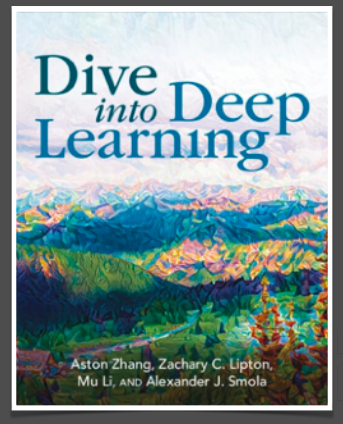
ROBOT INTELLIGENCE LAB

Introduction to Deep Learning

Convolutional Neural Networks

Sungjoon Choi, Korea University



ROBOT INTELLIGENCE LAB

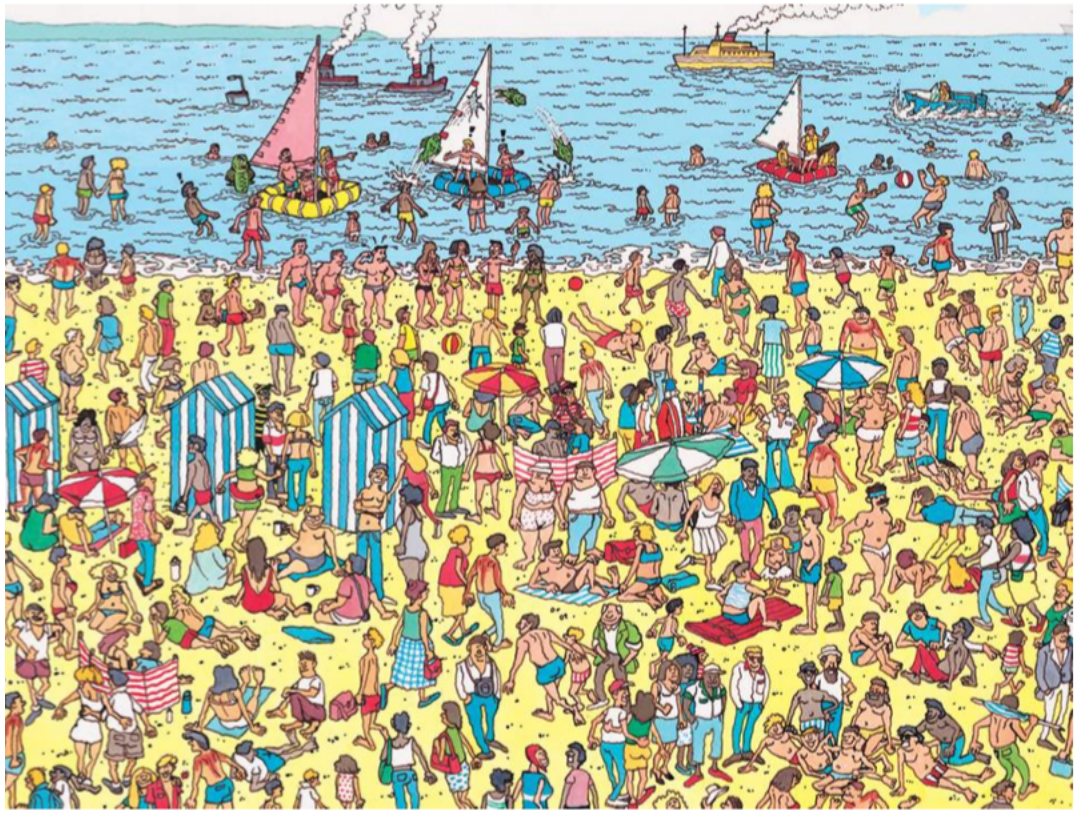
Convolution Operation

Fully Connected to Convolutions

ROBOT INTELLIGENCE LAB

Why do convolutional neural networks (CNNs) more appropriate than MLPs in image domains?

Invariance



- Imagine that we want to detect an object in an image (Wheres Waldo).
- It seems reasonable that whatever method we use, the precise location of the object in the image should not be overly concerned.

Invariance

ROBOT INTELLIGENCE LAB

- Despite Waldo's characteristic outfit, what Waldo looks like does not depend upon where is located.
- We could sweep the image with a Waldo detector that could assign a score to each patch, indicating the likelihood that the patch contains Waldo.
- Convolutional neural networks (CNNs) systematize this idea of spatial invariance, exploiting it to learn useful representations with fewer parameters.

Convolutions (in signal processings)

ROBOT INTELLIGENCE LAB

- In mathematics, the convolution between two functions, say f and g, is defined as

$$(f * g)(\mathbf{x}) = \int f(\mathbf{z})g(\mathbf{x} - \mathbf{z})d\mathbf{z}.$$

- That is, we measure the overlap between f and g when on function if flipped and shifted by $\mathbf{x}$ (i.e., $g(\mathbf{x} - \mathbf{z})$).
- When we have discrete objects in two-dimensional spaces:

$$(f * g)(i, j) = \sum_a \sum_b f(a, b)g(i - a, j - b).$$

Convolutions (in CNNs)

ROBOT INTELLIGENCE LAB

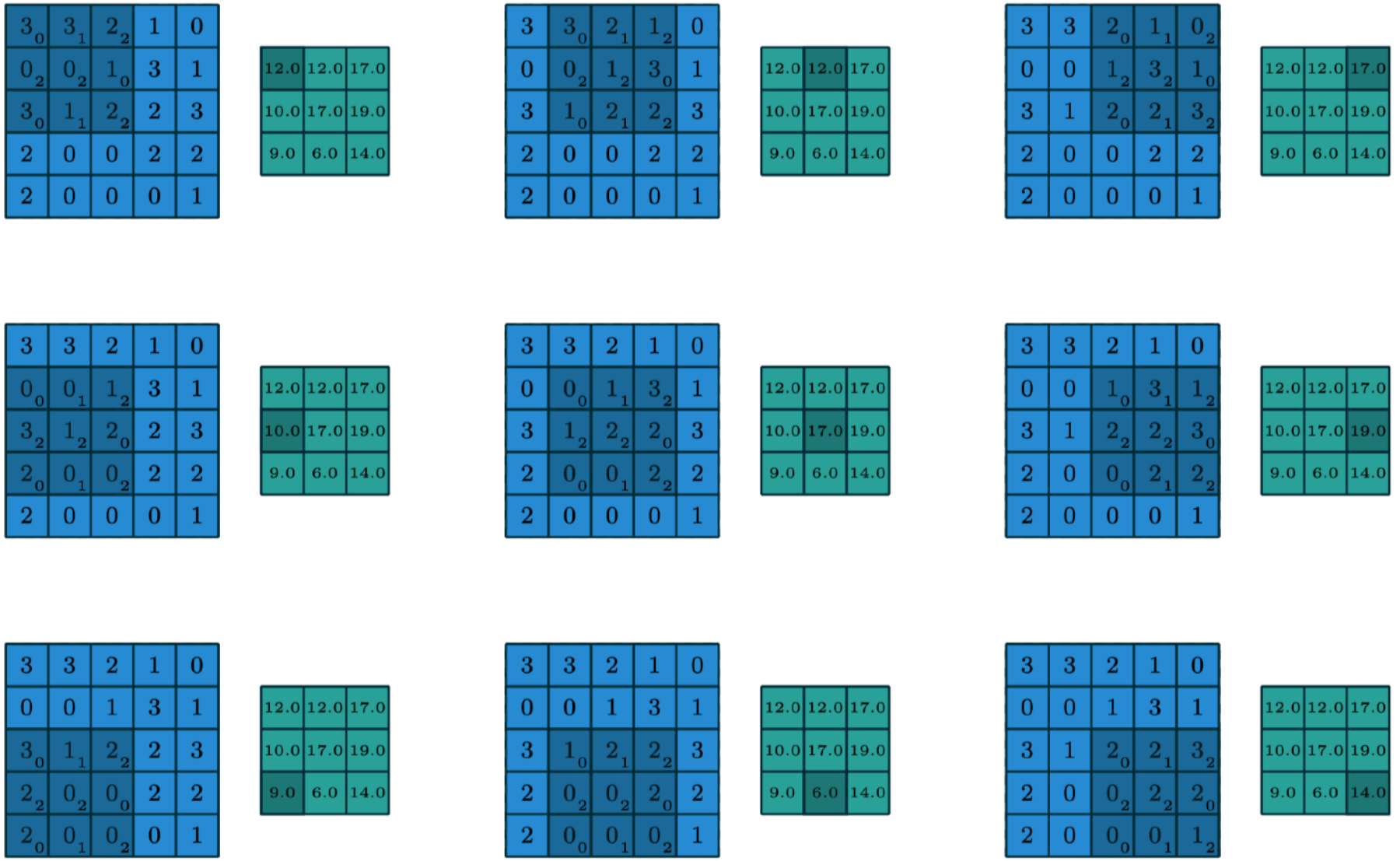
- To exploit the translation invariance, an output of a convolutional layer is computed as:

$$[\mathbf{H}]_{i,j} = u + \sum_{a=-\triangle}^{\triangle} \sum_{b=-\triangle}^{\triangle} [\mathbf{V}]_{a,b} [\mathbf{X}]_{i+a,j+b}$$

  where $\mathbf{V}$ is referred to as a convolution kernel or a filter.

- The number of parameters for the layer is $4\triangle^2$.
- This operation is more properly described as a cross-correlation.

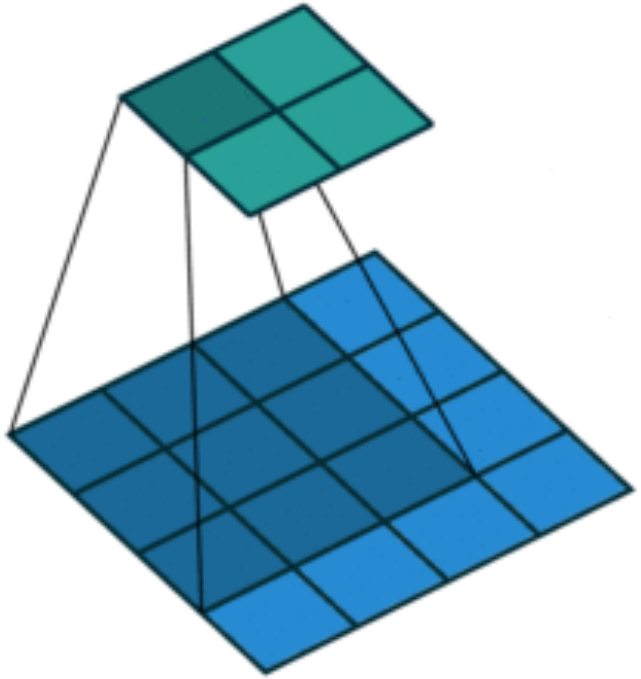
Convolution



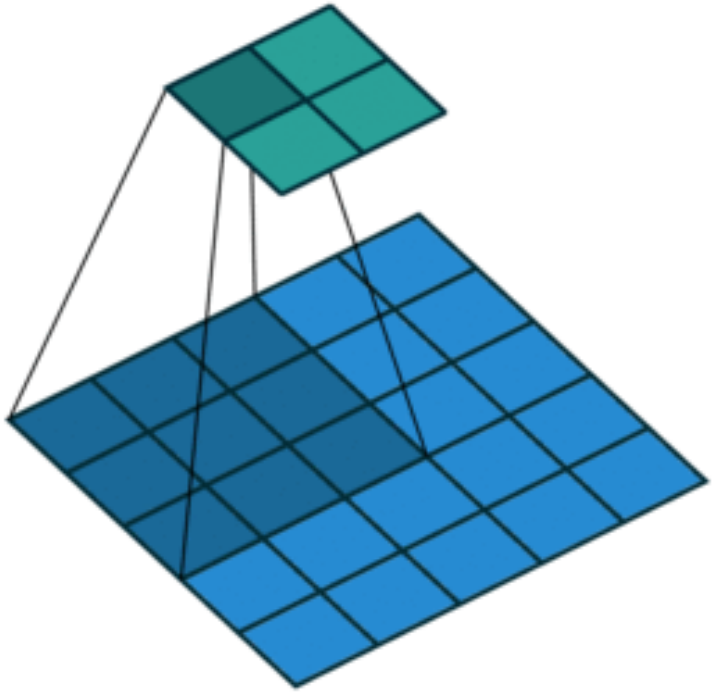
"A guide to convolution arithmetic for deep learning," 2018

Convolution (Padding and Strides)

ROBOT INTELLIGENCE LAB



padding=0 and stride=1



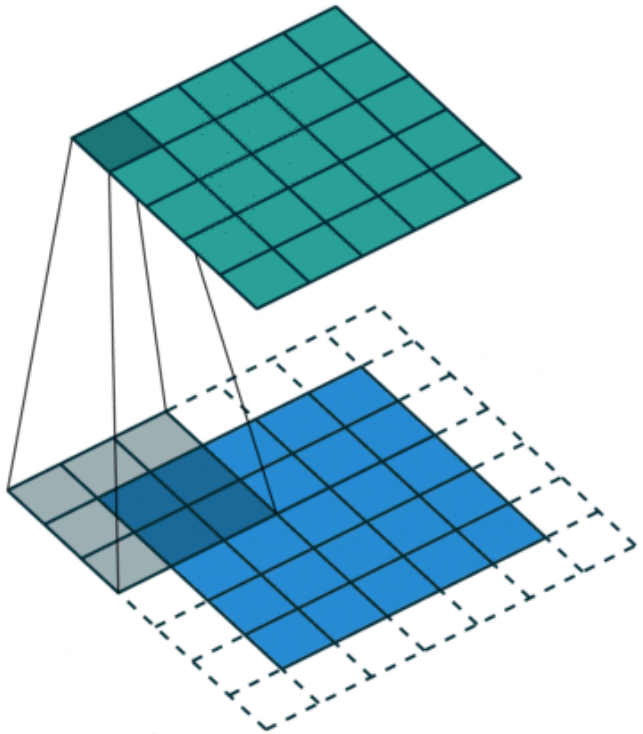
padding=0 and stride=2

Blue maps are inputs and cyan maps are outputs.

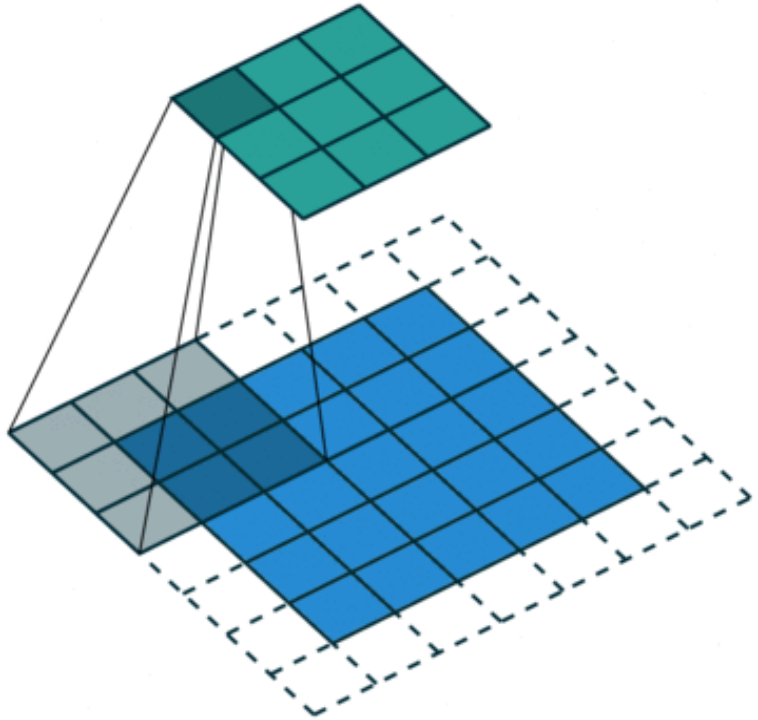
https://github.com/vdumoulin/conv_arithmetic

Convolution (Padding and Strides)

ROBOT INTELLIGENCE LAB



padding=1 and stride=1

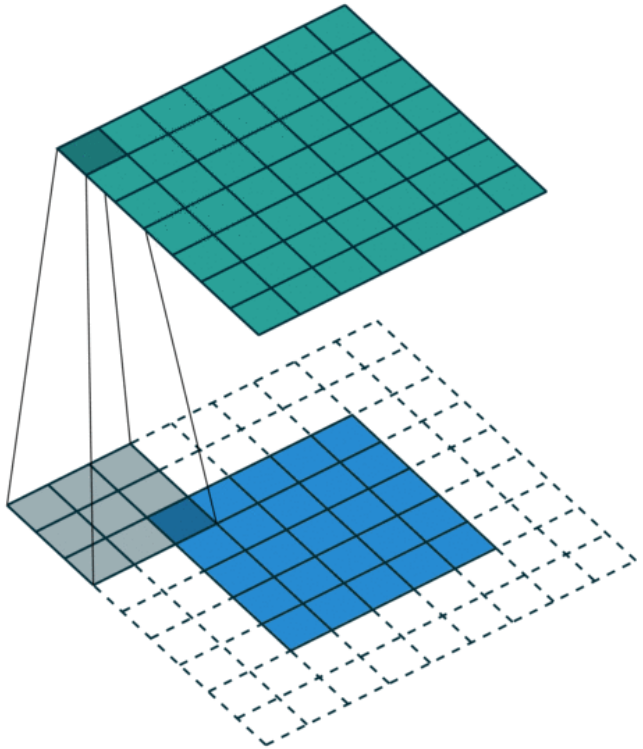


padding=1 and stride=2

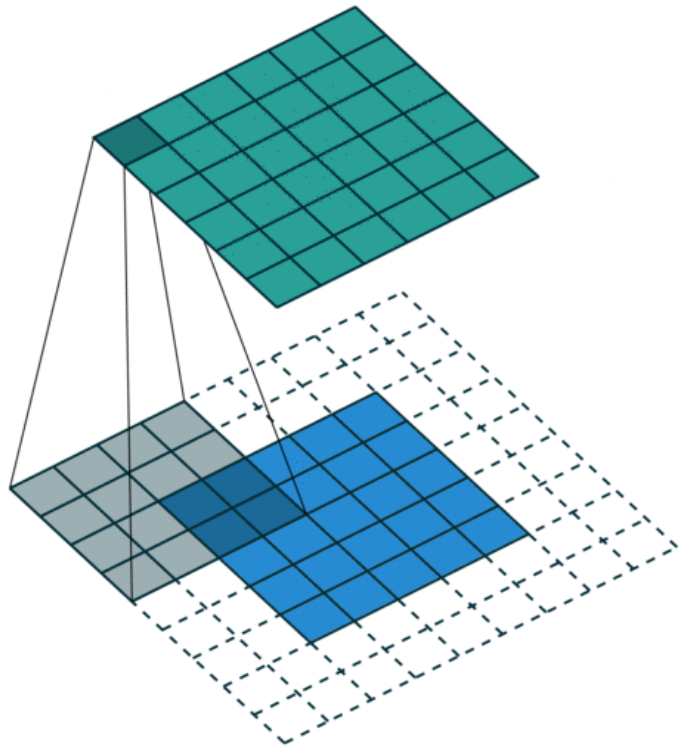
https://github.com/vdumoulin/conv_arithmetic

Convolution (Padding and Strides)

ROBOT INTELLIGENCE LAB



padding=2 and stride=1



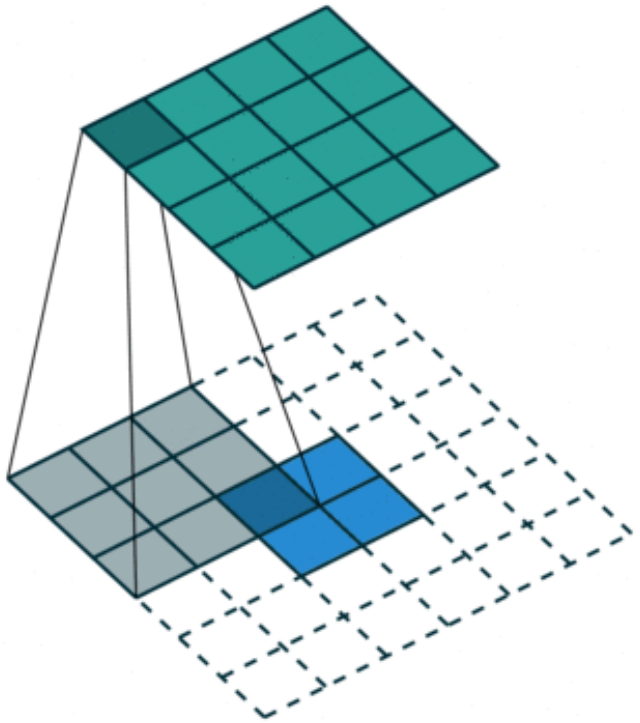
padding=2 and stride=1

Note that the sizes of the output features vary!

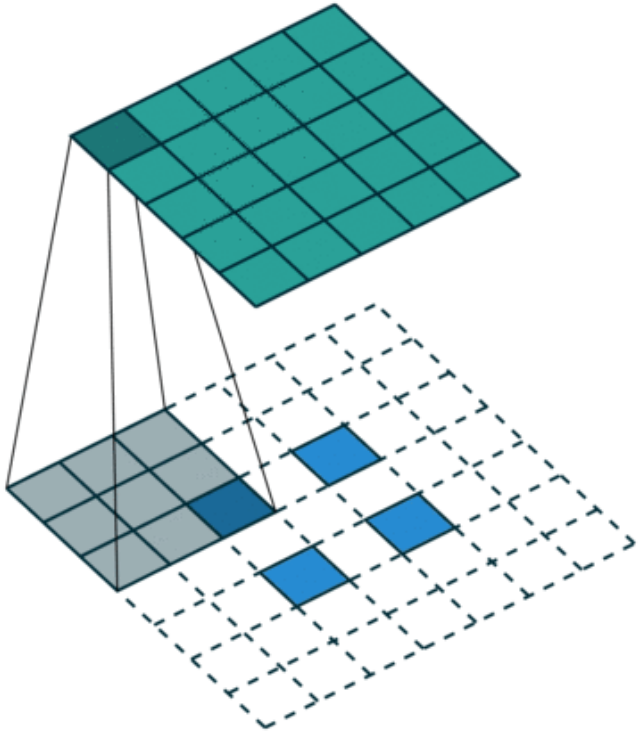
https://github.com/vdumoulin/conv_arithmetic

Transposed Convolution (Padding and Strides)

ROBOT INTELLIGENCE LAB



padding=0 and stride=0

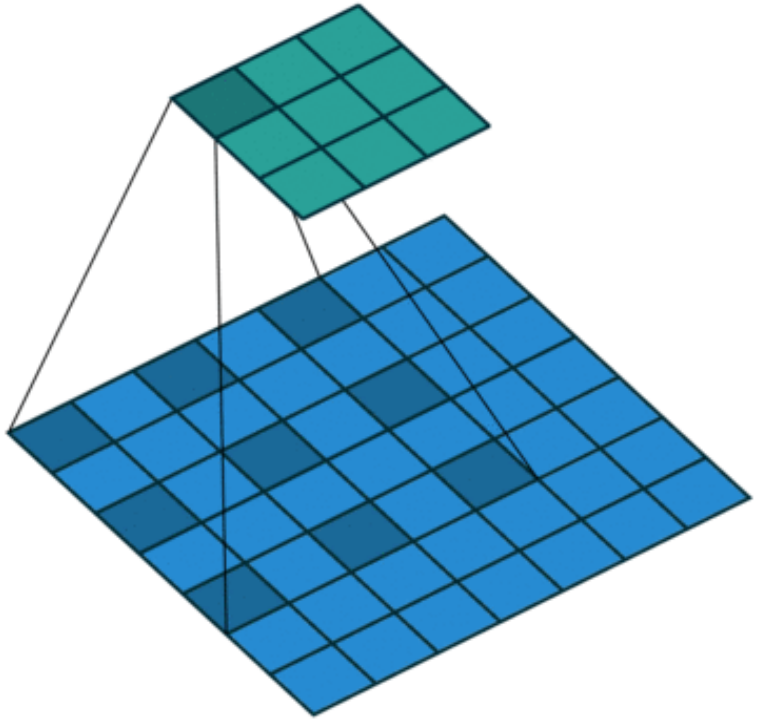


padding=0 and stride=1

Blue maps are inputs and cyan maps are outputs.

https://github.com/vdumoulin/conv_arithmetic

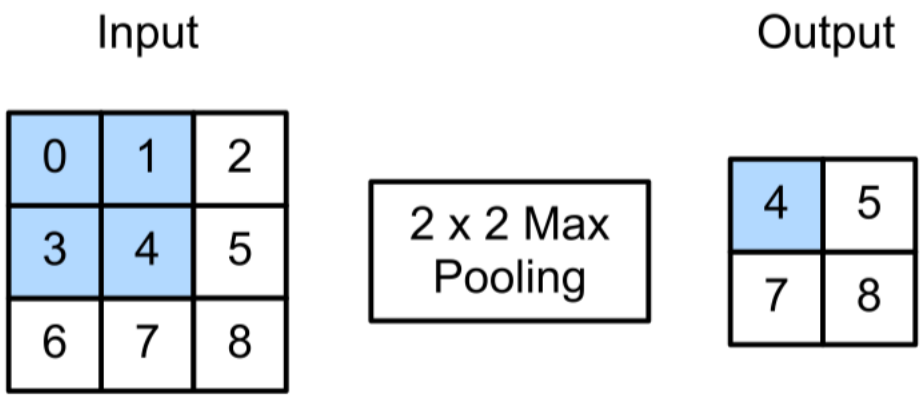
Dilated Convolution (Padding and Strides)



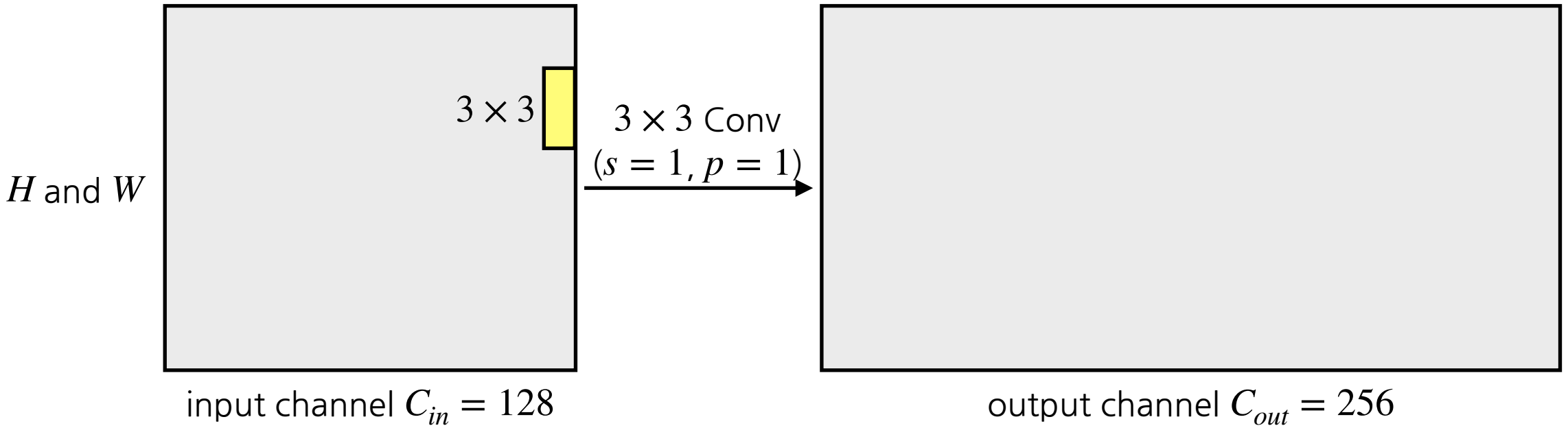
padding=0 and stride=1

https://github.com/vdumoulin/conv_arithmetic

Pooling

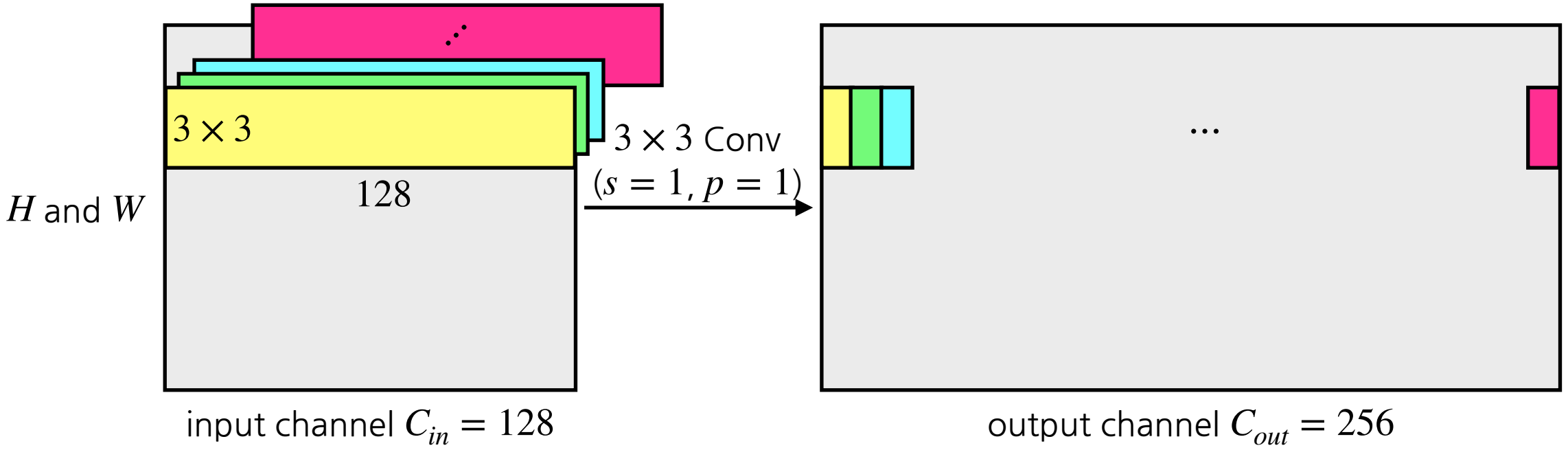


Channel



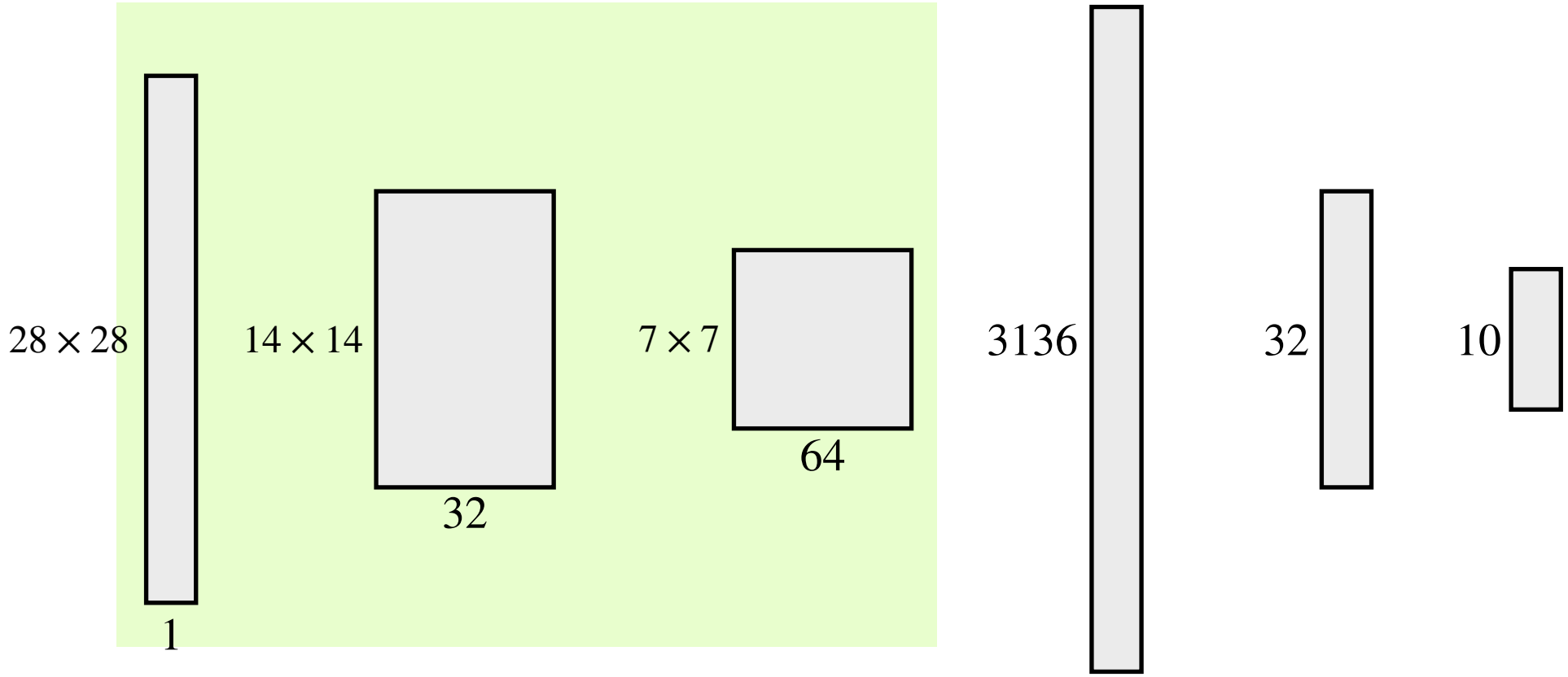
- What is the number of parameters of this 3×3 convolution layer?

Channel



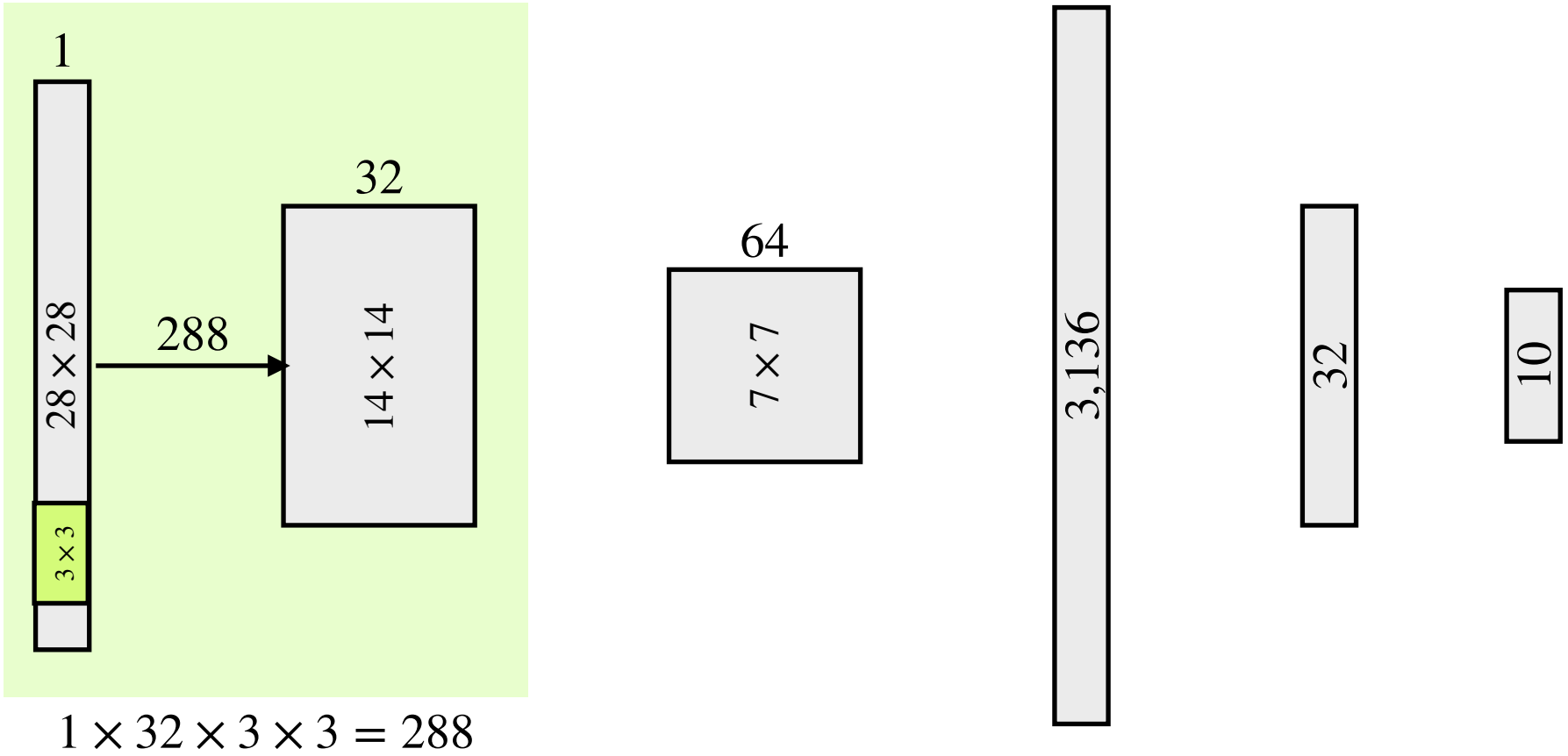
- What is the number of parameters of this 3×3 convolution layer?
  - Each kernel has a dimension of $128 \times 3 \times 3$
  - And we have 256 kernels.
  - The number of parameters is $128 \times 256 \times 3 \times 3 = 294{,}912$

Convolutional Neural Network

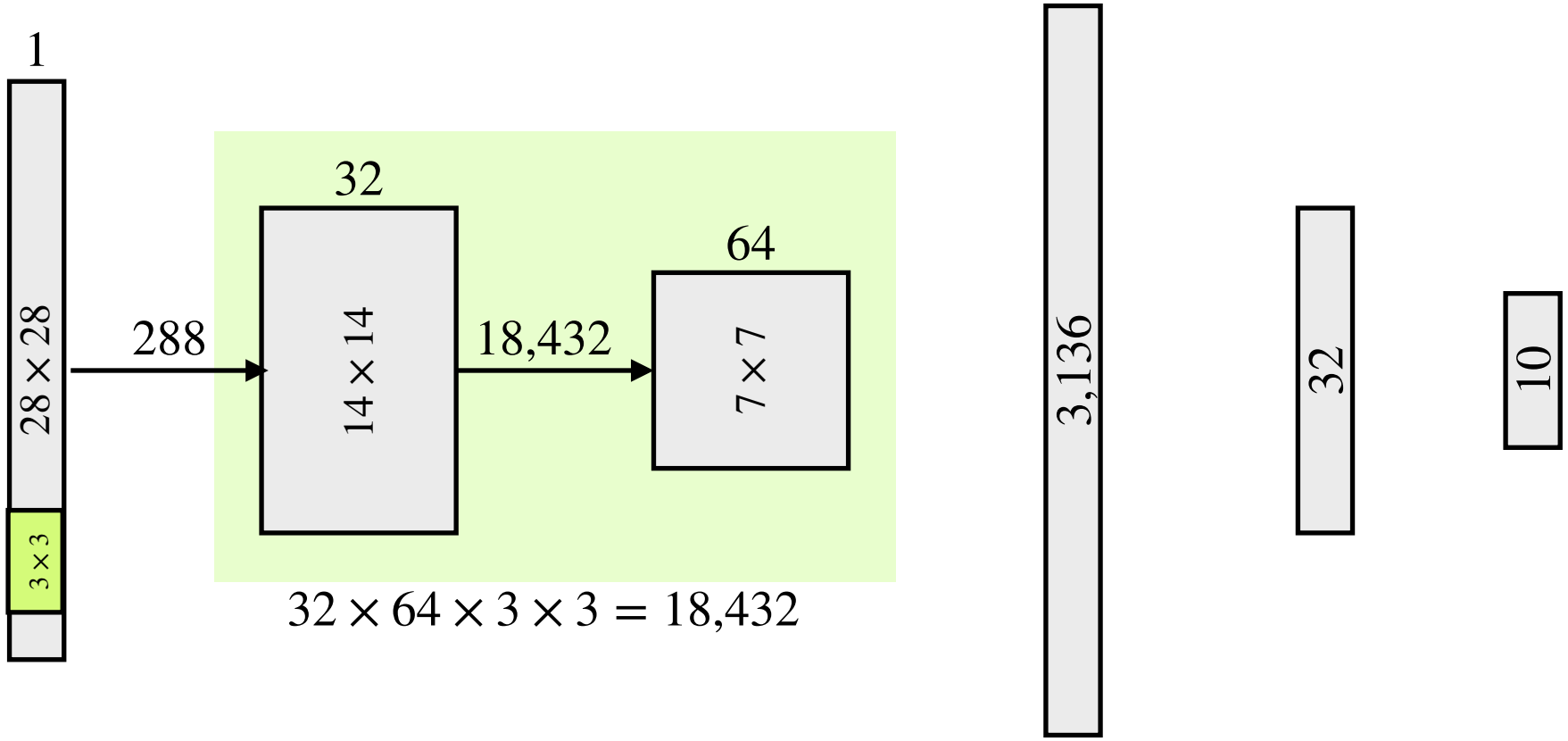


- Convolutions with 3×3 kernels, stride is two, and padding is one.
- Max-pooling with 2×2 kernels.

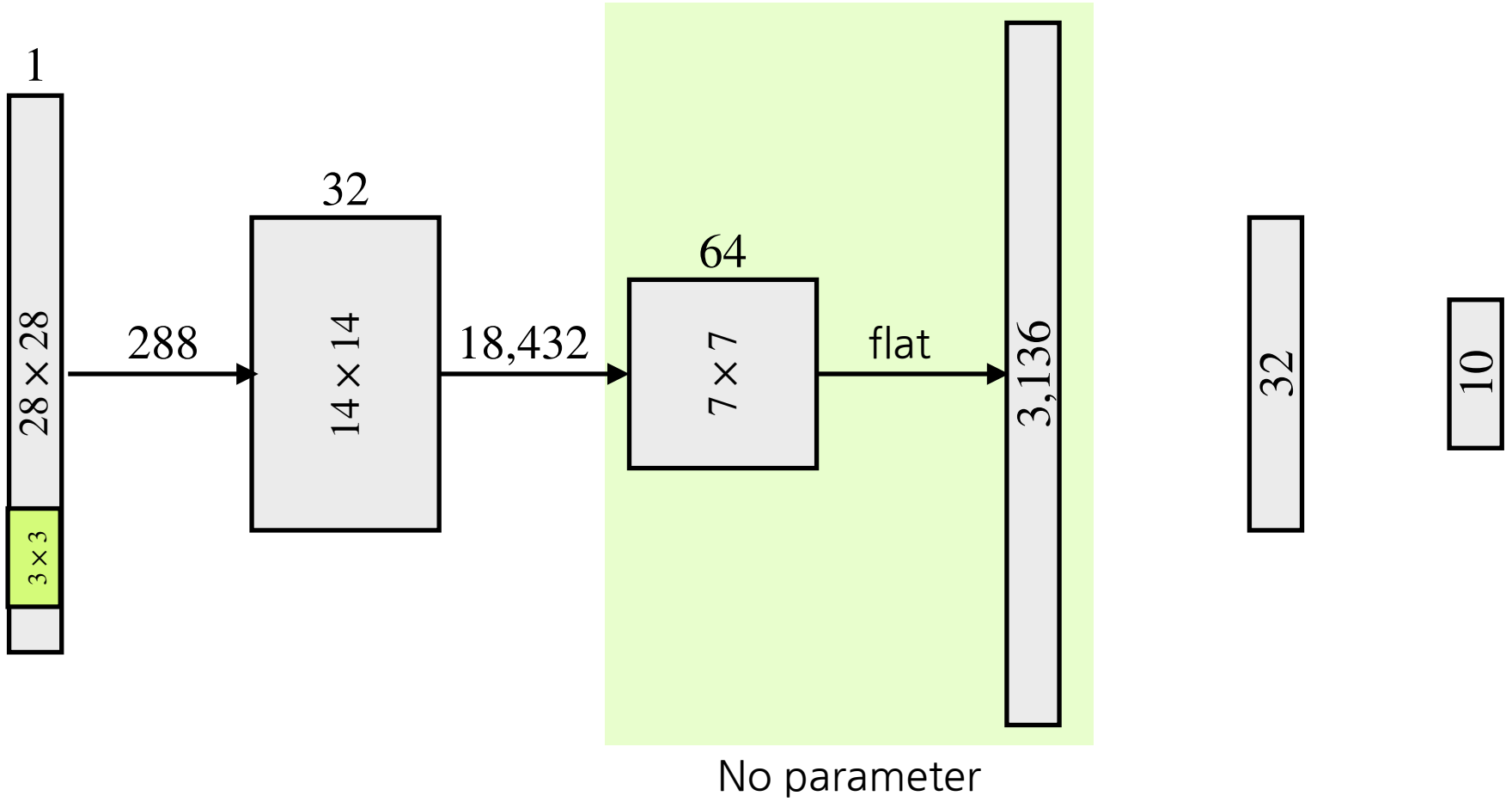
Convolutional Neural Network



Convolutional Neural Network

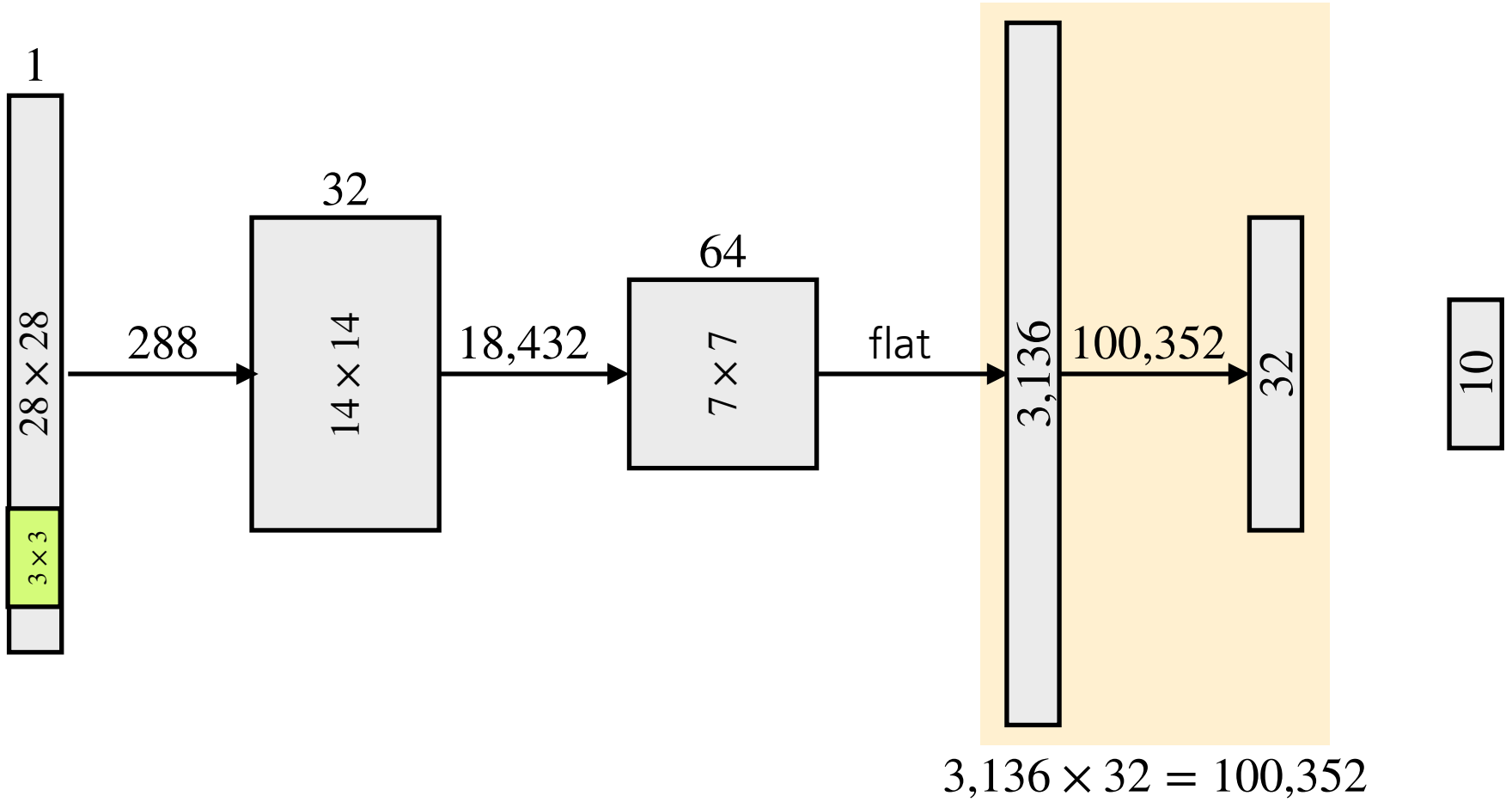


Convolutional Neural Network

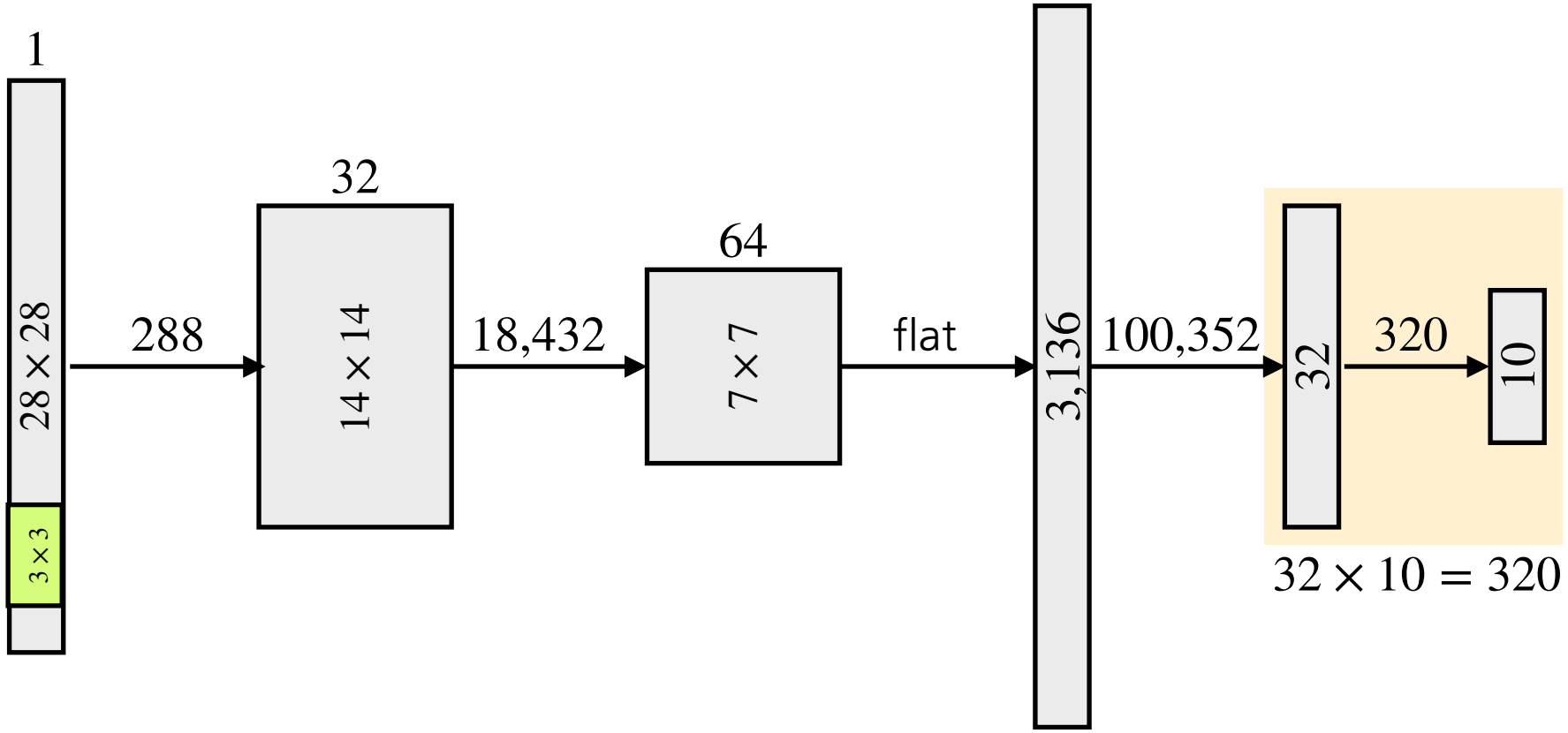


No parameter

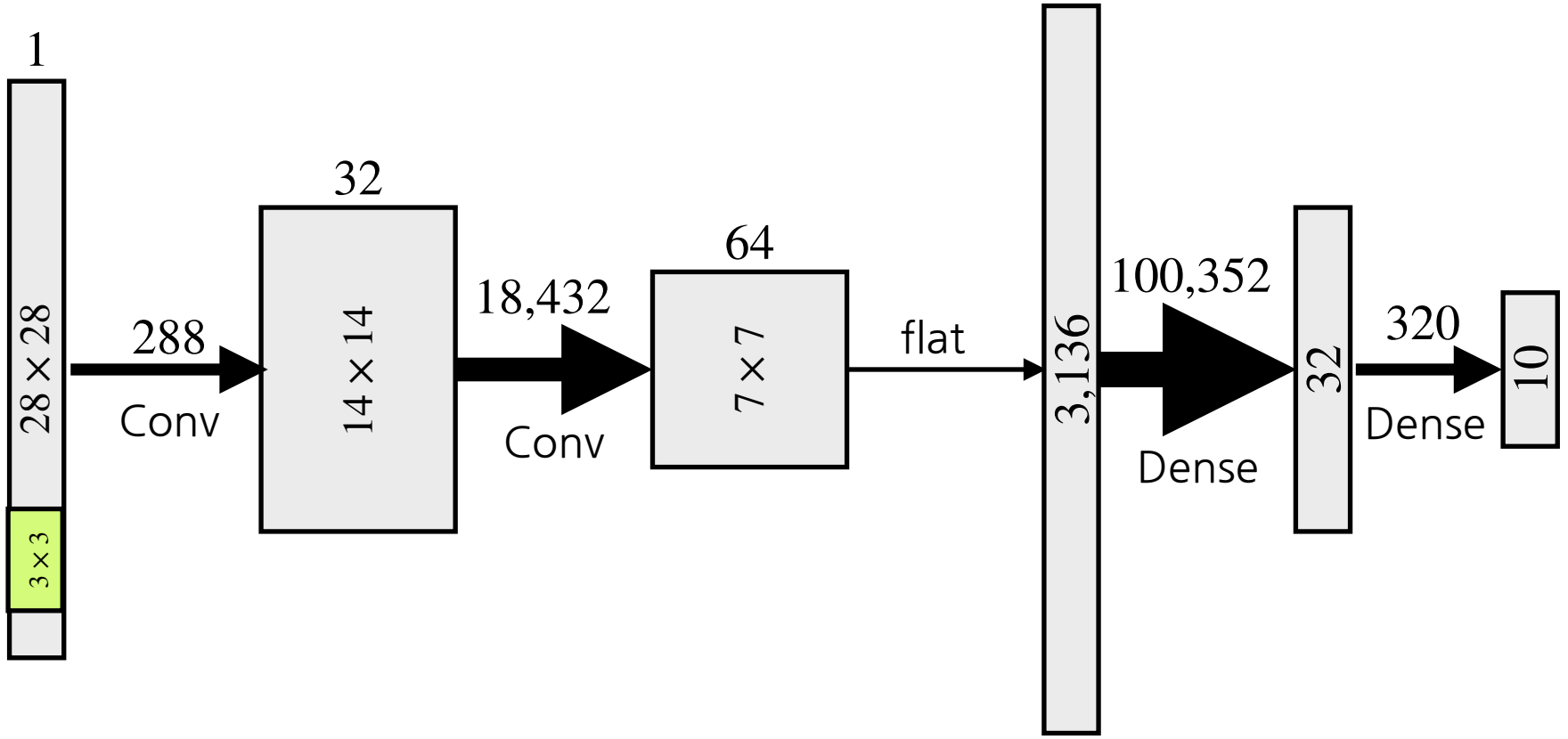
Convolutional Neural Network



Convolutional Neural Network



Convolutional Neural Network



PyTorch Implementation

ROBOT INTELLIGENCE LAB

https://github.com/sjchoi86/yet-another-pytorch-tutorial-v2

ROBOT INTELLIGENCE LAB

Modern CNNs

History

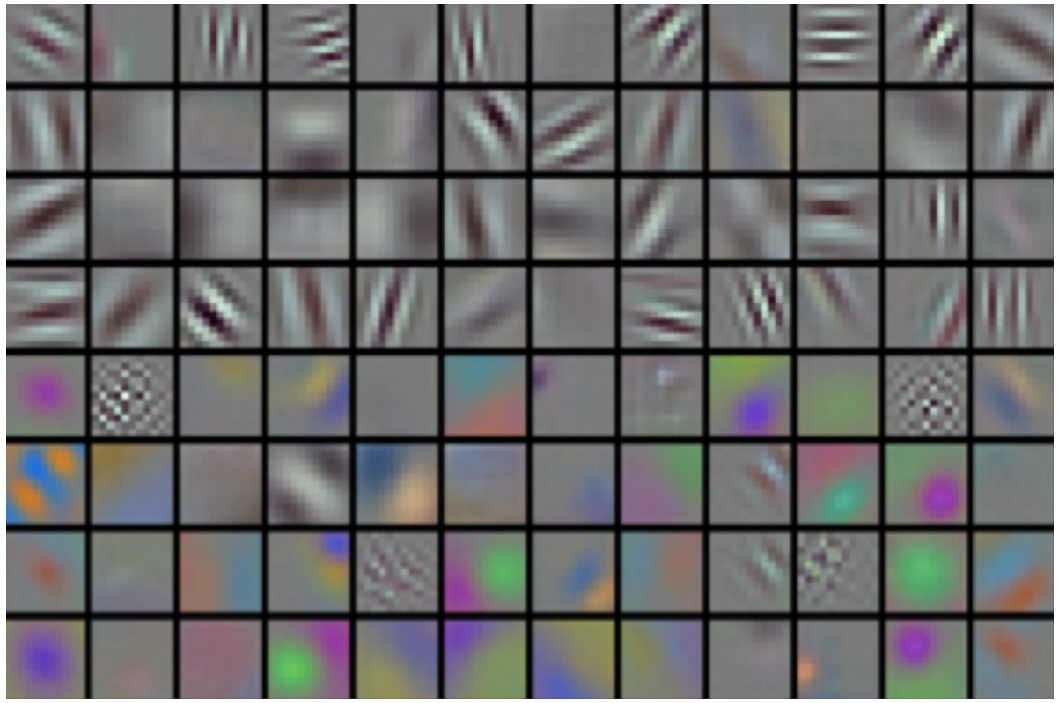
ROBOT INTELLIGENCE LAB

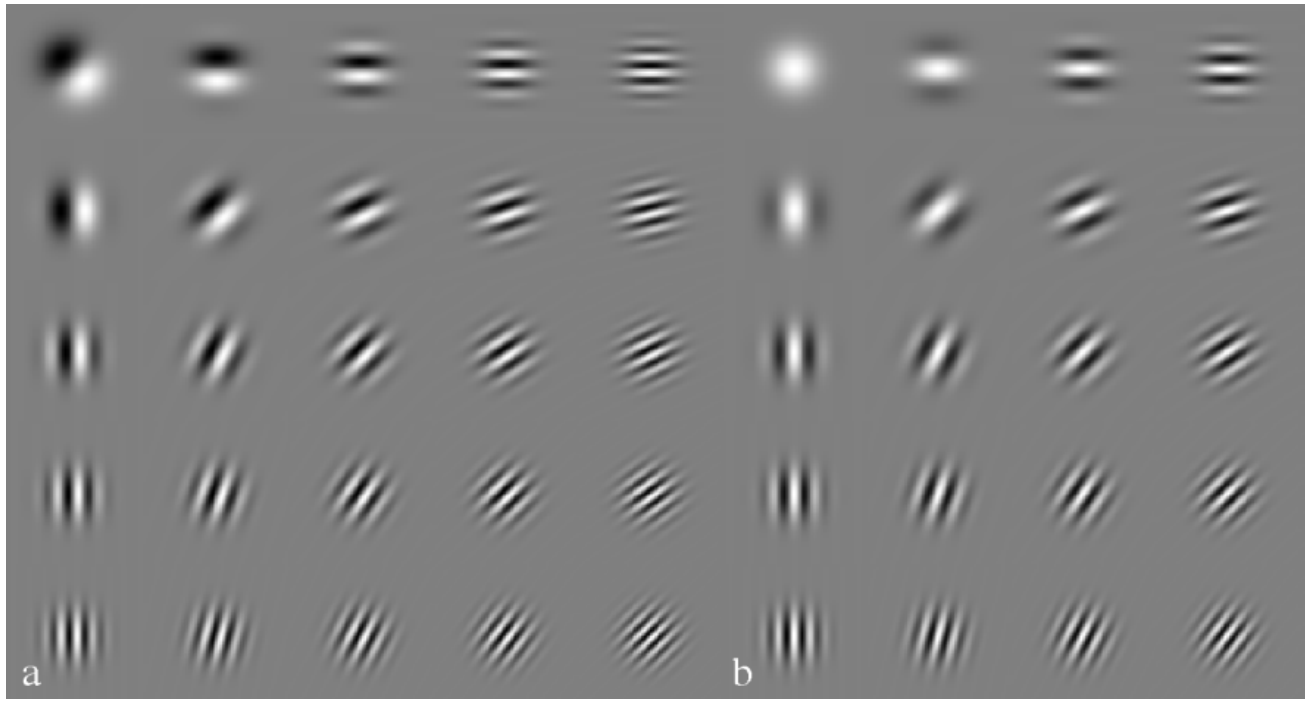
- The introduction of LeNet in 1995 introduced CNNs to computer vision and machine learning communities.
- However, CNNs did not immediately dominate the field until AlexNet in 2012.
- Back in the day, classical pipelines looked more like this:
  - Obtain an interesting dataset.
    - For instance, the Apple QuickTake 100 of 1994 supported 640x480 resolution (VGA), capable of storing up to 8 images, for the price of $1,000.
  - Preprocess the dataset with hand-crafted features based on some knowledge of optics and geometry and occasionally on the serendipitous discoveries of lucky graduate students.
    - SIFT (scale-invariant feature transform) or SURF (speeded-up robust features)
  - Dump the resulting representation into your favorite classifier
    - A linear model or kernel method

Apple QuickTake 100

Representation Learning

ROBOT INTELLIGENCE LAB





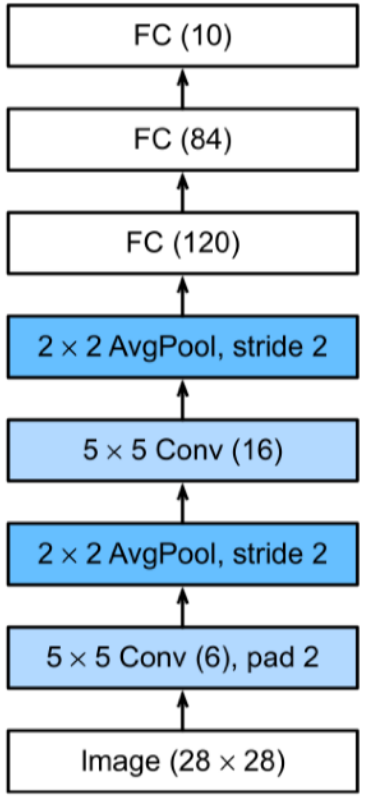
- The left figure shows image filters learned by the first layer of AlexNet.
- Interestingly, in the lowest layers of the network, the model learned feature extractors that resembled some traditional filters, as shown in the right figure.

ROBOT INTELLIGENCE LAB

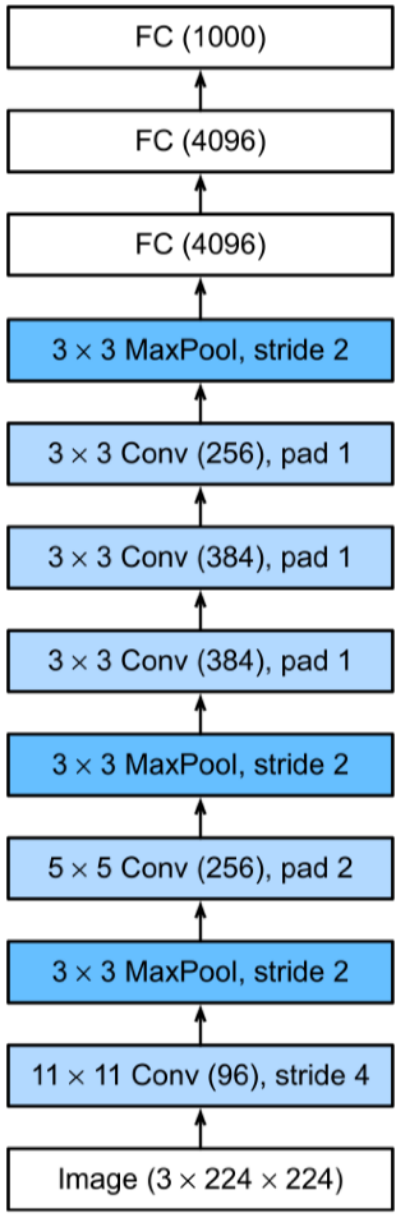
AlexNet

"ImageNet Classification with Deep Convolutional Neural Networks," 2012

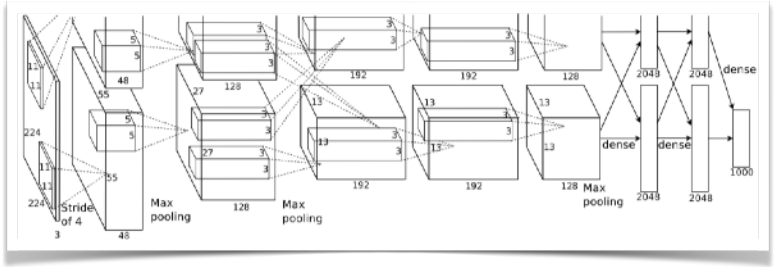
Architecture



LeNet



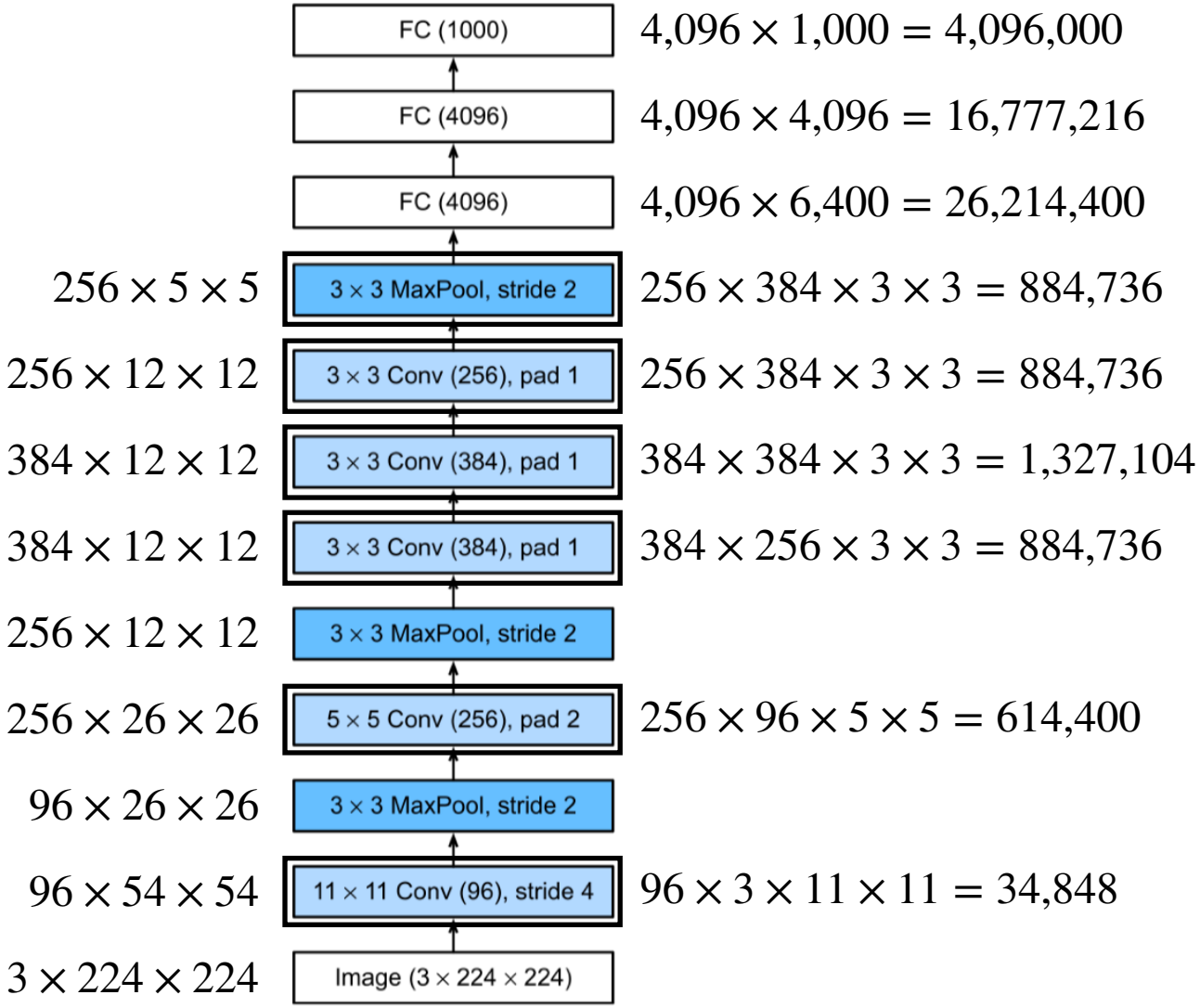
AlexNet



(Original) AlexNet

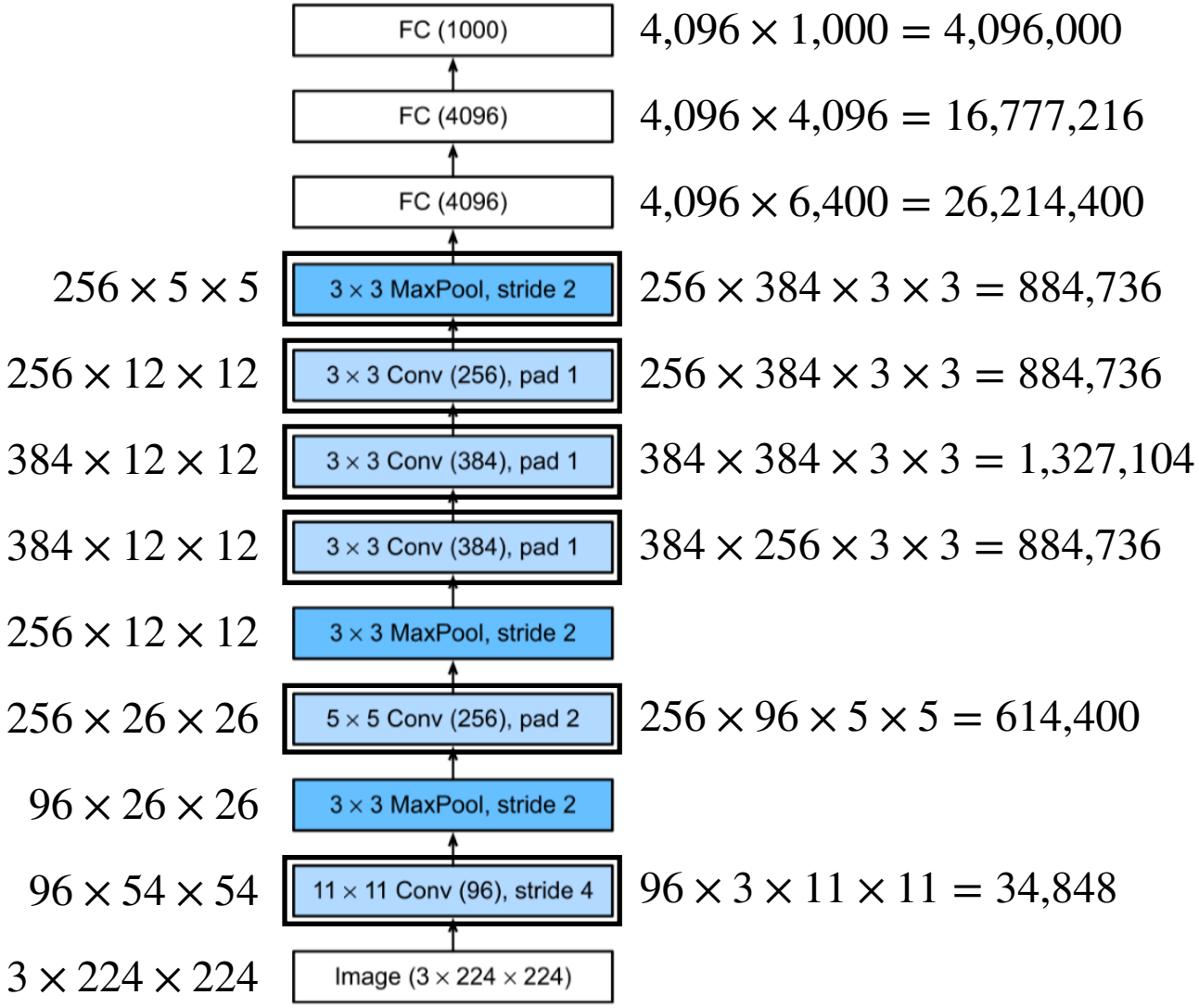
Architecture

- Kernel: [out_channels x in_channels x kernel_height x kernel_width]



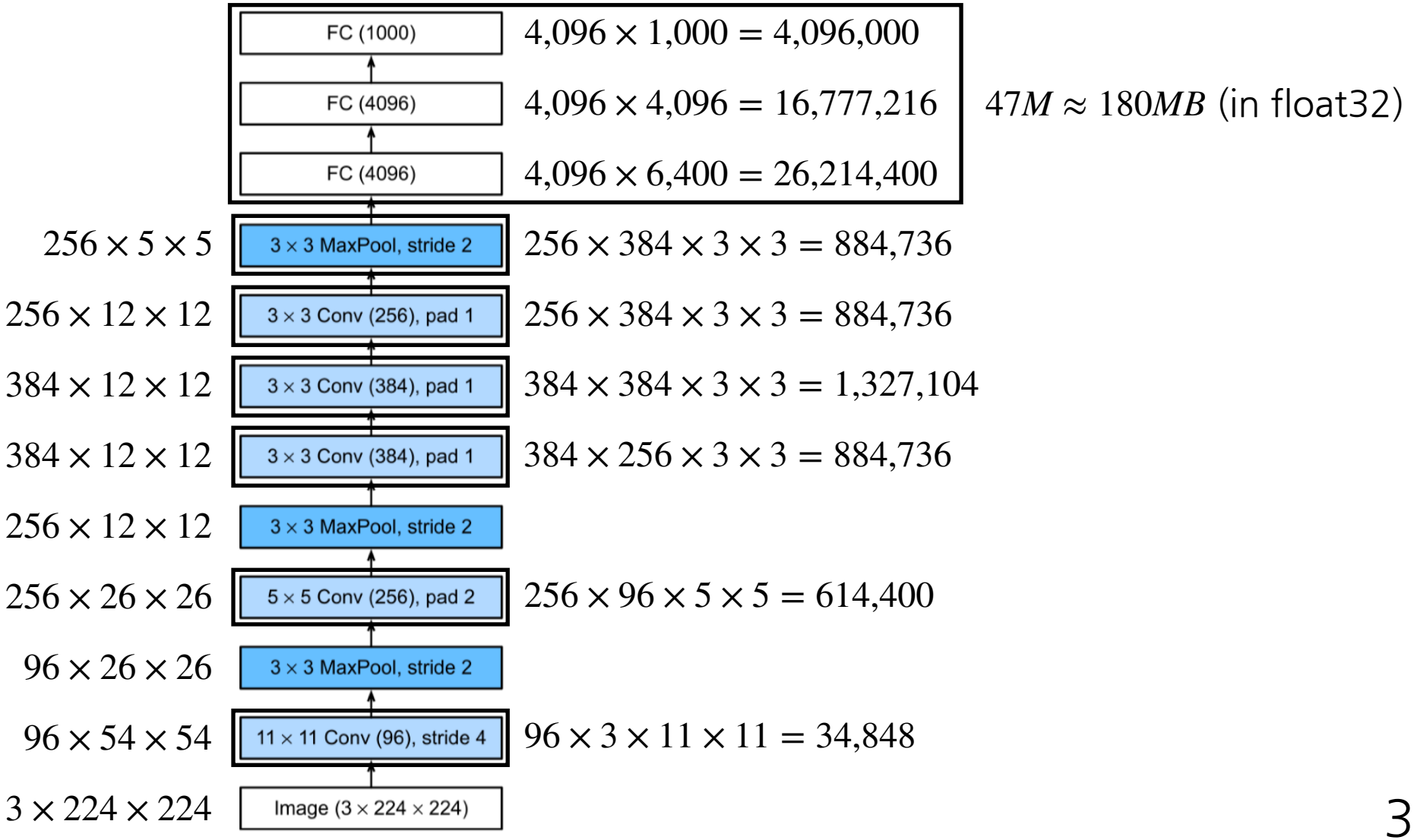
Architecture

- Kernel: [out_channels x in_channels x kernel_height x kernel_width]



Architecture

- Kernel: [out_channels x in_channels x kernel_height x kernel_width]



Implementation Details

ROBOT INTELLIGENCE LAB

- For activation functions, the ReLU activation function is used.
- Data augmentation is based on perturbing RGB pixel values.
- Dropout is used.
- The amount of parameters to be trained (40 million) by far exceeds the amount of training data.
    - However, there is hardly any overfitting.. why?

ROBOT INTELLIGENCE LAB

VGG

"Very Deep Convolutional Networks for Large-Scale Image Recognition," 2015

VGG

ROBOT INTELLIGENCE LAB

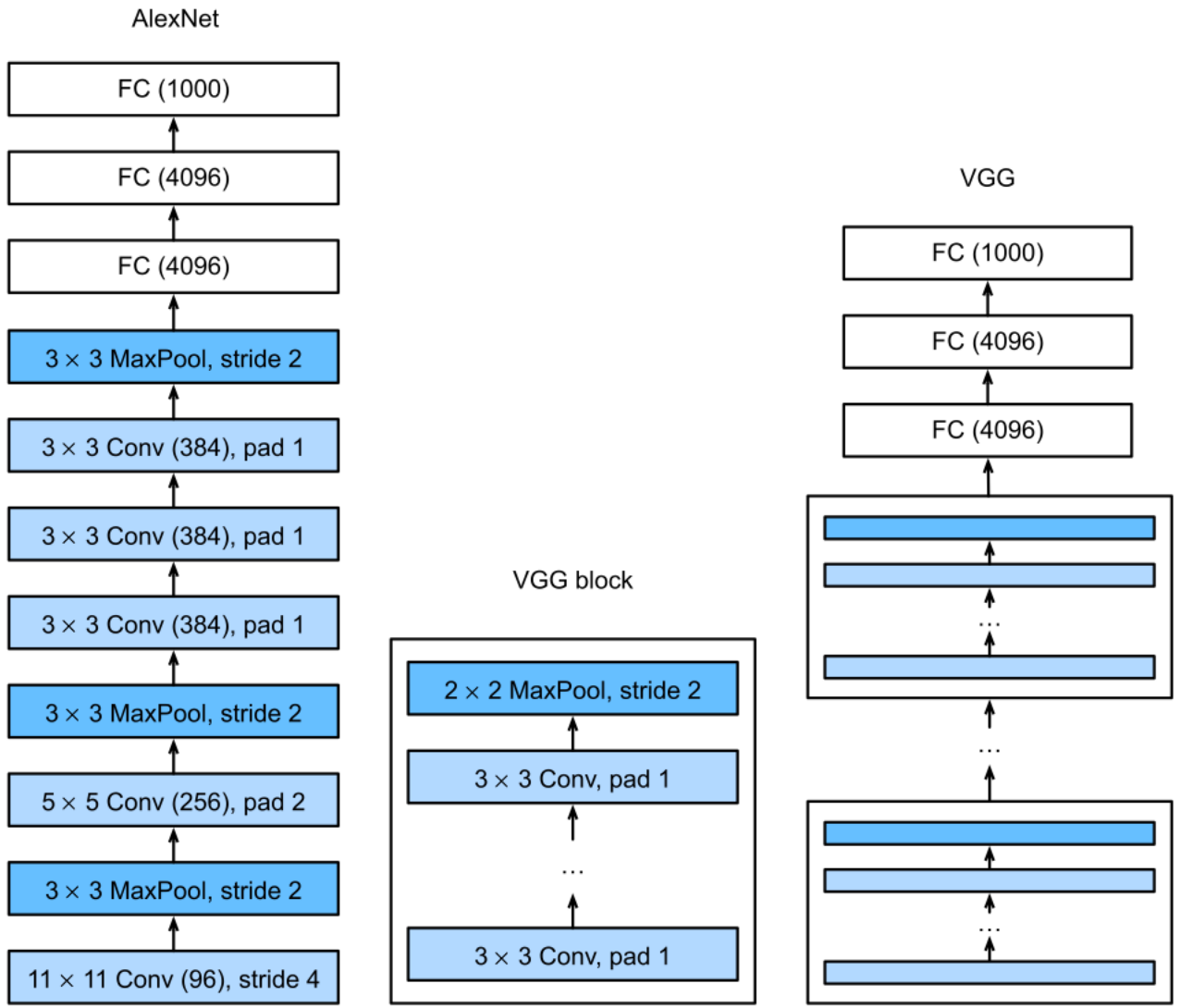
Visual **G**eometry **G**roup (VGG) at Oxford University

VGG Blocks

ROBOT INTELLIGENCE LAB

- Basic building blocks
  - a convolutional layer with padding to maintain the resolution
  - a nonlinearity such as a ReLU
  - a pooling layer such as max-pooling to reduce the resolution
- The key idea of VGG was to use multiple convolutions in between downsampling via max-pooling in the form of a block.

VGG Networks



Contribution

ROBOT INTELLIGENCE LAB

- While AlexNet introduced many of the components of what makes deep learning effective at scale, it is VGG that arguably introduced key properties such as blocks of multiple convolutions and a preference for deep and narrow networks.
- It is also the first network that is actually an entire family of similarly parametrized models, giving the practitioner ample trade-off between complexity and speed.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

ROBOT INTELLIGENCE LAB

GoogLeNet

"Going Deeper with Convolutions," 2015

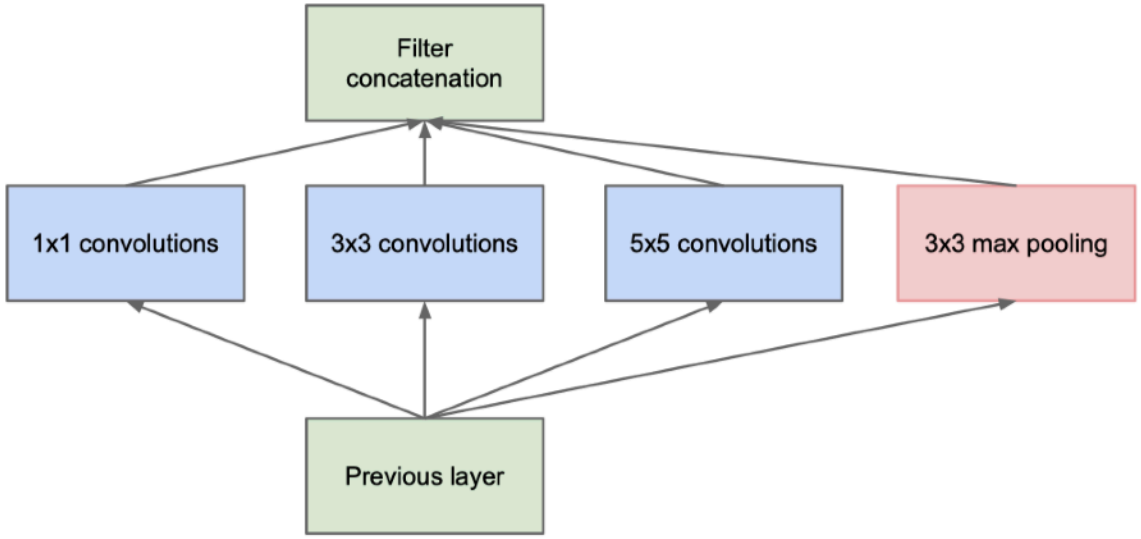

THAT'S NOT ENOUGH
WE HAVE TO GO DEEPER

Multi-Branch Networks

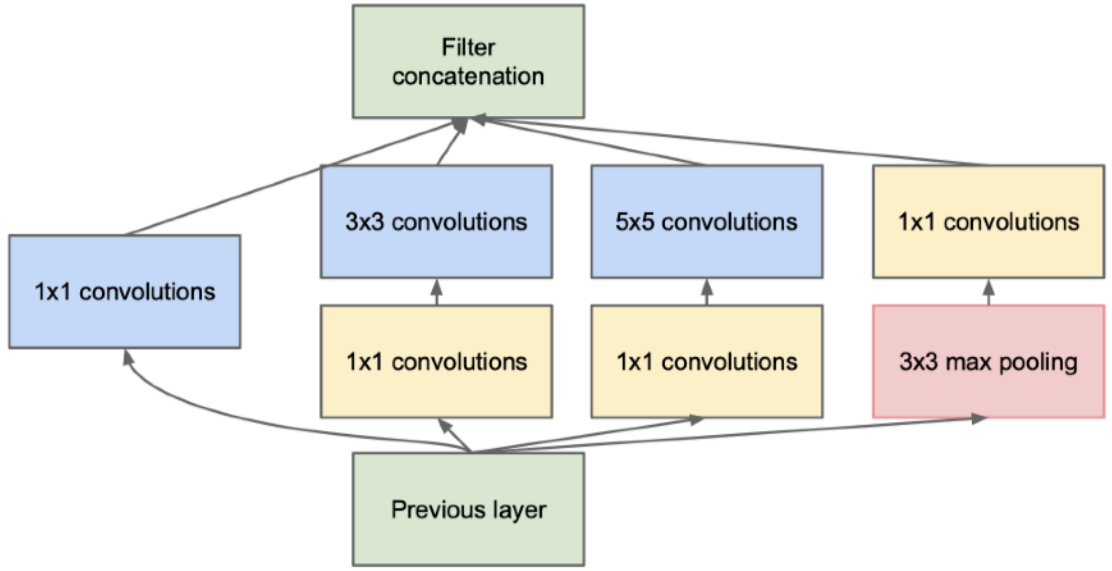
ROBOT INTELLIGENCE LAB

- In 2014, GooLeNet won the ImageNet Challenge.
- It is arguably also the first network that exhibits a clear distinction among the stem (data ingest), body (data processing), and head (prediction) in a CNN.
  - The **stem** is given by the first 2-3 convolutions that operate on the image
  - This is followed by a **body** of convolutional blocks.
  - Finally, the **head** maps the feature obtained so far to the required problem (e.g., classification or detection).
- The key contribution in GoogLeNet was the design of the **network body**.

Inception Blocks

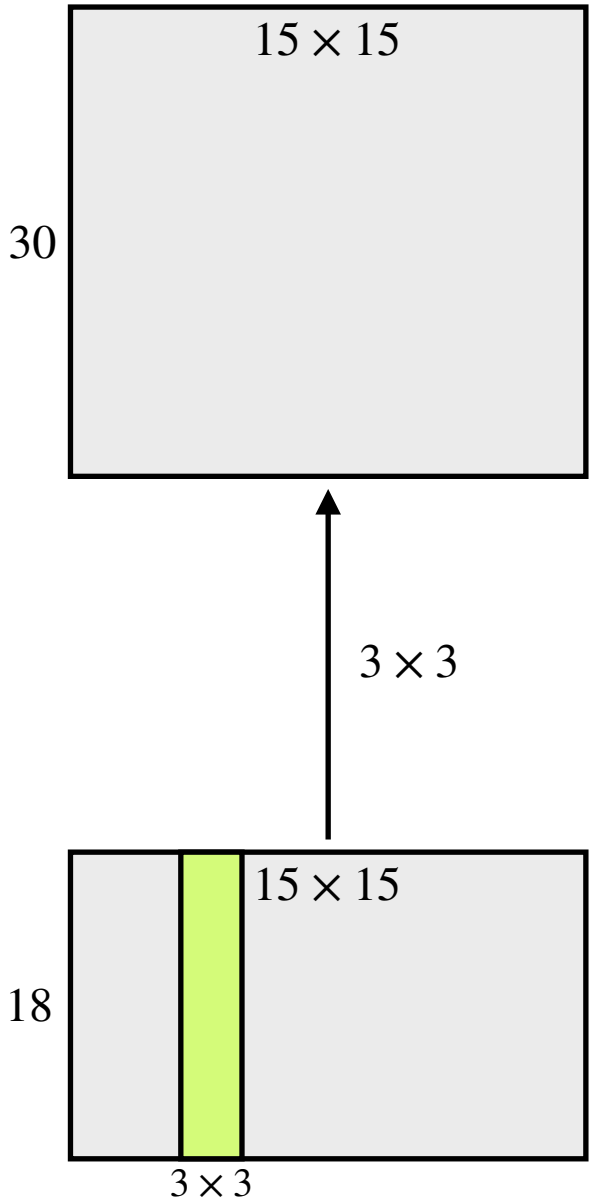


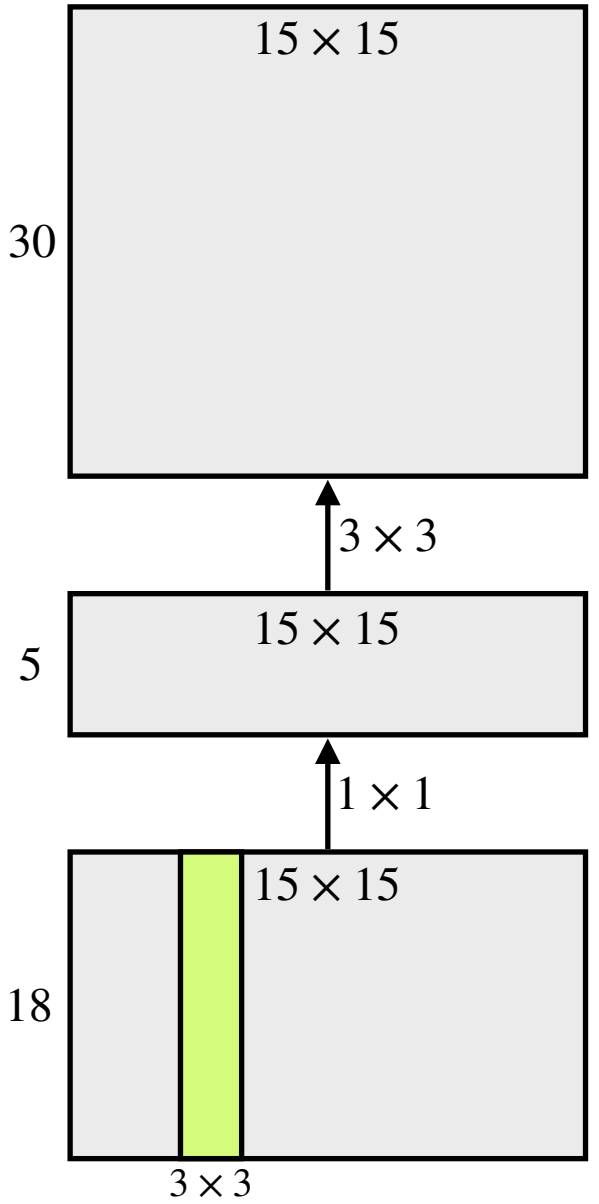
(a) Inception module, naïve version



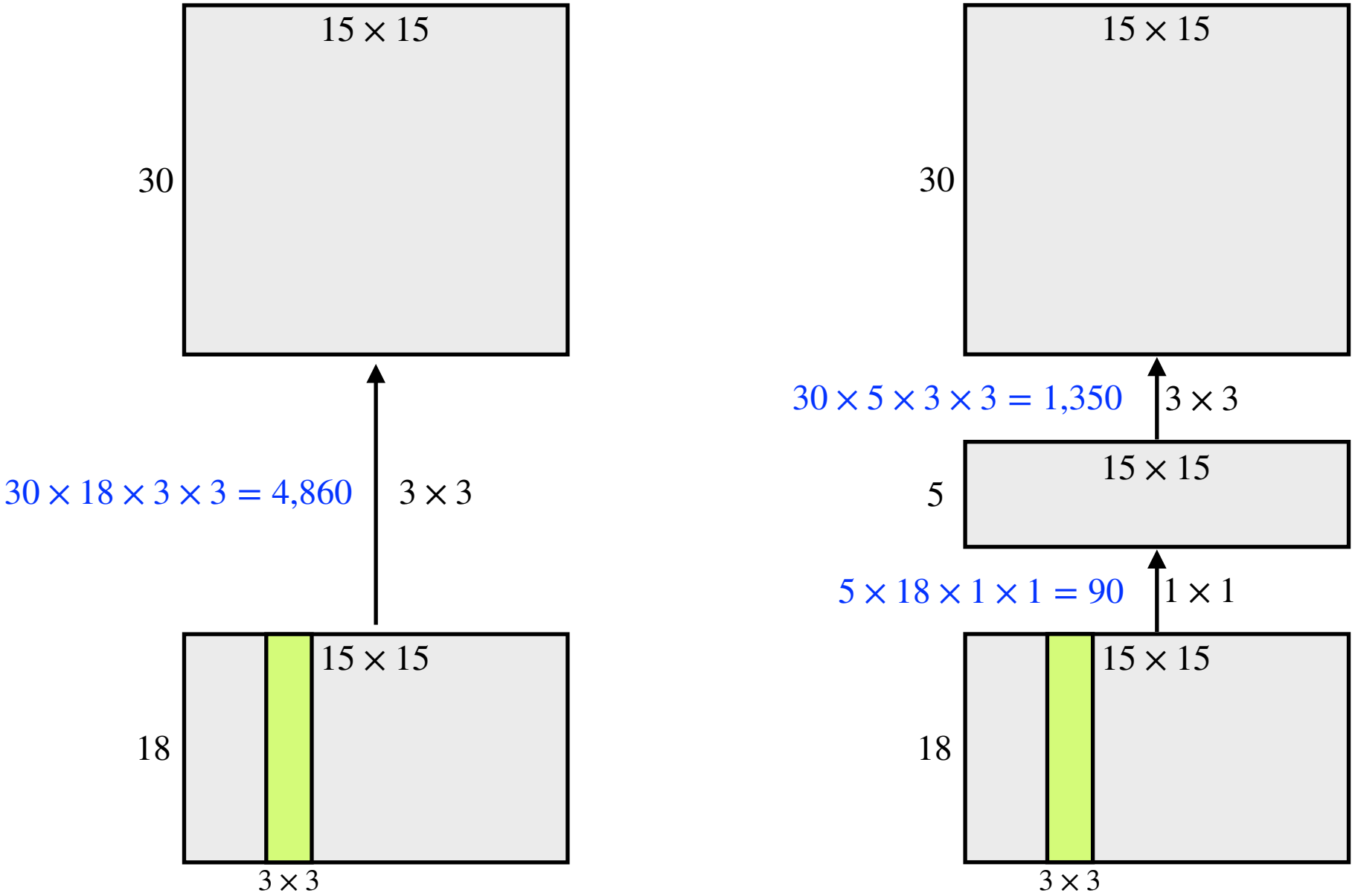
(b) Inception module with dimension reductions

Why adding 1x1 Convolution?

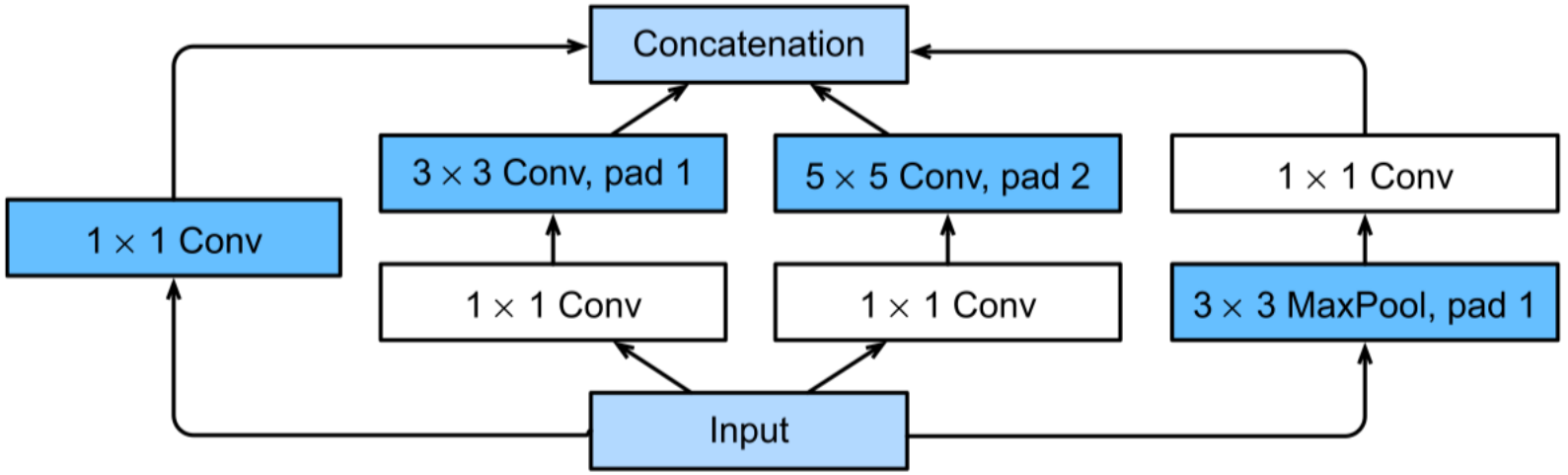




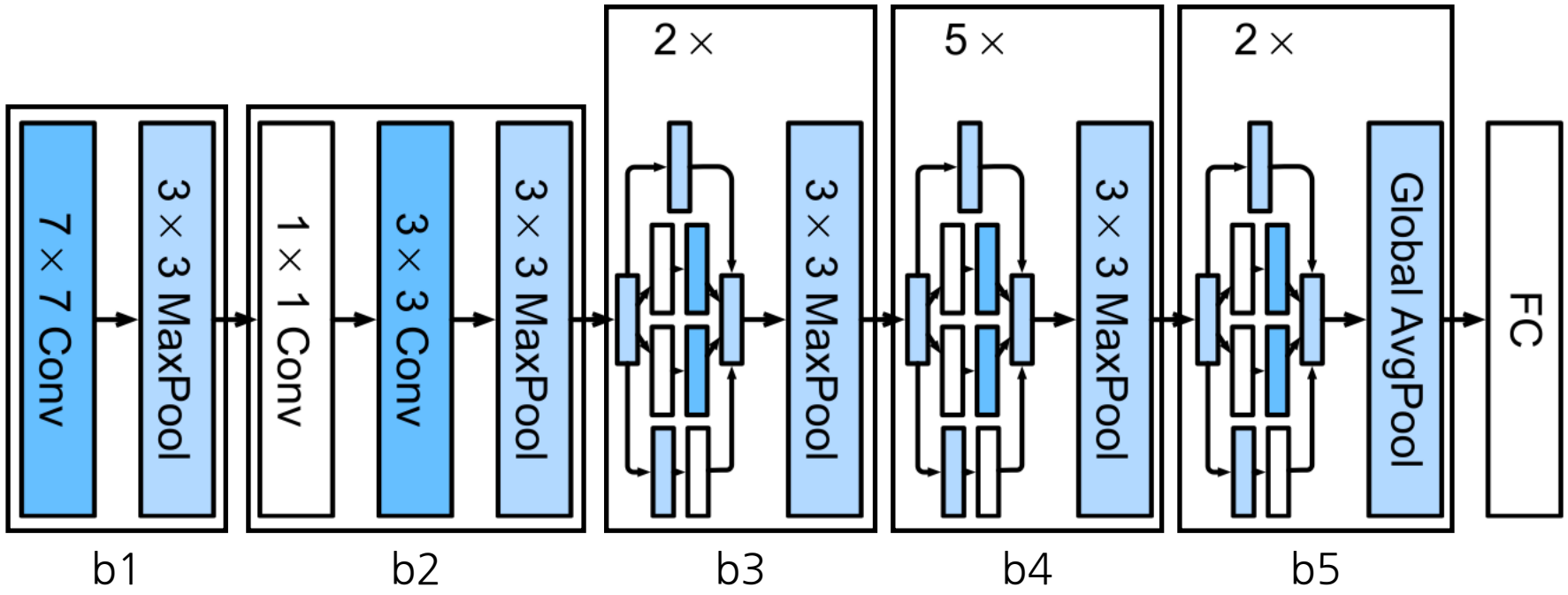
Why adding 1x1 Convolution?



Inception Blocks



Model


2 ×
5 ×
2 ×
7 × 7 Conv
3 × 3 MaxPool
1 × 1 Conv
3 × 3 Conv
3 × 3 MaxPool
3 × 3 MaxPool
3 × 3 MaxPool
Global AvgPool
FC
b1
b2
b3
b4
b5

ROBOT INTELLIGENCE LAB

ResNet

"Deep Residual Learning for Image Recognition," 2016

A degradation problem

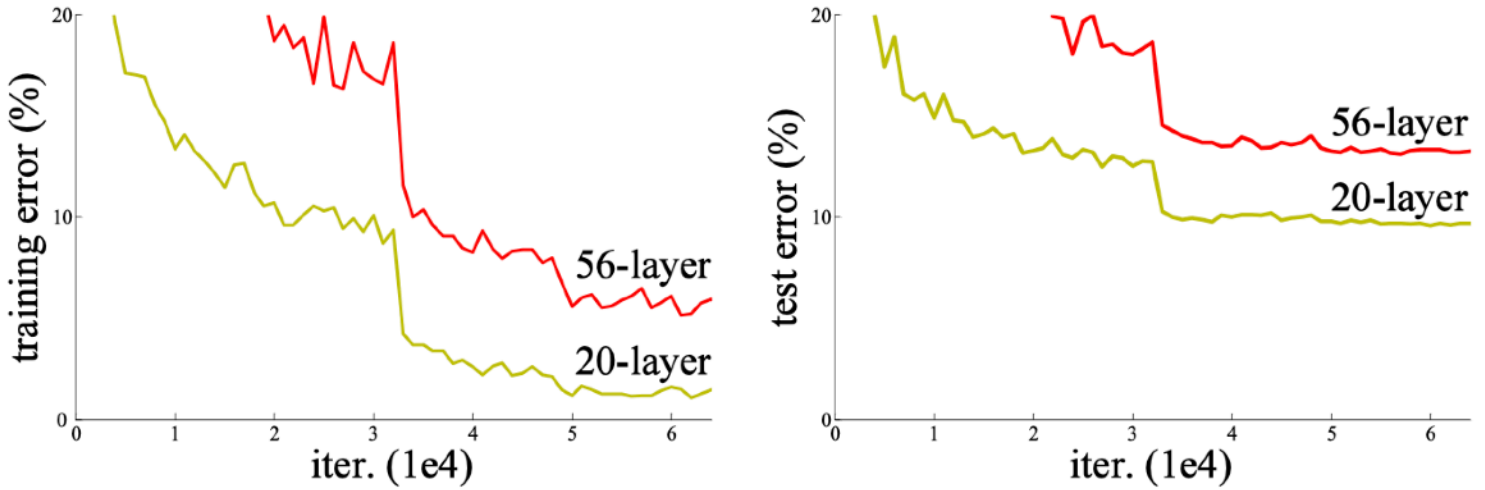


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

- The **degradation** is not caused by overfitting, and adding more layers leads to higher training errors (and also higher test errors).

ResNet

ROBOT INTELLIGENCE LAB

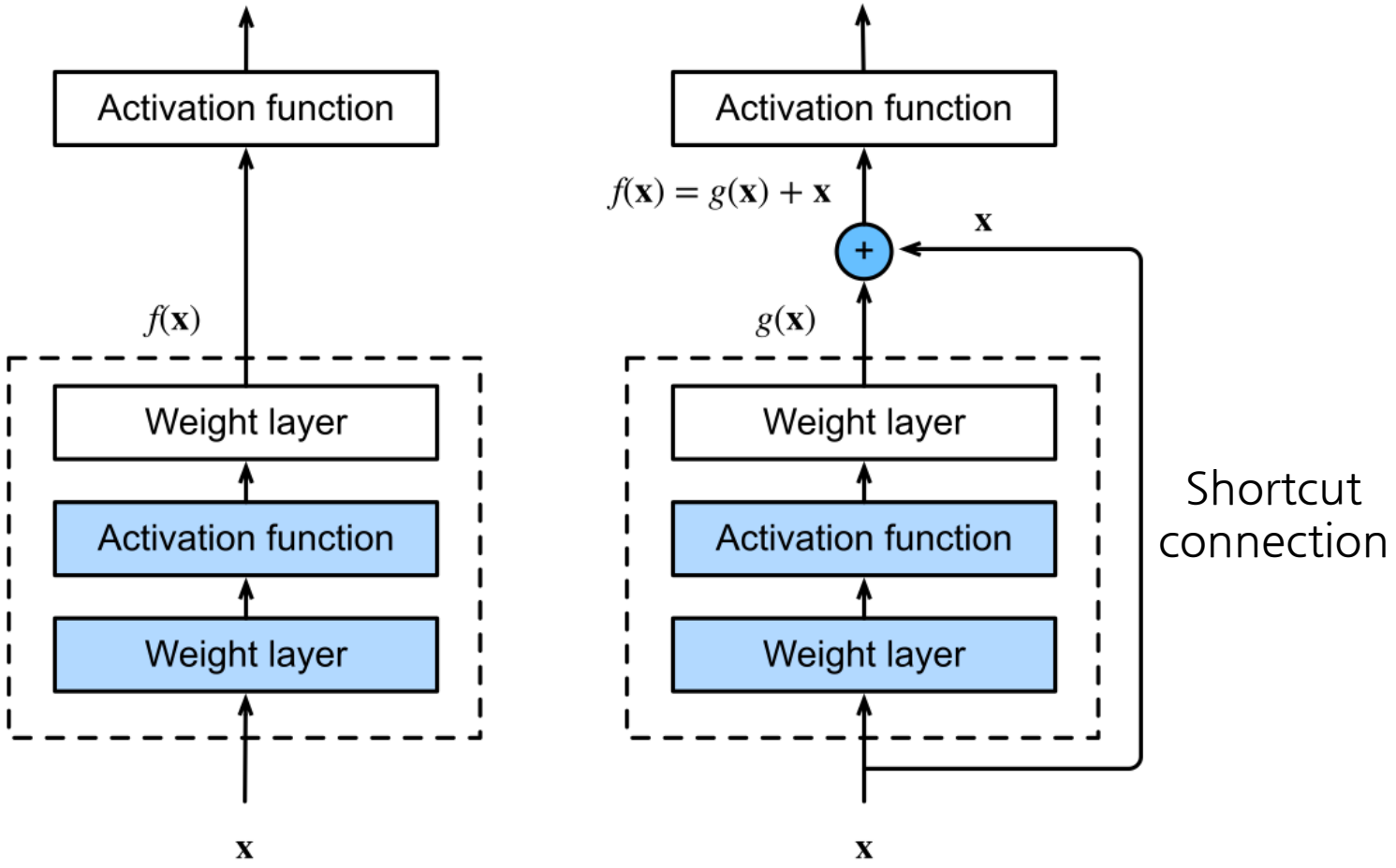
- ResNet won the ILSVRC in 2015.
- The key idea is that every additional layer should contain the identity function as one of its elements.

ResNet

ROBOT INTELLIGENCE LAB

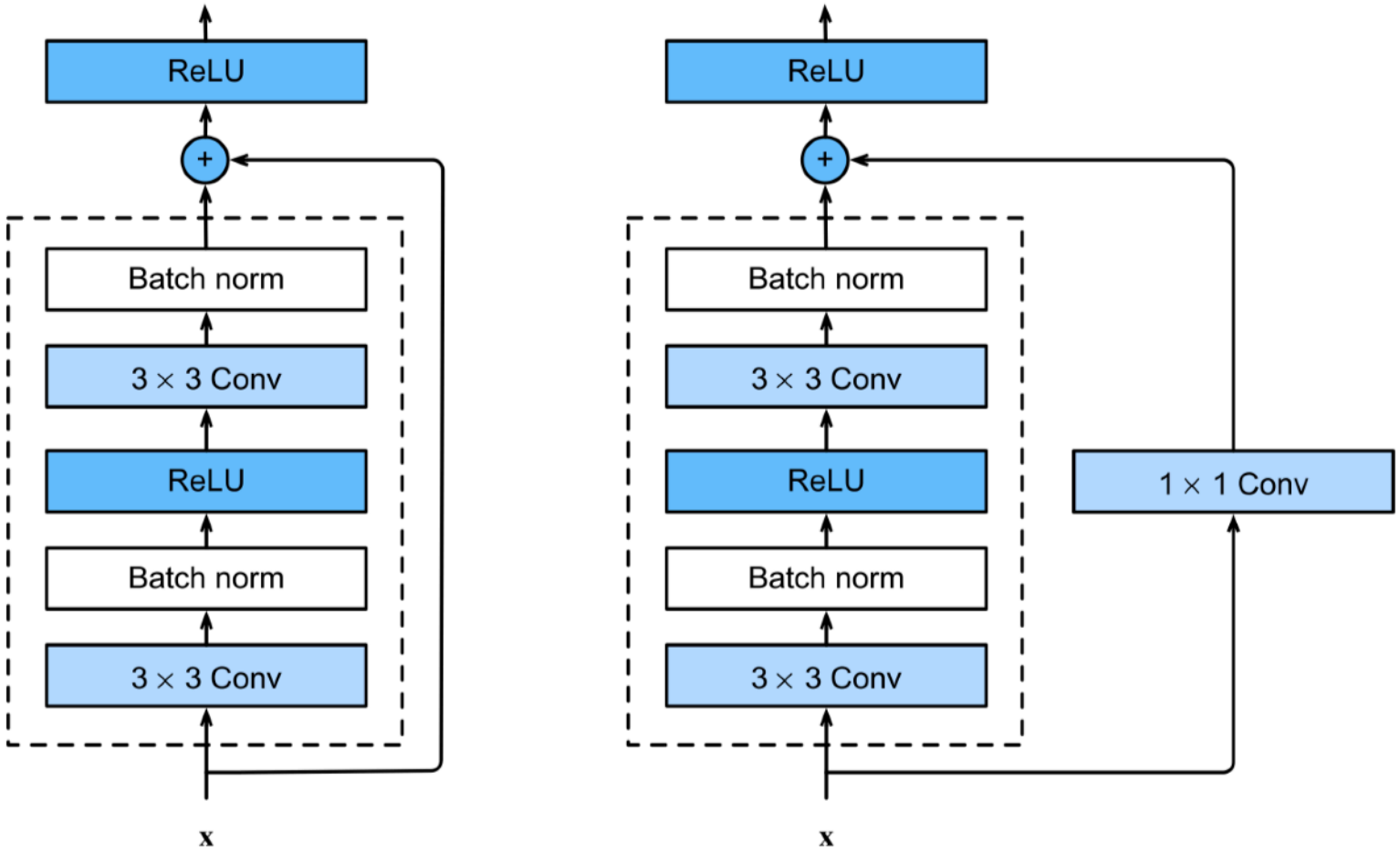
We hypothesize that it is easier to optimize the residual mapping than to optimize the original mapping.

Residual Blocks



Shortcut connection

Residual Blocks



- The order of operations matters.

Bottleneck Blocks

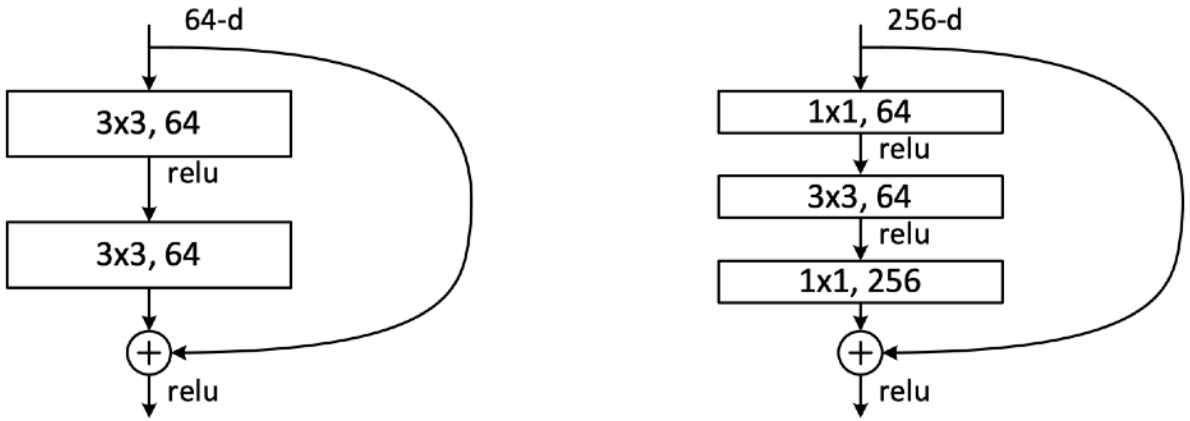
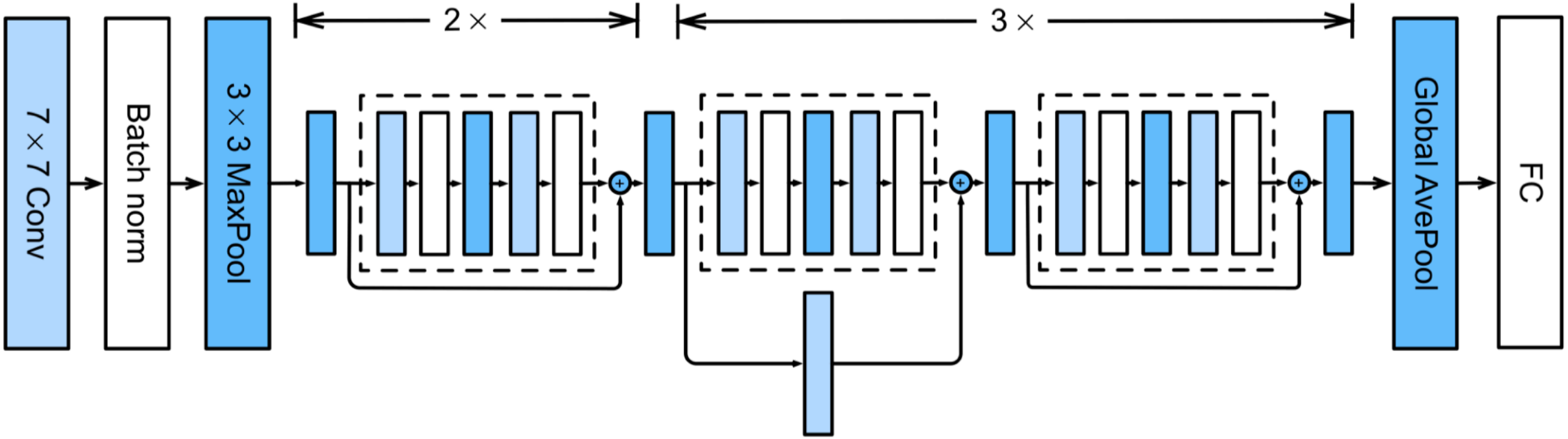


Figure 5. A deeper residual function $\mathcal{F}$ for ImageNet. Left: a building block (on 56×56 feature maps) as in Fig. 3 for ResNet-34. Right: a "bottleneck" building block for ResNet-50/101/152.

ResNet-18



ImageNet Results

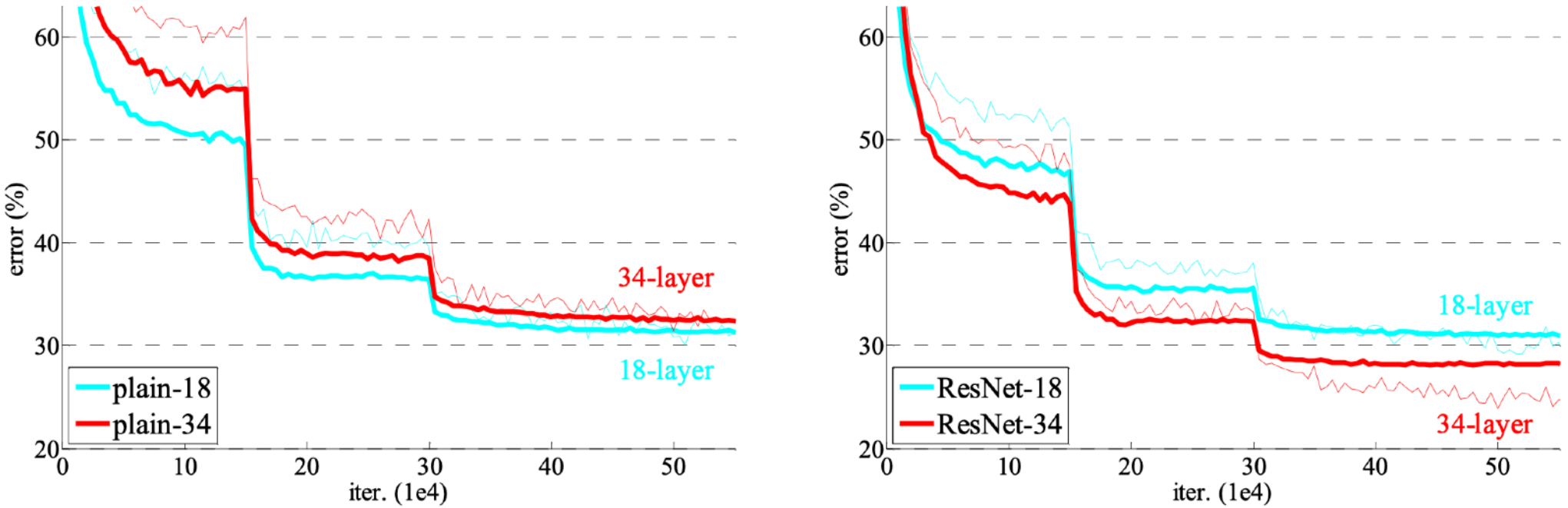


Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

CIFAR-10 Results

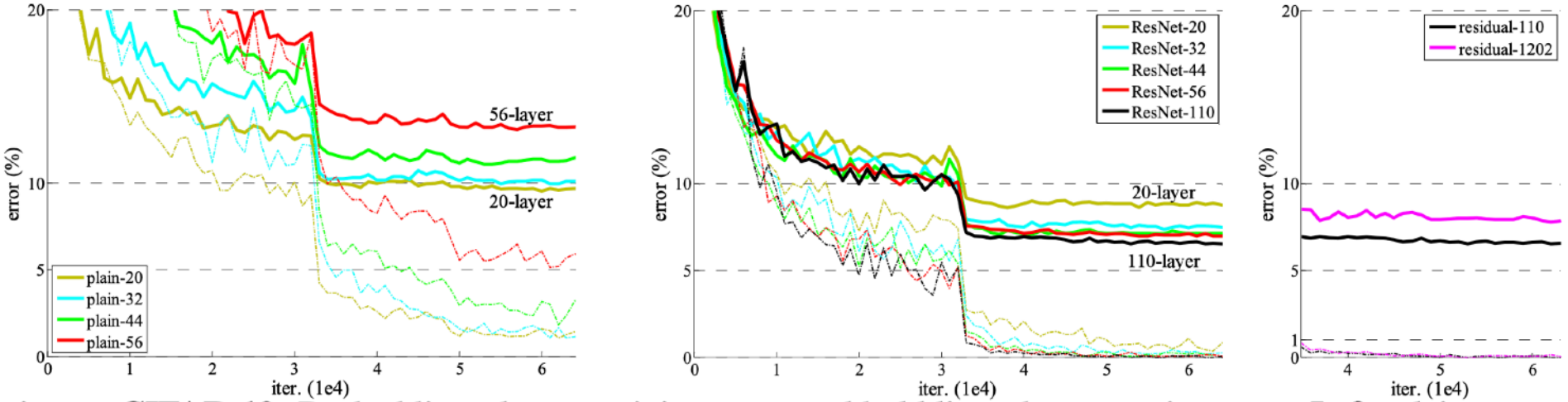


Figure 6. Training on **CIFAR-10**. Dashed lines denote training error, and bold lines denote testing error. **Left**: plain networks. The error of plain-110 is higher than 60% and not displayed. **Middle**: ResNets. **Right**: ResNets with 110 and 1202 layers.


ROBOT INTELLIGENCE LAB