CNN Architectures

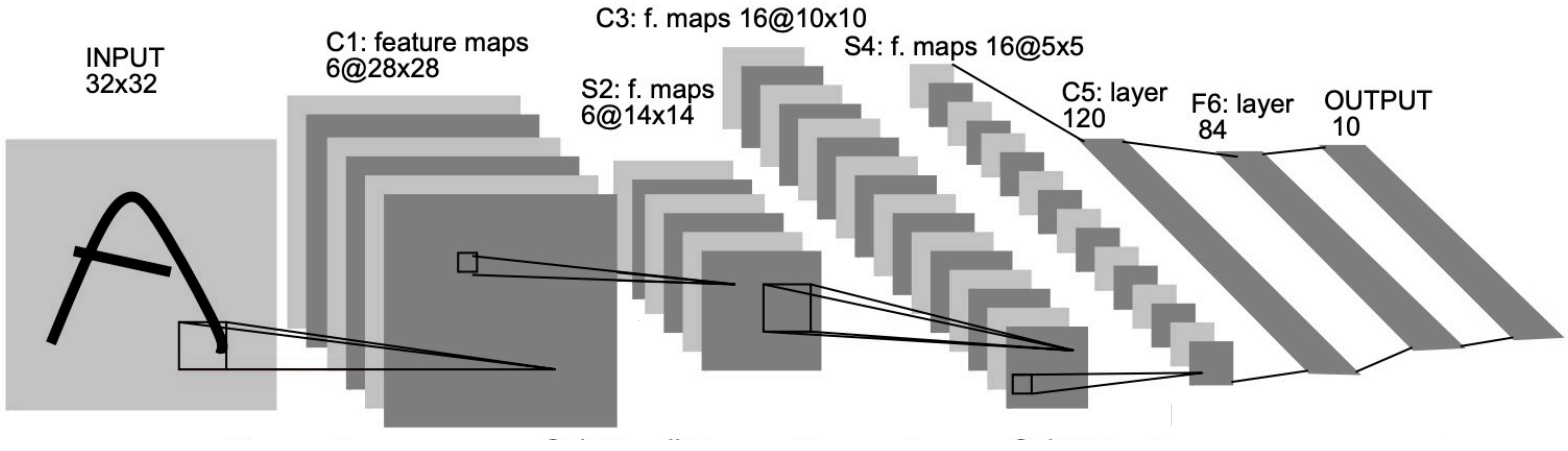
COMP90051 Statistical Machine Learning

Qiuhong Ke

Outline

- LeNet5
- AlexNet
- VGG
- GoogleNet
- ResNet

Architecture



LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

1st layer

32x32x1

Convolutional layer
No.filters: 6
Filter size: 5x5
Padding: 0
Stride:1

? 28x28x6

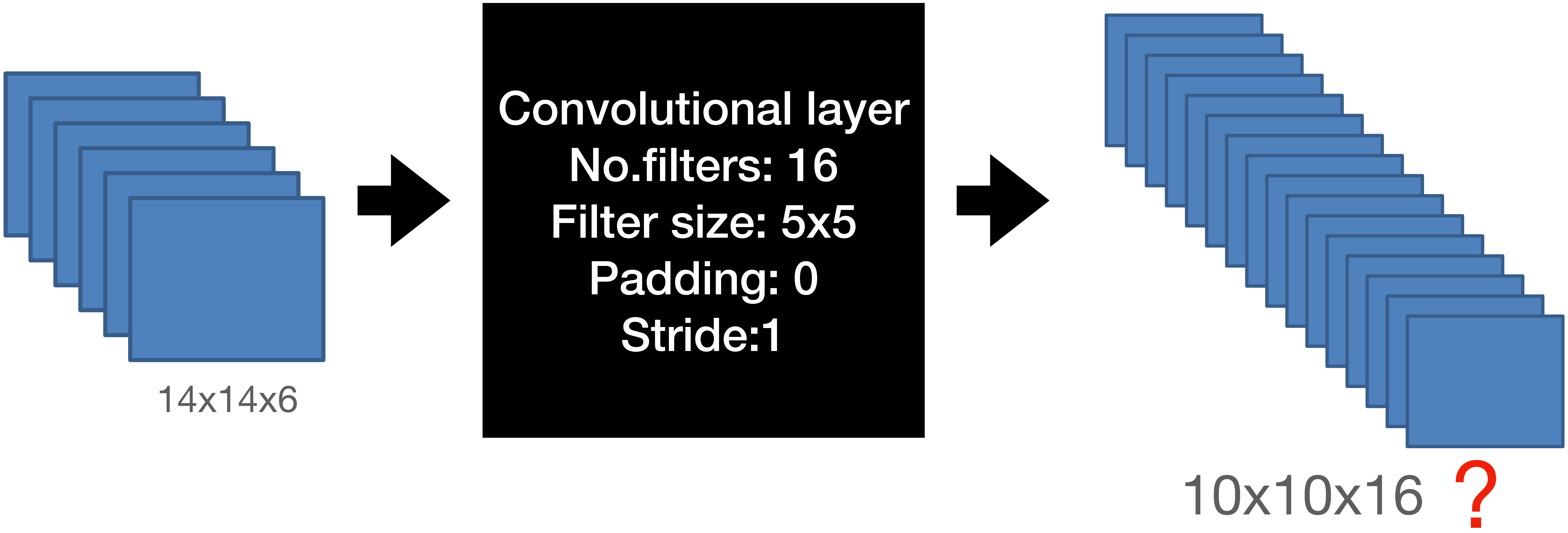
No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

2nd layer

- Take the sum of all units in the 2x2 window
- output of each patch = sum*coefficient (trainable) + bias (trainable)

No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

3rd layer



No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

Convolution on Multiple-channel input

Kernel: same channel (depth)

R

$* K$(Depth 1)

G

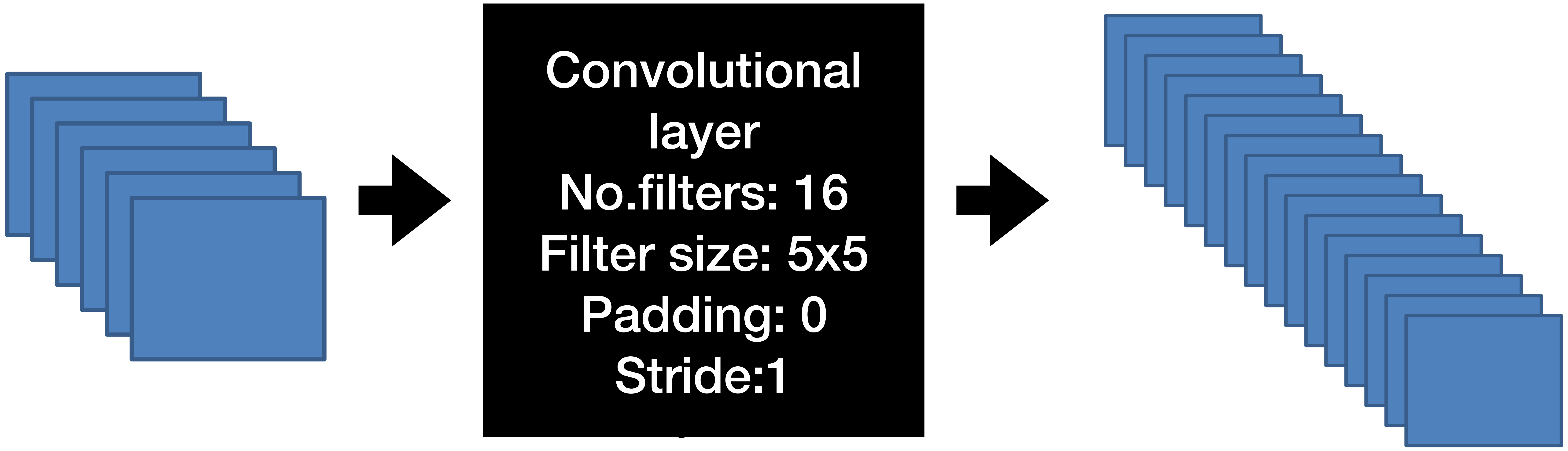
$* K$(Depth 2) → Element-wise sum → One channel

B

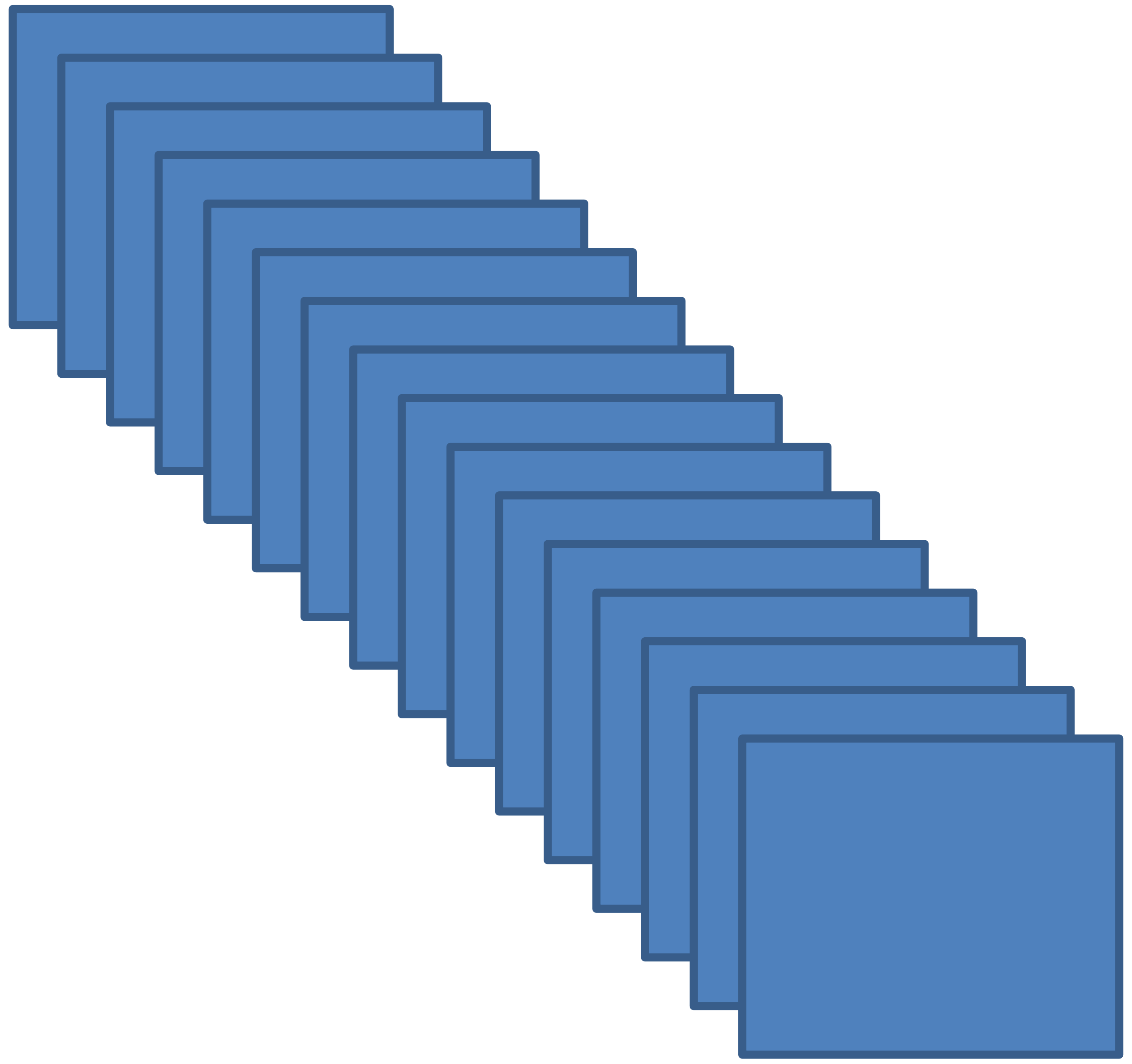
$* K$(Depth 3)

3rd layer: Non-complete connection scheme

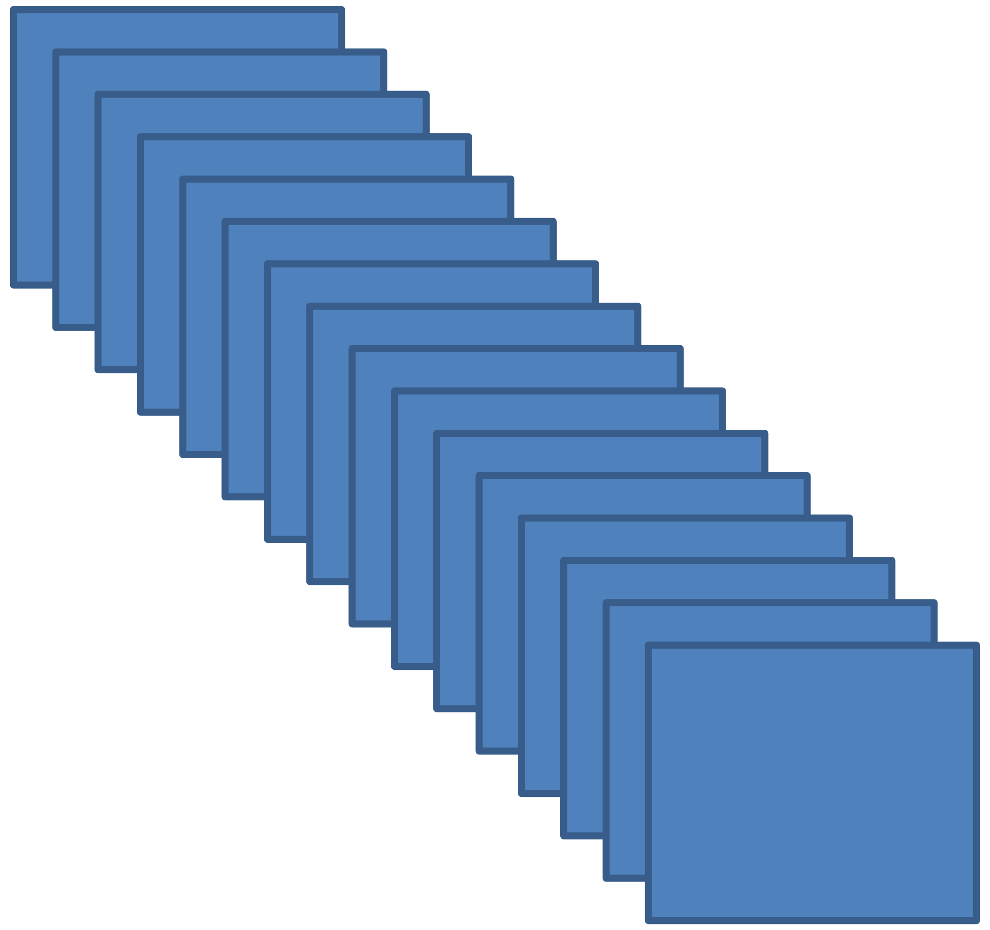
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X



4th layer

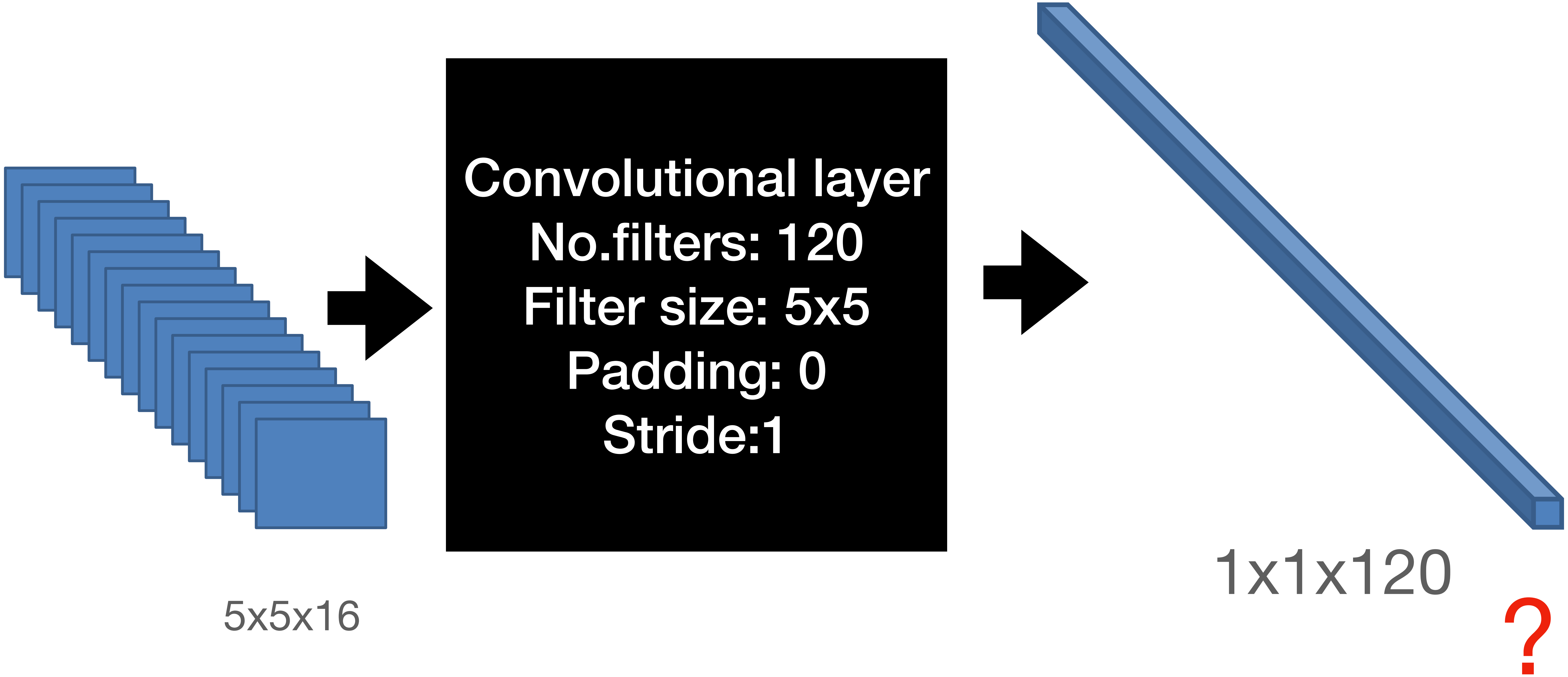


Subsampling layer
Filter size: 2x2
Padding: 0
Stride:2



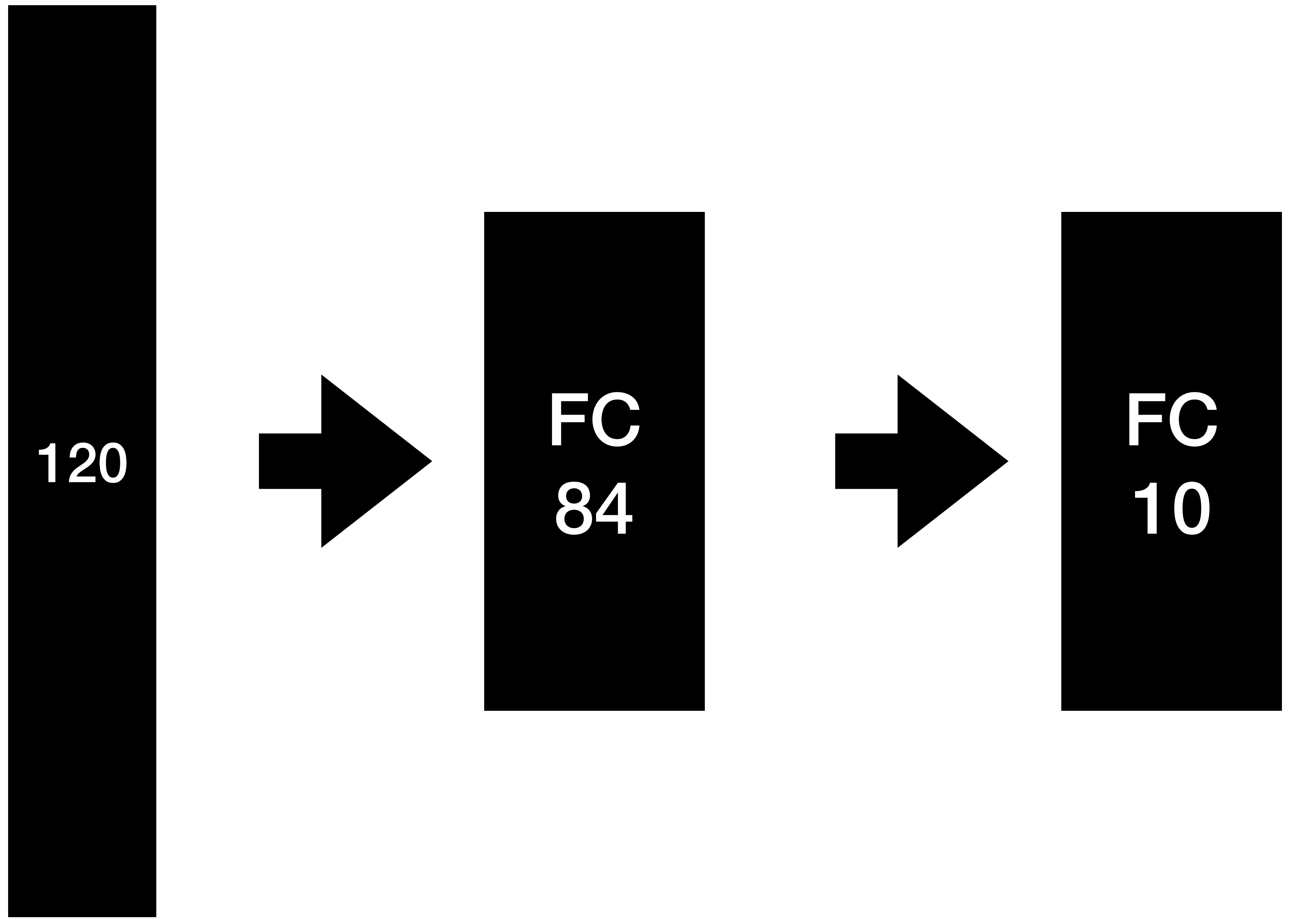
No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

5th layer



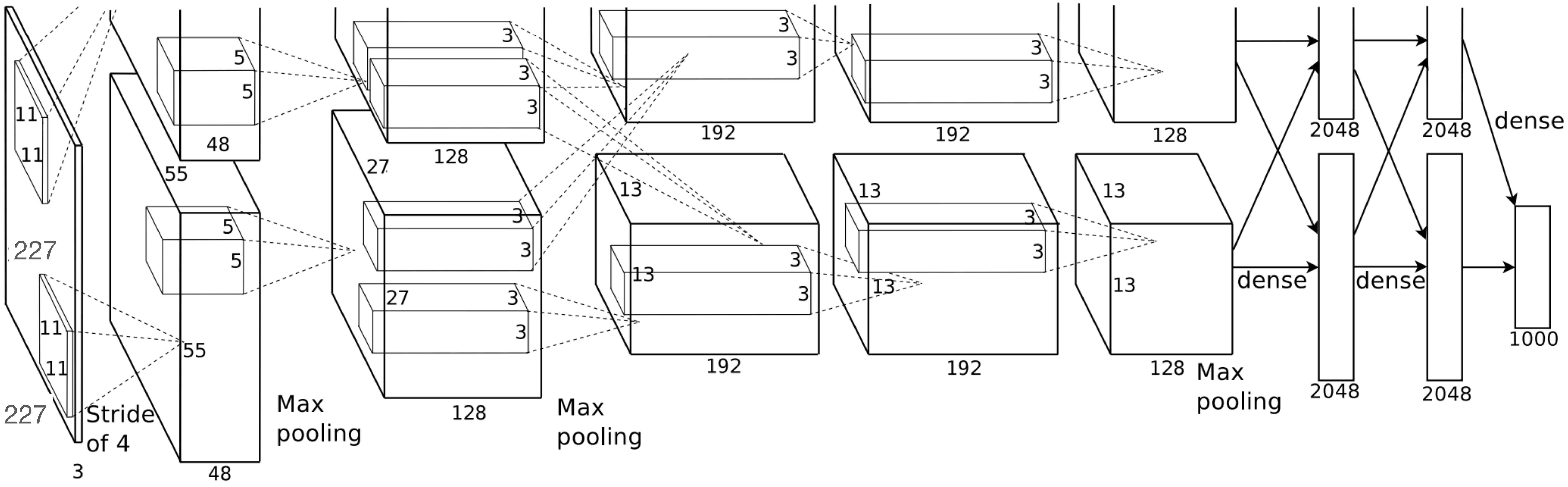
No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

Following: Fully connected layers



Architecture

This training on 2 GPU where each stream represent a training stream on one GPU



Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

1st layer

Input channels No. filters

How many parameters?

Filter size

11x11x3x96=34,848

227x227x3

Convolutional layer
No.filters: 96
Filter size: 11x11
Padding: 0
Stride: 4

No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

Convolution on Multiple-channel input

R

G

B

Kernel: same channel (depth)

$* K$(Depth 1)

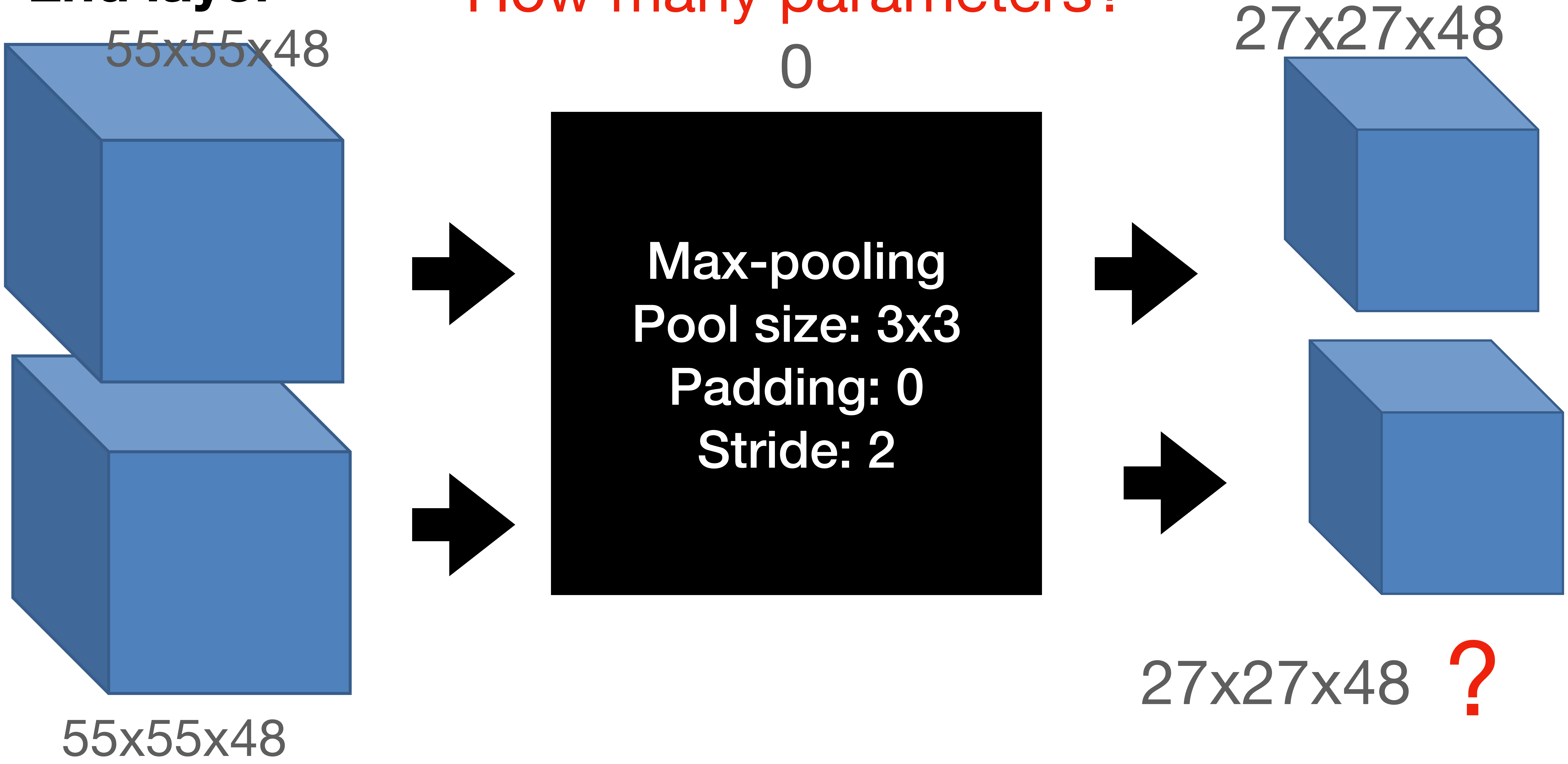
$* K$(Depth 2) → Element-wise sum → One channel

$* K$(Depth 3)

2nd layer

How many parameters?

0

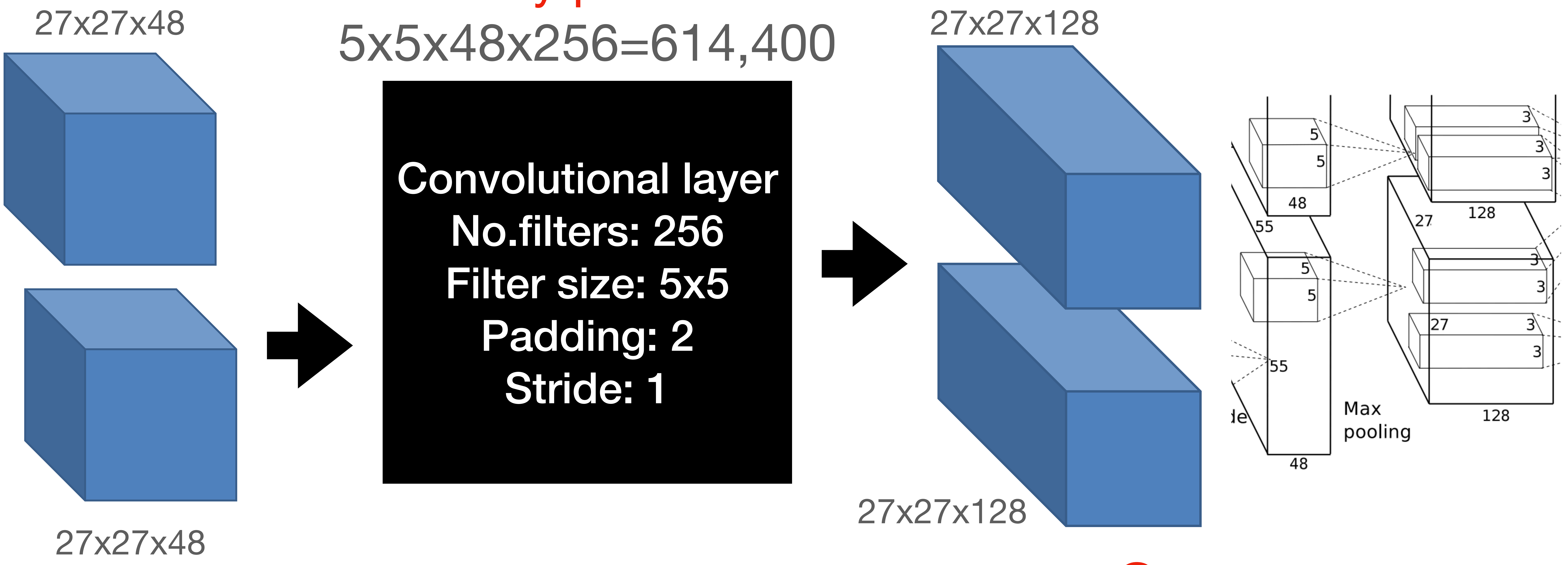


No padding: output_size=ceiling((input_size-kernel_size+1)/stride)

3rd layer

How many parameters?

5x5x48x256=614,400



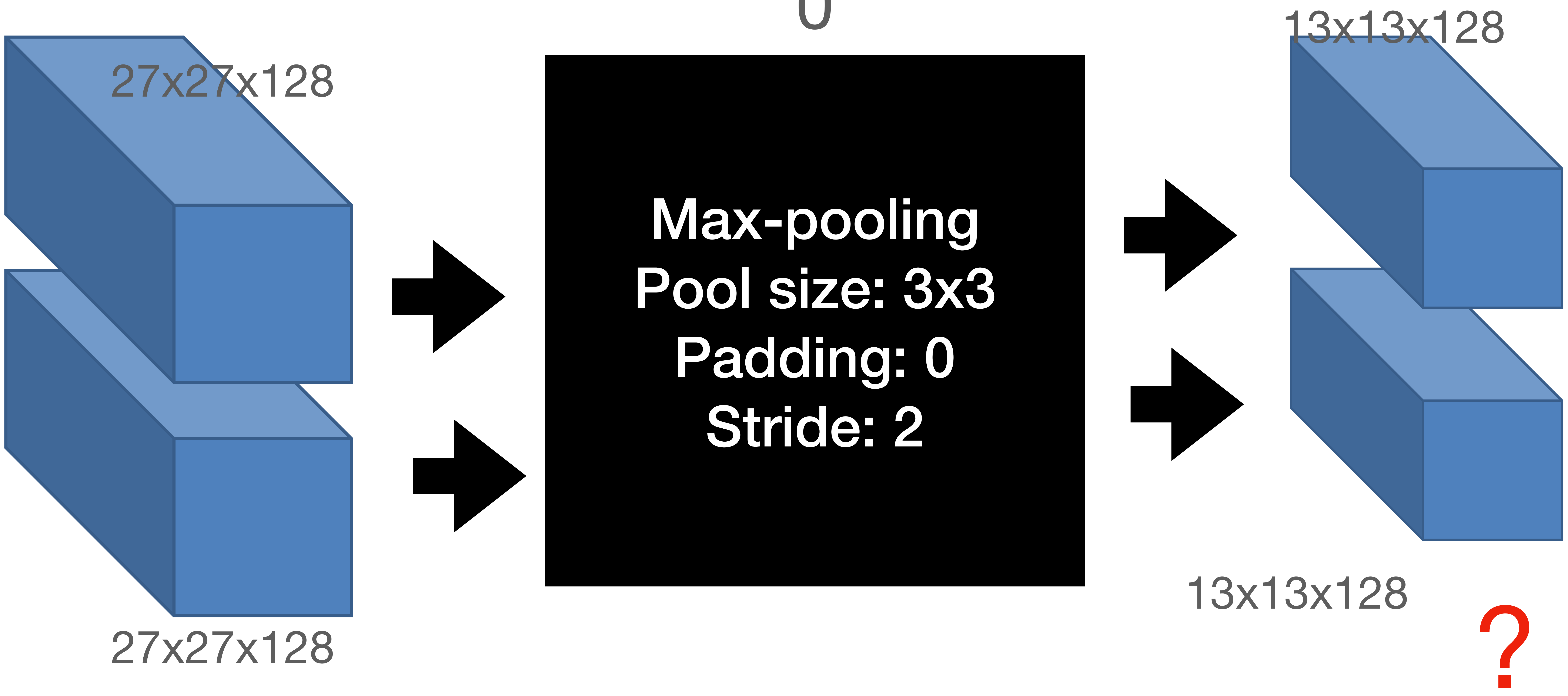
padding: output_size=ceiling((input_size)/stride)

?

4th layer

How many parameters?

0

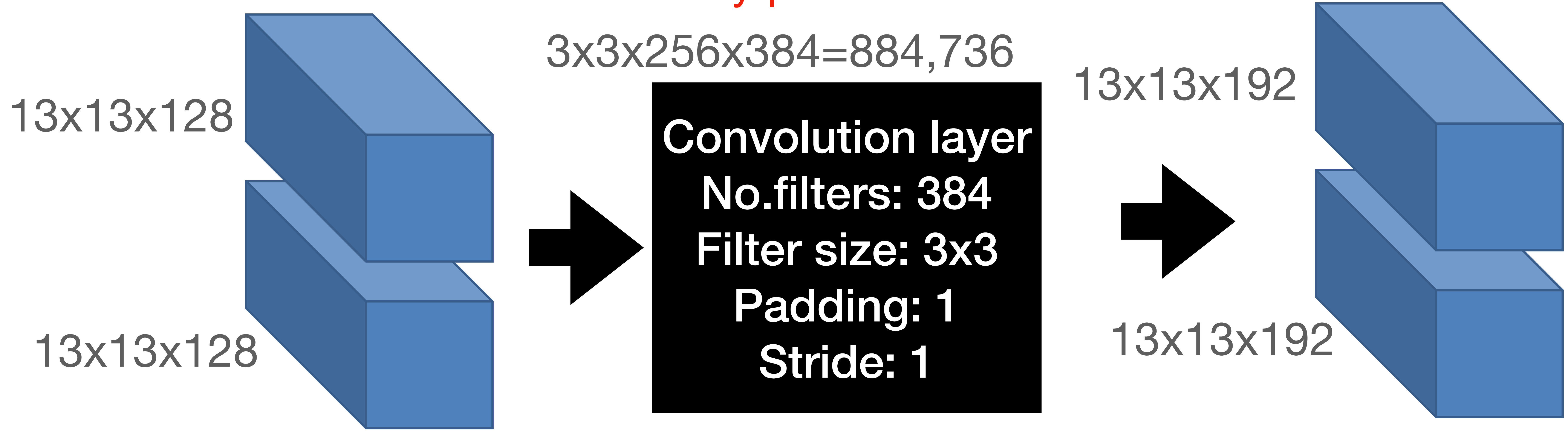


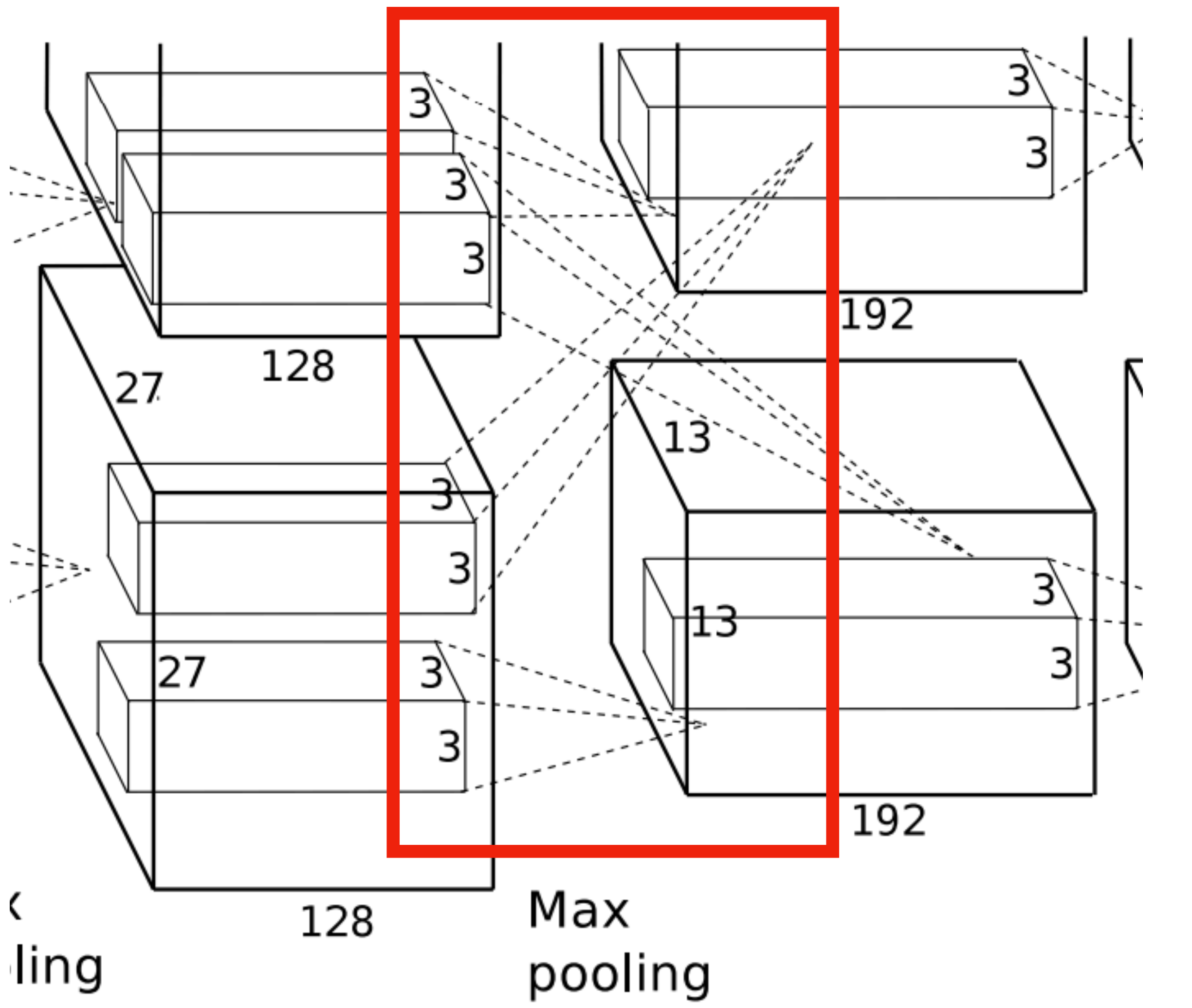
output_size=ceiling((input_size-kernel_size+1)/stride)

5th layer

How many parameters?

3x3x256x384=884,736





Following convolutional layers

How many parameters?

3x3x192x384=663,552

13x13
X192

13x13
X192

Convolution layer
No.filters: 384
Filter size: 3x3
Padding: 1
Stride: 1

13x13
X192

13x13
X192

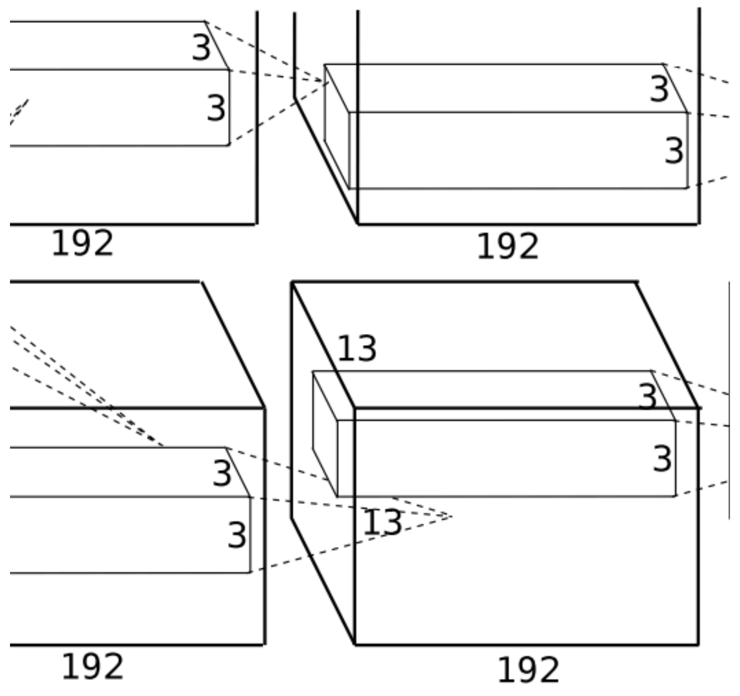
How many parameters?

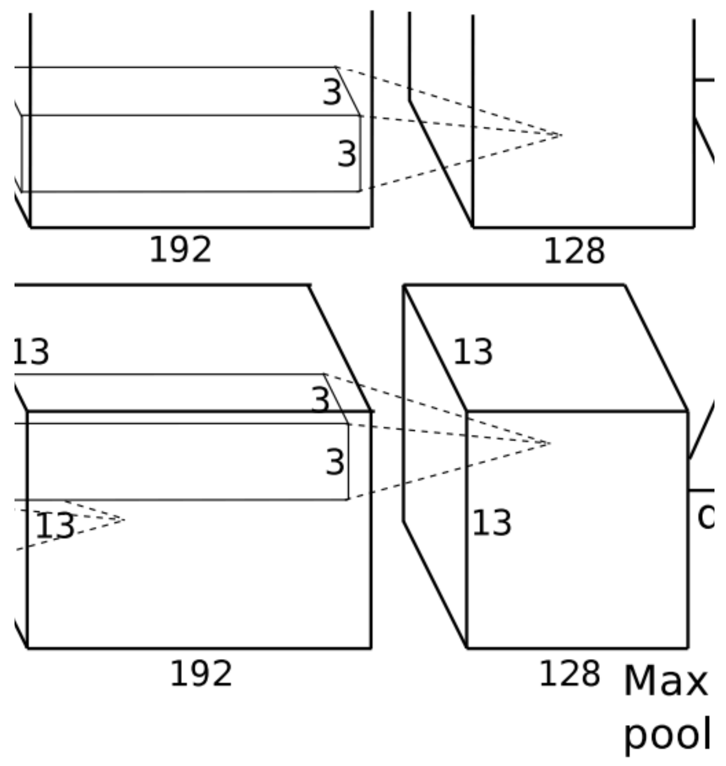
3x3x192x256=442,368

Convolution layer
No.filters: 256
Filter size: 3x3
Padding: 1
Stride: 1

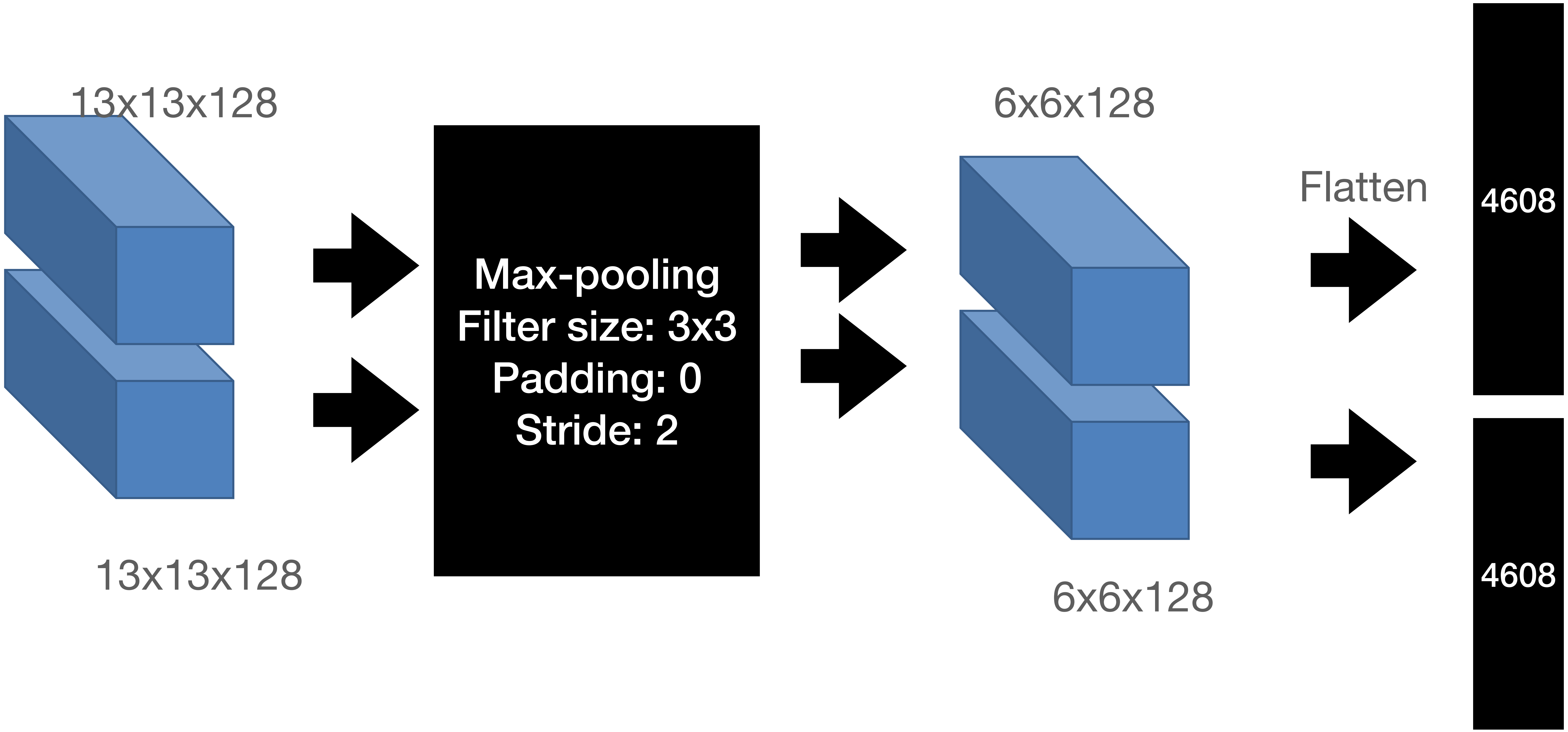
13x13
X128

13x13
X128





Max-pooling and flatten



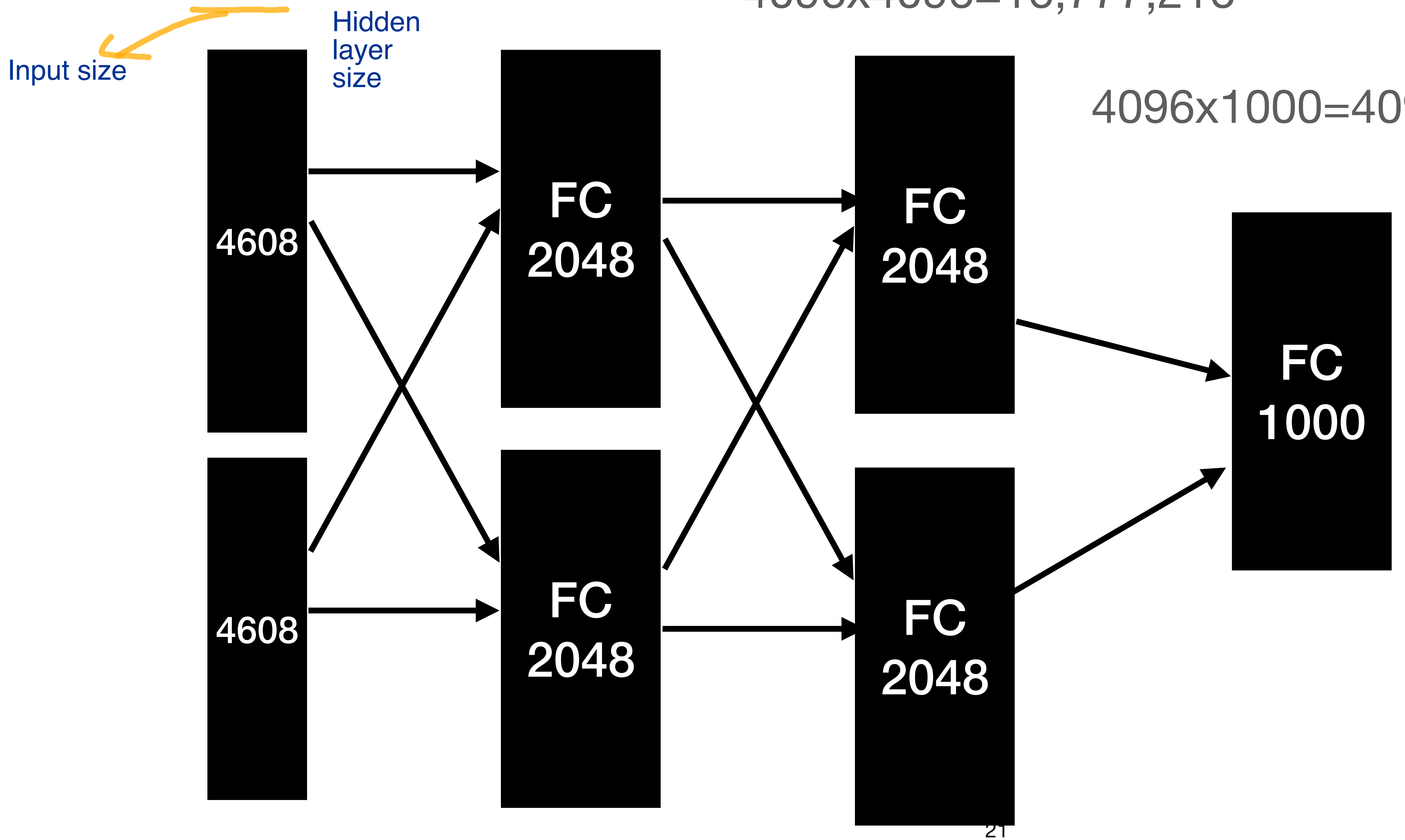
Following fully connected layers

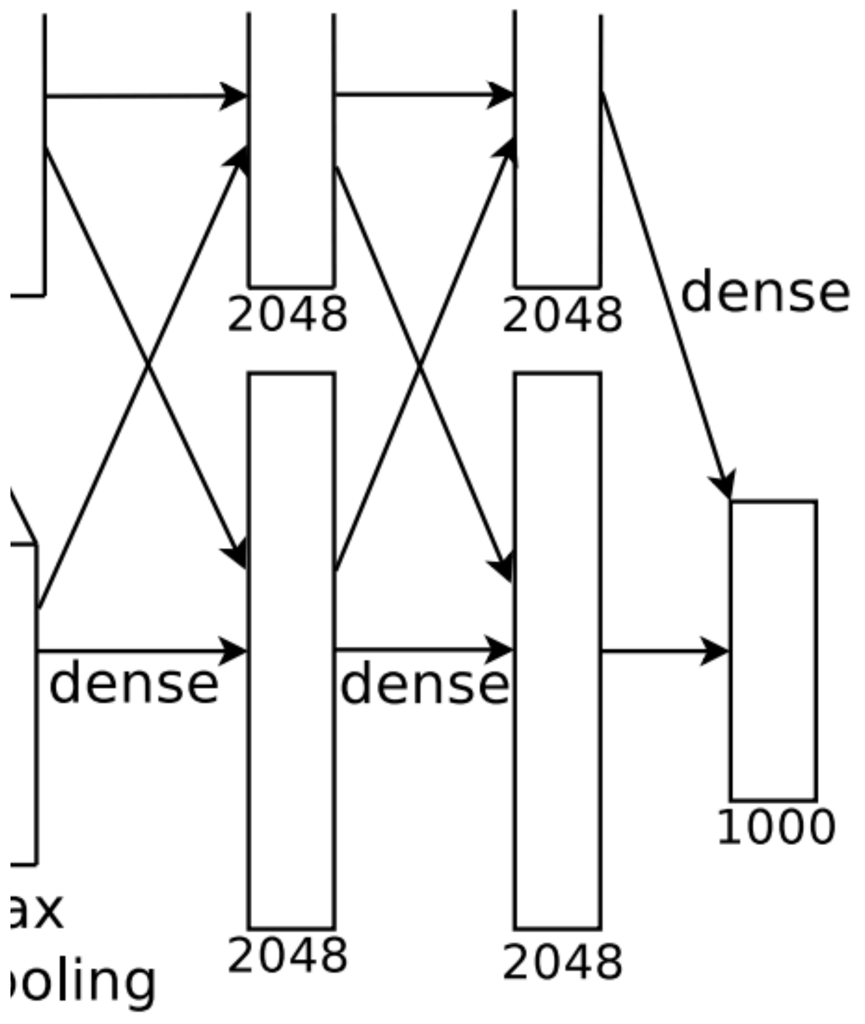
How many parameters?

9216x4096=37,748,736

4096x4096=16,777,216

4096x1000=4096000





Architecture

- VGG16: 16 weight layers

224 × 224x3

→ conv3-64, conv3-64, Maxpool → 112×112x64

→ Conv3-128, Conv3-128, Maxpool → 56×56x128

→ Conv3-256, Conv3-256, Conv3-256, Maxpool → 28×28x256

→ Conv3-512, Conv3-512, Conv3-512, Maxpool → 14×14x512

→ Conv3-512, Conv3-512, Conv3-512, Maxpool → 7×7x512

→ FC4096, FC4096, FC1000

Conv layer: kernel size 3x3, pad 1, stride 1

Maxpooling layer: 2x2,stride 2

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

Architecture

- VGG19: 19 weight layers

224 × 224x3

→ conv3-64 conv3-64 Maxpool → Conv3-128 Conv3-128 Maxpool → Conv3-256 Conv3-256 Conv3-256 Conv3-256 Maxpool → Conv3-512 Conv3-512 Conv3-512 Conv3-512 Maxpool → Conv3-512 Conv3-512 Conv3-512 Conv3-512 Maxpool → FC4096 FC4096 FC1000

Conv layer: kernel size 3x3, pad 1, stride 1

Maxpooling layer: 2x2,stride 2

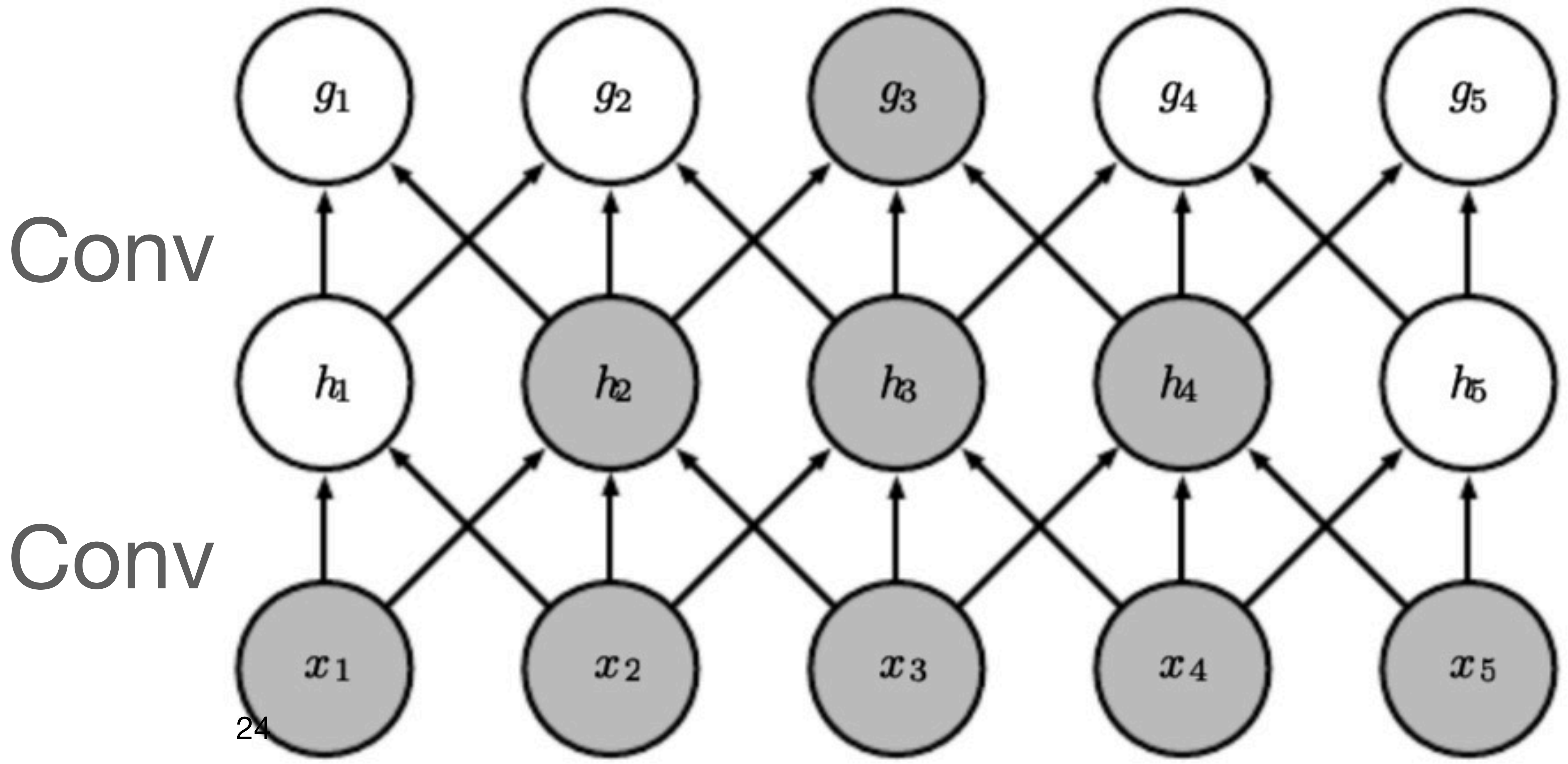
Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

Stacking multiple 3x3 conv layers

- a stack of two 3×3 conv. layers has an effective receptive field of 5×5
- a stack of 3 3×3 conv. layers has an effective receptive field of 7×7

If you add an additional convolutional layer with kernel size K, the receptive field is increased by (K-1)

More layers: larger size of receptive field (larger window of the input is seen)



Why Stacking multiple 3x3 conv layers instead of large filter size?

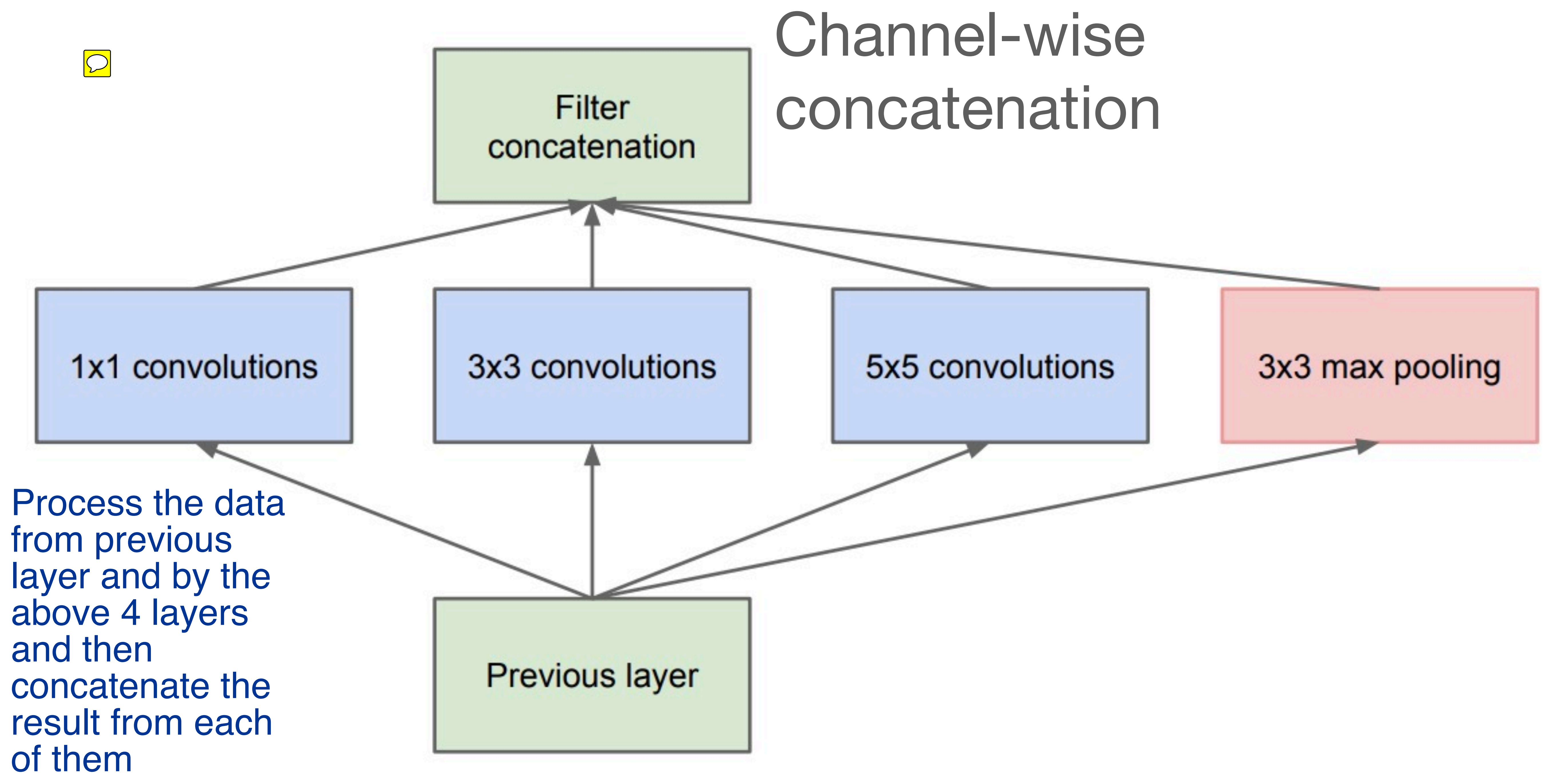
- Reduce parameter:
  - 5x5=25, 3x3x2=18
  - 7x7=49, 3x3x3=27

We could achieve a similar receptive field by stacking 3 3*3 conv layer but with less parameters but more powerful

- Each conv use ReLU as the activation function. More layers, more non-linear rectification layers → More powerful network

Architecture

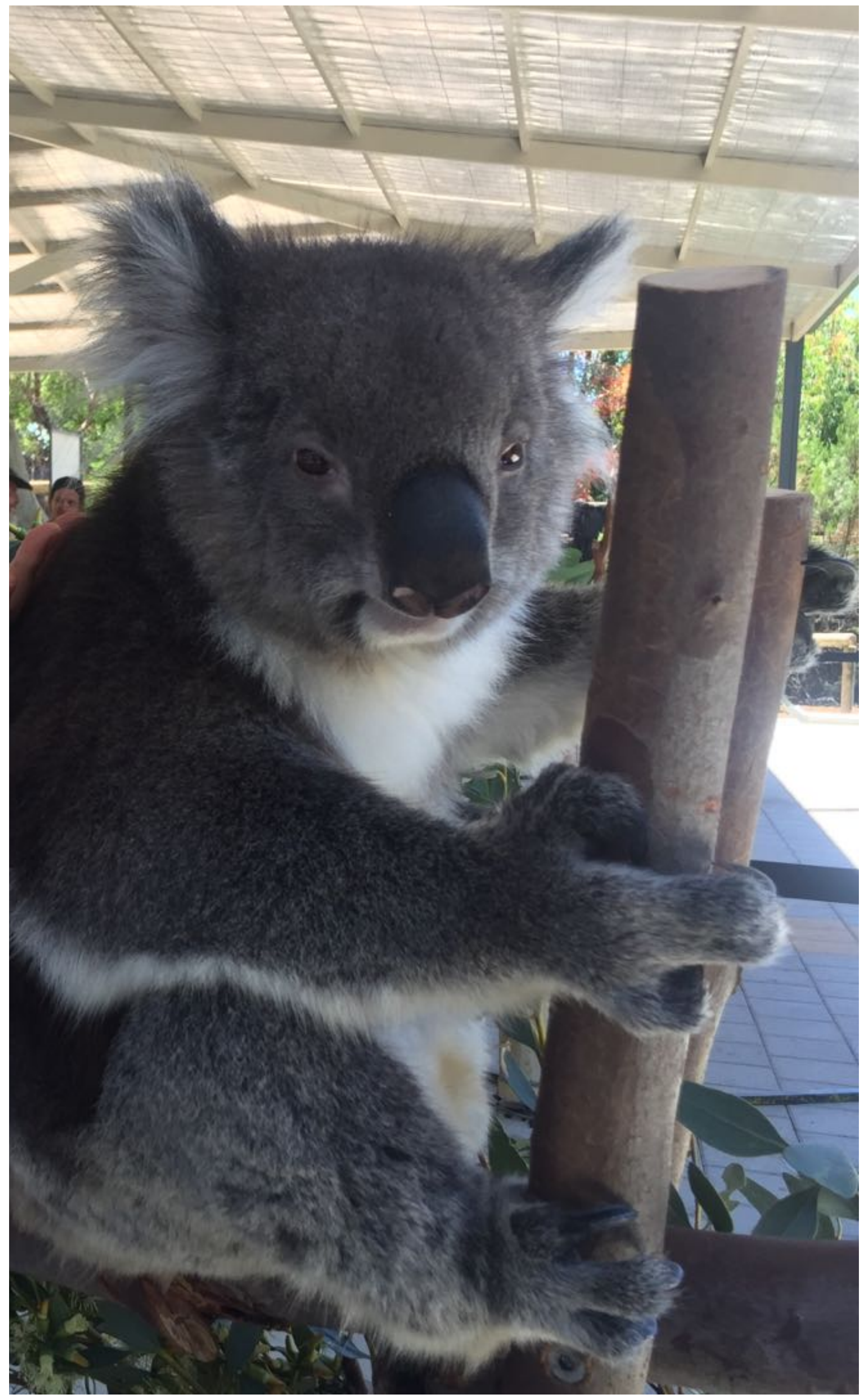
Inception module (Naive version):



type	patch size/ stride	output size	depth
convolution	7×7/2	112×112×64	1
max pool	3×3/2	56×56×64	0
convolution	3×3/1	56×56×192	2
max pool	3×3/2	28×28×192	0
inception (3a)		28×28×256	2
inception (3b)		28×28×480	2
max pool	3×3/2	14×14×480	0
inception (4a)		14×14×512	2
inception (4b)		14×14×512	2
inception (4c)		14×14×512	2
inception (4d)		14×14×528	2
inception (4e)		14×14×832	2
max pool	3×3/2	7×7×832	0
inception (5a)		7×7×832	2
inception (5b)		7×7×1024	2
avg pool	7×7/1	1×1×1024	0
dropout (40%)		1×1×1024	0
linear		1×1×1000	1
softmax		1×1×1000	0

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015

Different scales of data require different convolutional filter sizes



Inception module (Naive version)

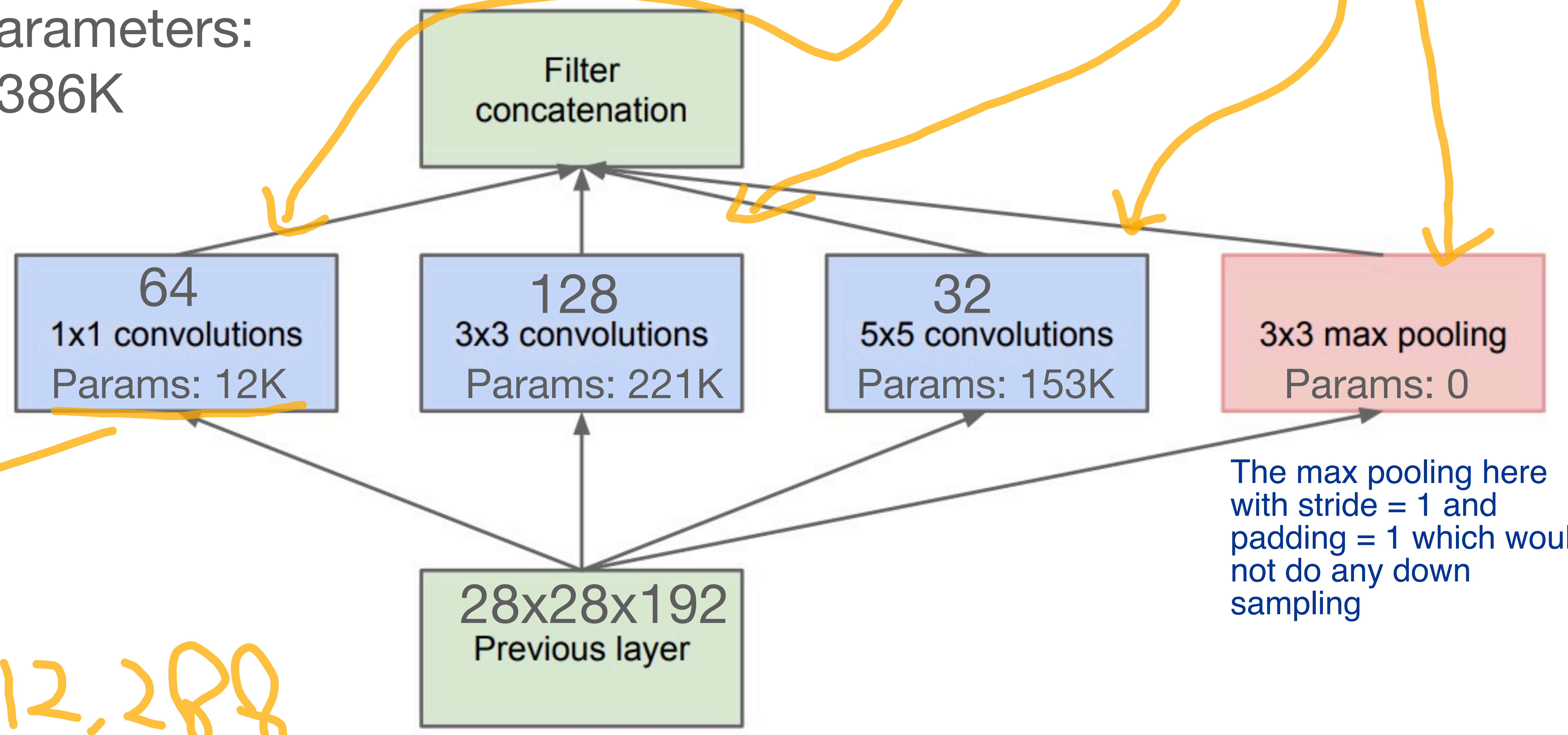
How many parameters?

Input_channel x
Kernel_size x
NO.filters
(output_channel)

Total parameters:
~386K

Channel-wise
concatenation

28x28x(64+128+32+192)
=28x28x416



The max pooling here with stride = 1 and padding = 1 which would not do any down sampling

192x1x1x64=12,288

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015

Inception module (Naive version)

More modules

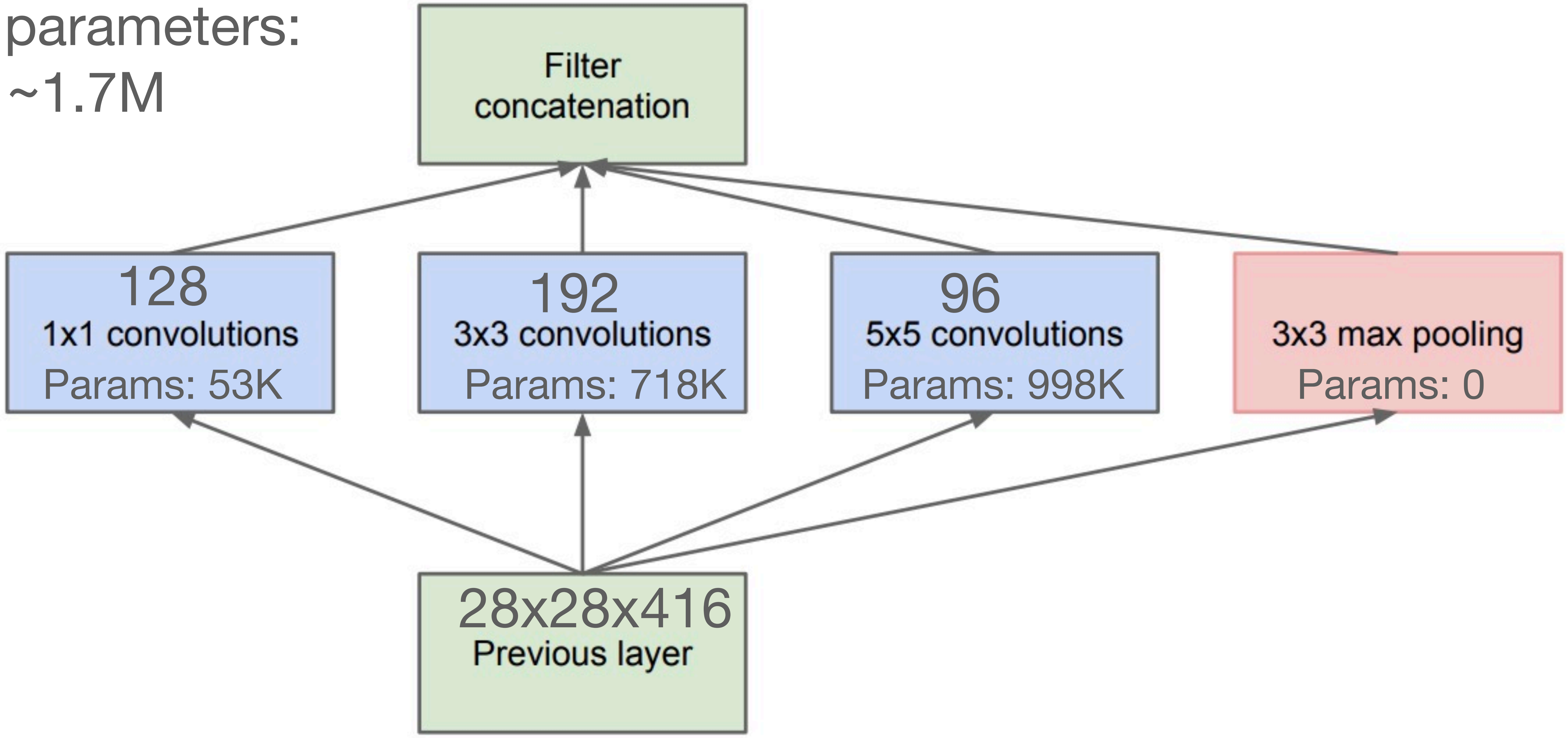
More parameters

More computations

Total parameters:
~1.7M

Channel-wise
concatenation

28x28x(128+192+96+416)
=28x28x832



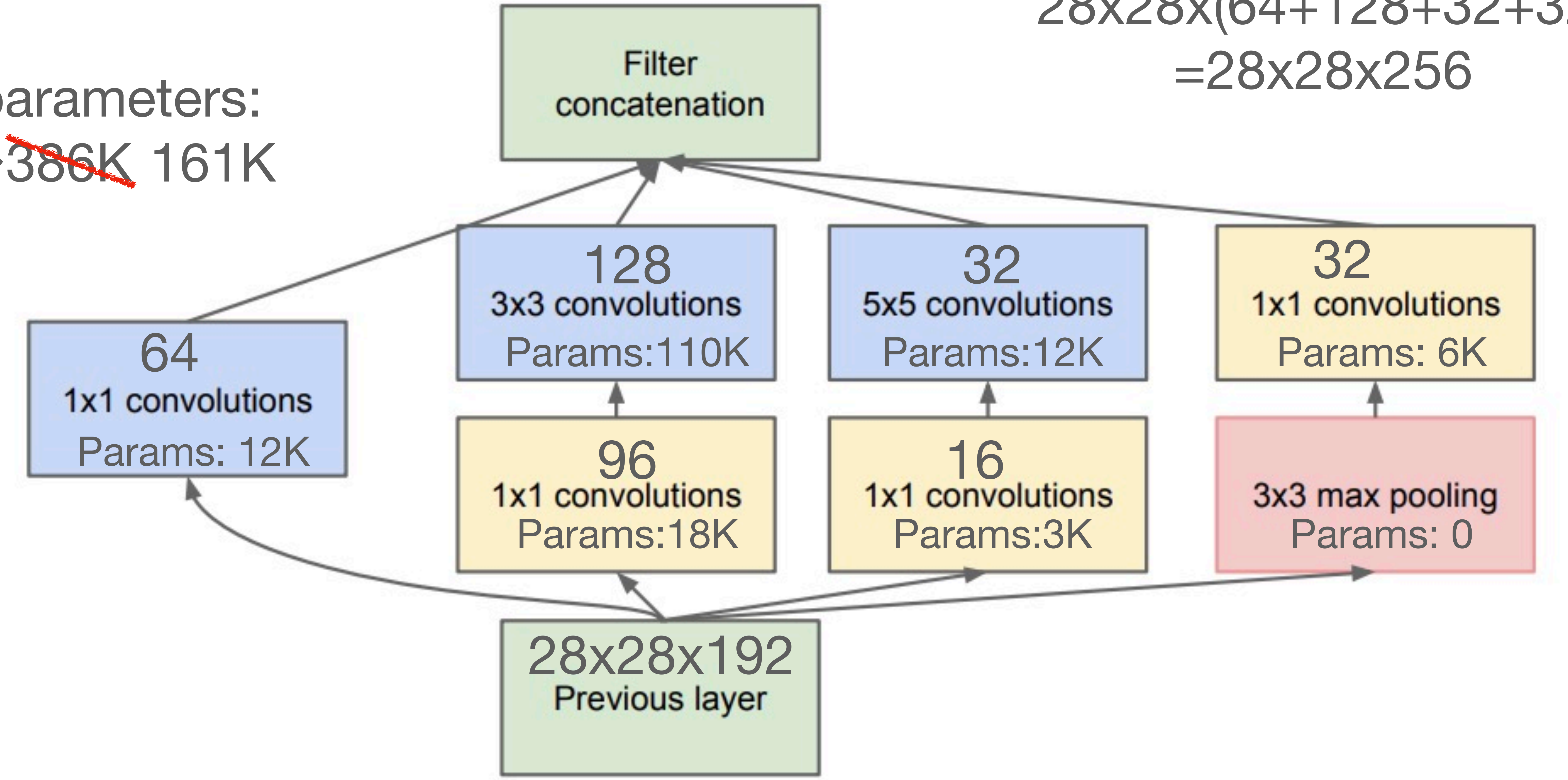
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015

Inception module with dimensionality reduction

Use 1x1 convolution to reduce channels

Total parameters:
~~~386K~~ 161K

28x28x(64+128+32+32)
=28x28x256



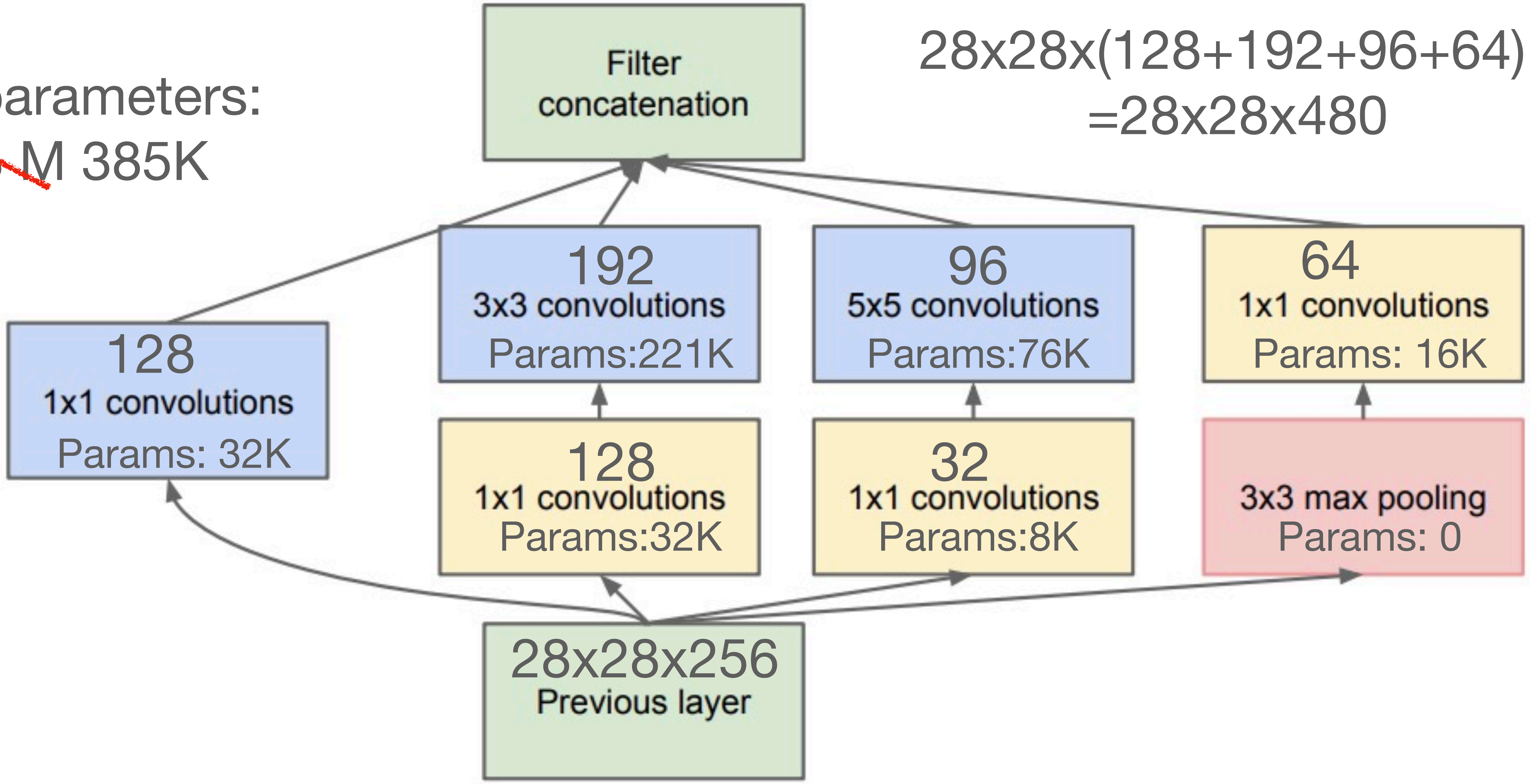
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015
~~~

Inception module with dimensionality reduction

Use 1x1 convolution to reduce channels

Total parameters:
~~~1.3 M~~ 385K

28x28x(128+192+96+64)
=28x28x480



Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015
~~~

Large Scale Visual Recognition Challenge

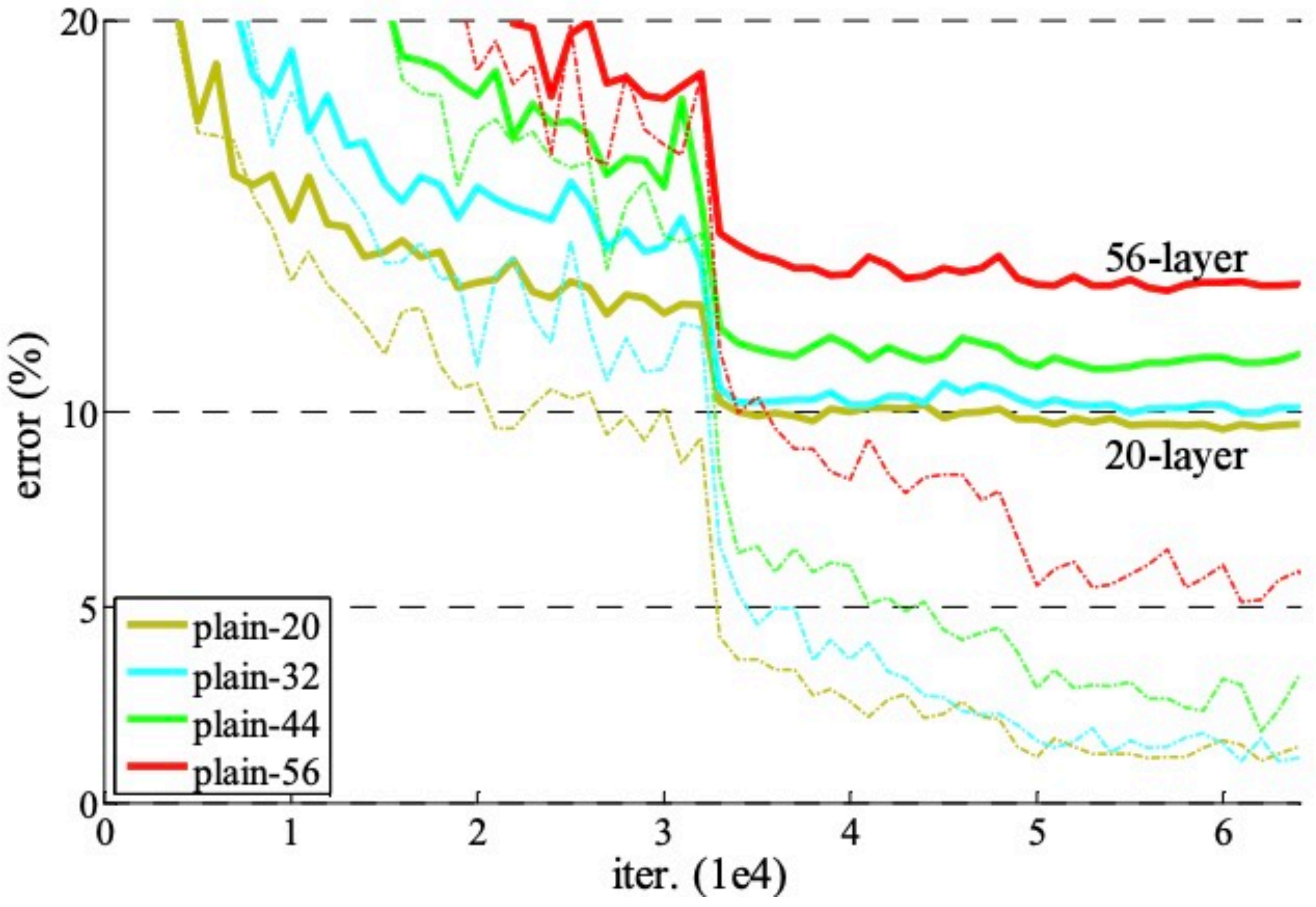
Comparison of different architectures

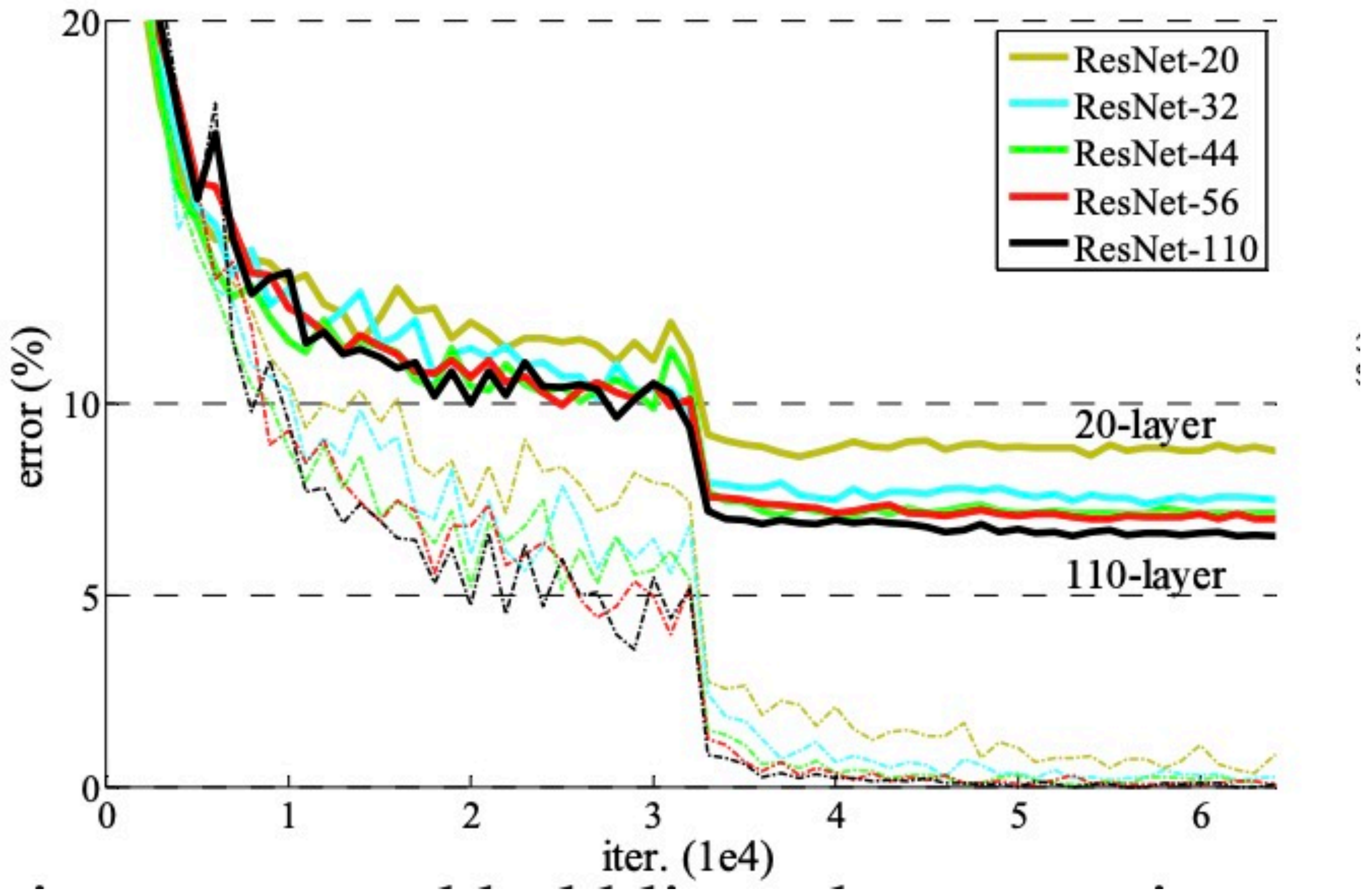
- Top-5 error: the proportion of images that the ground-truth category is outside the top-5 predicted categories of the model.

CNN Architecture	Layers	Top-5 error
AlexNet	8	16.4%
VGG-19	19	7.3%
GoogleNet	22	6.7%

More layers?

Residual network: more layers & better performance



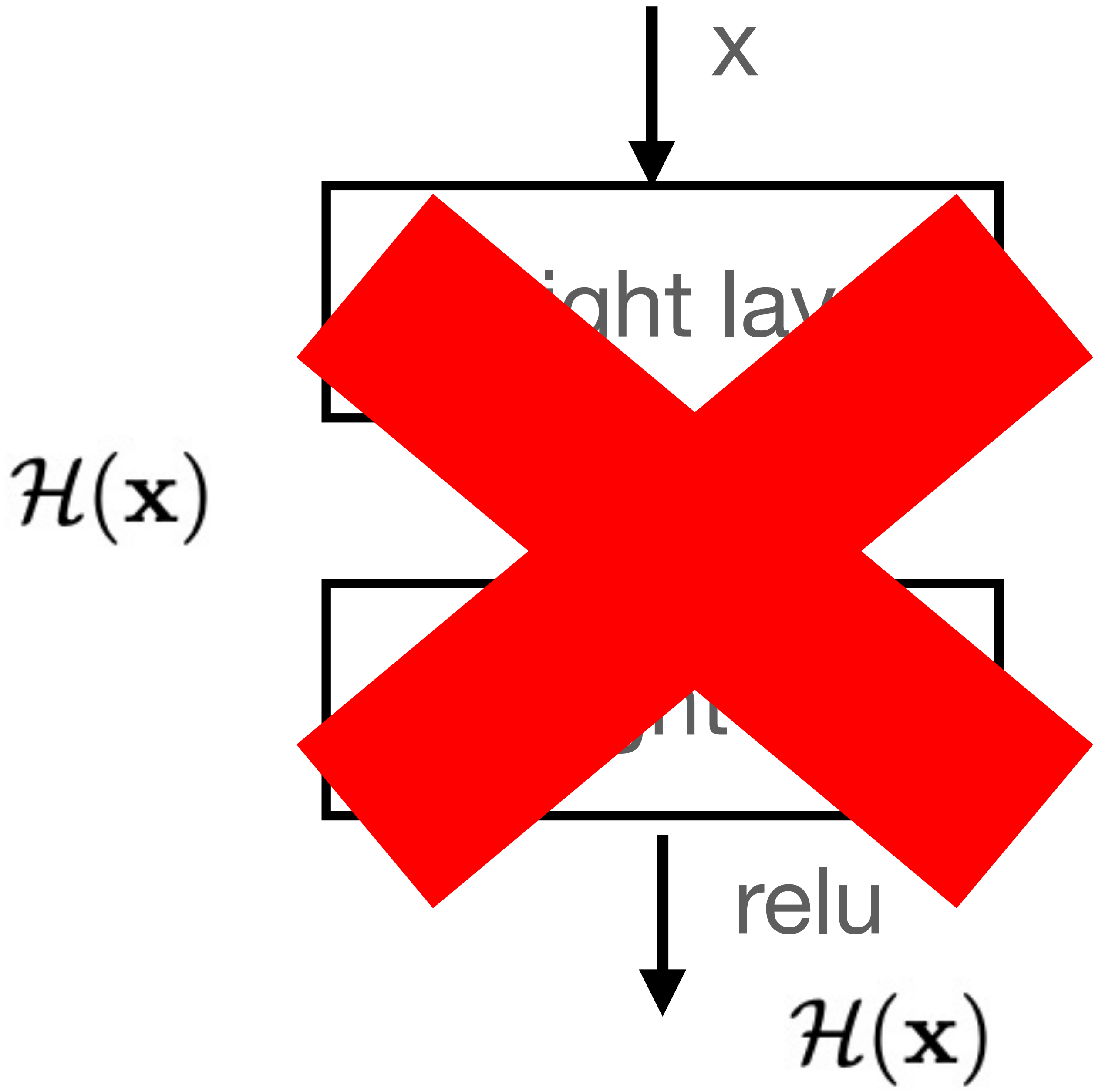


Training on CIFAR-10. Dashed lines denote training error, and bold lines denote testing error

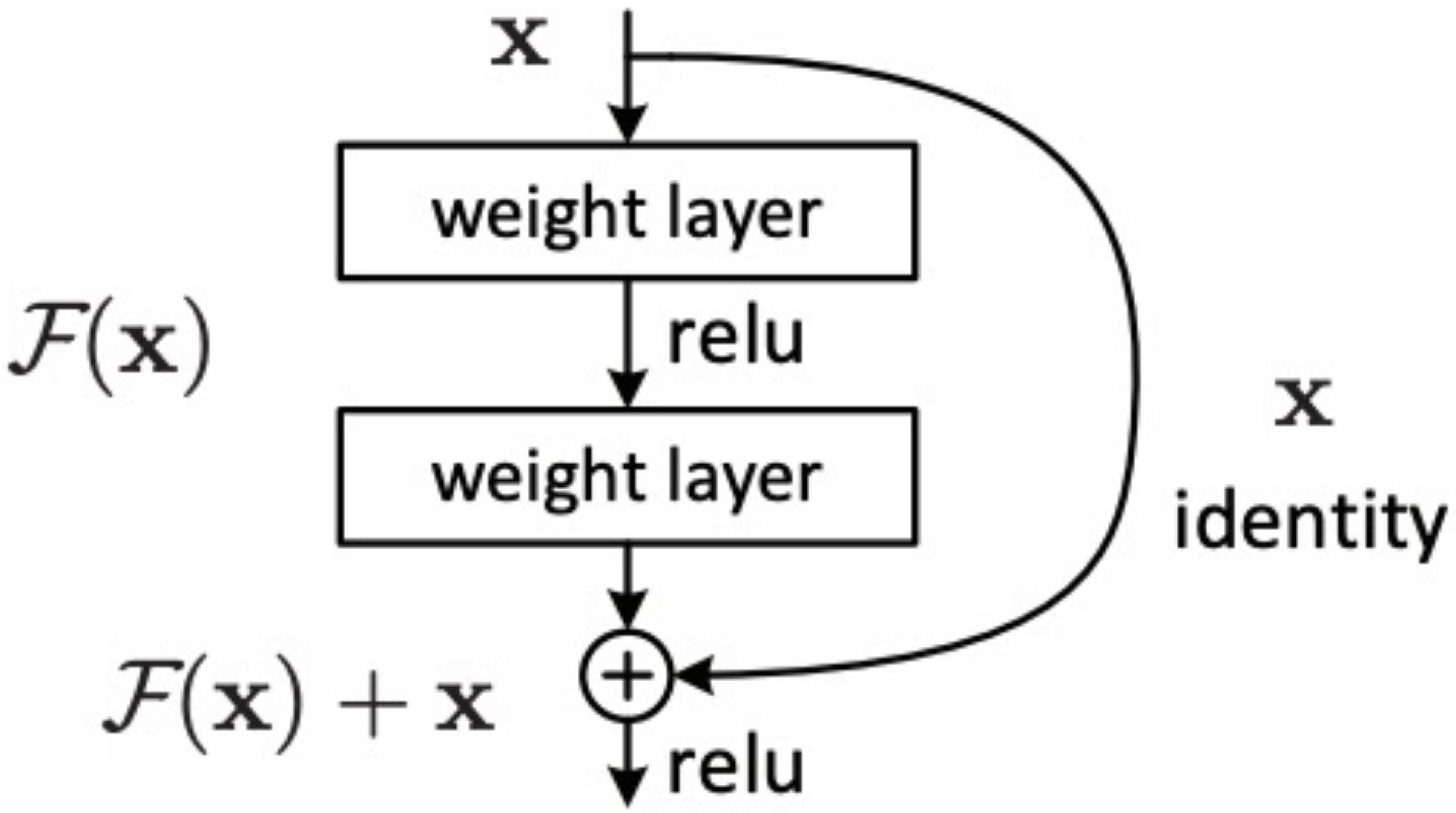
He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Residual network

Hypothesis: residual mapping $\mathcal{F}(\mathbf{x}) := \mathcal{H}(\mathbf{x}) - \mathbf{x}$ is easier to optimise



Unreferenced mapping:
directly fit the desired underlying mapping



Residual learning:
let these layers fit a residual mapping

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Residual network (34 layers)

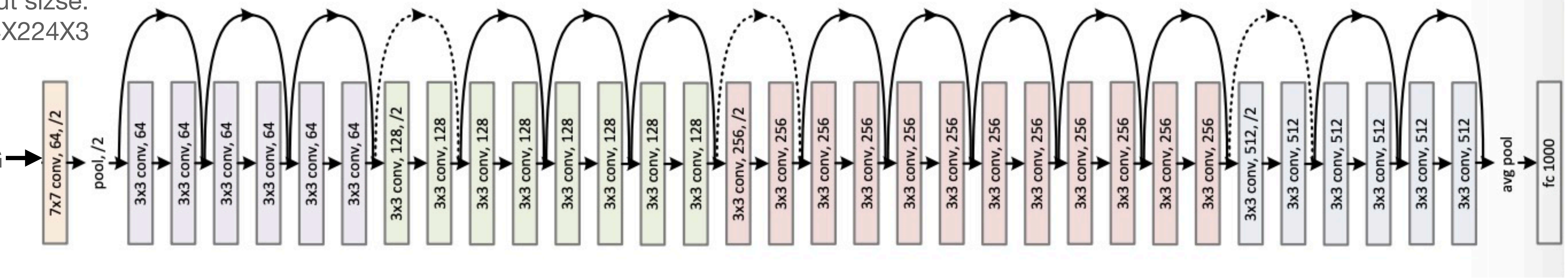
conv 1x1,128

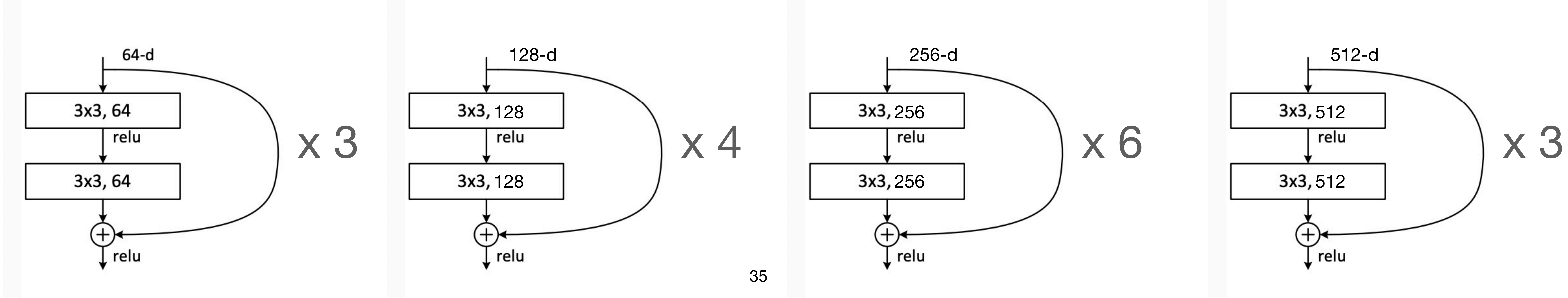
conv 1x1,256

conv 1x1,512

Input sizse: 224X224X3

IMG





Residual network

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				

Downsampling is performed by conv3 1, conv4 1, and conv5 1 with a stride of 2

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Large Scale Visual Recognition Challenge

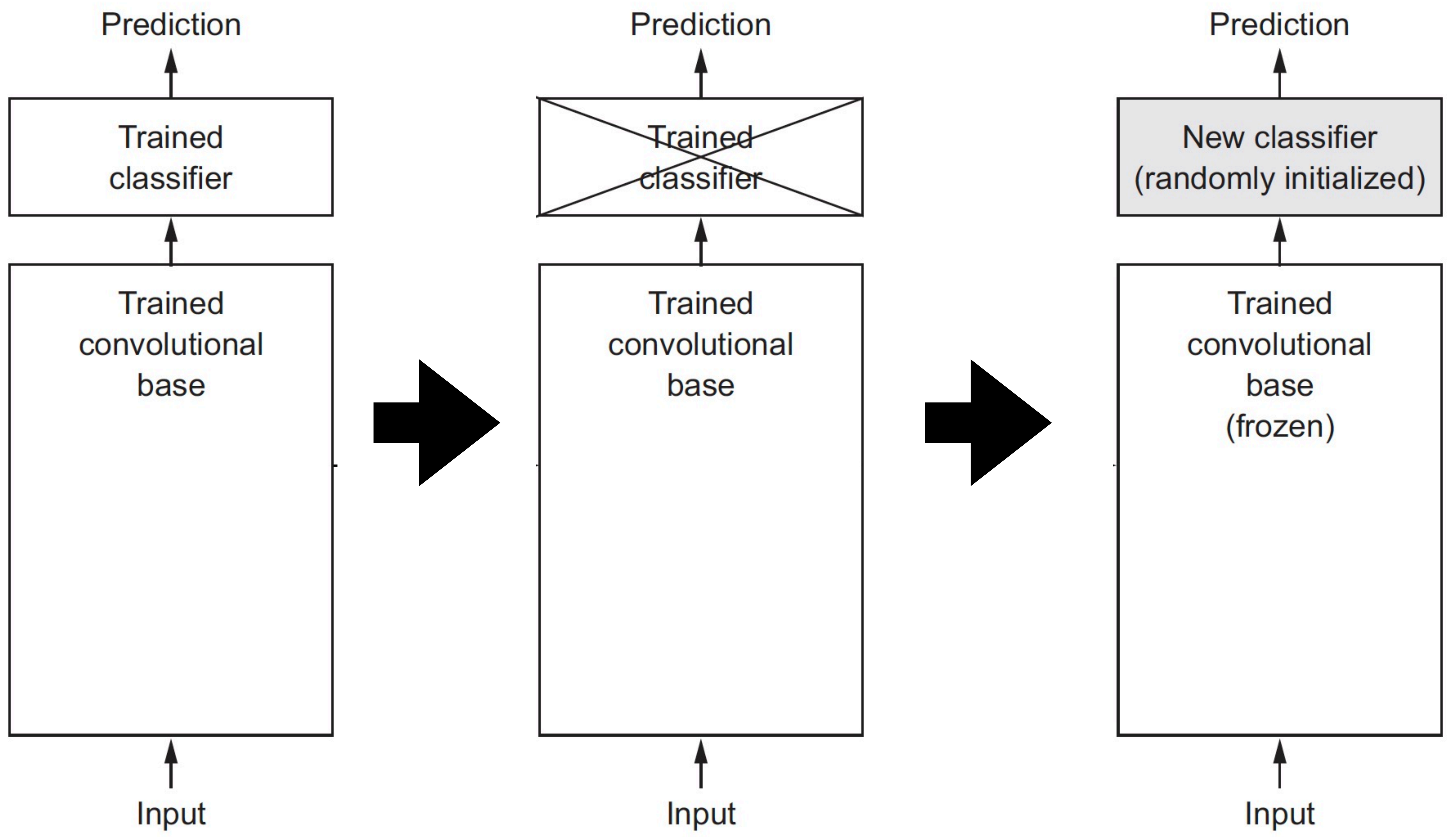
Comparison of different architectures

- Top-5 error: the proportion of images that the ground-truth category is outside the top-5 predicted categories of the model.

CNN Architecture	Layers	Top-5 error
AlexNet	8	16.4%
VGG-19	19	7.3%
GoogleNet	22	6.7%
ResNet	152	3.57%

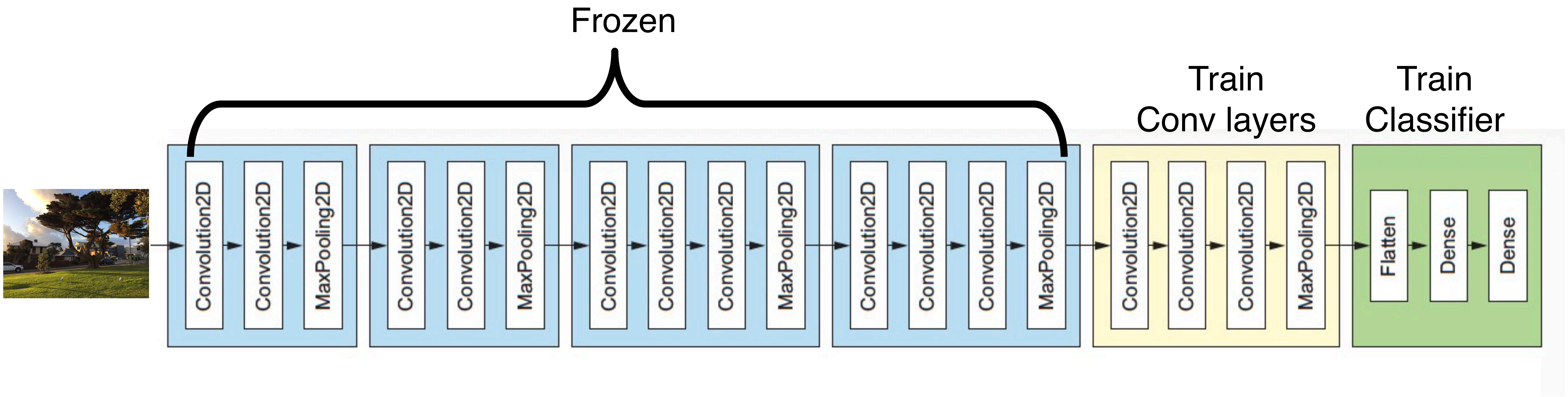
Use the pretrained CNN model as feature extractor

Train a new classifier for output



If you have quite a lot of data: fine-tuning

Slightly train a few more top layers



Summary

- How to calculate the NO. parameters & size of output feature map ?
- Difference of the architectures
- Key idea of VGG: how to increase receptive field?
- Key idea of GoogleNet: how to reduce parameters?
- Key idea of ResNet: how to increase layers with better performance?