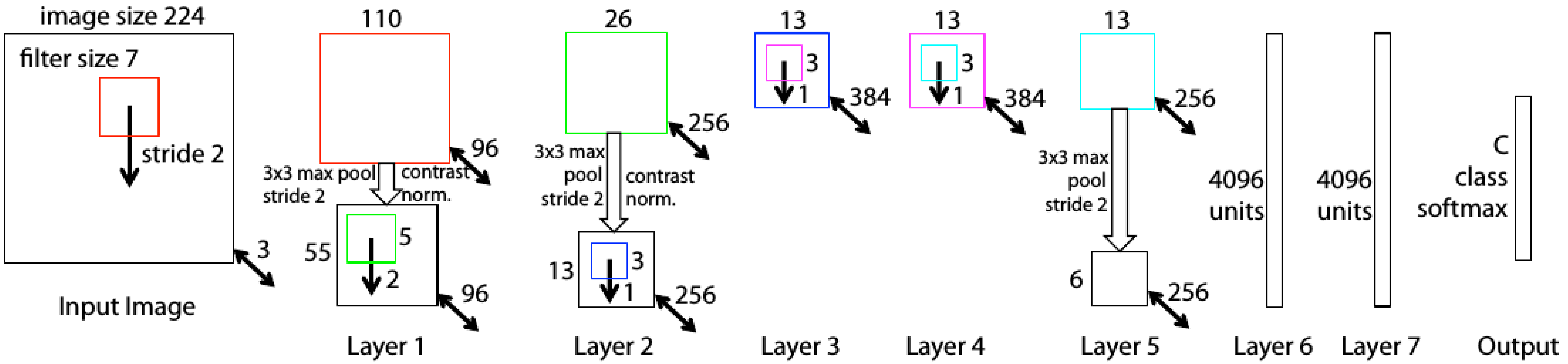


ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners

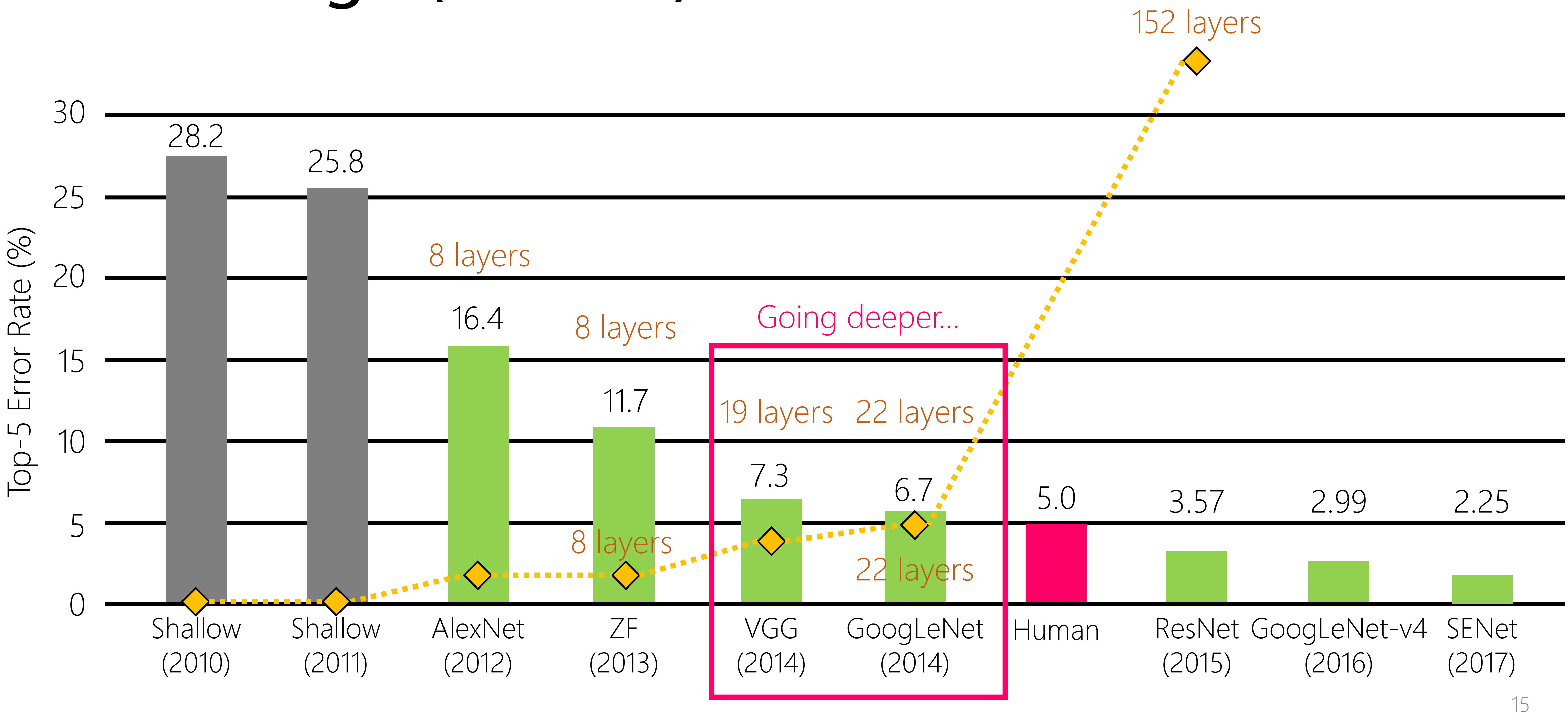


ZFNet

- Zeiler & Fergus (2013)
- Basically AlexNet, but
  - The first conv layer is changed from (11x11 stride 4) to (7x7 stride 2)
  - The third~fifth conv layer: instead of 384, 384, 256 kernels, ZFNet uses 512, 1024, 512 kernels.

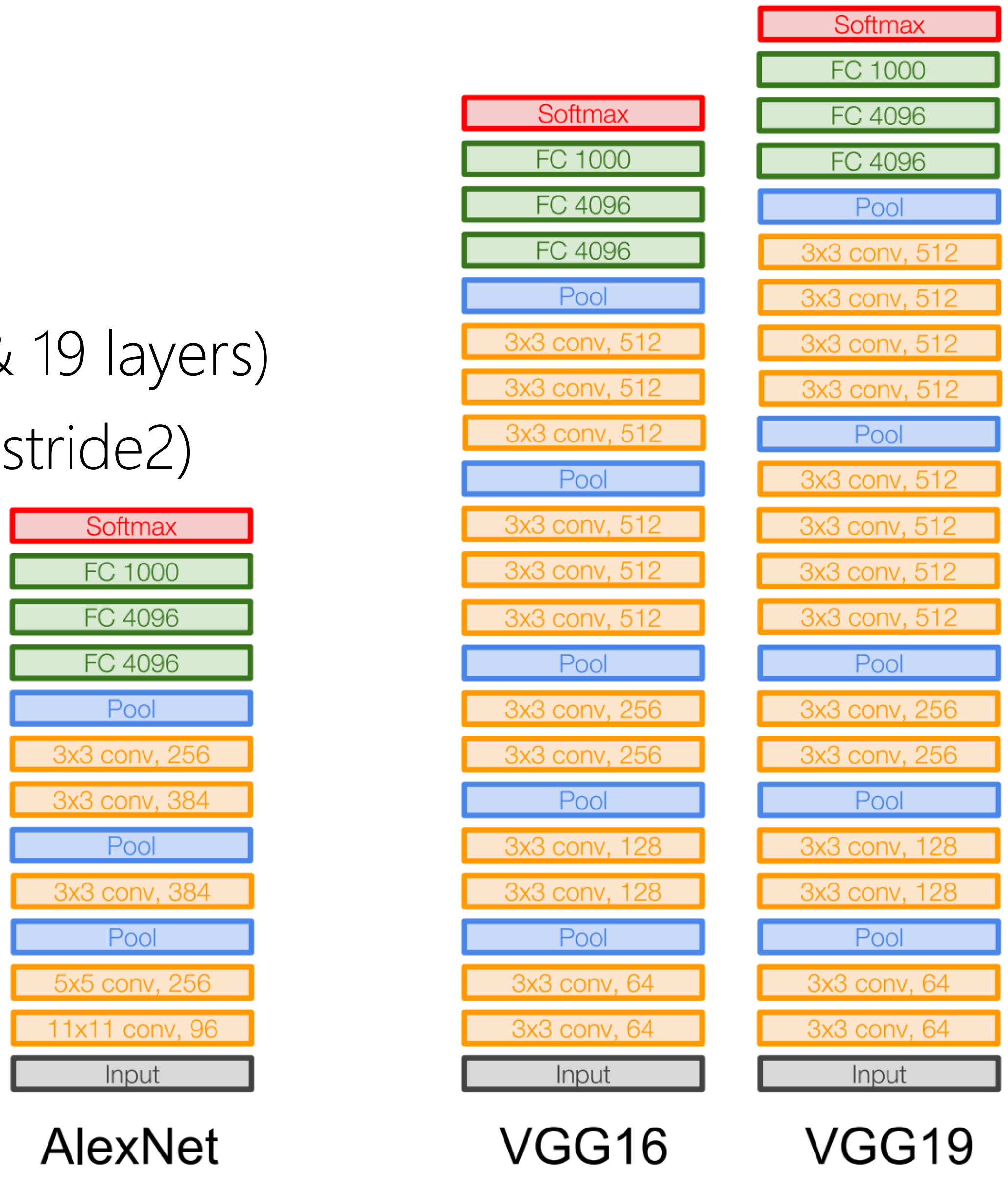


ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



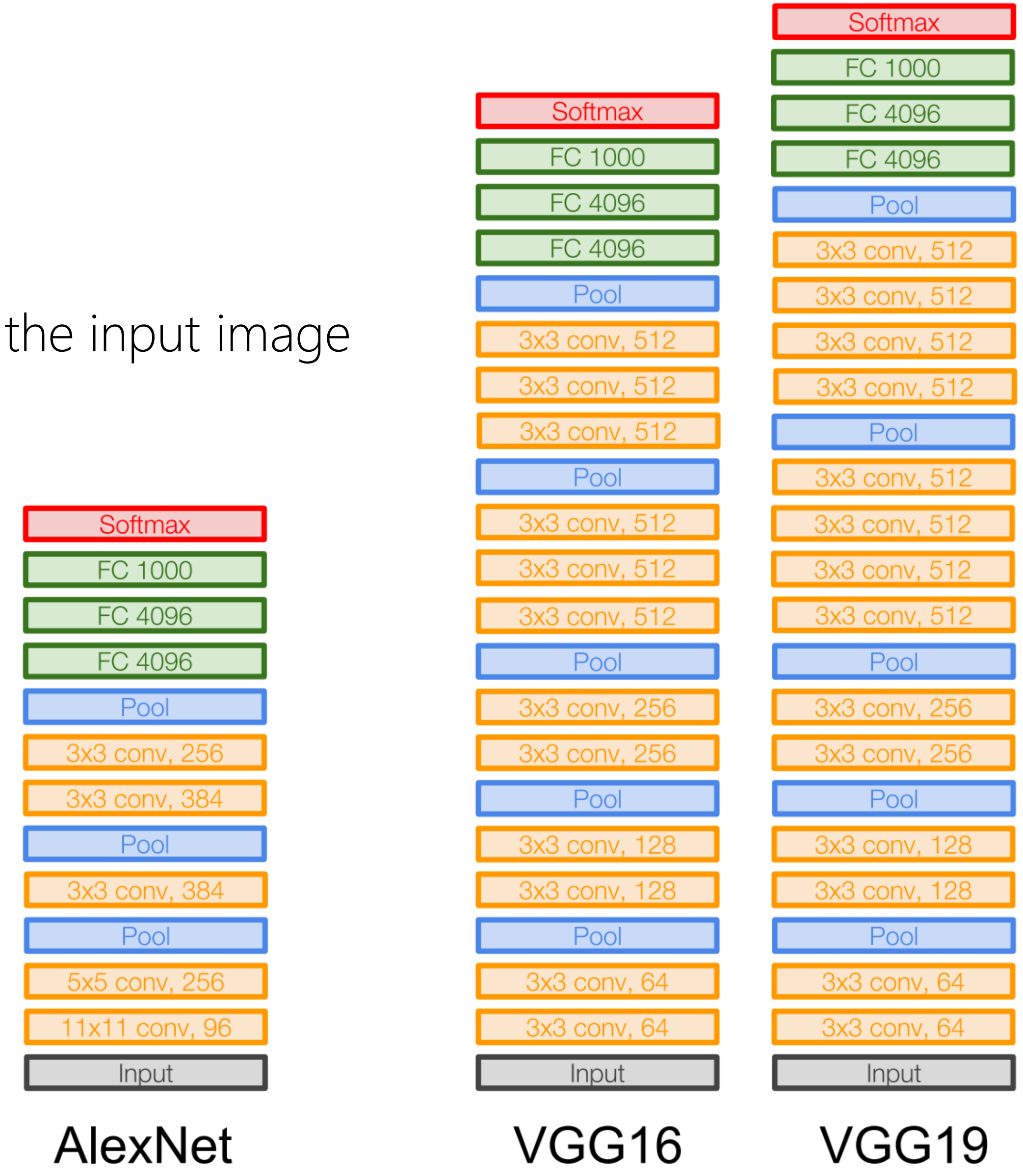
VGGNet

- Go deeper with smaller filters!
- AlexNet (8 layers) → VGG16 & VGG19 (16 & 19 layers)
- Only (Conv3-stride1-pad1) and (MaxPool2-stride2)
- First time went below 10% error.



VGGNet

- Why use small filters?
  - Stack of three Conv3 layers covers 7x7 area in the input image (receptive field)
  - Same RF, but deeper, more nonlinearity
  - Obviously, fewer parameters! (7*7 vs 3*3*3)



VGGNet

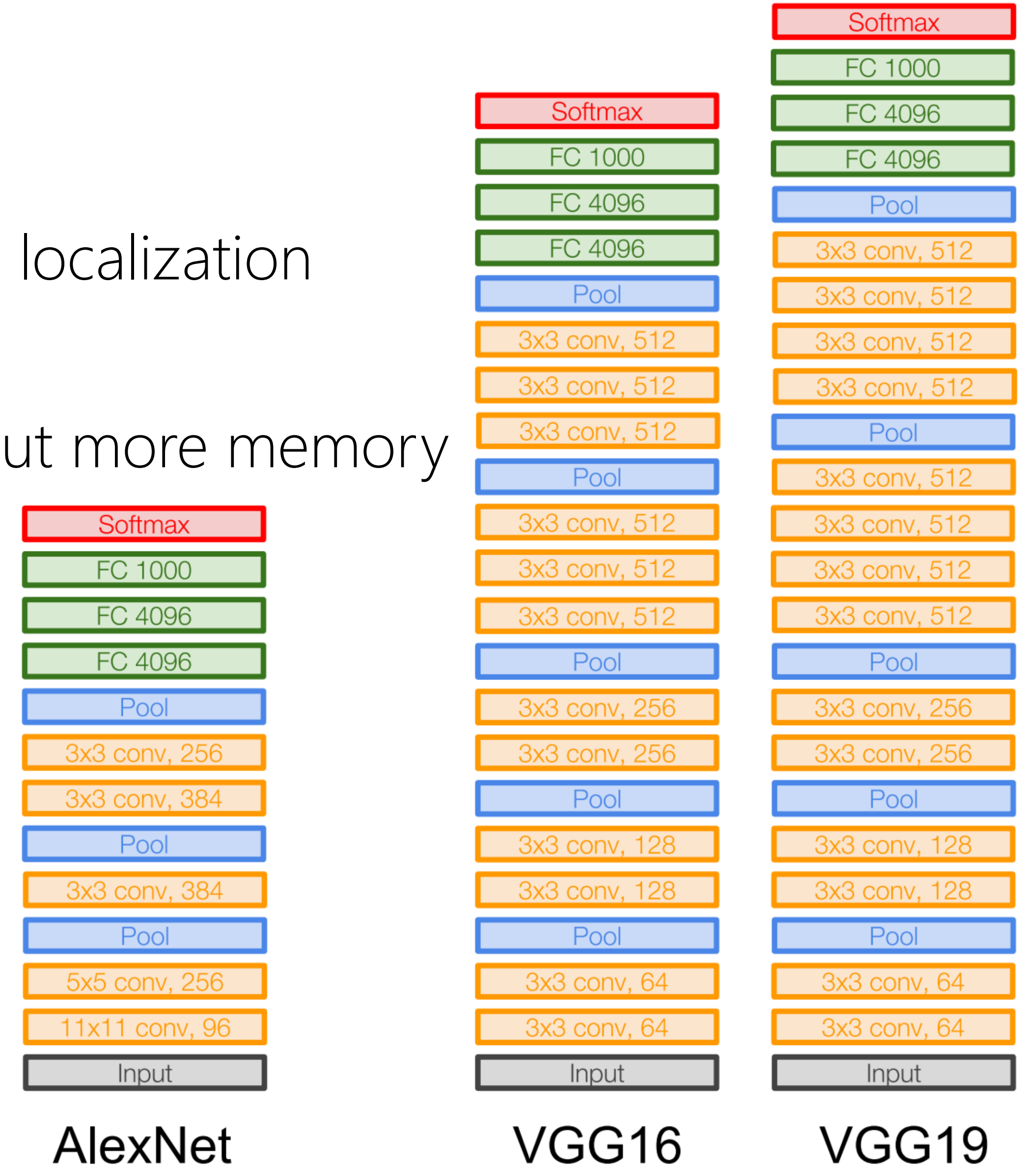
INPUT: [224x224x3] memory: 224*224*3=150K params: 0 (not counting biases)
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K params: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K params: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K params: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K params: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K params: 0
FC: [1x1x4096] memory: 4096 params: 7*7*512*4096 = 102,760,448
FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 96MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters

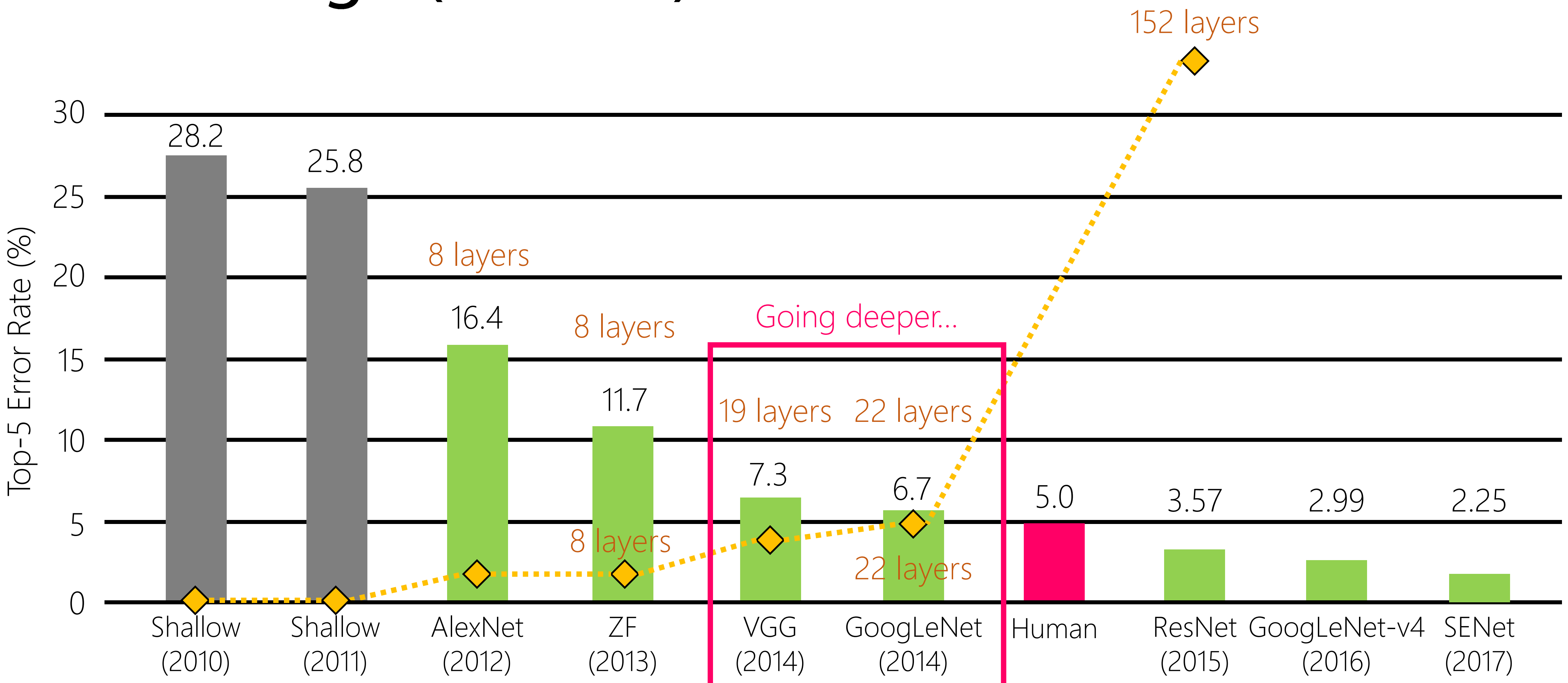
VGG16

VGGNet

- ILSVRC'14 runner-up in classification, 1st in localization
- No local response normalization
- VGG19 is only slightly better than VGG16 but more memory
- Simple and straightforward, the second FC layer generalizes quite well when transferred → "garlic salt"

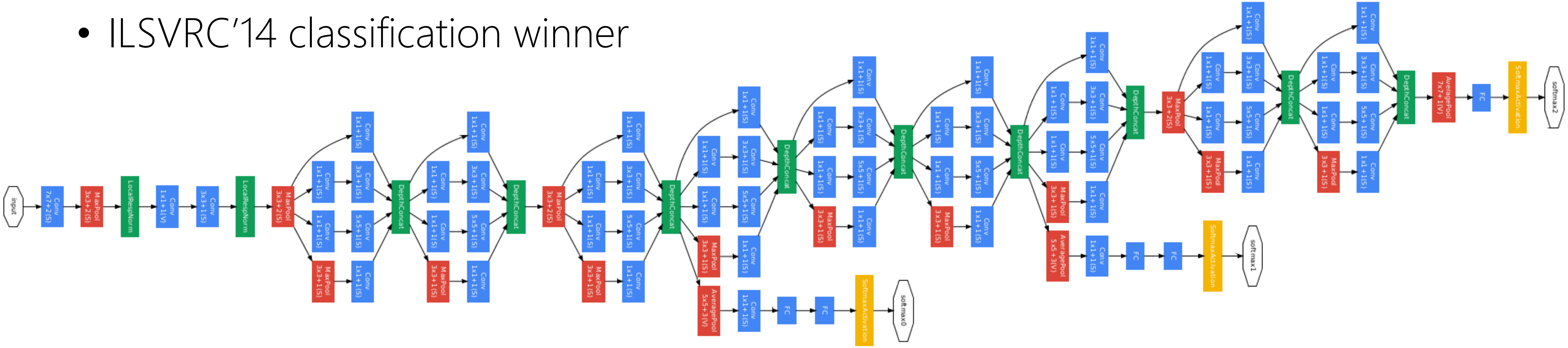


ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



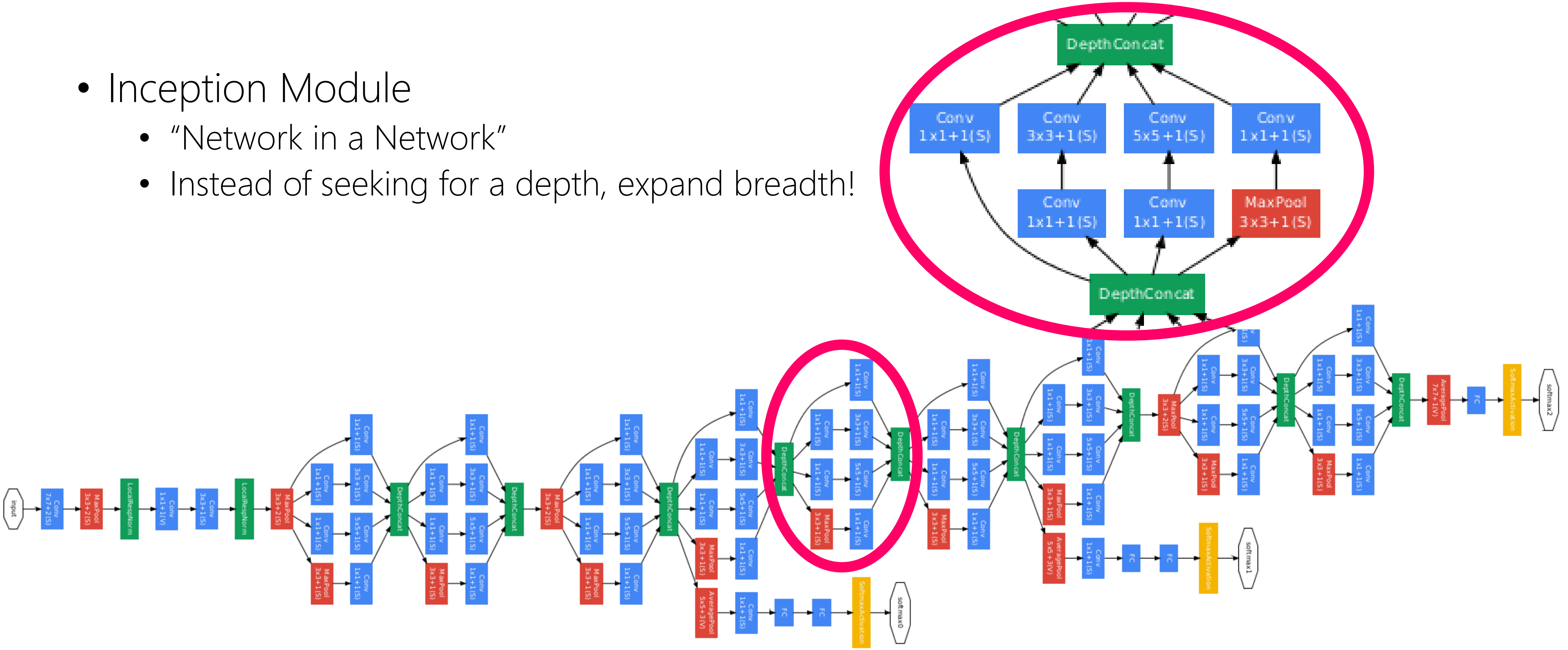
GoogLeNet a.k.a. Inception Network

- Deeper but efficient computation!
- Efficiency via inception module
- Reduced FC layers
- Only 5M, as opposed to 62M (AlexNet) & 138M (VGG16)
- ILSVRC'14 classification winner



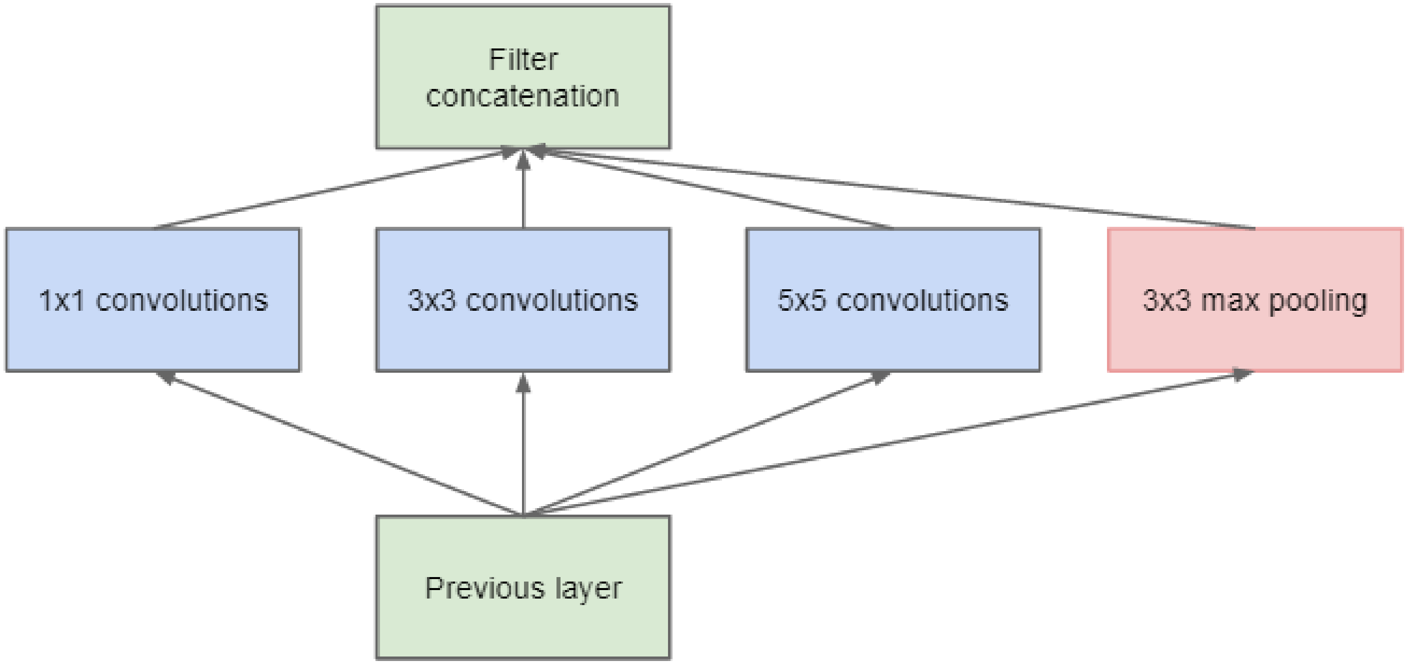
GoogLeNet a.k.a. Inception Network

- Inception Module
  - “Network in a Network”
  - Instead of seeking for a depth, expand breadth!



GoogLeNet a.k.a. Inception Network

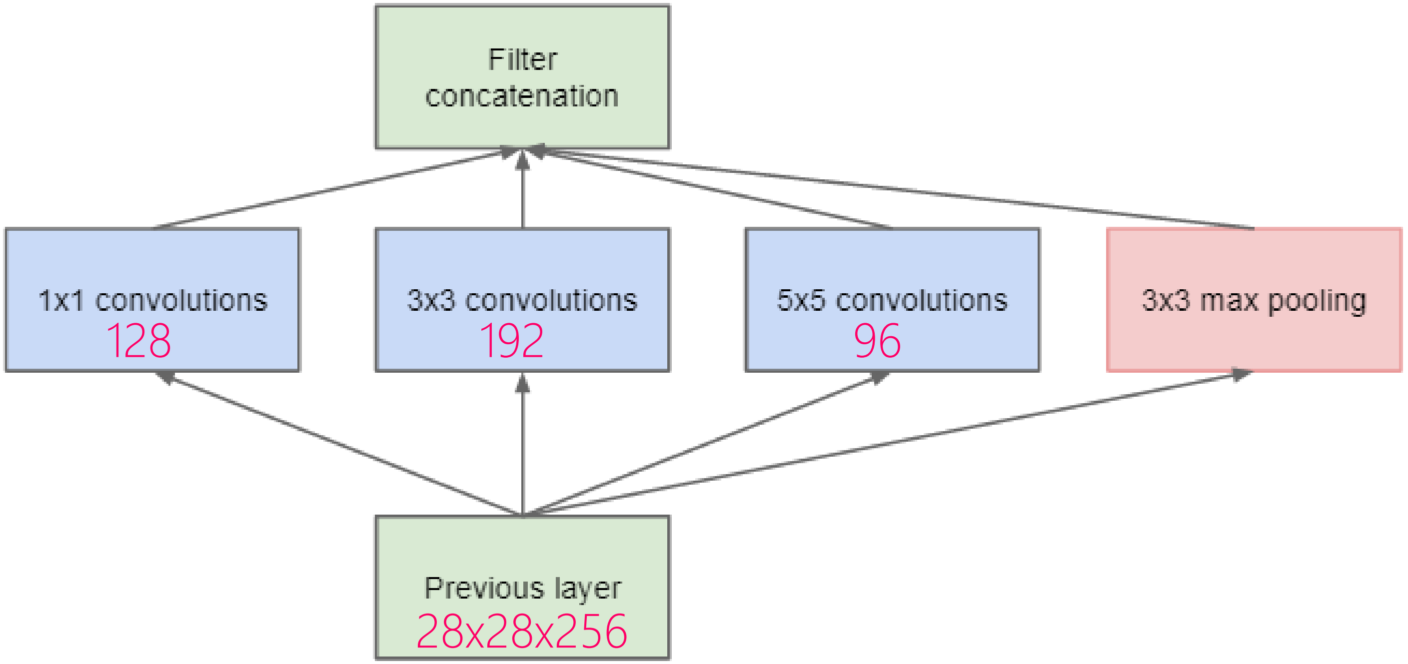
- Naïve Inception Module
  - "Why bother to decide which size of convolution to use when you can have them all?"



(a) Inception module, naïve version

GoogLeNet a.k.a. Inception Network

- Naïve Inception Module
  - "Why bother to decide which size of convolution to use when you can have them all?"

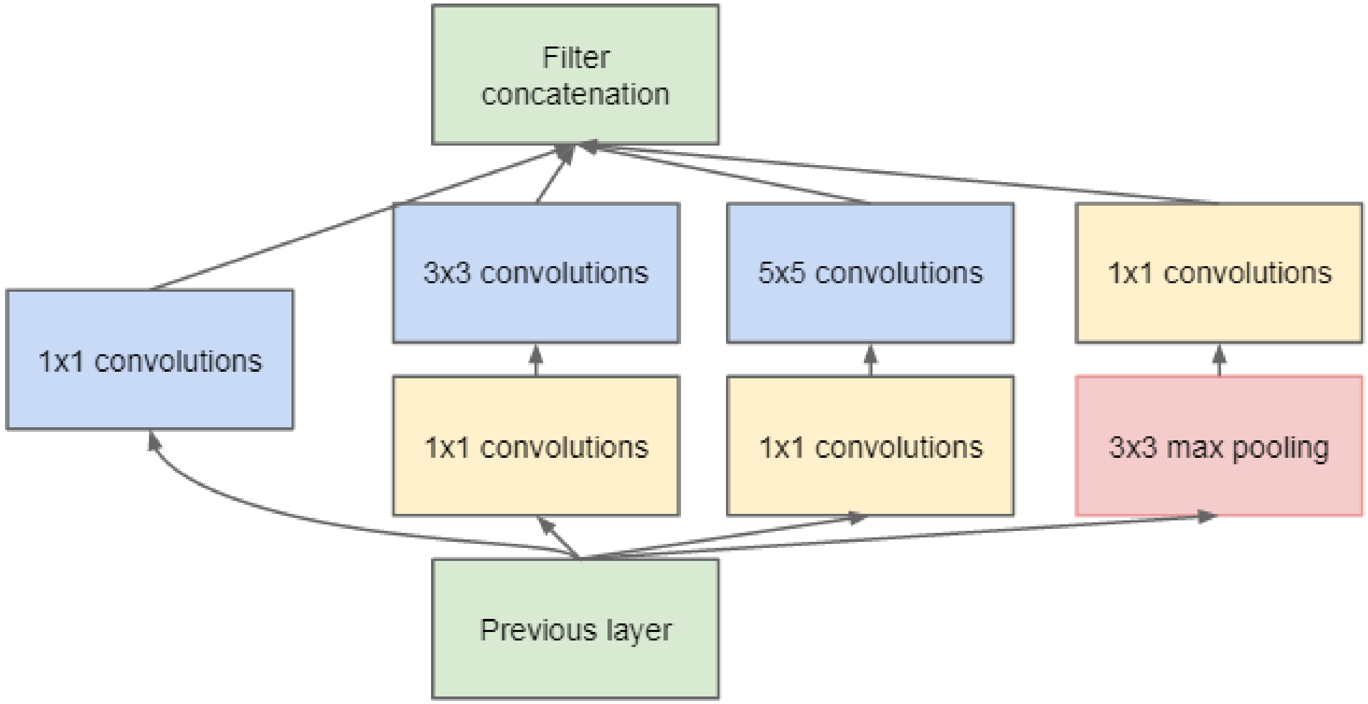


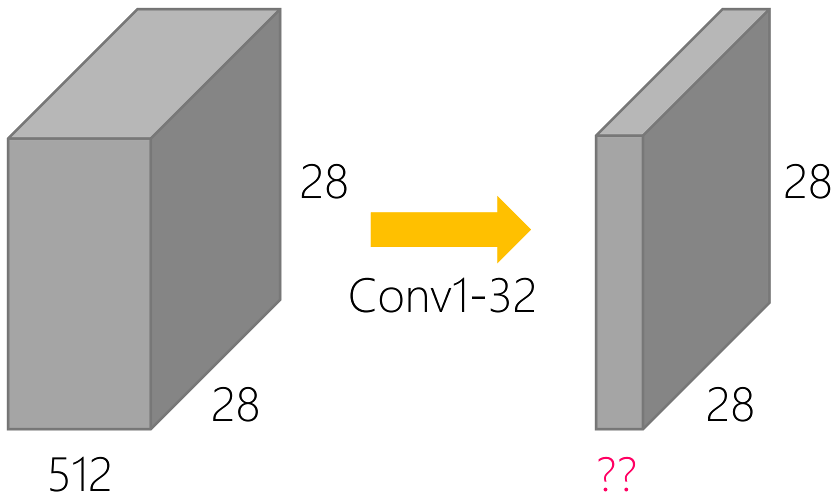
(a) Inception module, naïve version

Q. What's wrong with the naïve version?

GoogLeNet a.k.a. Inception Network

- Inception module with bottleneck layers

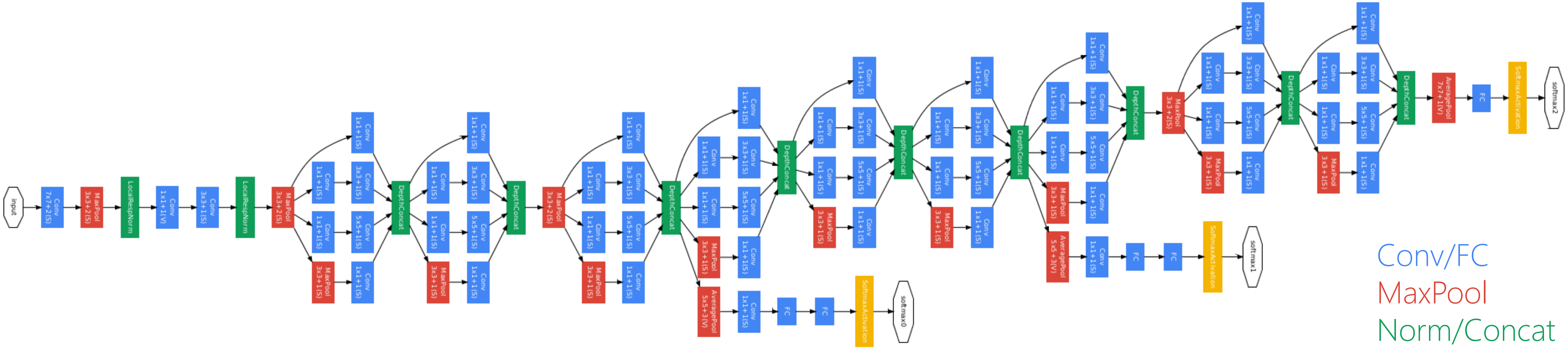




(b) Inception module with dimension reductions

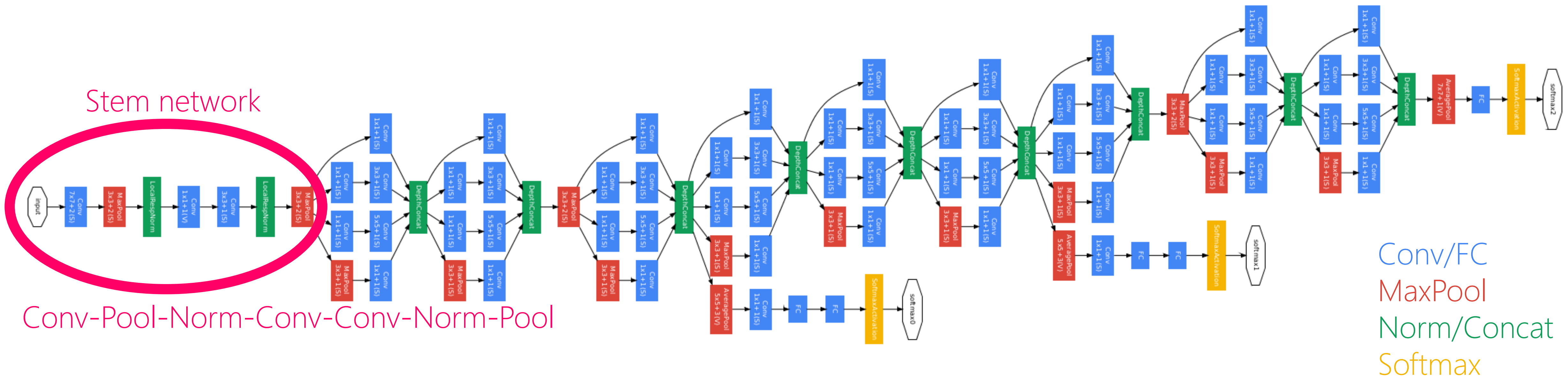
GoogLeNet a.k.a. Inception Network

- Putting them all together…



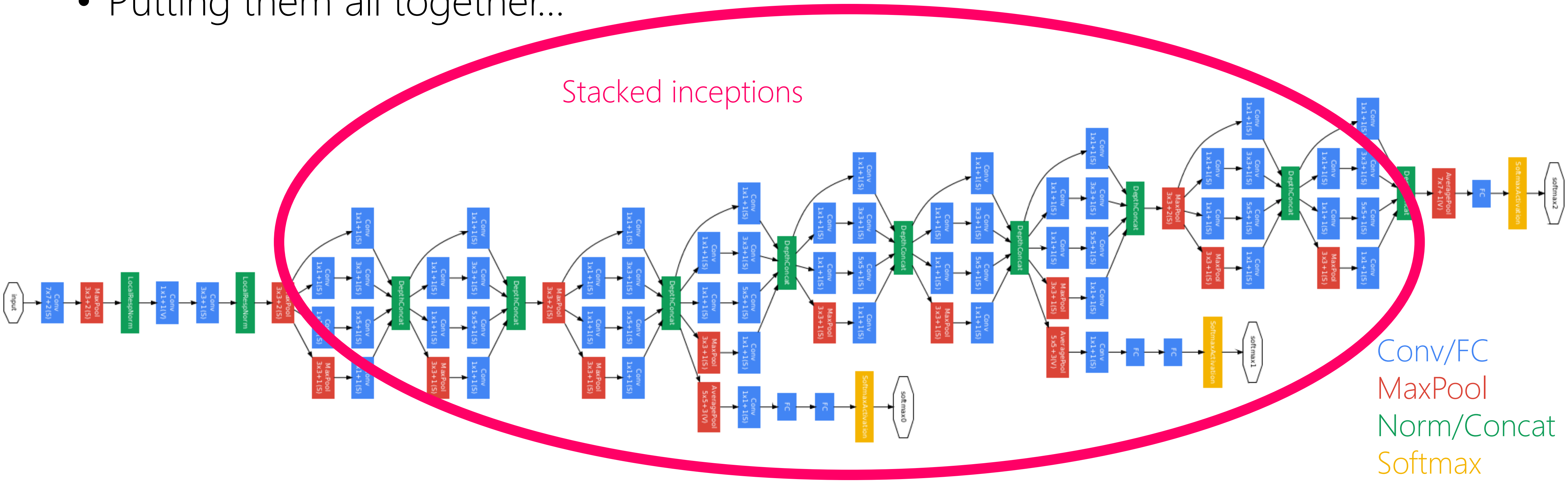
GoogLeNet a.k.a. Inception Network

- Putting them all together…



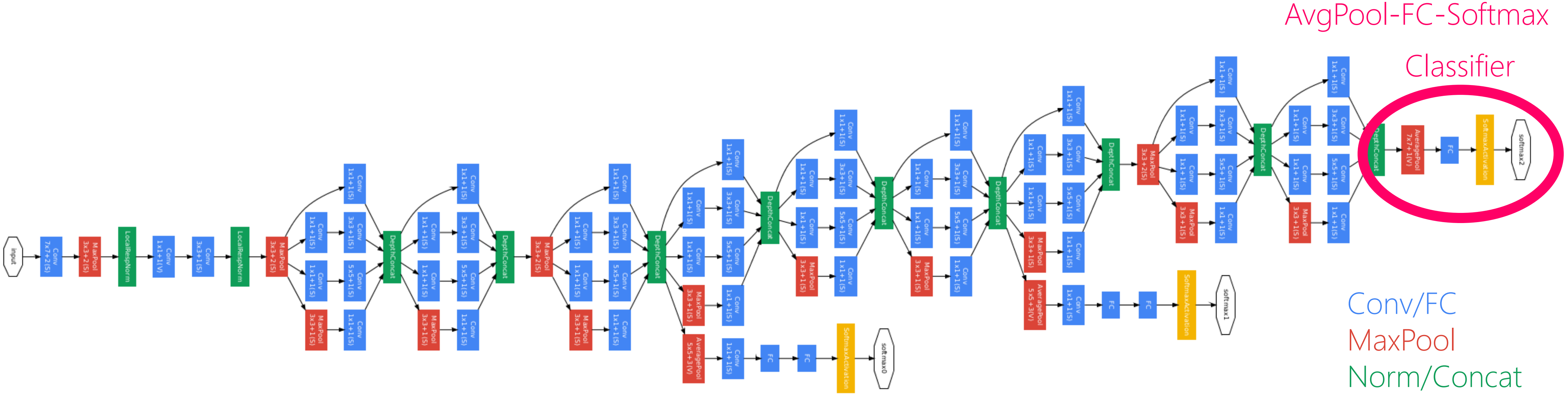
GoogLeNet a.k.a. Inception Network

- Putting them all together…



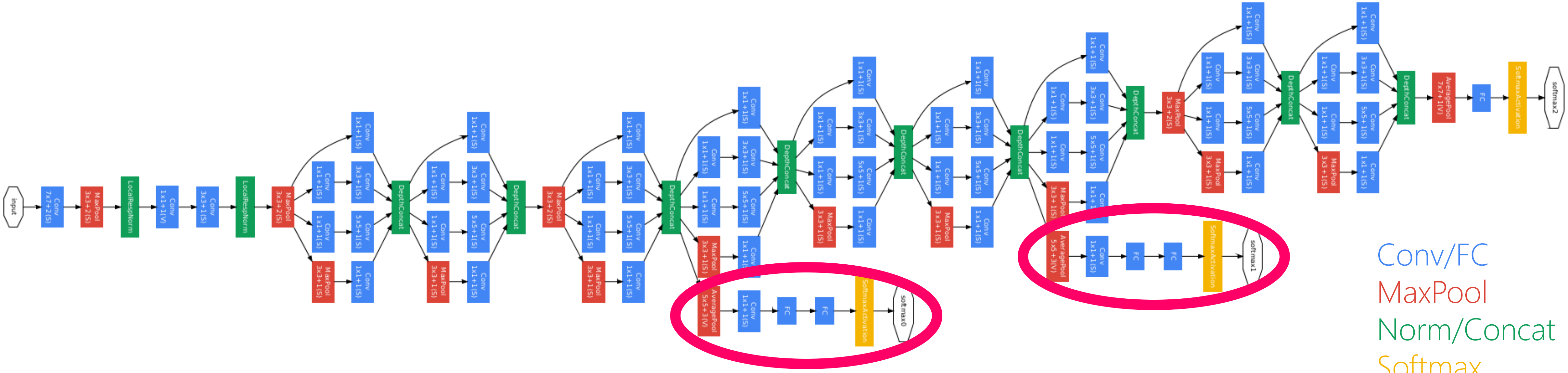
GoogLeNet a.k.a. Inception Network

- Putting them all together...



GoogLeNet a.k.a. Inception Network

- Putting them all together…

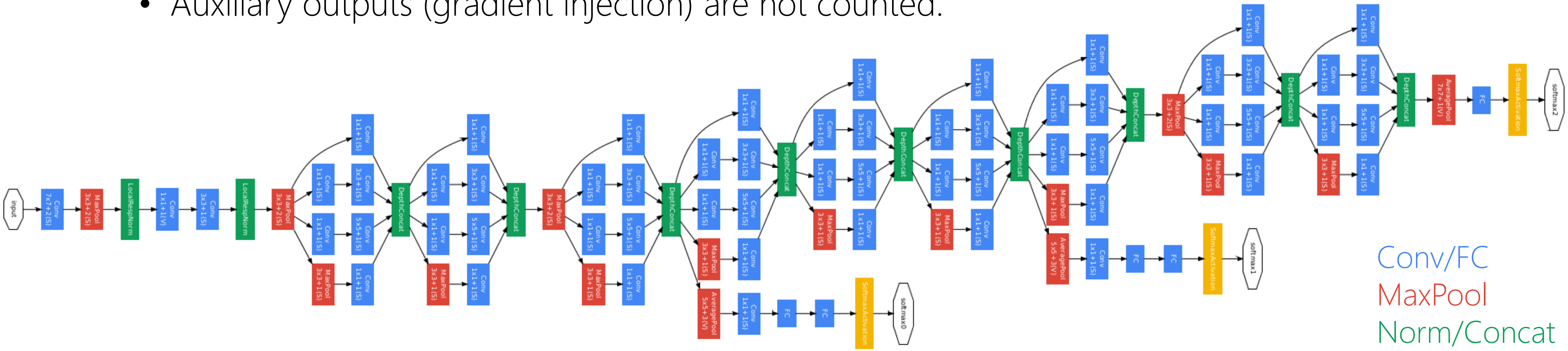


Conv/FC
MaxPool
Norm/Concat
Softmax

Additional classification losses to inject additional gradients in order to prevent vanishing gradient.

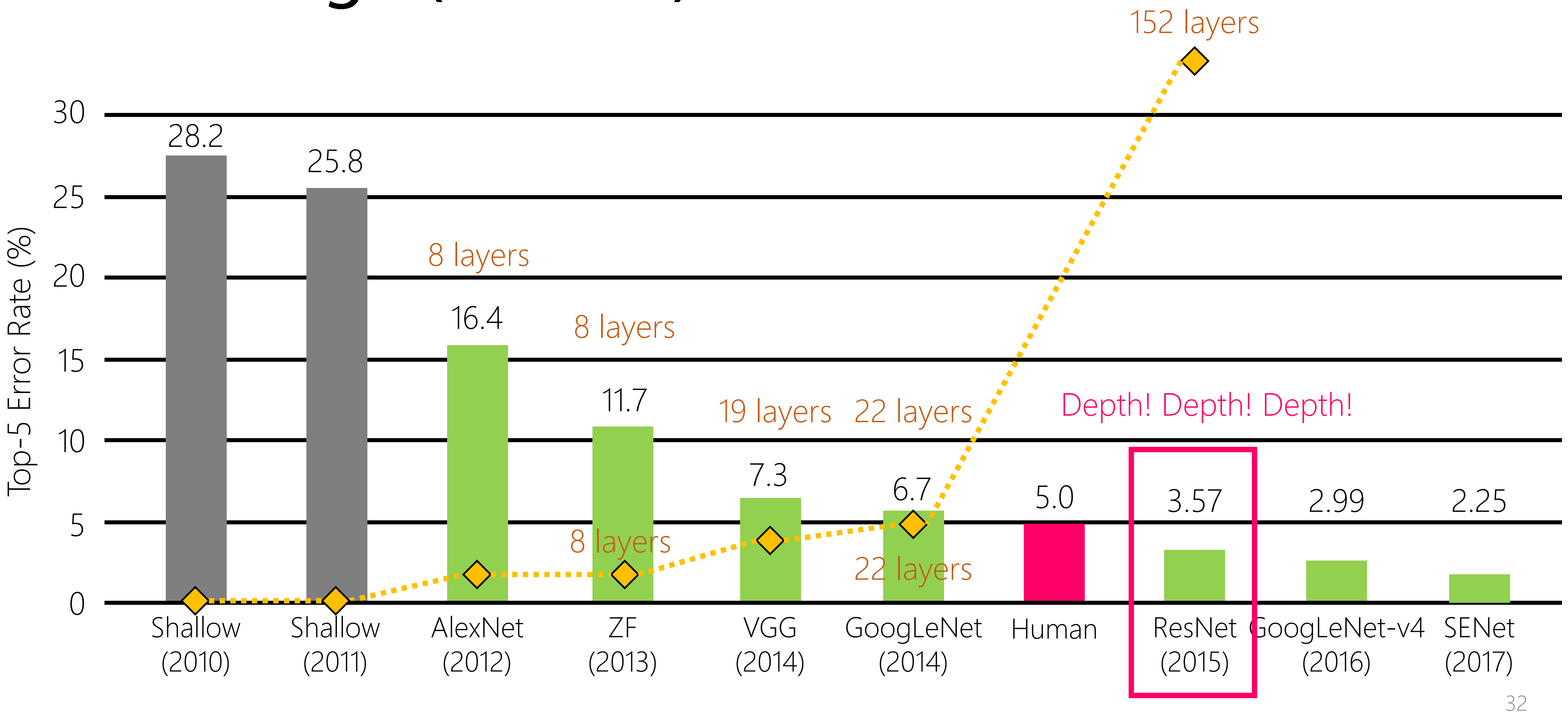
GoogLeNet a.k.a. Inception Network

- How to count 22 layers?
  - Parallel layers are considered as one. (i.e. 2 layers per inception)
  - Auxiliary outputs (gradient injection) are not counted.



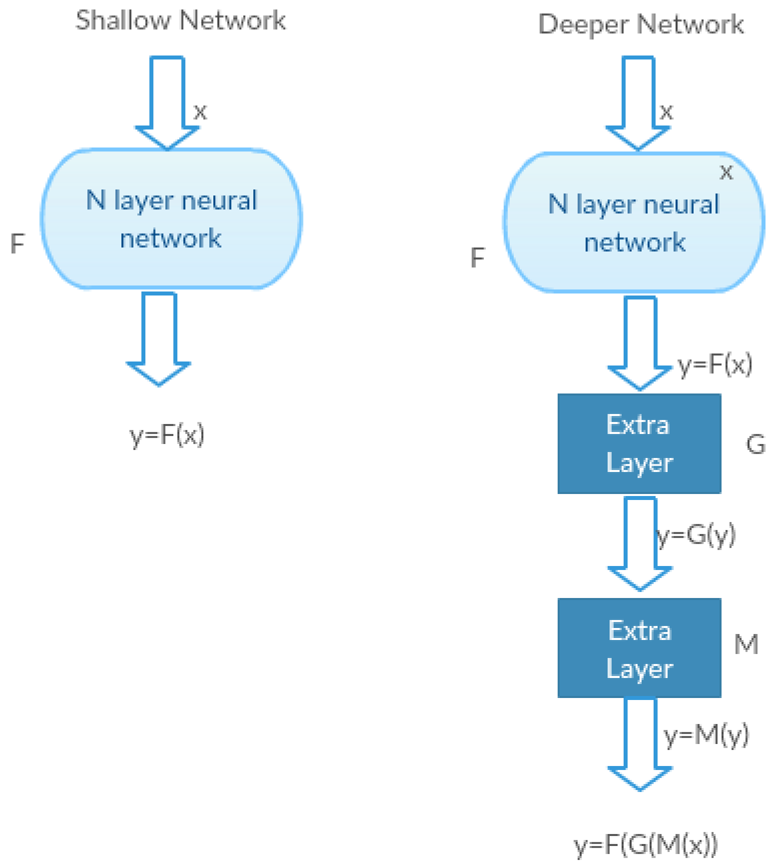
Conv/FC
MaxPool
Norm/Concat
Softmax

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



Adding more layers seems to work well, but…

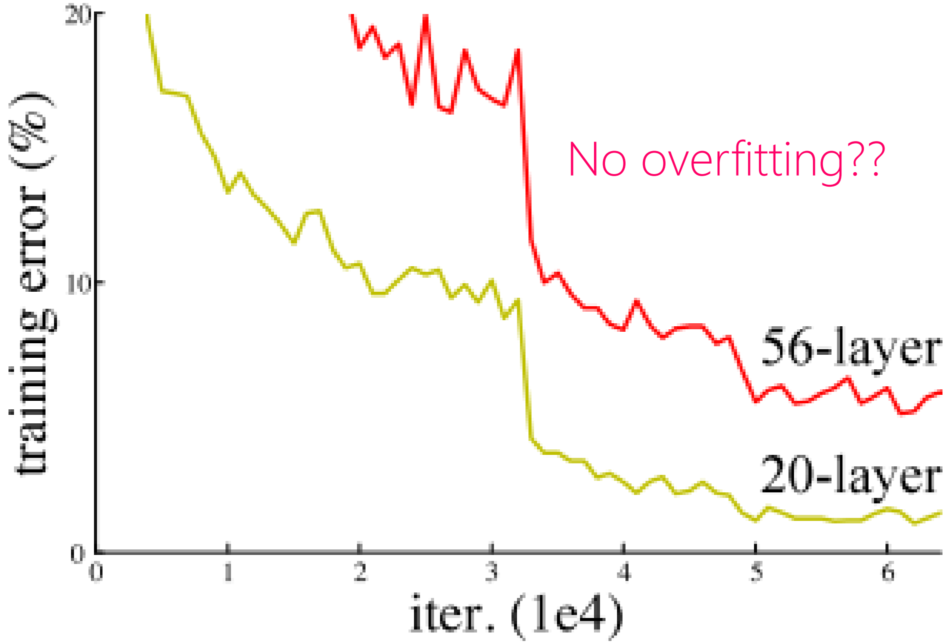
- Deeper models should be able to perform at least as good as shallow models…
- Why?

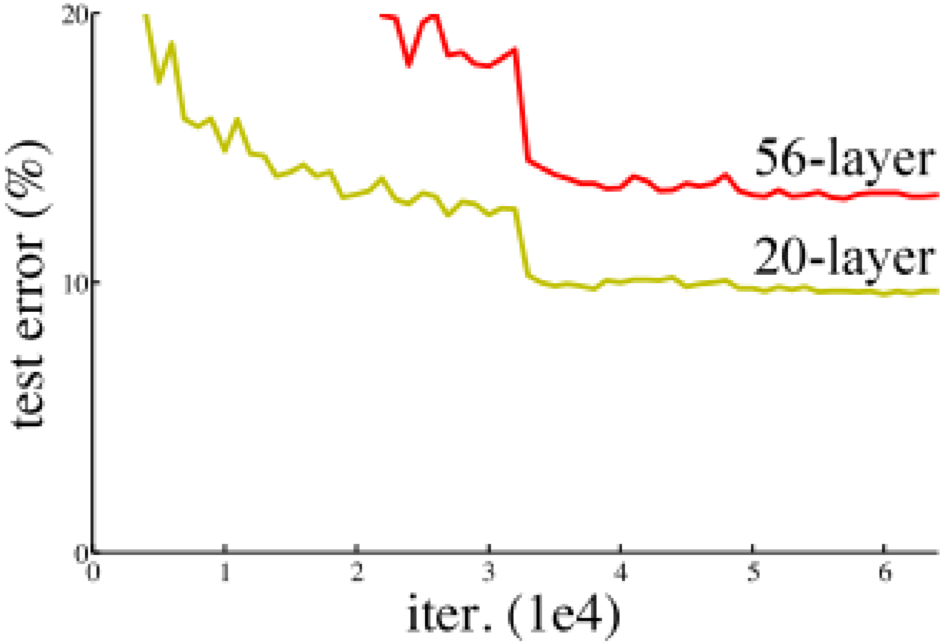


https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624

Adding more layers seems to work well, but…

- Degradation of accuracy with more layers…

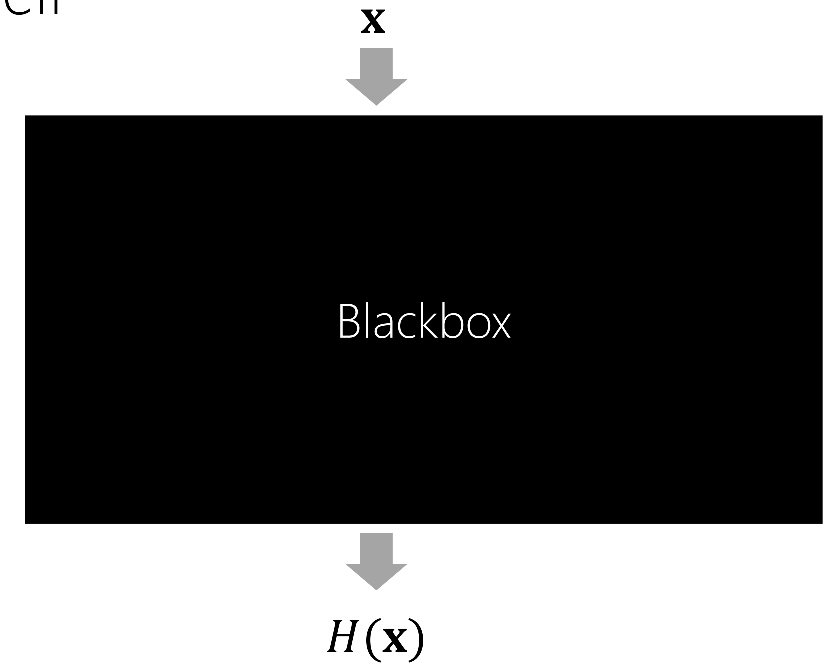




https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

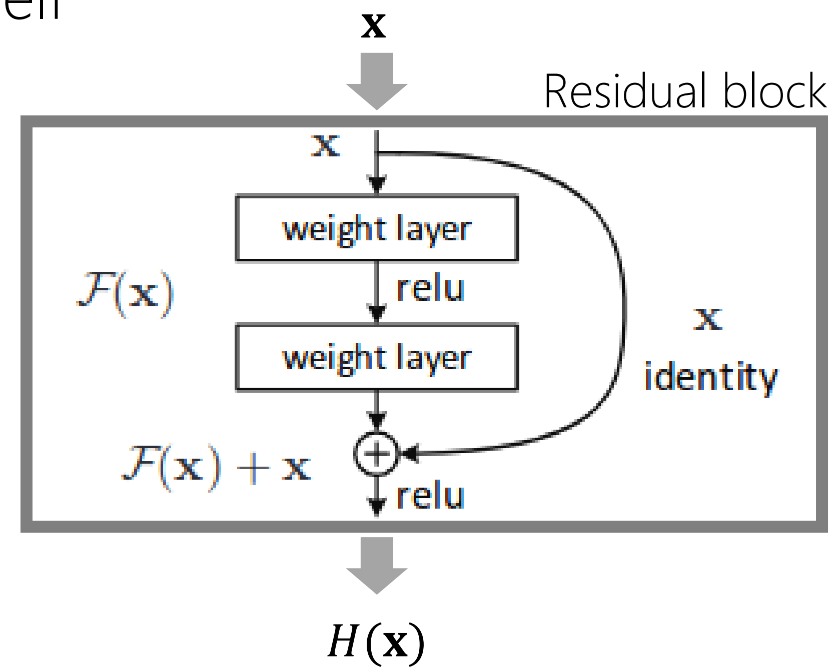
ResNet

- Hypothesis: the degradation problem is due to an optimization problem (deeper models are harder to optimize)
- Proposed solution: Let’s fit the residual (i.e. the extra accuracy gain of a conv block compared to the identity mapping) instead of directly fitting the underlying mapping itself



ResNet

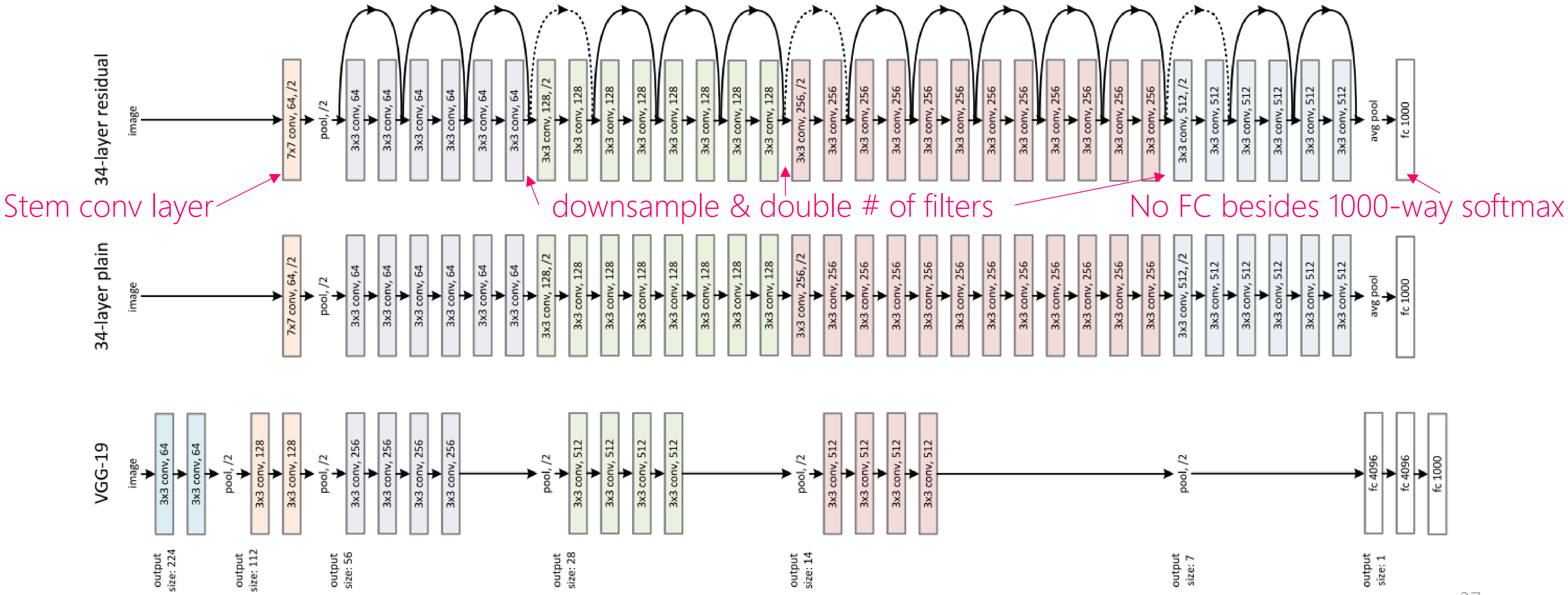
- Hypothesis: the degradation problem is due to an optimization problem (deeper models are harder to optimize)
- Proposed solution: Let's fit the residual (i.e. the extra accuracy gain of a conv block compared to the identity mapping) instead of directly fitting the underlying mapping itself



Fit F(x)=H(x)-x (residual) instead of H(x) directly!

ResNet

- Full ResNet architecture: stack residual blocks each of which has two Conv3.



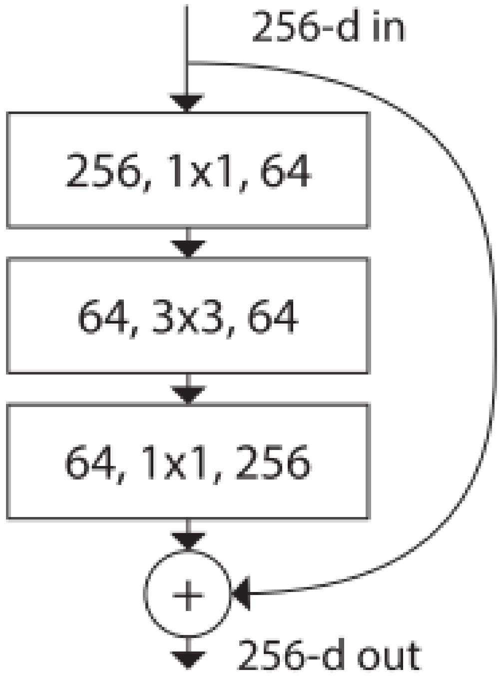
https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035

ResNet

- ResNet-34, ResNet-50, ResNet-101, ResNet-152 available…
- ResNet-50+ uses bottleneck layers to improve efficiency

- Batch normalization after every conv
- SGD with Momentum 0.9
- Learning rate: 0.1, divided by 10 when val error plateaus
- Mini-batch size 256
- No dropout

→ Can be trained without degradation

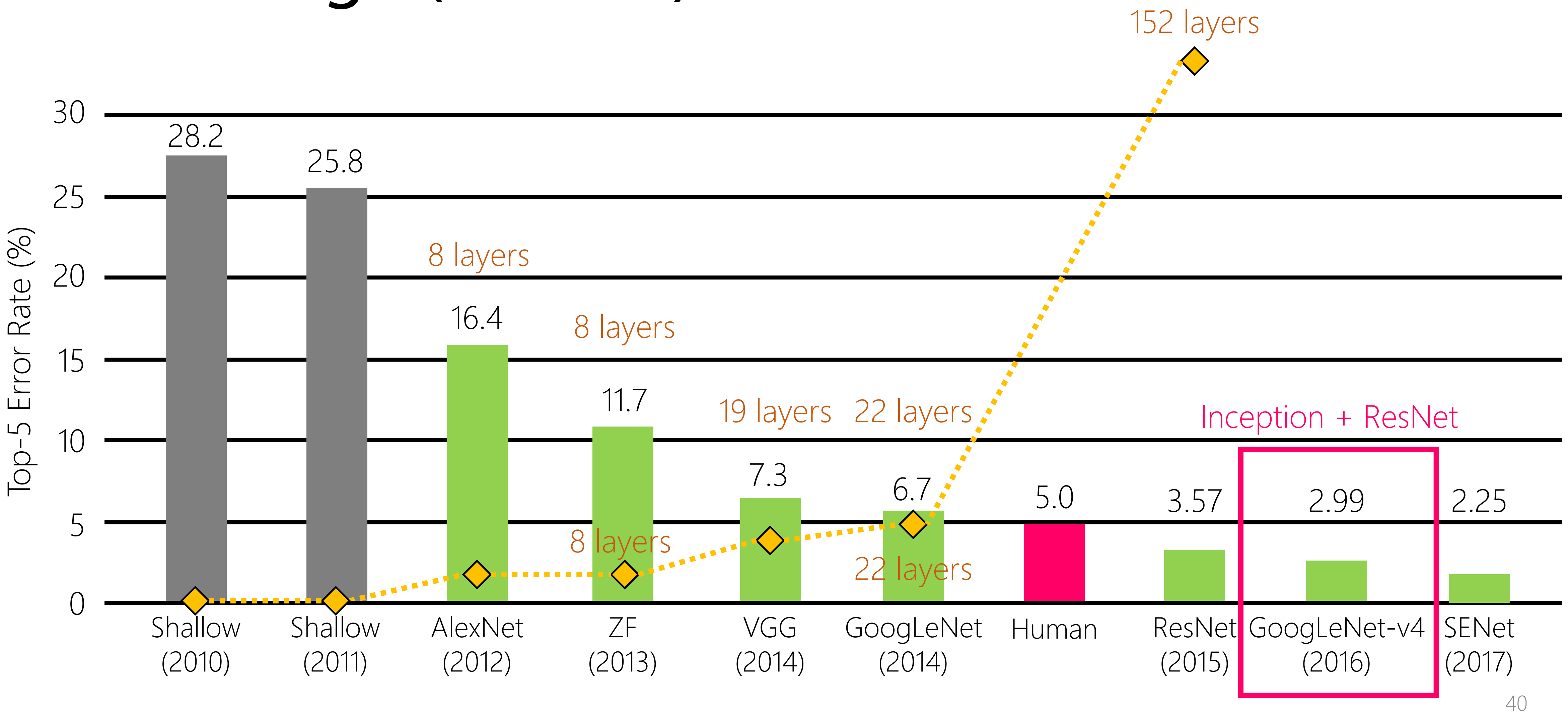


ResNet

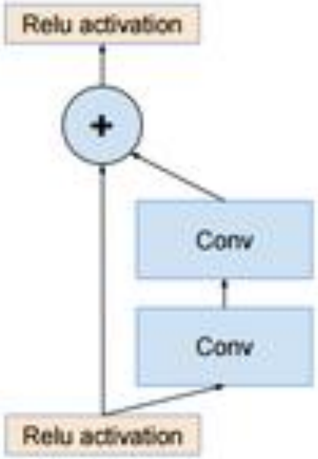
- Swept 1st place in all ILSVRC and COCO 2015 Competitions

- Better than human performance now!

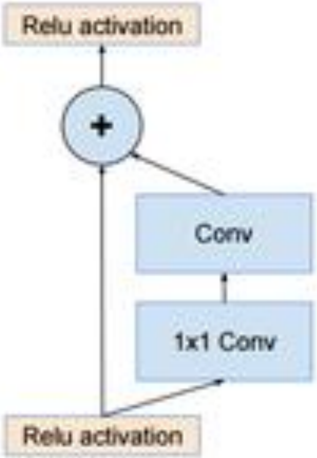
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



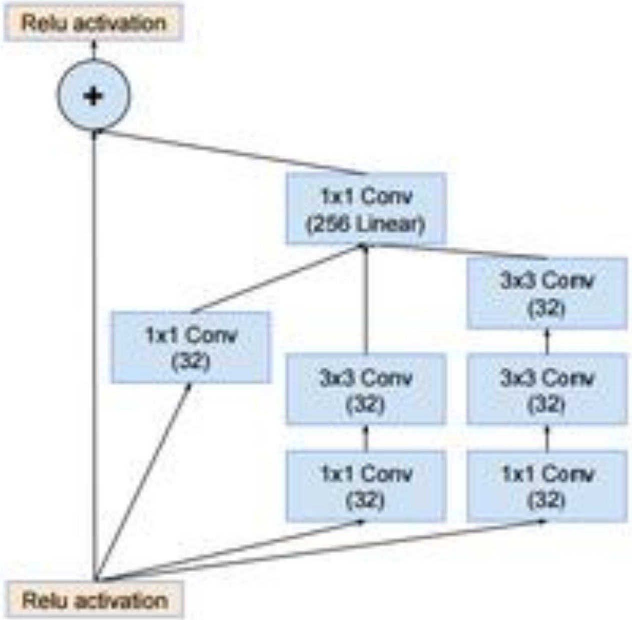
Inception v4 (a.k.a. Inception-ResNet)



(a) Residual module introduced by He et al. (2015)

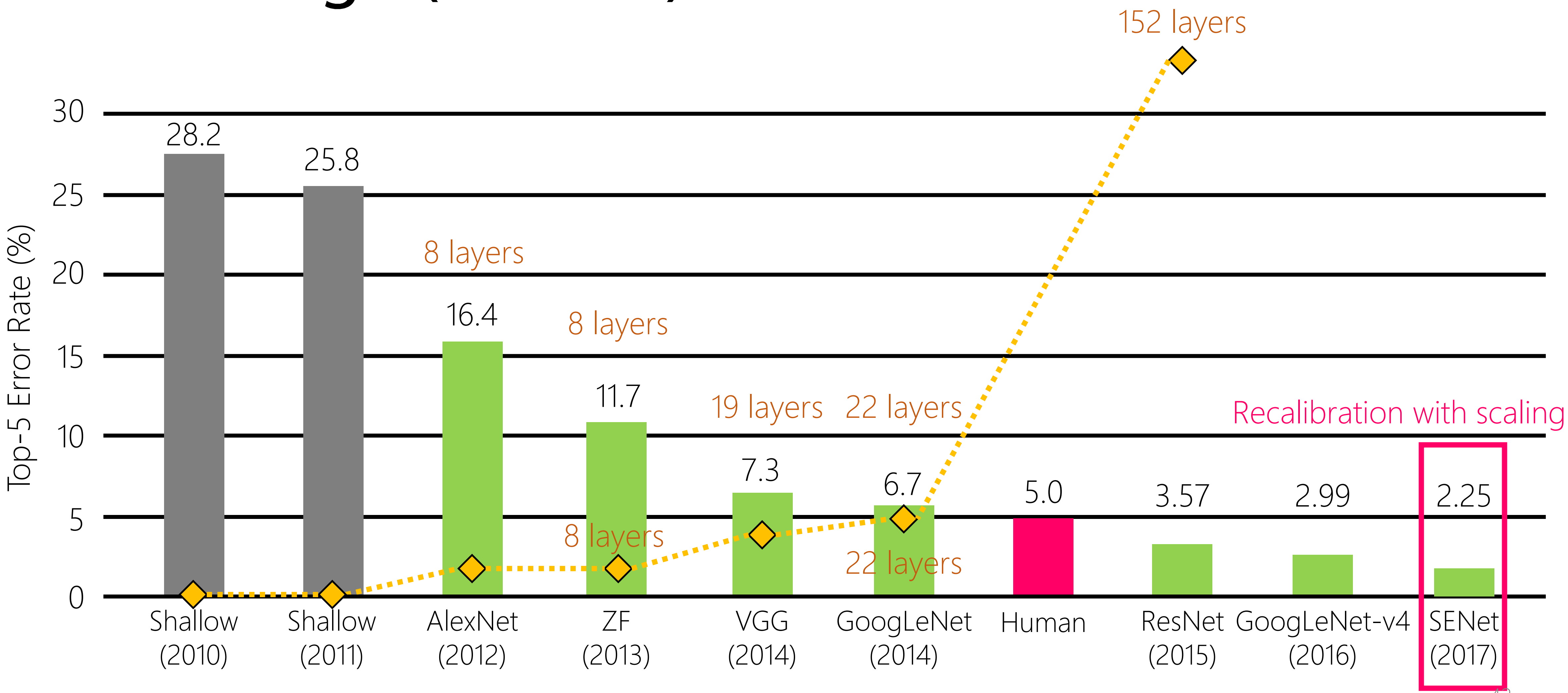


(b) Optimized version of residual modules to shield computation, like in Lin et al. (2013).



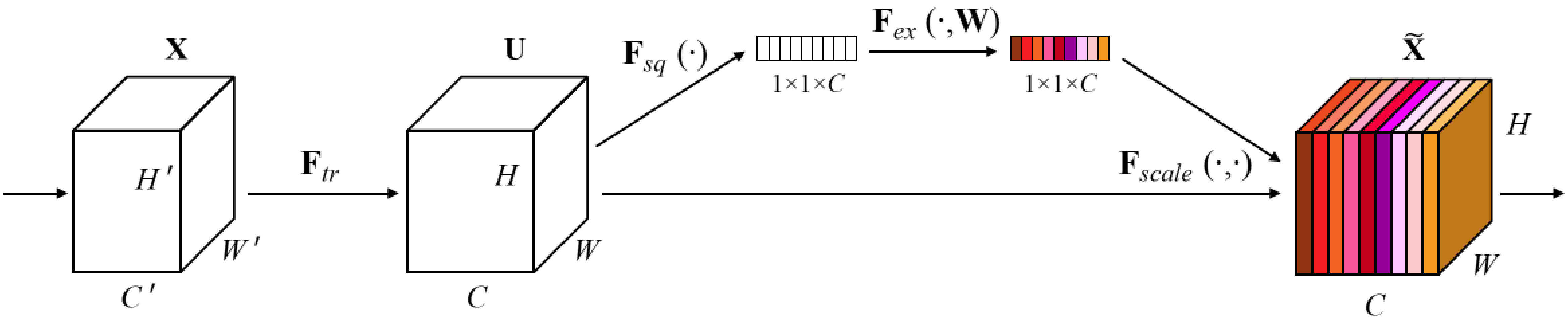
(c) Inception-ResNet example module for the 35 × 35 grid.

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



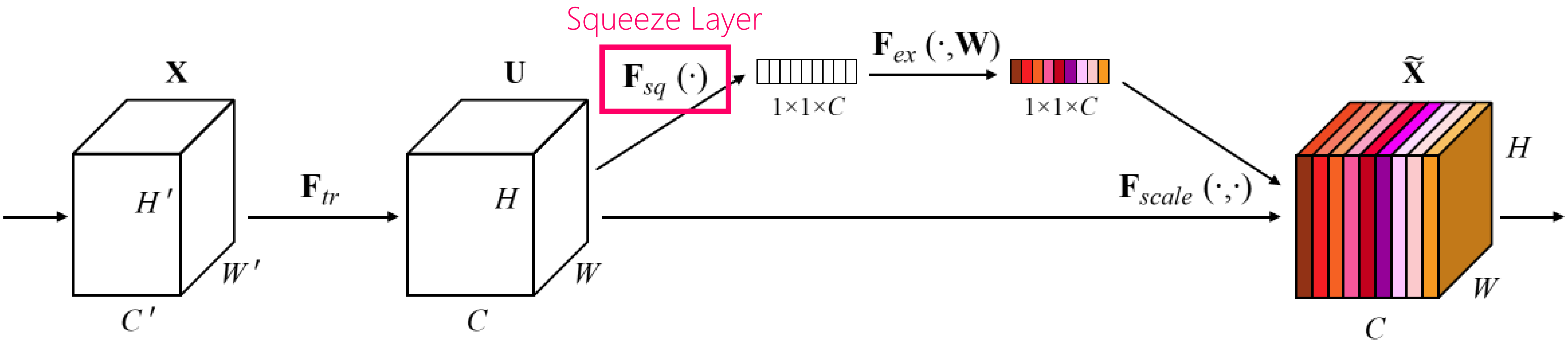
Squeeze and Excitation (SE) Network

- Recalibration of the activation maps with the global average



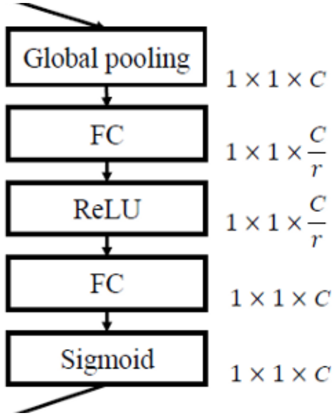
Squeeze and Excitation (SE) Network

- Recalibration of the activation maps with the global average

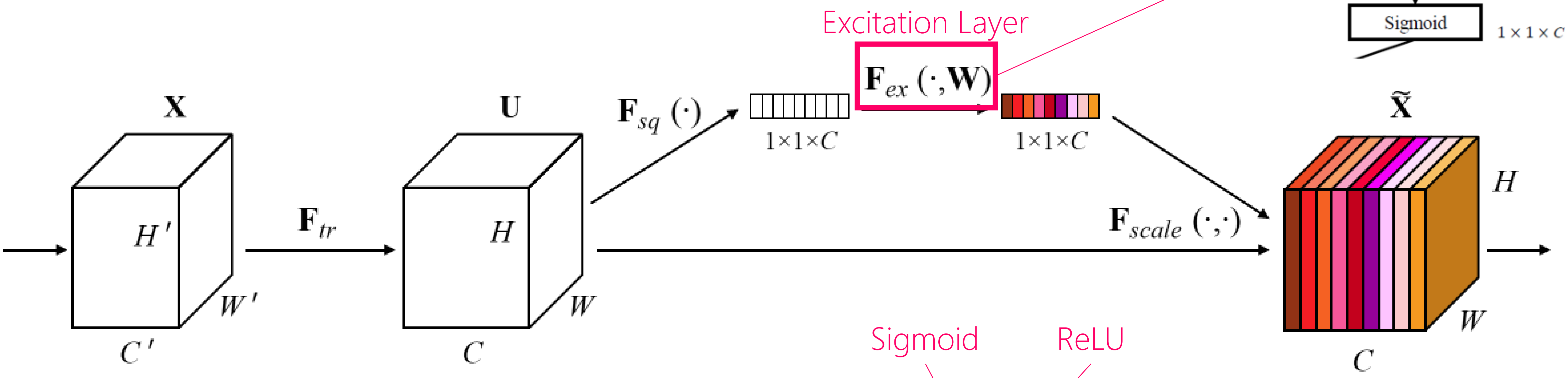


$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j).$$

Squeeze and Excitation (SE) Network



- Recalibration of the activation maps with the global average



Sigmoid ReLU

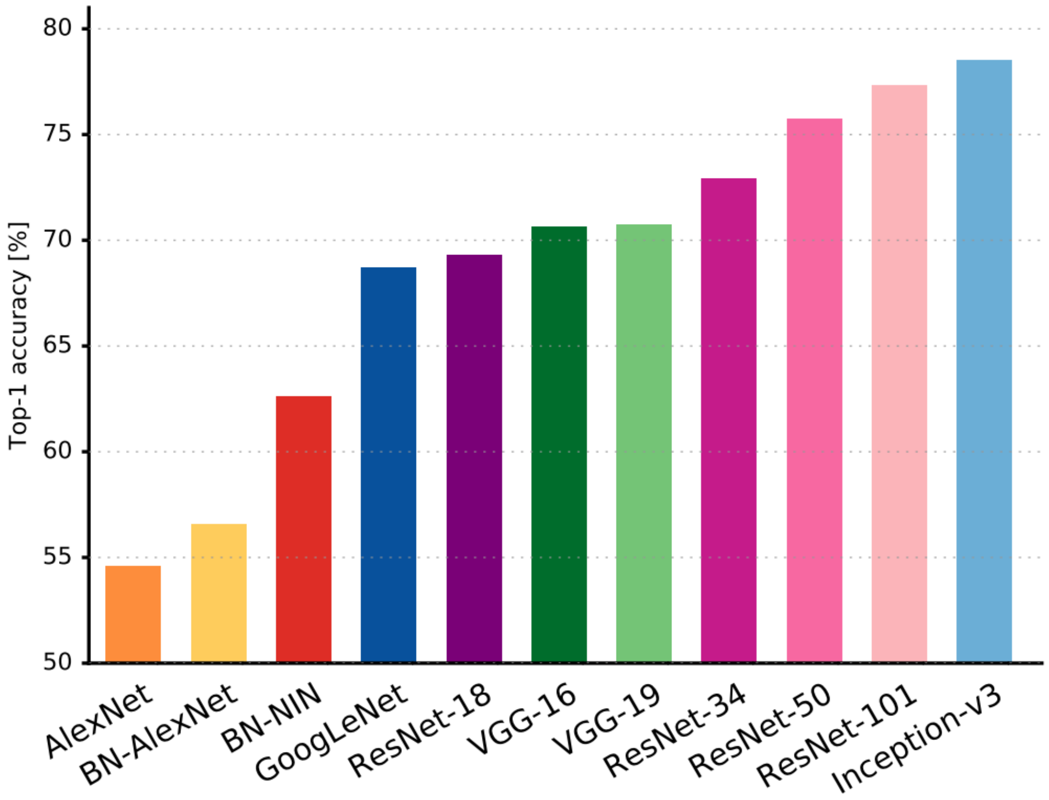
$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})),$$

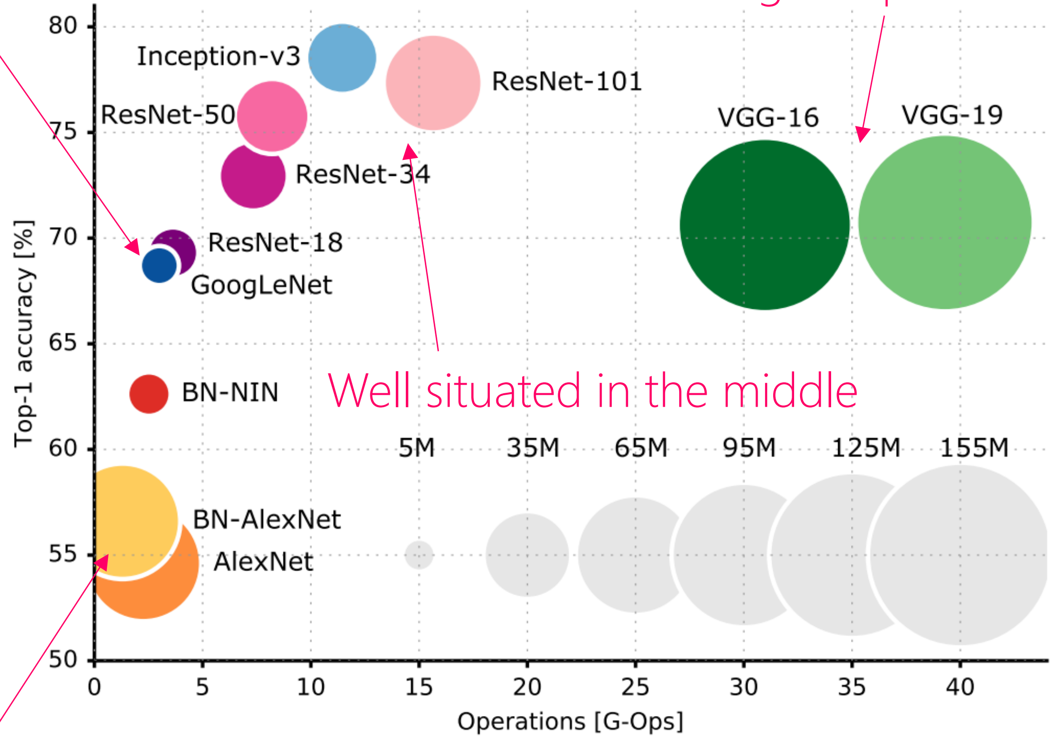
r: Dimensionality reduction ratio (4, 8, 16, 32)
Captures channel-wise dependencies
Learns a nonlinear and non-mutually-exclusive relationship between channels.

Alex, VGG, ResNet, Inception at a glance

Most efficient

Largest memory,
Largest operations

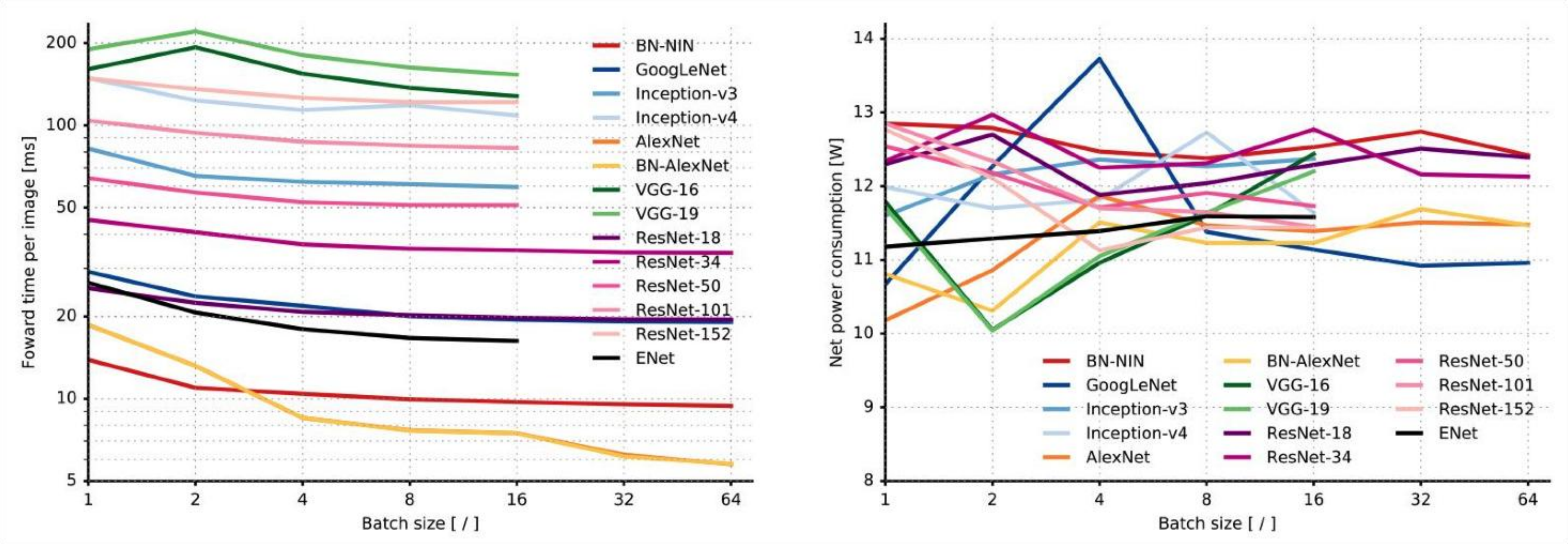




Smaller computation, but memory heavy
Lower accuracy

Canziani et al. (2016). An Analysis of Deep Neural Network Models for Practical Applications

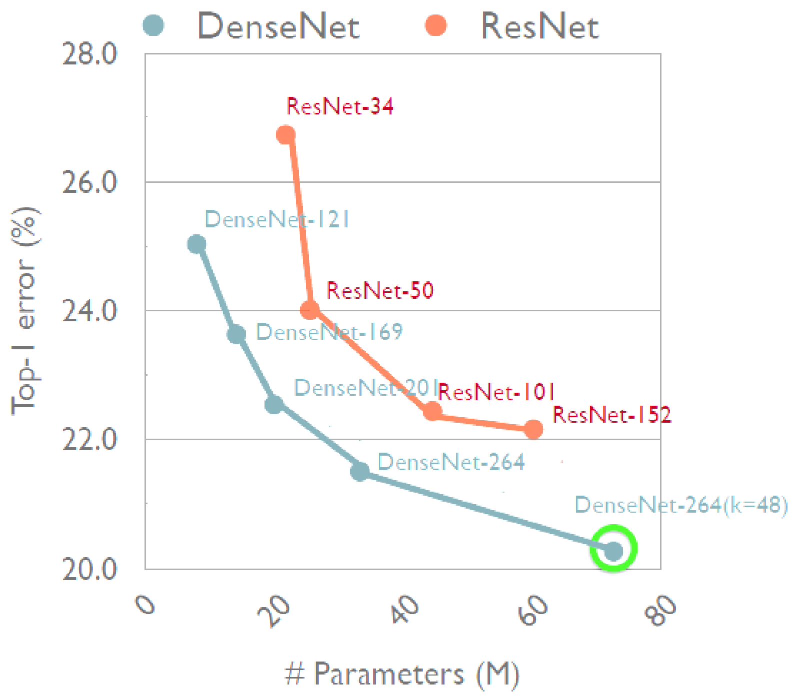
Alex, VGG, ResNet, Inception at a glance

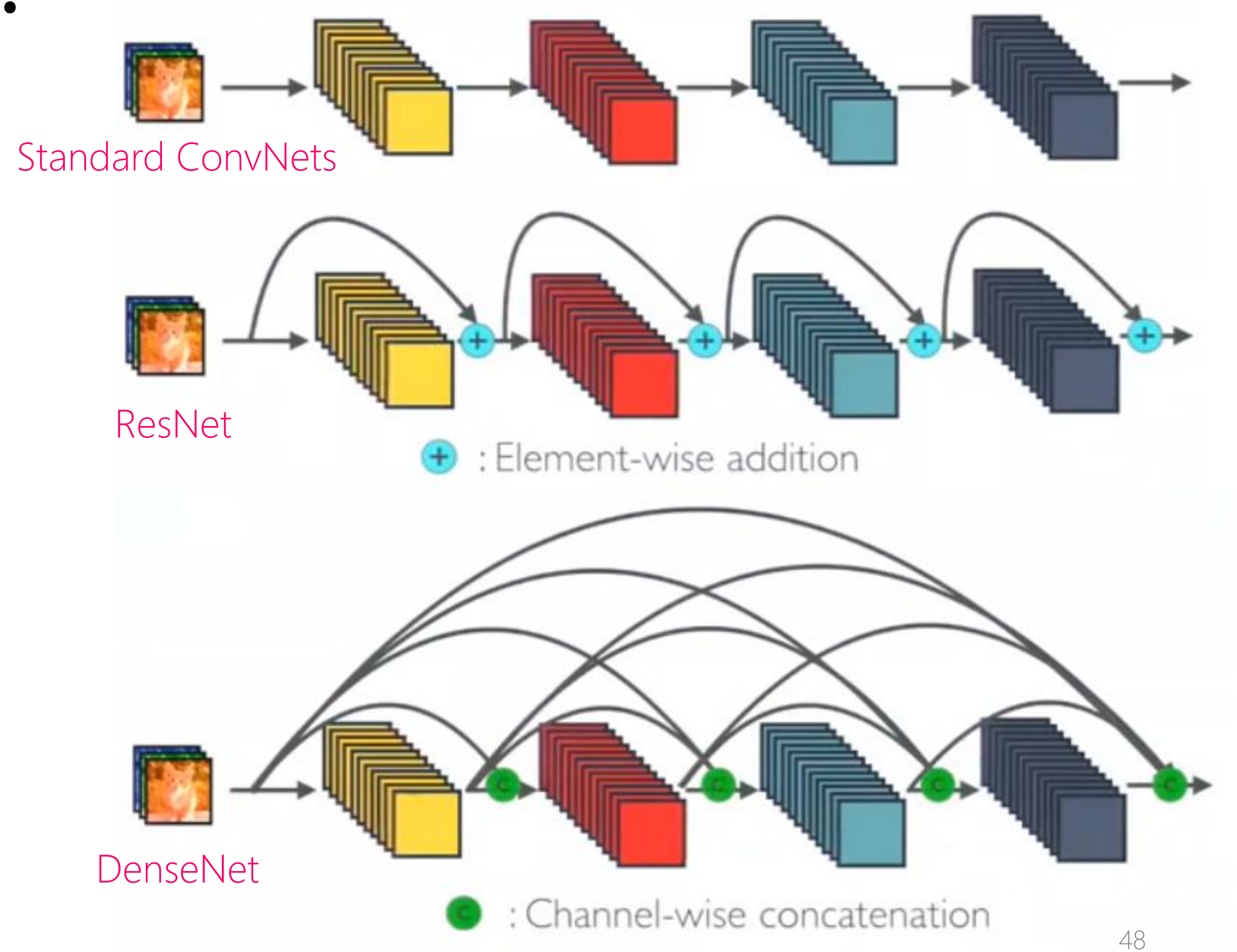


Canziani et al. (2016). An Analysis of Deep Neural Network Models for Practical Applications

Other Variants…

- DenseNet (CVPR 2017)





Other Variants…

- ResNeXt (CVPR 2017)

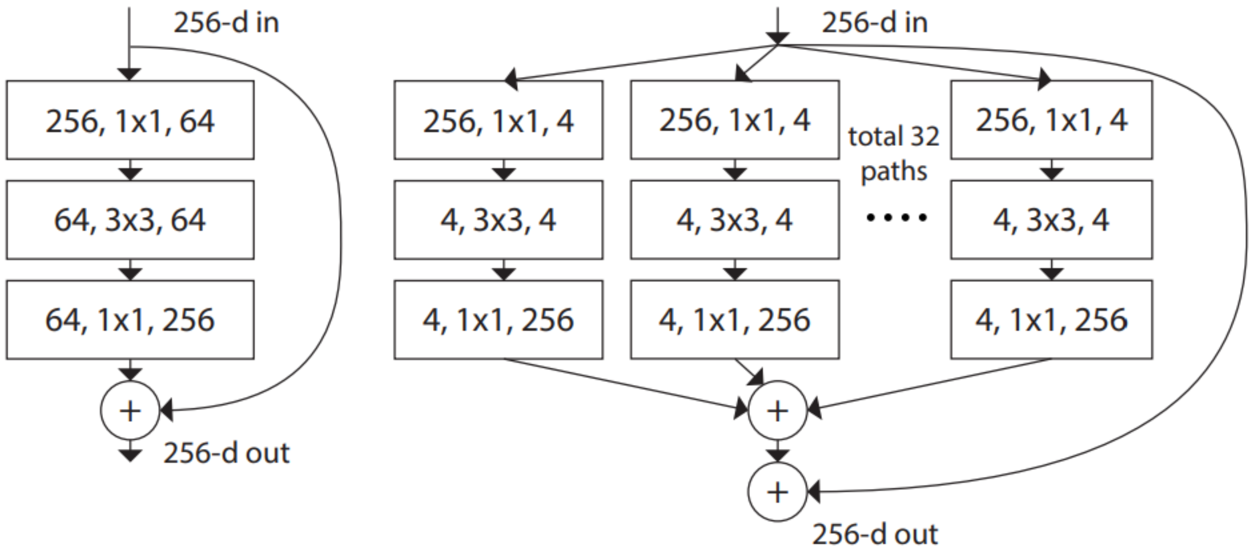


Figure 1. **Left**: A block of ResNet [14]. **Right**: A block of ResNeXt with cardinality = 32, with roughly the same complexity. A layer is shown as (# in channels, filter size, # out channels).

Other Variants…

- MobileNet-V1 (2017)

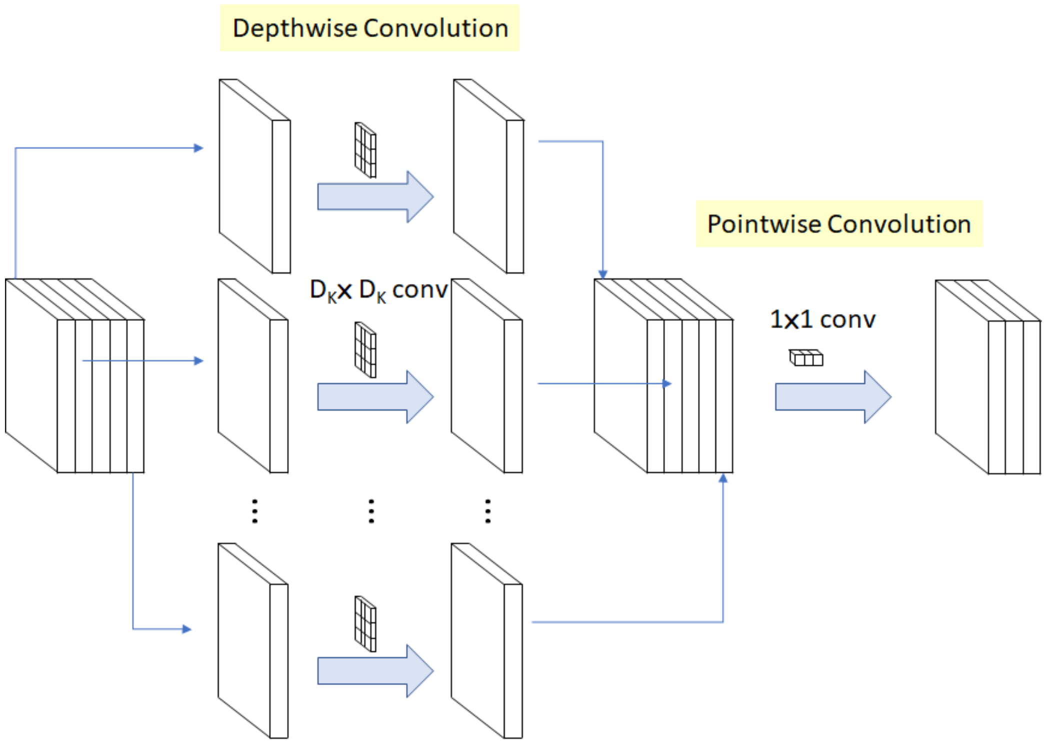
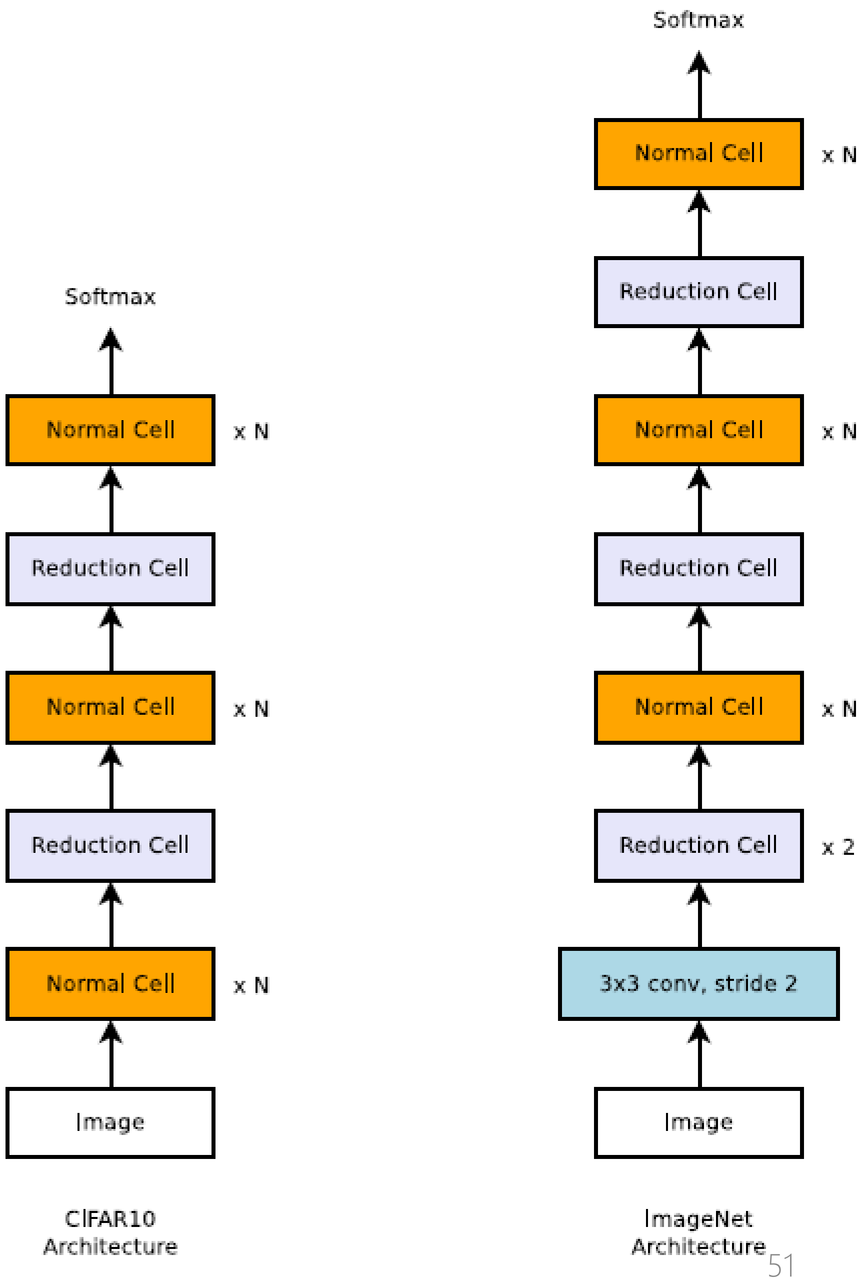


Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5× Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
5× Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Other Variants…

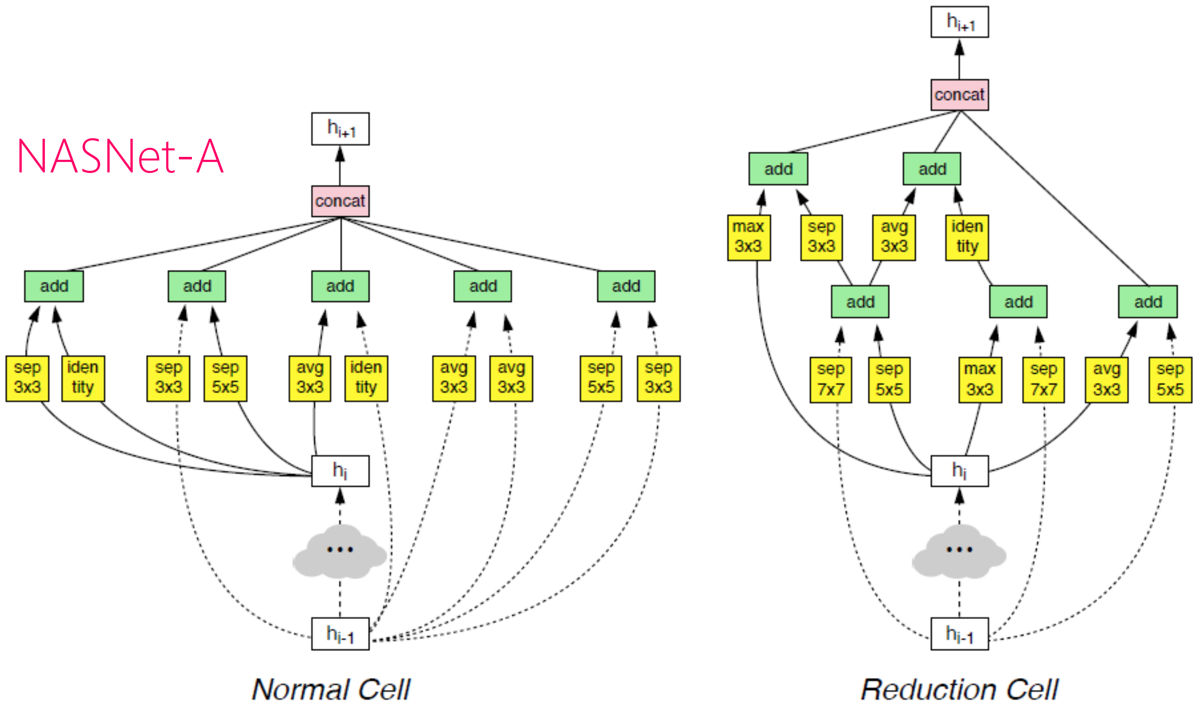
- NASNet (CVPR 2018)
  - Neural Architecture Search
    - Overall architecture is predefined. Specific blocks and cells are not.
    - Normal Cell: Conv layers that return a feature map of the same size.
    - Reduction Cell: Conv layers that return a feature map of the size reduced by a factor of two
    - Structures of the Normal and Reduction Cells are searched by a recurrent neural network (Lecture 9)



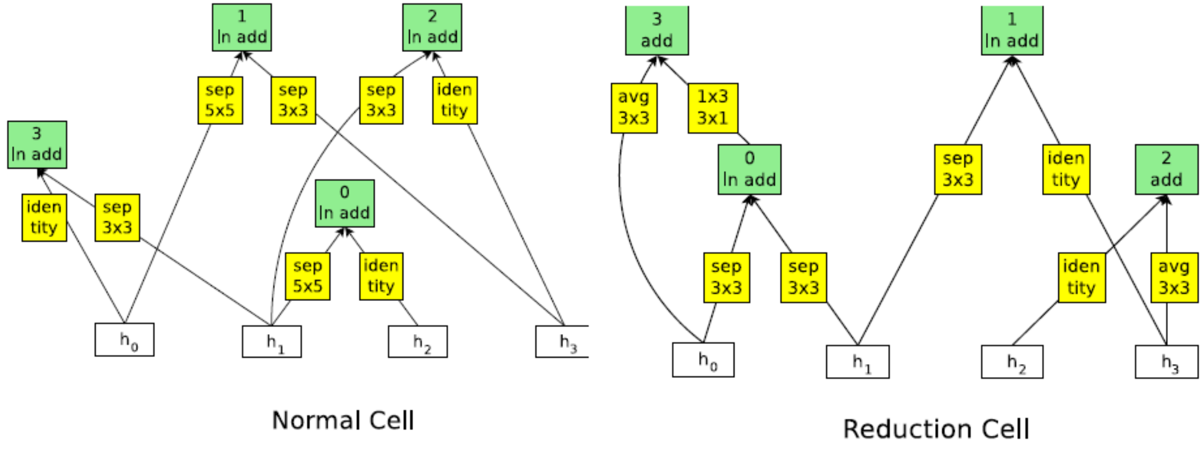
Other Variants…

- NASNet (CVPR 2018)

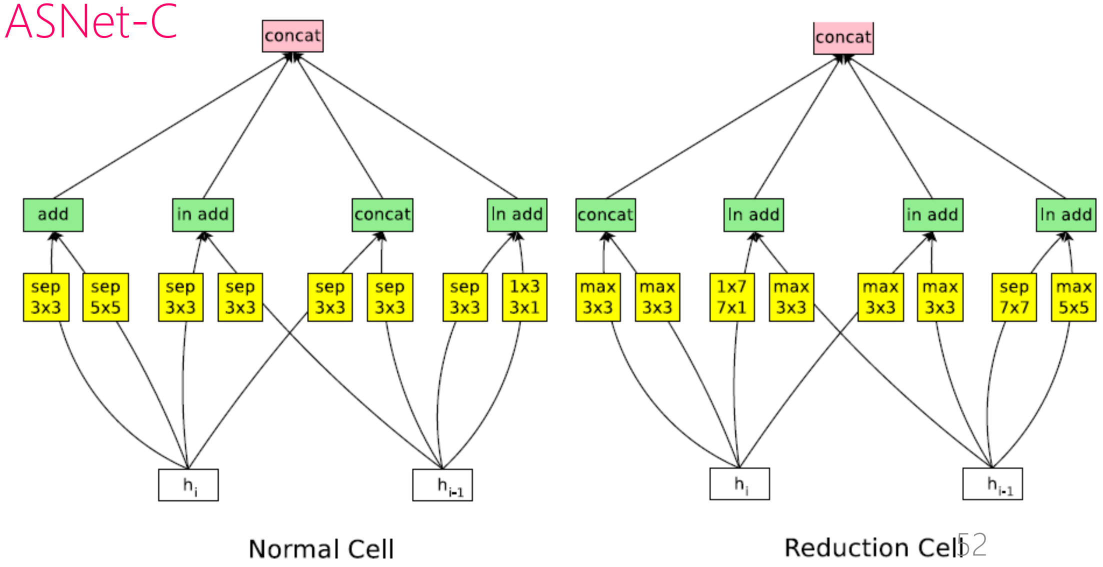
NASNet-A



NASNet-B

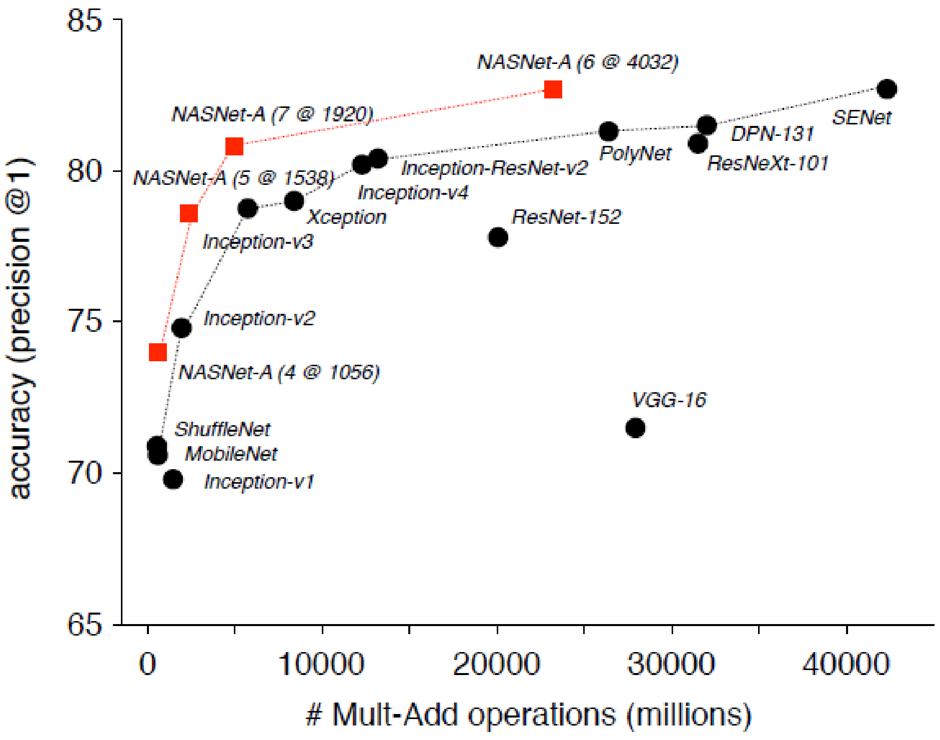


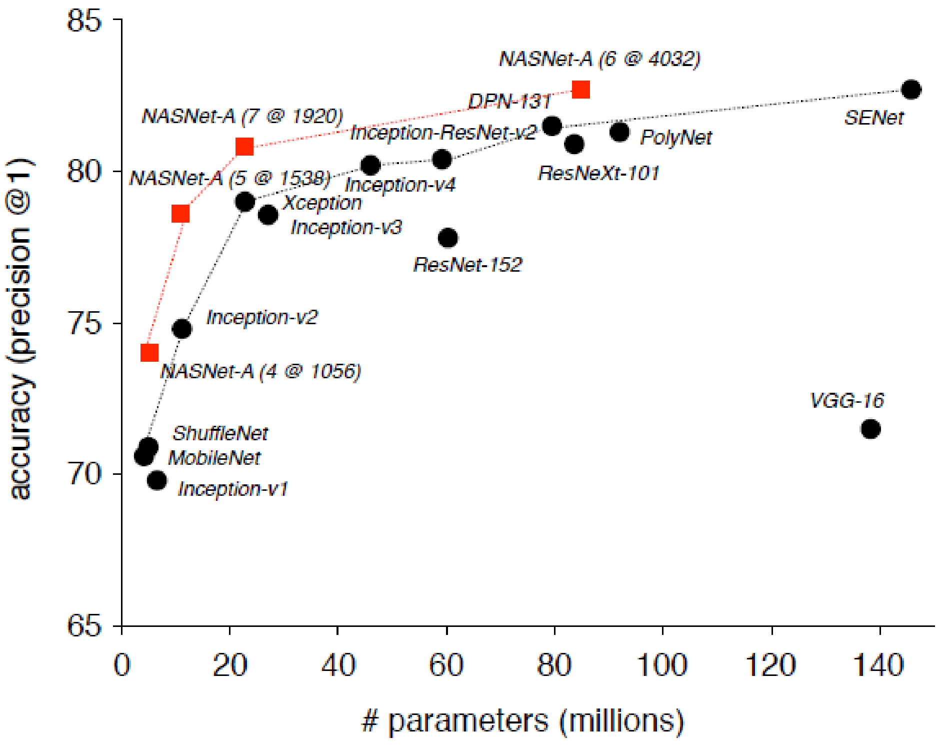
NASNet-C



Other Variants…

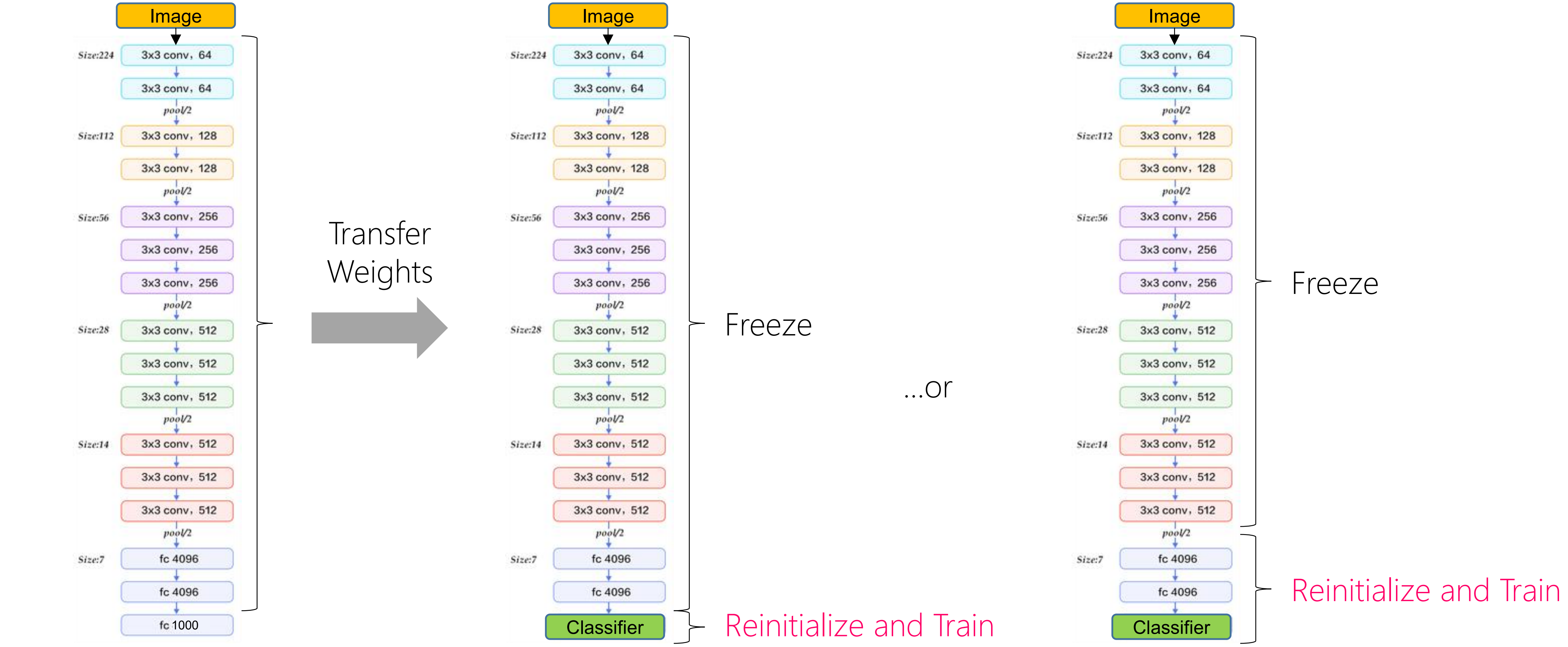
- NASNet (CVPR 2018)





Transfer Learning

Transfer Learning



VGG trained on imagenet

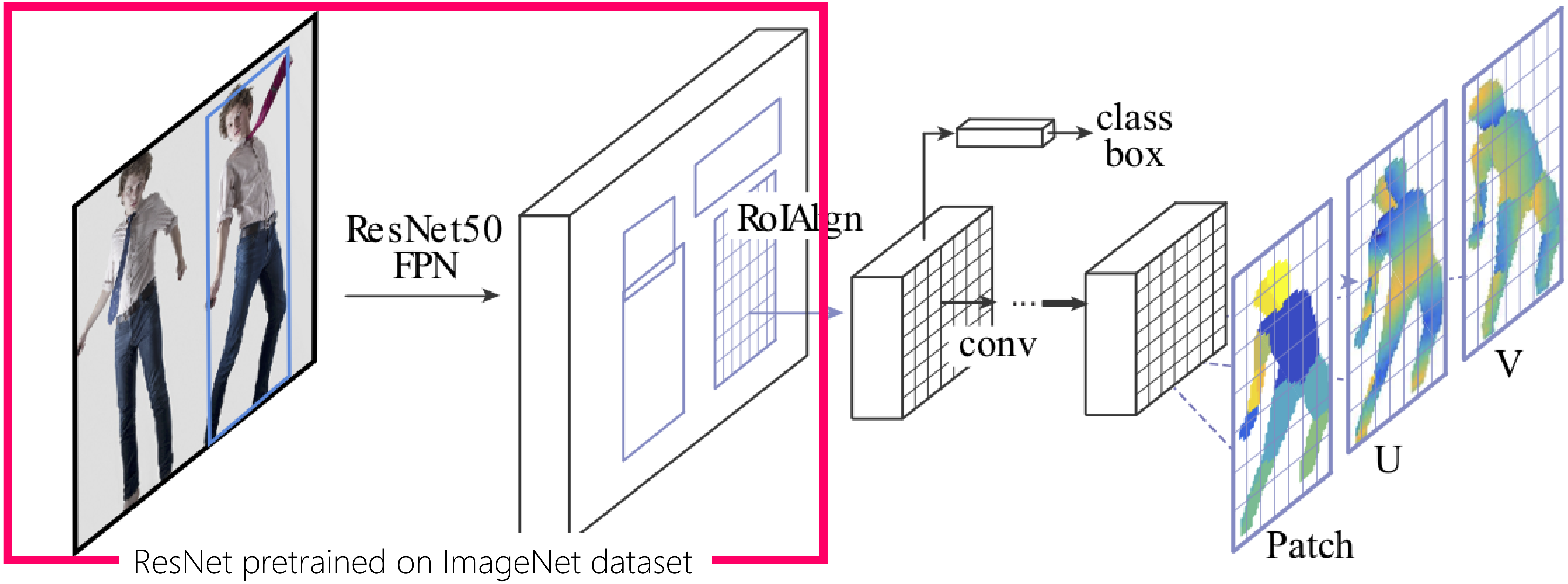
Your small dataset

Transfer Learning

- You may not need to train a large network on your own ☺
  - Save time
  - Save money (to buy an expensive computer)
- You may not need a large dataset ☺

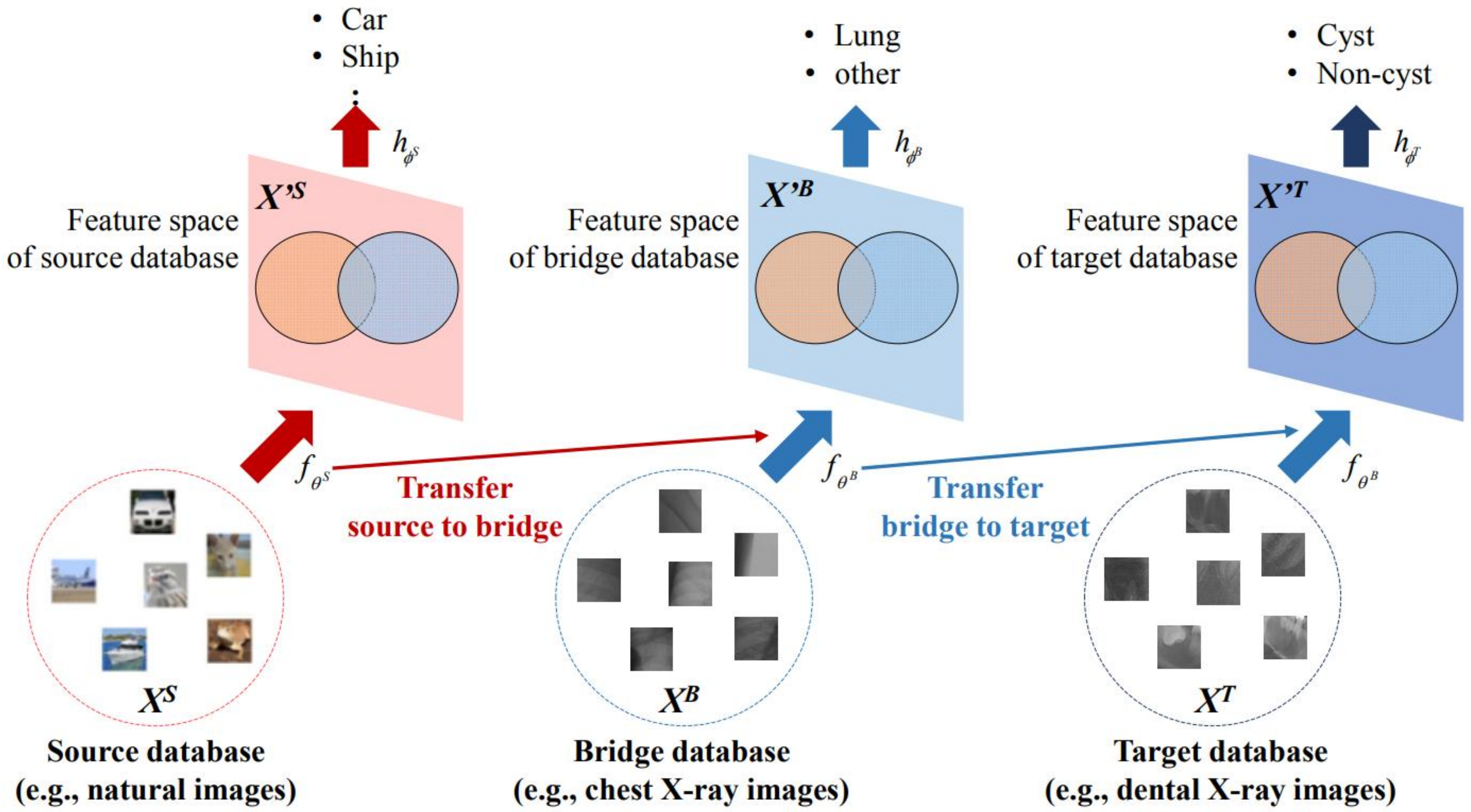
Transfer Learning is VERY Common

- DensePose (Facebook Research, 2018)



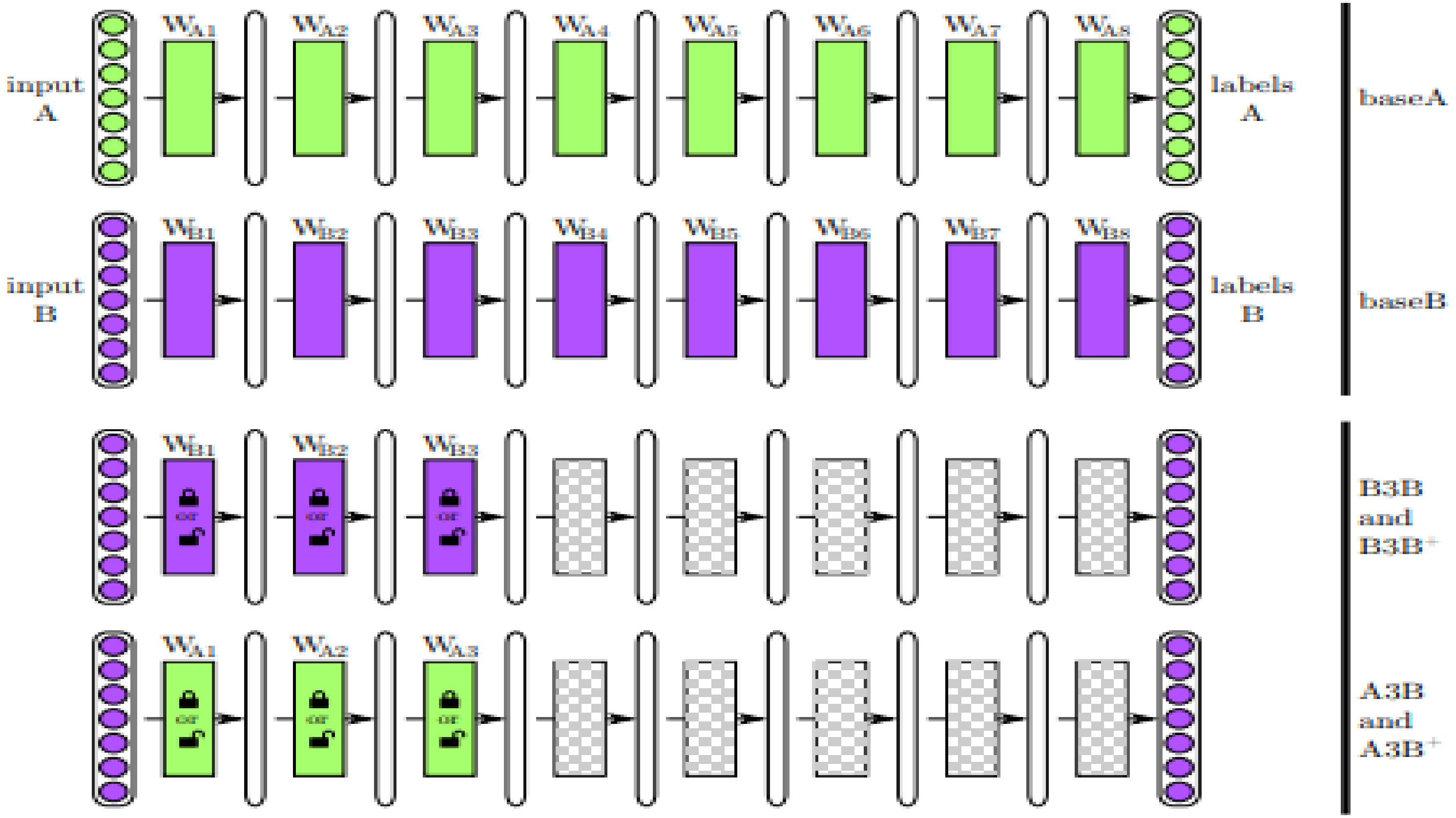
Transfer Learning is VERY Common

- Modality Bridging (Kim, Choi, & Ro 2017)



Transferability of the Layers

- Yosinski et al. (2014)



Transferability of the Layers

- Yosinski et al. (2014)
  - First few layers are more general/universal, later layers are more problem-specific.
  - Transfer + fine-tuning improves generalization