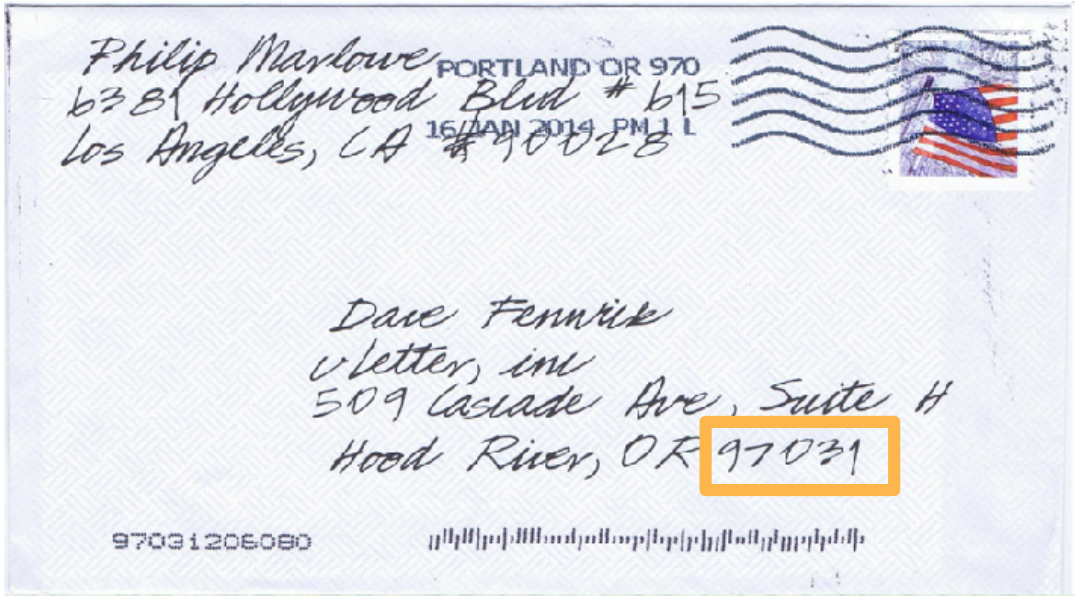
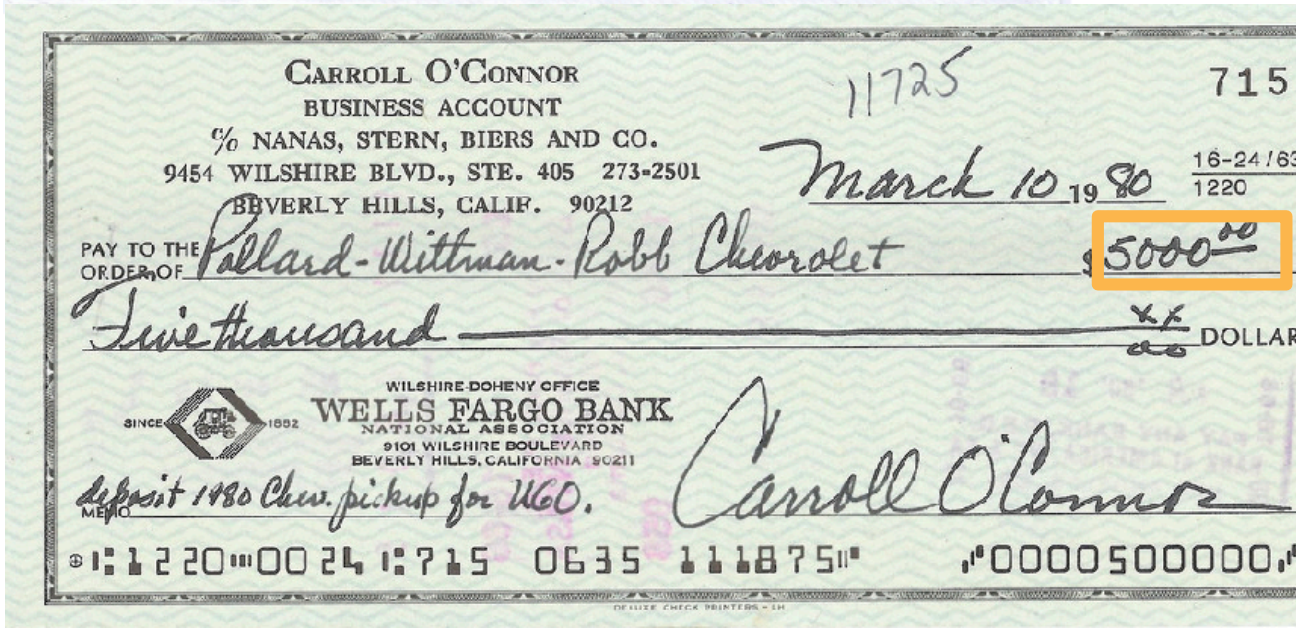
Handwritten Digit Recognition



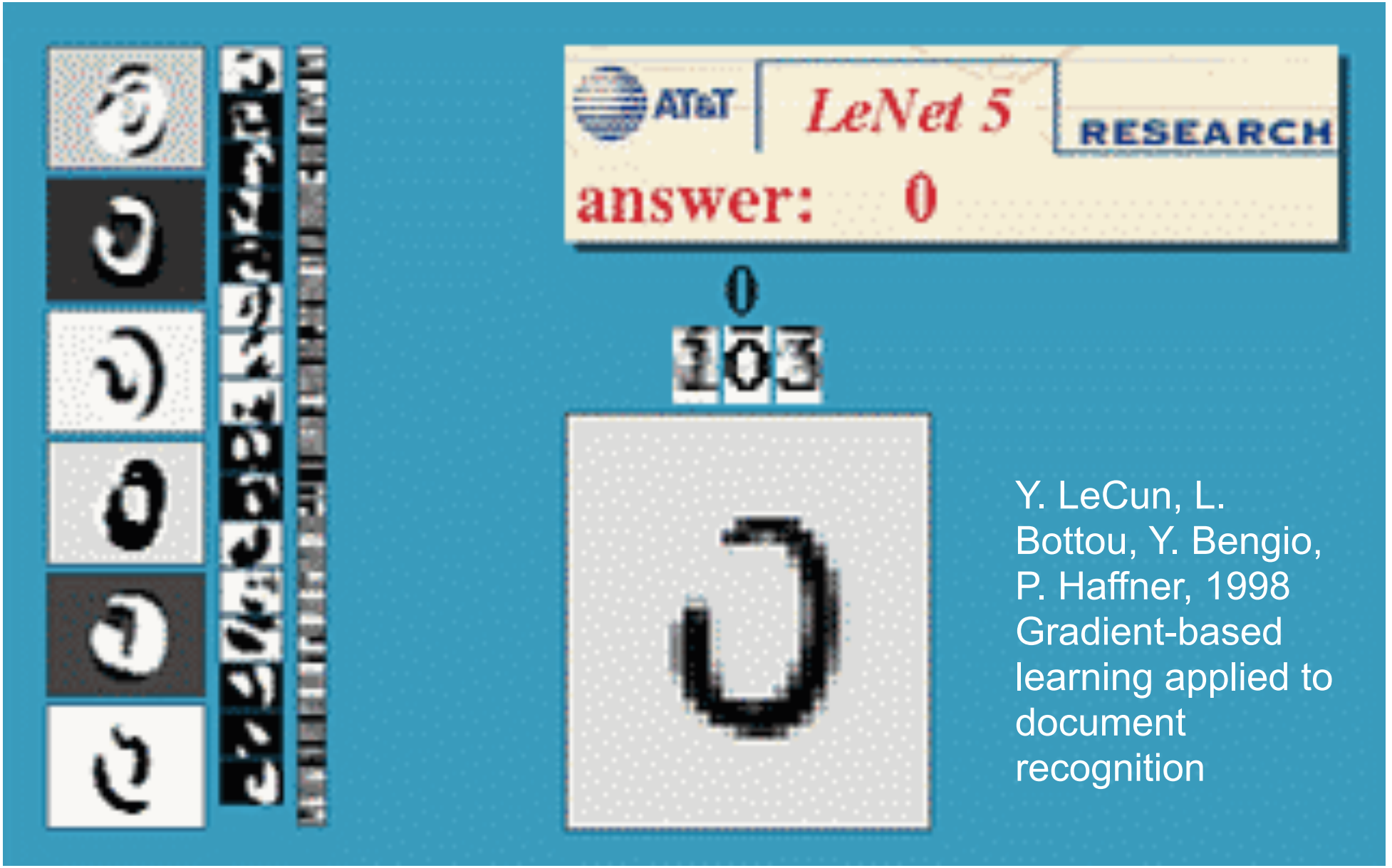


adapted from courses.d2l.ai/berkeley-stat-157

MNIST

- Centered and scaled
- 50,000 training data
- 10,000 test data
- 28 x 28 images
- 10 classes

adapted from courses.d2l.ai/berkeley-stat-157



Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, 1998 Gradient-based learning applied to document recognition

Summary

- Convolutional layer
  - Reduced model capacity compared to dense layer
  - Efficient at detecting spatial pattens
  - High computation complexity
  - Control output shape via padding, strides and channels
- Max/Average Pooling layer
  - Provides some degree of invariance to translation

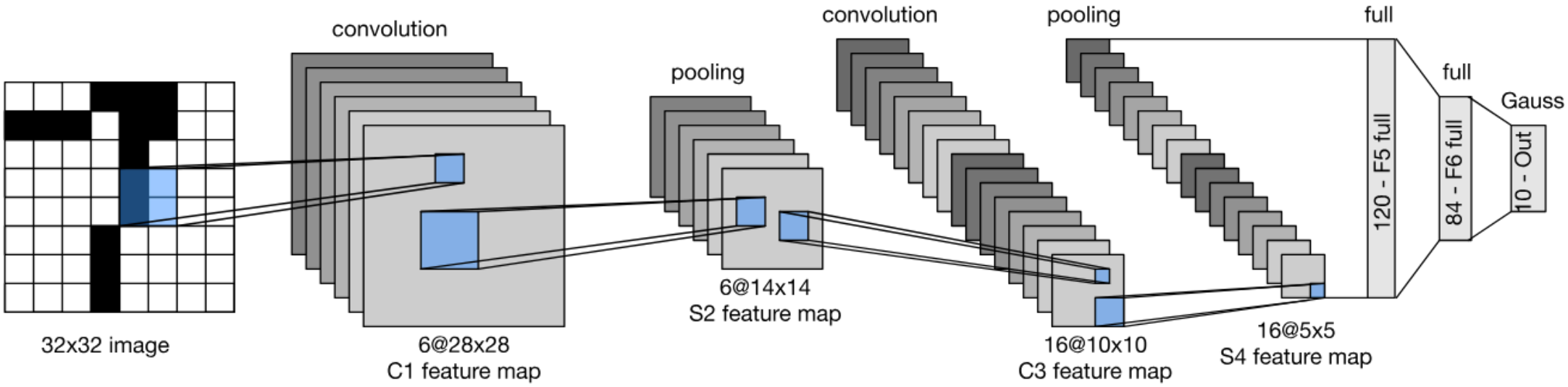
adapted from courses.d2l.ai/berkeley-stat-157

Introduction to (Deep) Learning

The trend/revolution LeNet, AlexNet, VGG (and NiN)

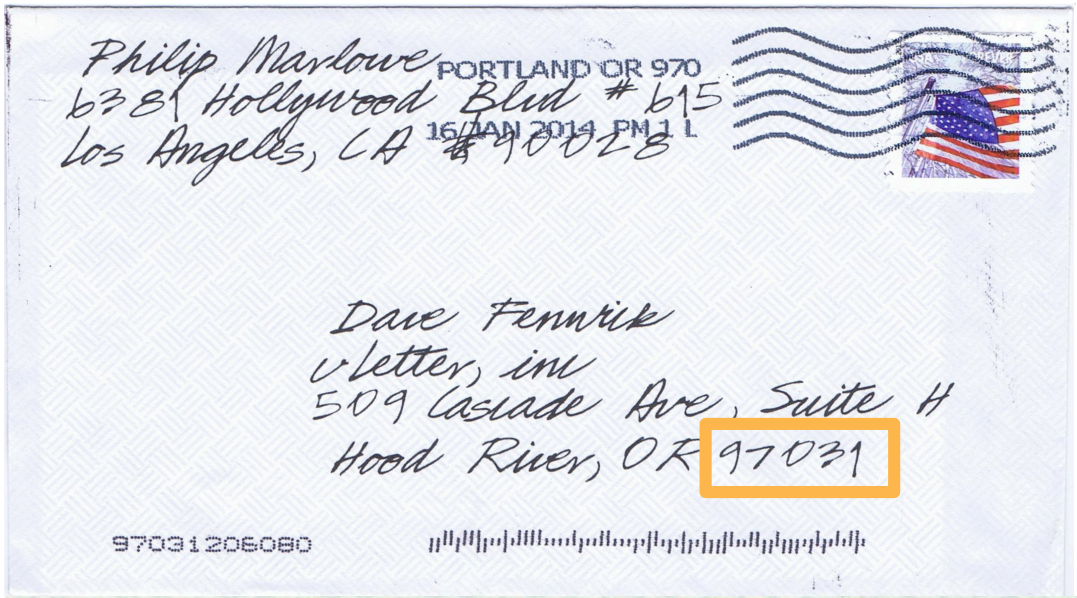
adapted from courses.d2l.ai/berkeley-stat-157

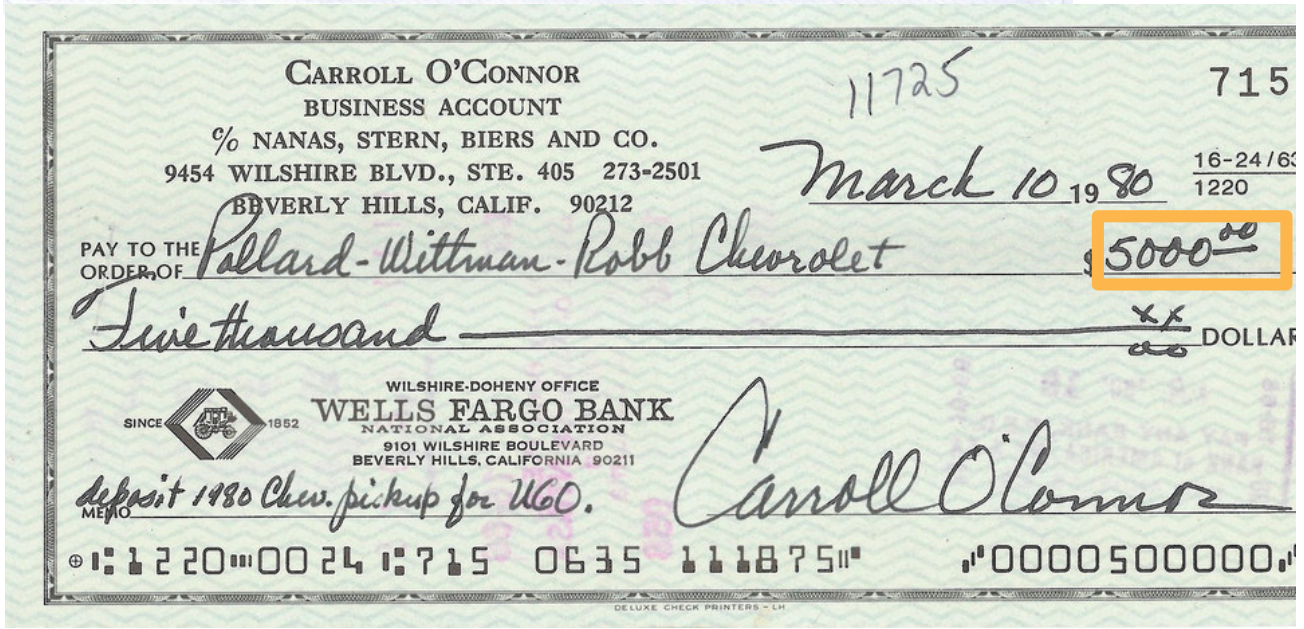
LeNet Architecture



adapted from courses.d2l.ai/berkeley-stat-157

Handwritten Digit Recognition



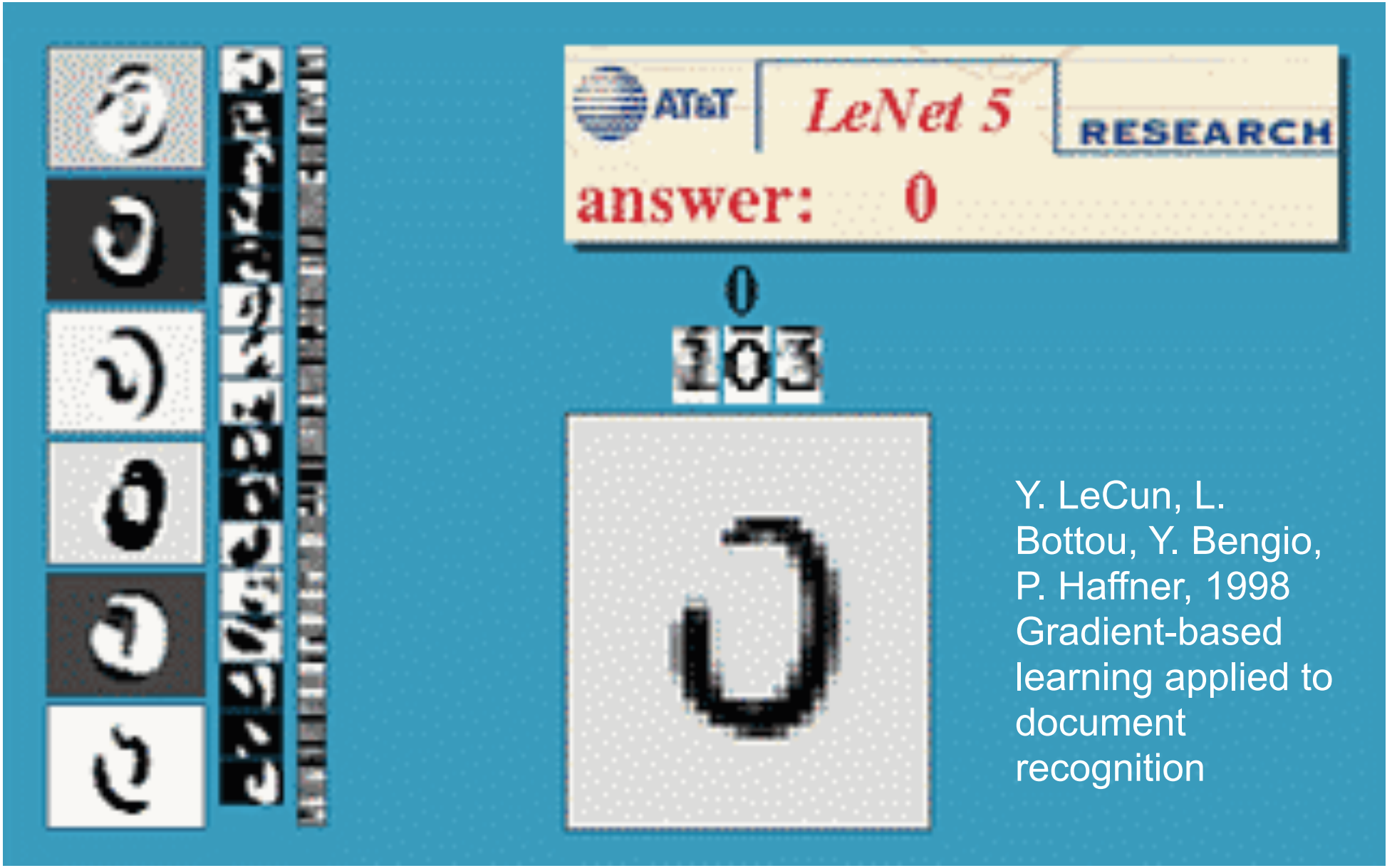


adapted from courses.d2l.ai/berkeley-stat-157

MNIST

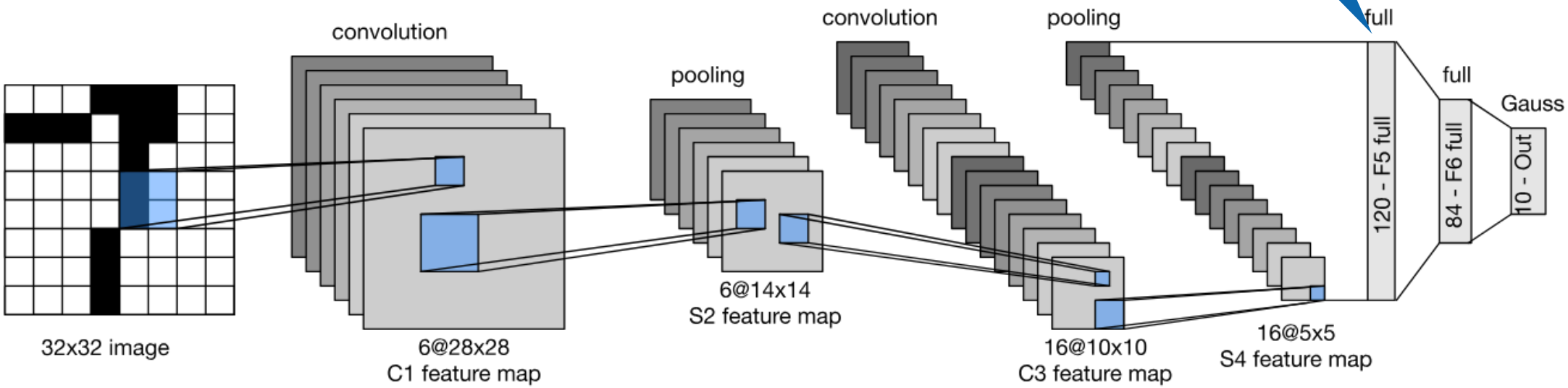
- Centered and scaled
- 60,000 training data
- 10,000 test data
- 28 x 28 images
- 10 classes

adapted from courses.d2l.ai/berkeley-stat-157



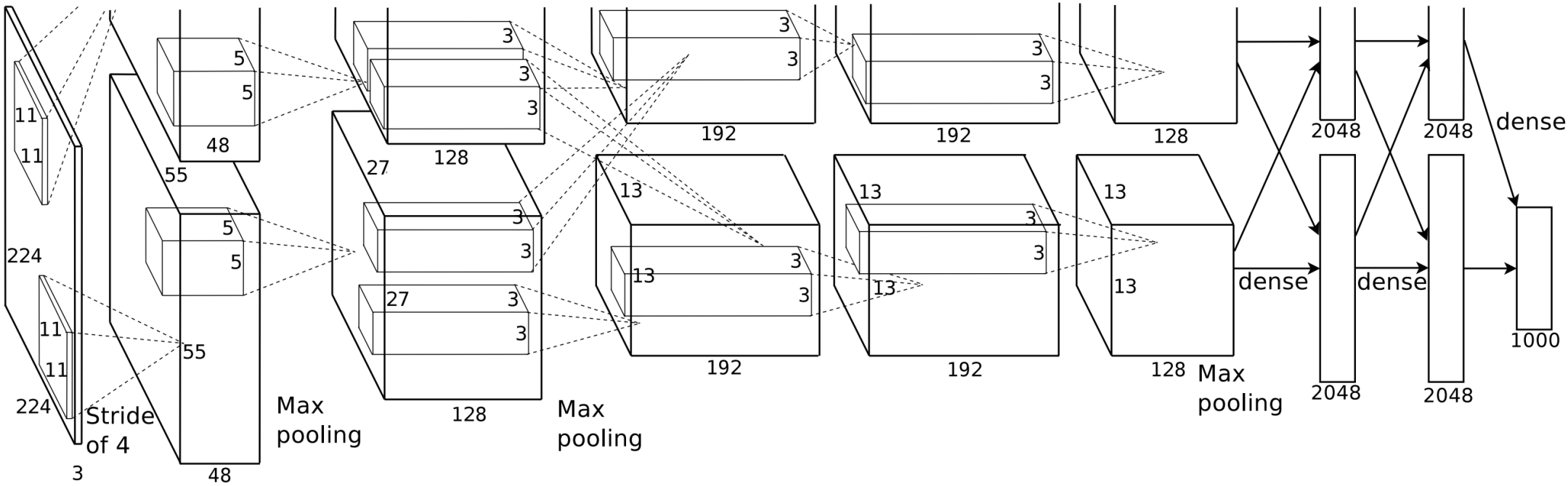
Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, 1998 Gradient-based learning applied to document recognition

Expensive if we have many outputs

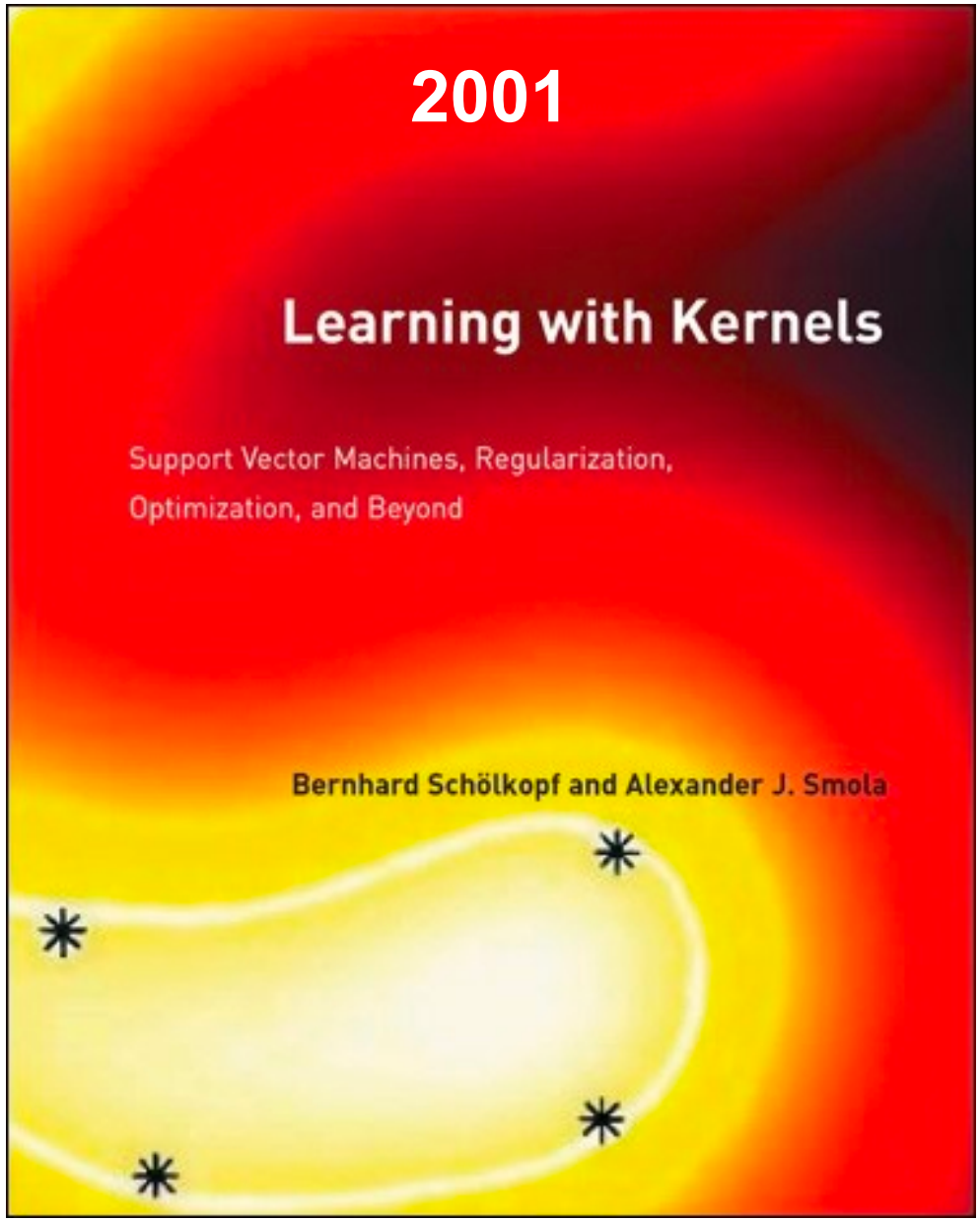


adapted from courses.d2l.ai/berkeley-stat-157

AlexNet



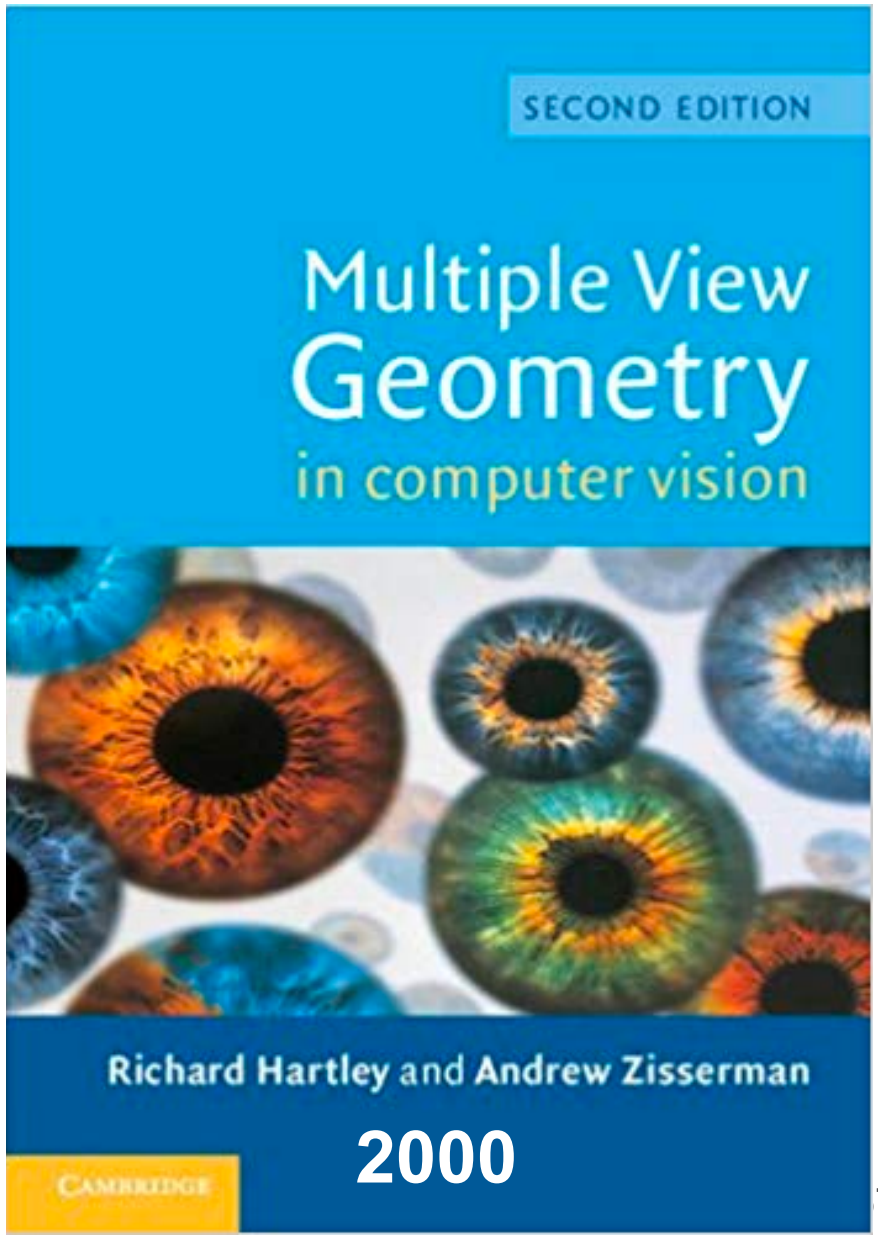
adapted from courses.d2l.ai/berkeley-stat-157



Function Classes

In the 1990s, a new type of learning algorithm was developed, based on results from statistical learning theory: the Support Vector Machine (SVM). This gave rise to a new class of theoretically elegant learning machines that use a central concept of SVMs – -kernels – for a number of

- Extract features
- Pick kernel for similarity
- **Convex** optimization problem
- Many beautiful theorems …



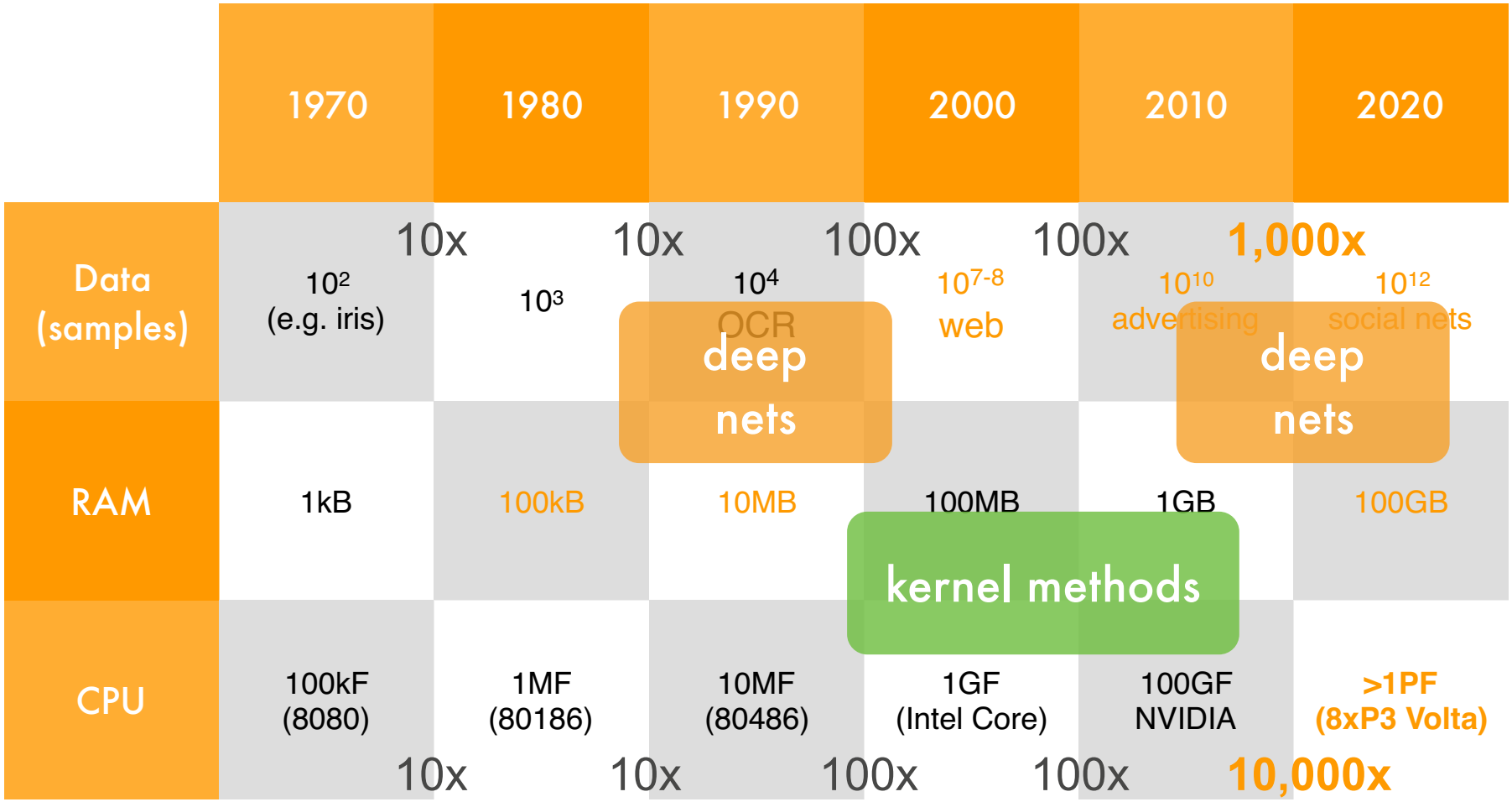
Geometry

- Extract features
- Describe geometry (e.g. multiple cameras) analytically
- **(Non)Convex** optimization problems
- Many beautiful theorems …
- Works very well in theory when the assumptions are satisfied

Feature Engineering

- Feature engineering is crucial
- Feature descriptors, e.g. SIFT (Scale-invariant feature transform), SURF
- Bag of visual words (clustering)
- Then apply SVM …

Hardware



aws

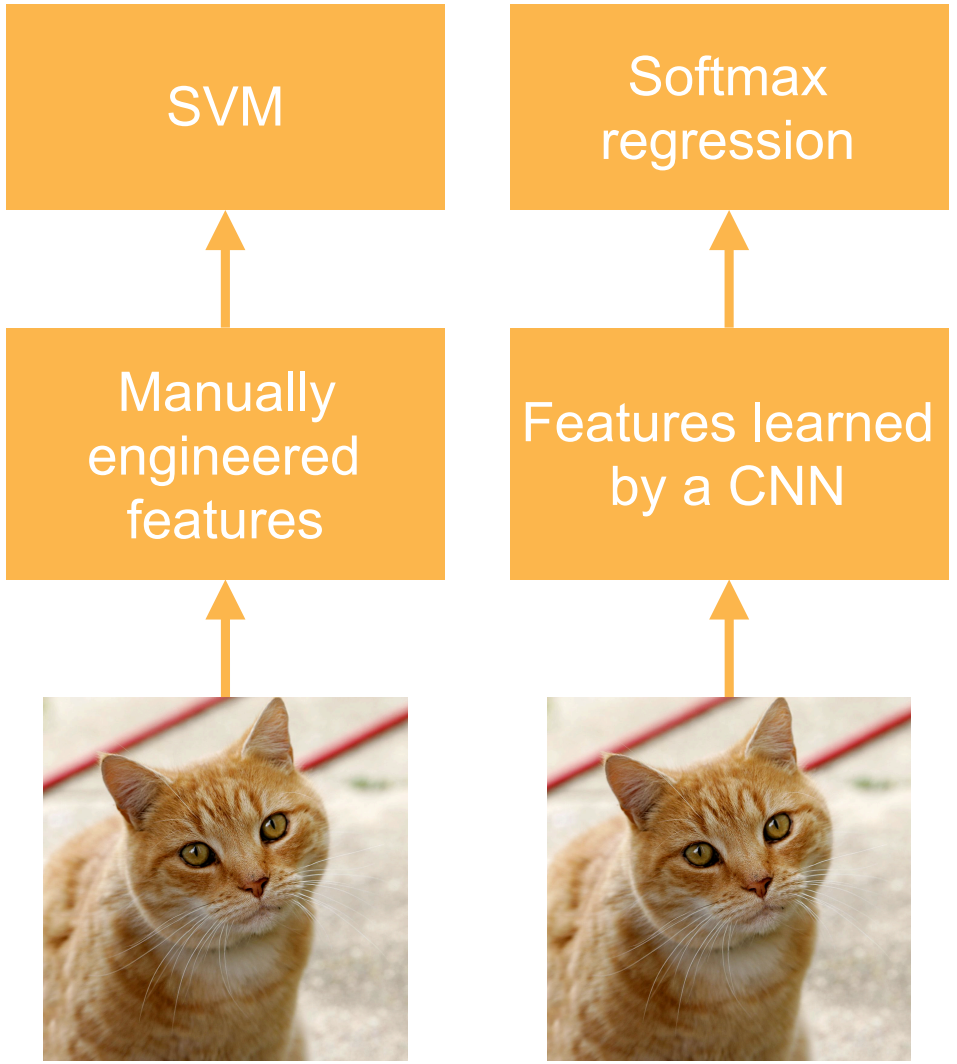
ImageNet (2010)

Images	Color images with nature objects	Gray image for hand-written digits
Size	469 x 387	28 x 28
# examples	1.2 M	60 K
# classes	1,000	10

adapted from courses.d2l.ai/berkeley-stat-157

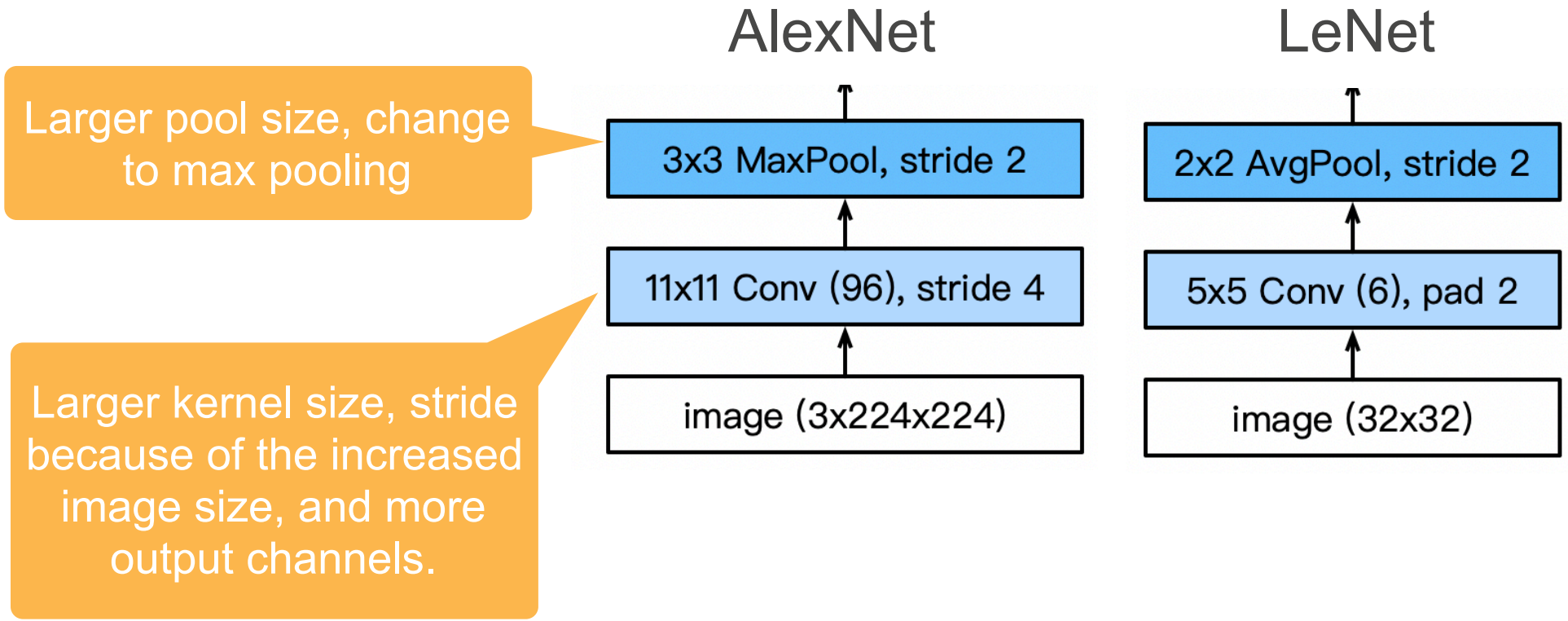
AlexNet

- AlexNet won ImageNet competition in 2012
- Deeper and bigger LeNet
- Key modifications
  - Dropout (regularization)
  - ReLu (training)
  - MaxPooling
- Paradigm shift for computer vision



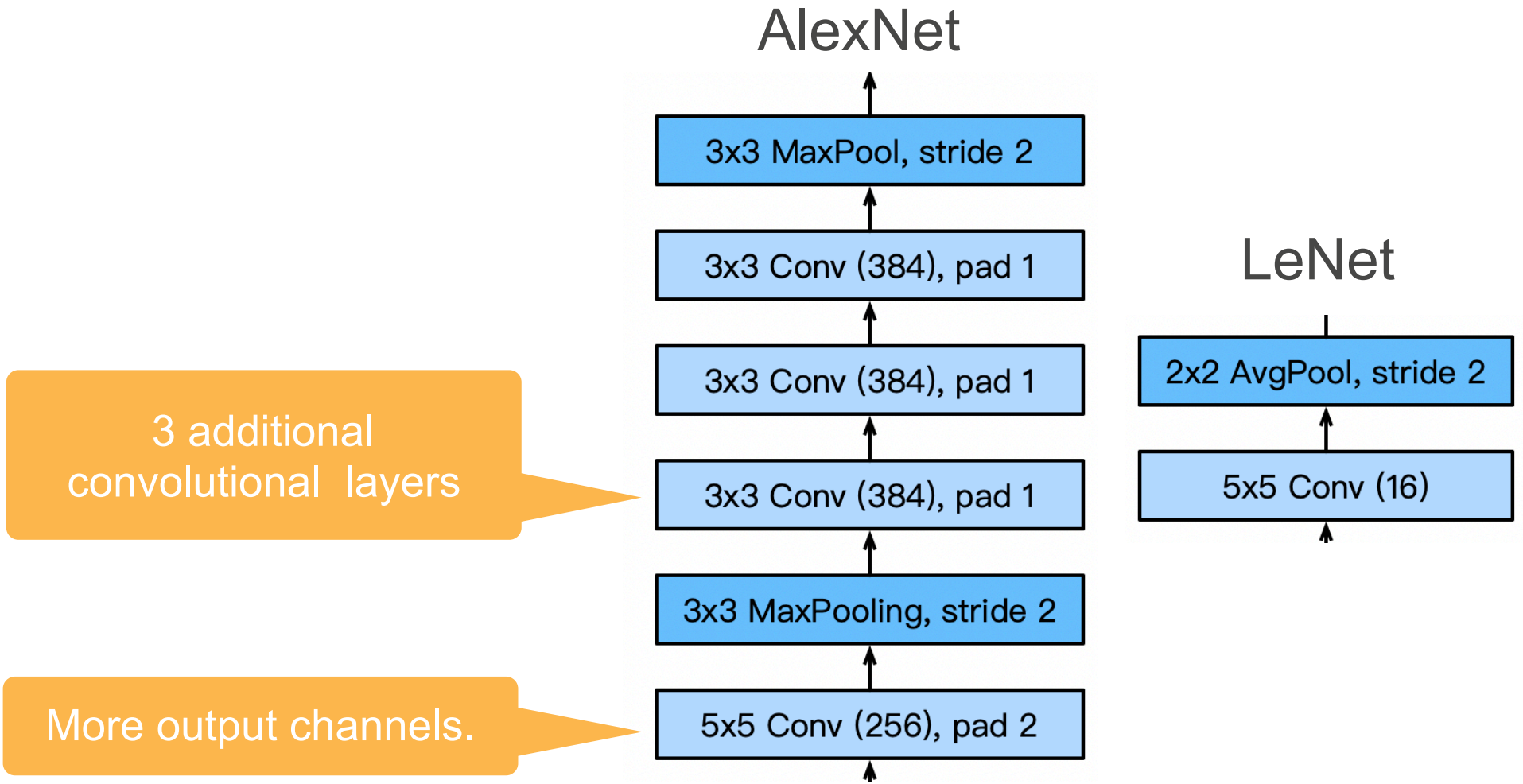
adapted from courses.d2l.ai/berkeley-stat-157

AlexNet Architecture



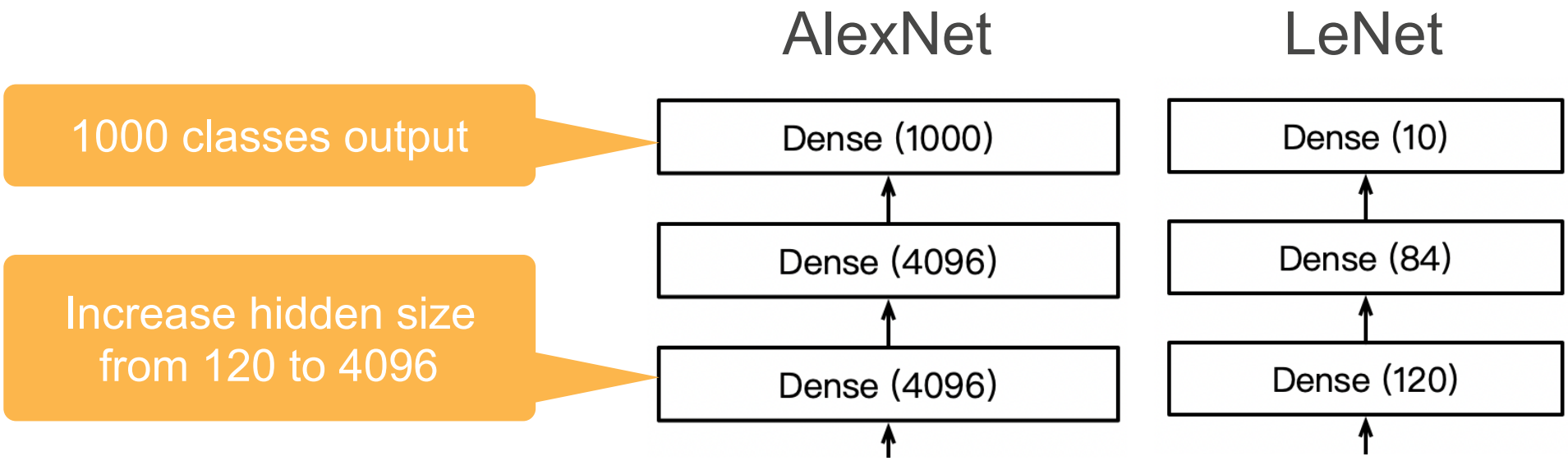
adapted from courses.d2l.ai/berkeley-stat-157

AlexNet Architecture



adapted from courses.d2l.ai/berkeley-stat-157

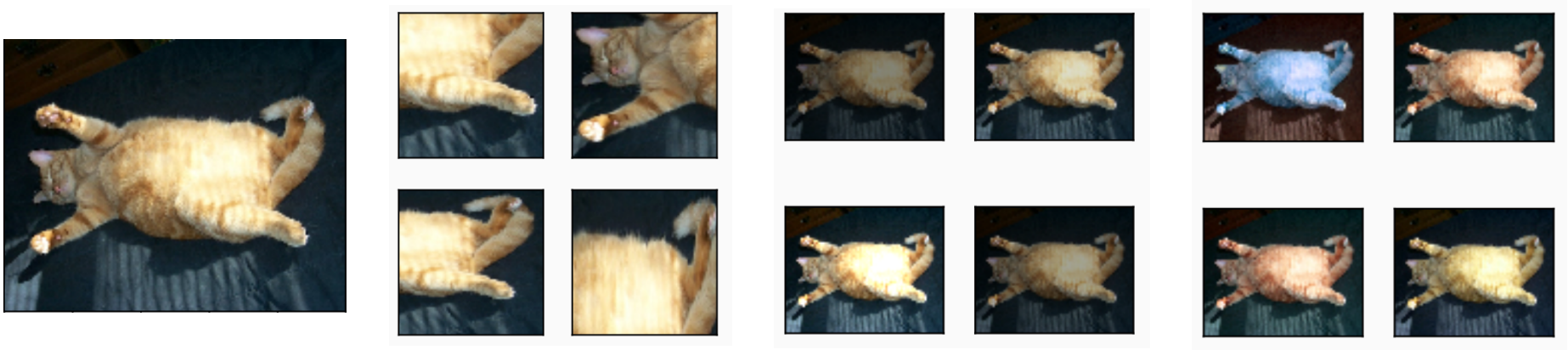
AlexNet Architecture



adapted from courses.d2l.ai/berkeley-stat-157

More Tricks

- Change activation function from sigmoid to ReLu (no more vanishing gradient)
- Add a dropout layer after two hidden dense layers (better robustness / regularization)
- Data augmentation

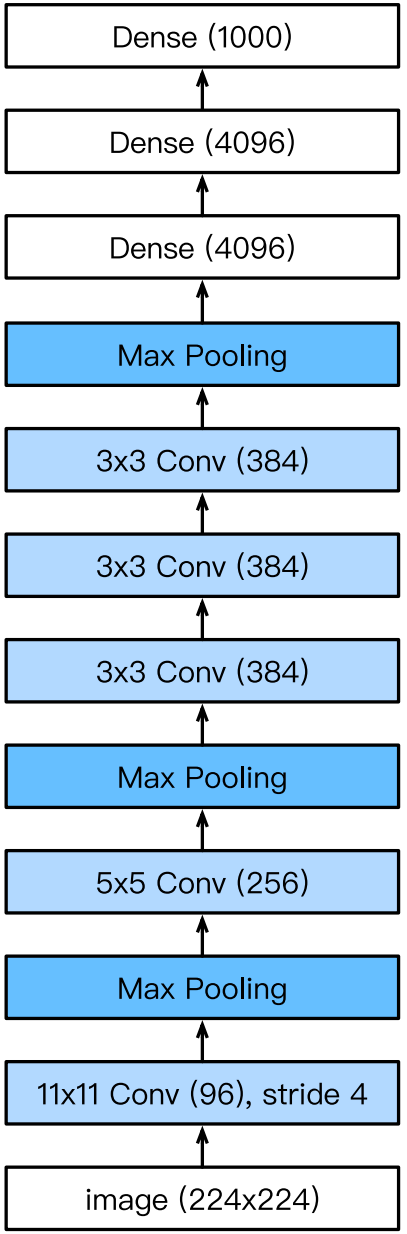


adapted from courses.d2l.ai/berkeley-stat-157

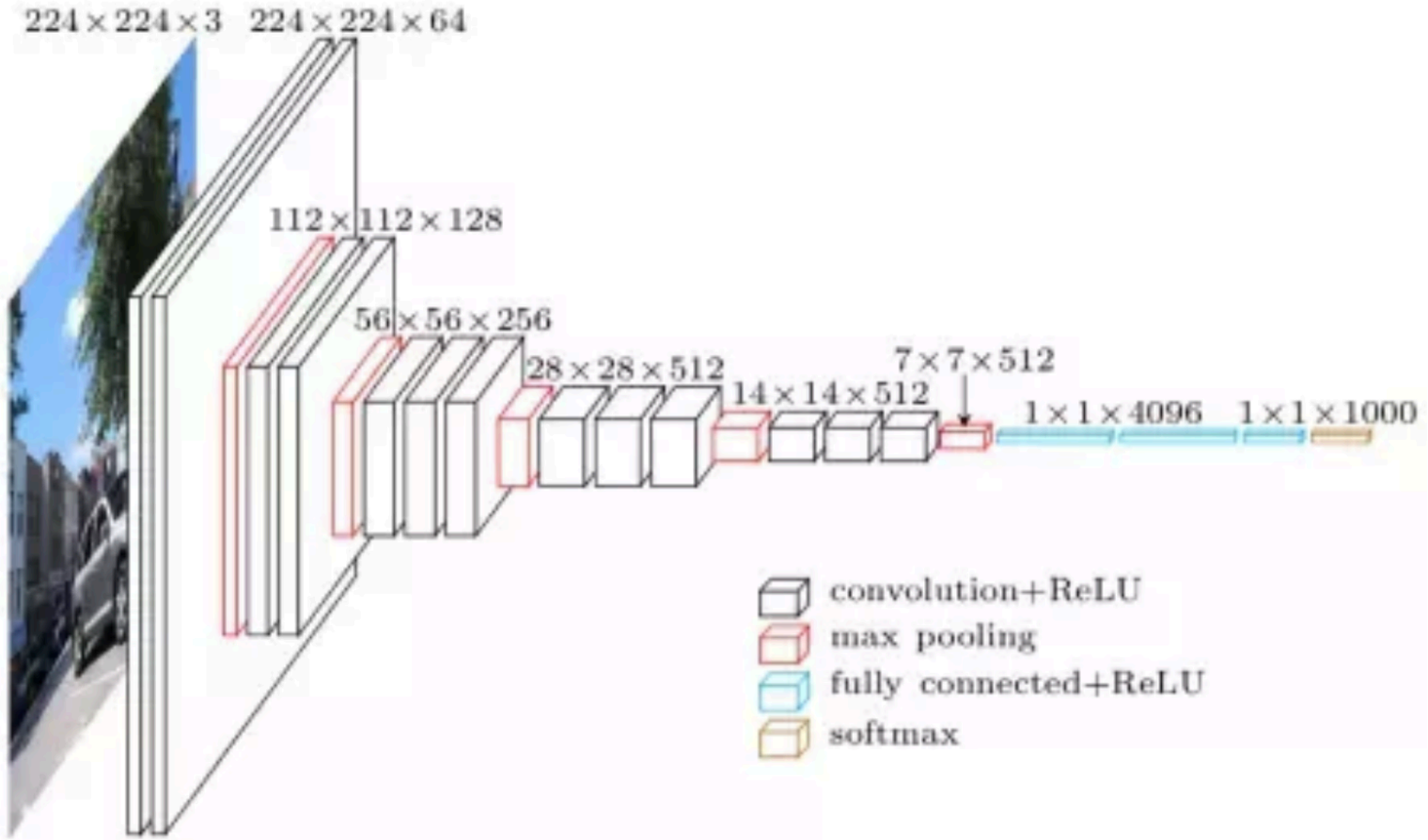
Complexity

	#parameters		FLOP	
	AlexNet	LeNet	AlexNet	LeNet
Conv1	35K	150	101M	1.2M
Conv2	614K	2.4K	415M	2.4M
Conv3-5	3M		445M	
Dense1	26M	0.48M	26M	0.48M
Dense2	16M	0.1M	16M	0.1M
Total	46M	0.6M	1G	4M
Increase	11x	1x	250x	1x

adapted from courses.d2l.ai/berkeley-stat-157



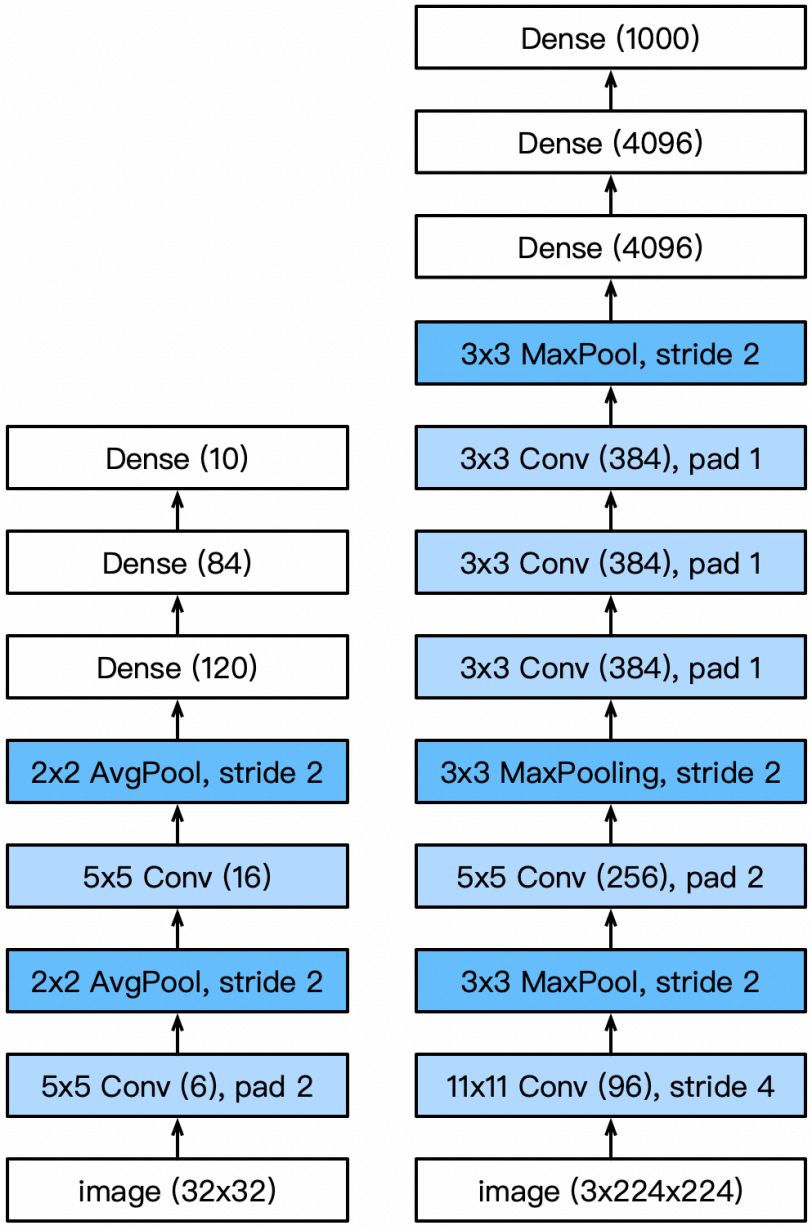
VGG



adapted from courses.d2l.ai/berkeley-stat-157

VGG

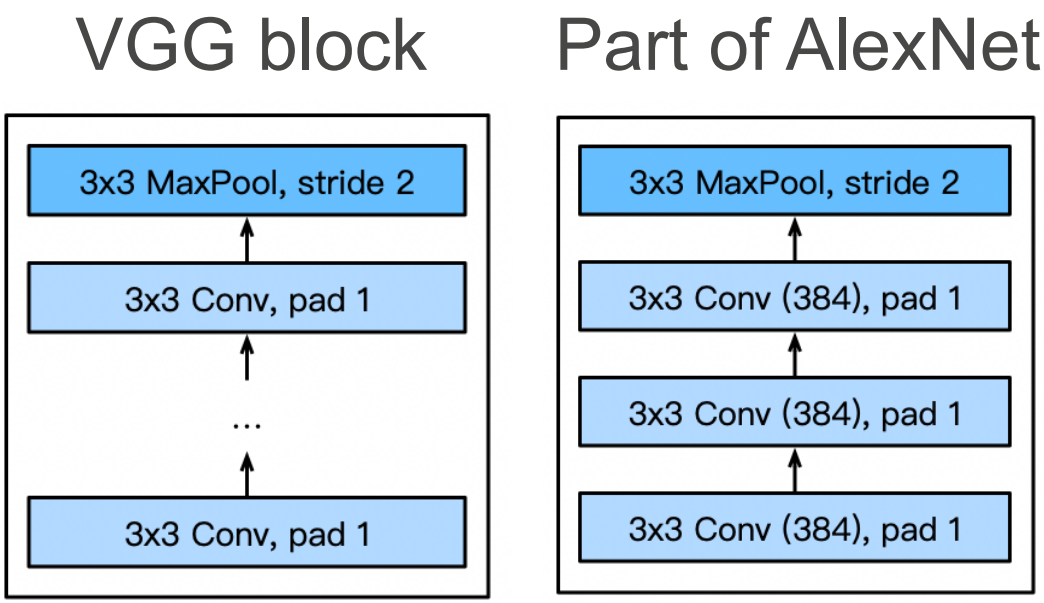
- AlexNet is deeper and bigger than LeNet to get performance
- Go even bigger & deeper?
- Options
  - More dense layers (too expensive)
  - **More** convolutions
  - Group into **blocks**



adapted from courses.d2l.ai/berkeley-stat-157

VGG Blocks

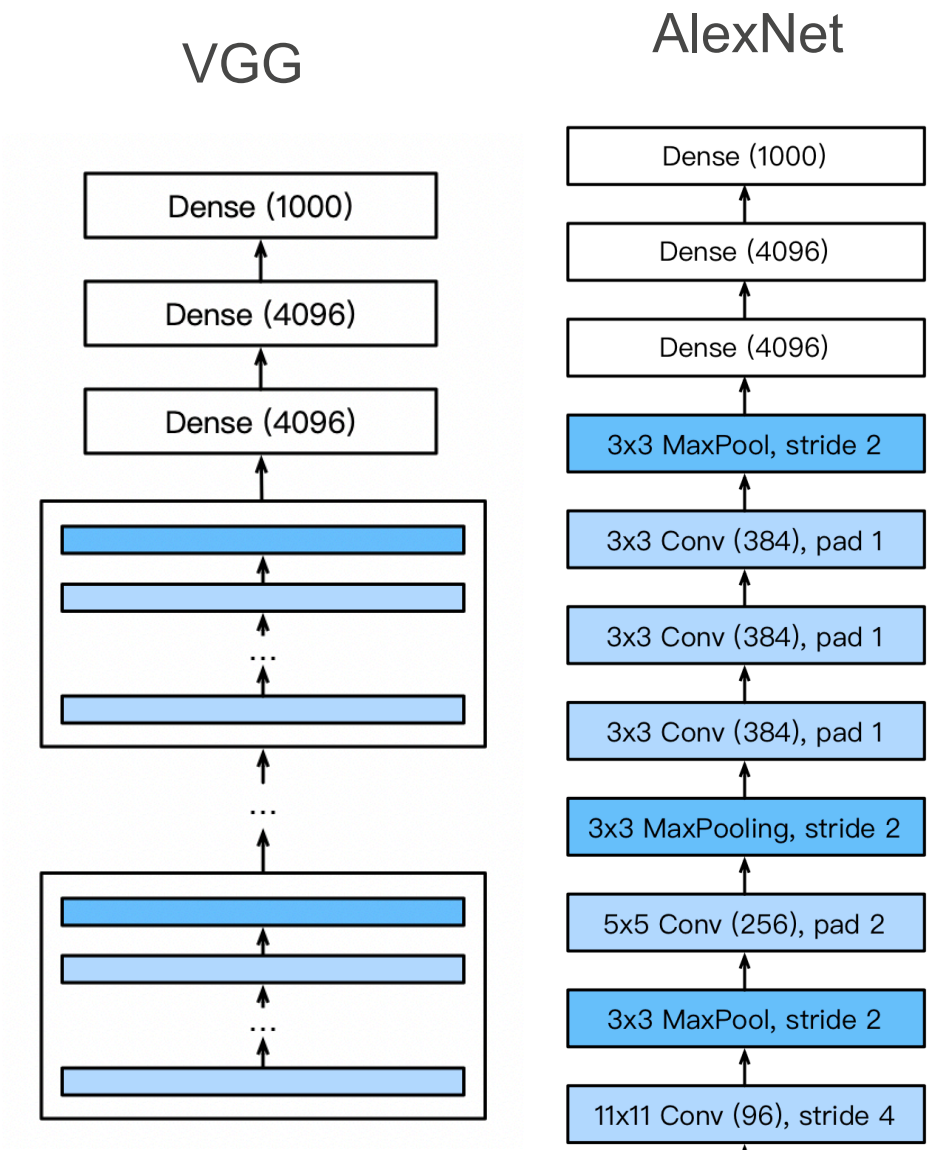
- Deeper vs. wider?
  - 5x5 convolutions
  - 3x3 convolutions (more)
  - **Deep & narrow better**
- VGG block
  - *3x3* convolutions (pad 1) **(n layers, m channels)**
  - 2x2 max-pooling (stride 2)



adapted from courses.d2l.ai/berkeley-stat-157

VGG Architecture

- Multiple VGG blocks followed by dense layers
- Vary the repeating number to get different architectures, such as VGG-16, VGG-19, …

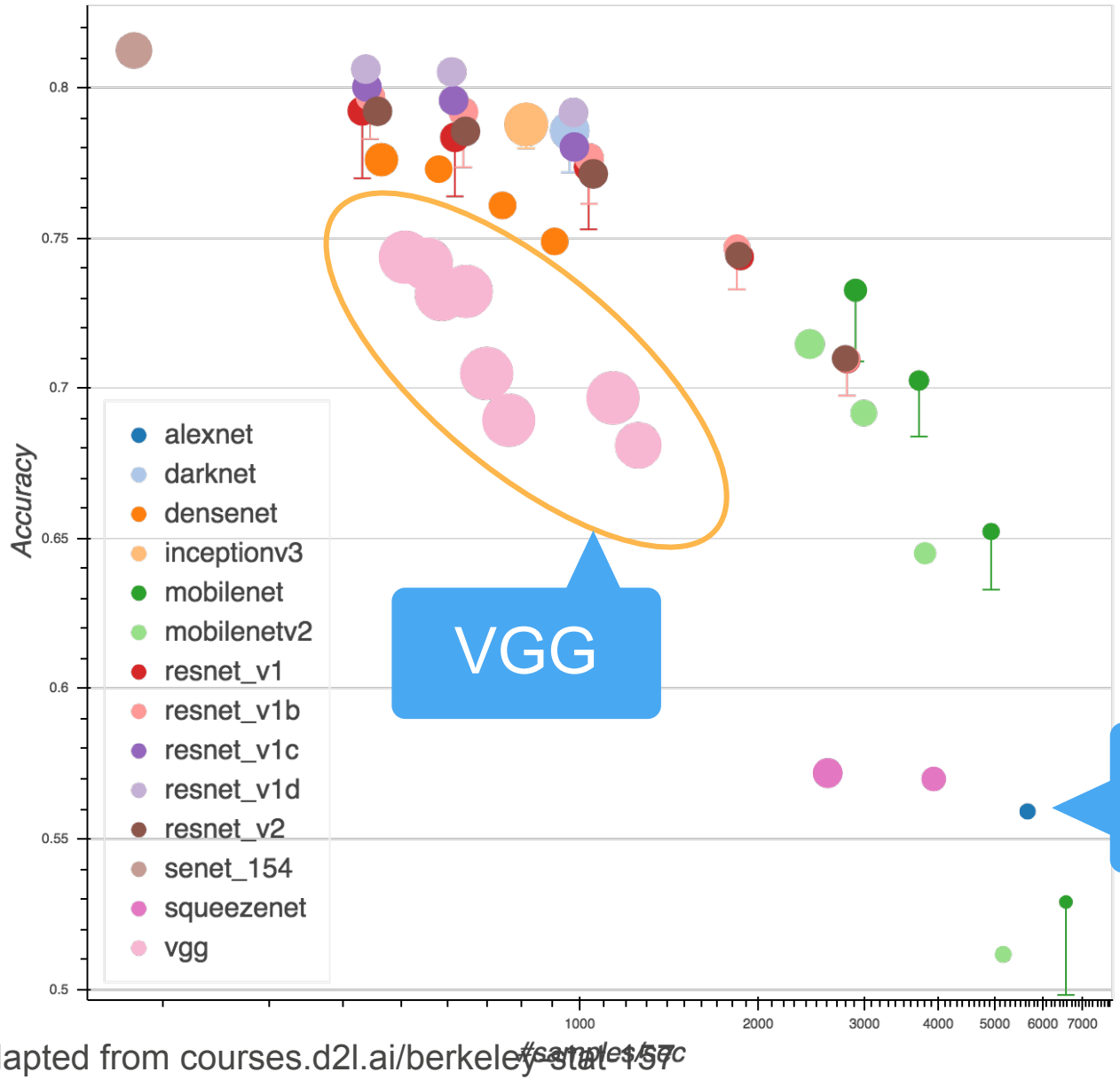


adapted from courses.d2l.ai/berkeley-stat-157

Progress

- LeNet (1995)
  - 2 convolution + pooling layers
  - 2 hidden dense layers
- AlexNet
  - Bigger and deeper LeNet
  - ReLu, Dropout, preprocessing
- VGG
  - Bigger and deeper AlexNet (repeated VGG blocks)

adapted from courses.d2l.ai/berkeley-stat-157

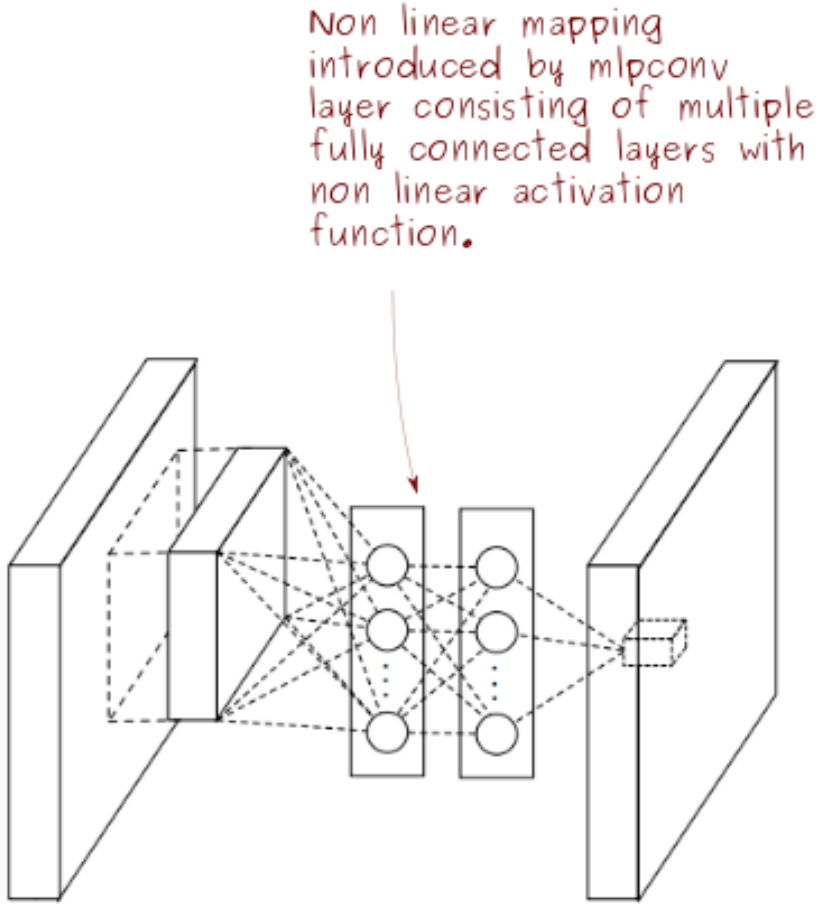


GluonCV Model Zoo

gluon-cv.mxnet.io/model_zoo/classification.html

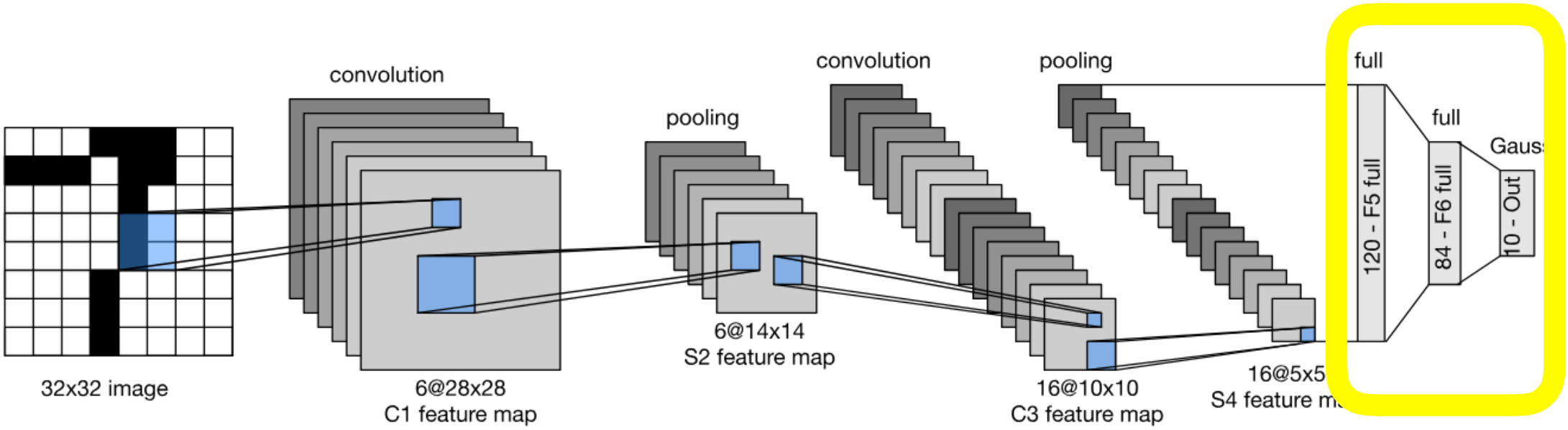
adapted from courses.d2l.ai/berkeley-stat-157

Network in Network



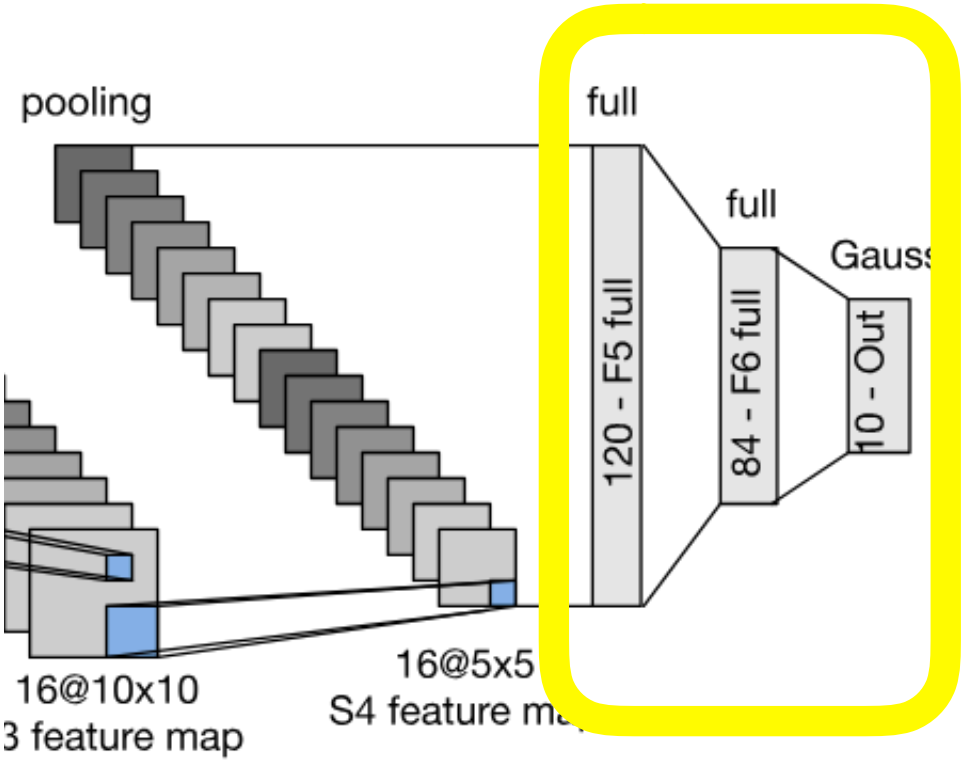
adapted from courses.d2l.ai/berkeley-stat-157

The Curse of the Last Layer(s)



adapted from courses.d2l.ai/berkeley-stat-157

The Last Layer(s)



- Convolution layers need relatively few parameters

$$c_i \times c_o \times k^2$$

- Last layer needs many parameters for n classes

$$c \times m_w \times m_h \times n$$

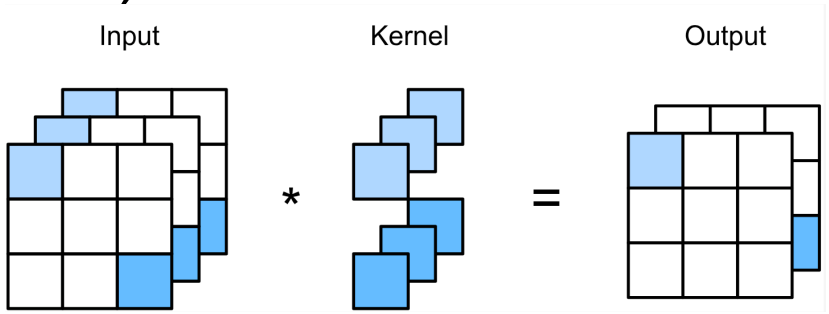
- LeNet 16x5x5x120 = 48k
- AlexNet 256x5x5x4096 = 26M
- VGG 512x7x7x4096 = 102M

gluon-cv.mxnet.io

aws

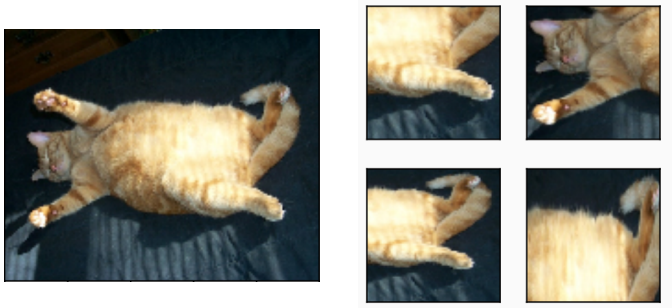
Breaking the Curse of the Last Layer

- Key Idea
  - **Get rid of the fully connected last layer(s)**
  - Convolutions and pooling reduce resolution (e.g. stride of 2 reduces resolution 4x)
- Implementation details
  - Reduce resolution progressively
  - Increase number of channels
  - Use **1x1 convolutions** (they only act per pixel)
- **Global average pooling in the end**

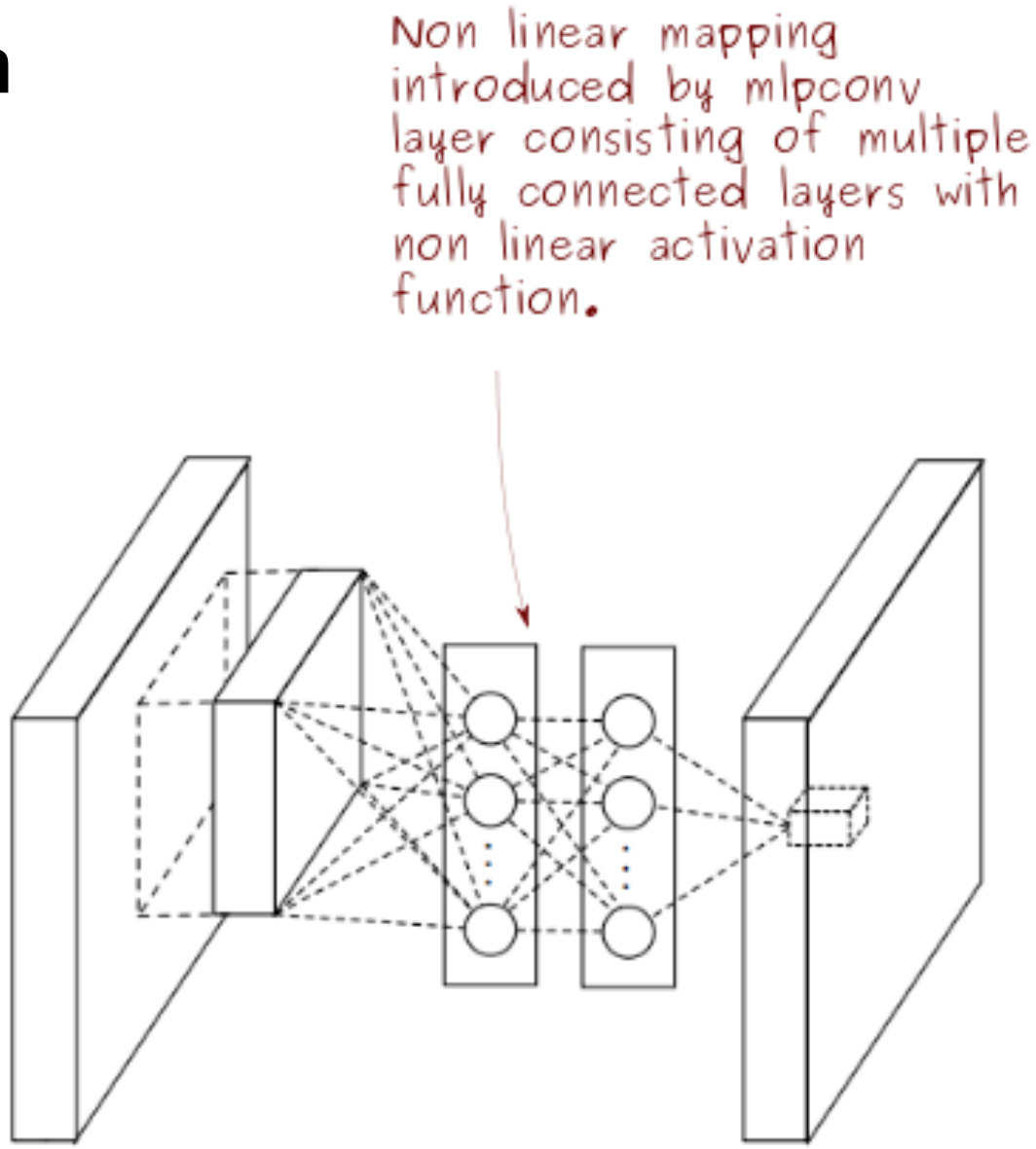


What's a 1x1 convolution anyway?

- Extreme case
  1x1 image with n channels
- Equivalent to MLP
- Pooling allows for translation invariance of detection (e.g. 5x5)

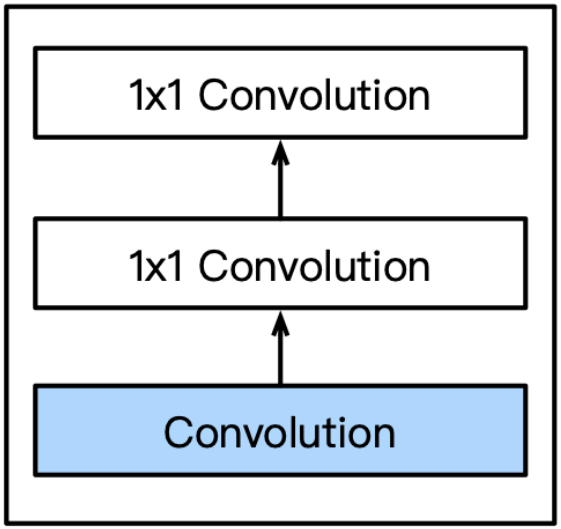


adapted from courses.d2l.ai/berkeley-stat-157



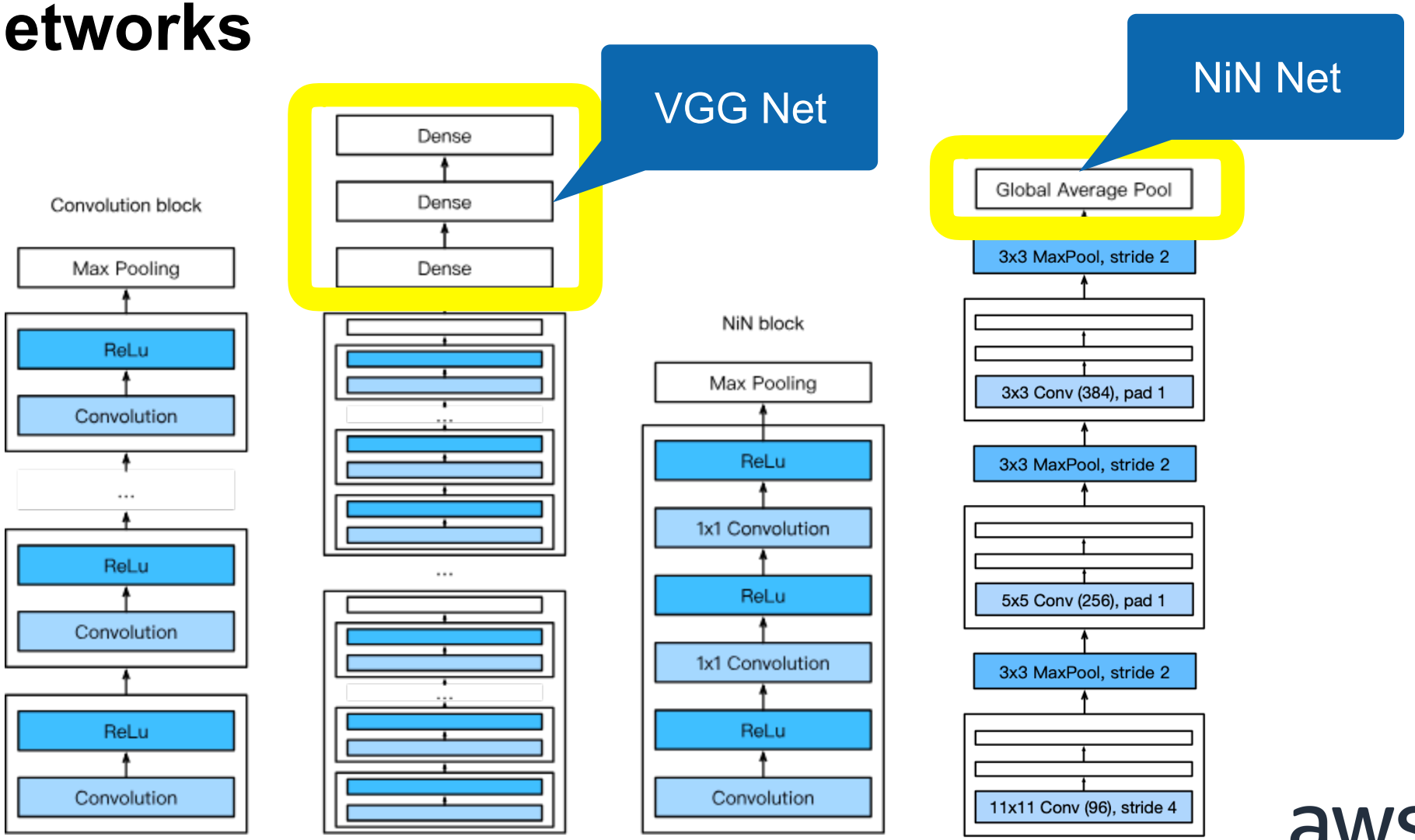
NiN Block

- A convolutional layer
  - kernel size, stride, and padding are hyper-parameters
- Following by two 1x1 convolutions
  - 1 stride and no padding, share the same output channels as first layer
  - Act as dense layers



adapted from courses.d2l.ai/berkeley-stat-157

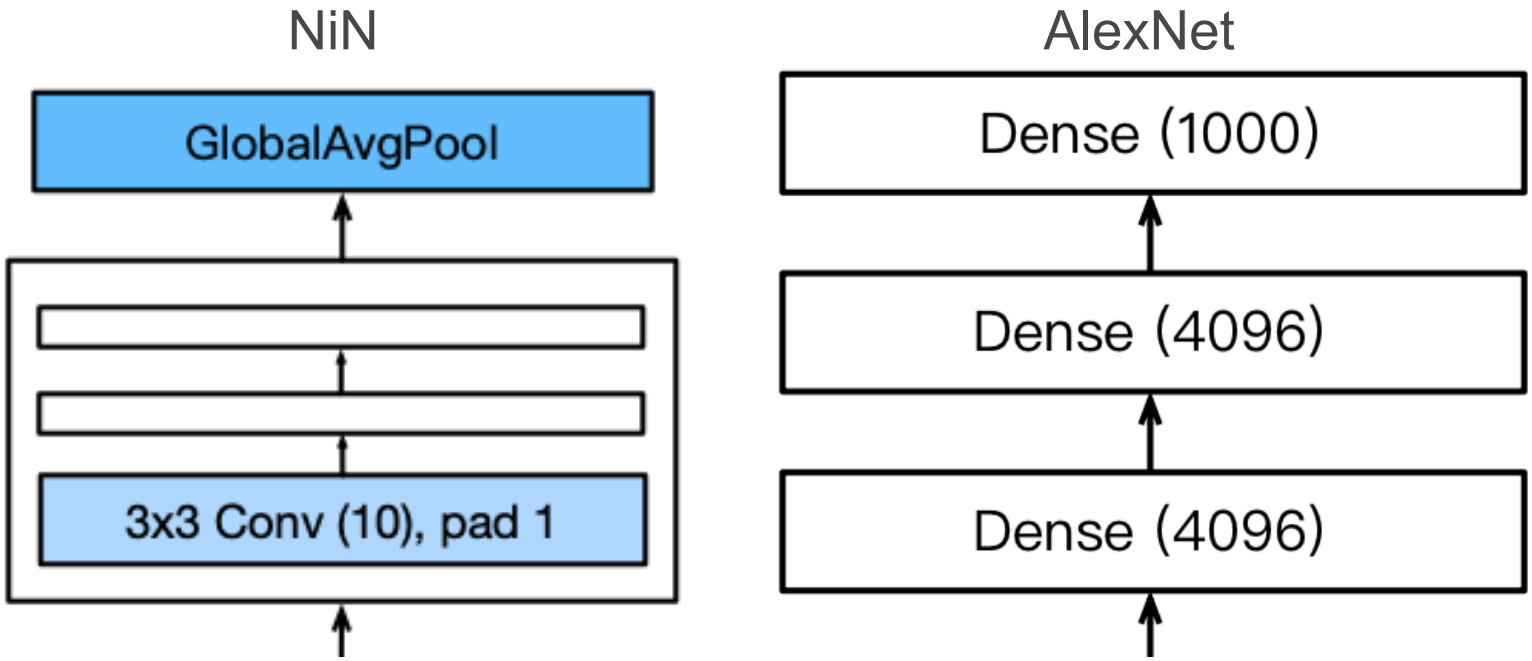
NiN Networks



gluon-cv.mxnet.io

NiN Last Layers

- Replaced AlexNet's dense layers with a NiN block
- Global average pooling layer to combine outputs



adapted from courses.d2l.ai/berkeley-stat-157

Summary

- **LeNet** (the first convolutional neural network)
- **AlexNet**
  - More of everything
  - ReLu, Dropout, Invariances
- **VGG**
  - Even more of everything (narrower and deeper)
  - Repeated blocks
- **NiN**
  - 1x1 convolutions + global pooling instead of dense

adapted from courses.d2l.ai/berkeley-stat-157