THE HONG KONG POLYTECHNIC UNIVERSITY
香港理工大學

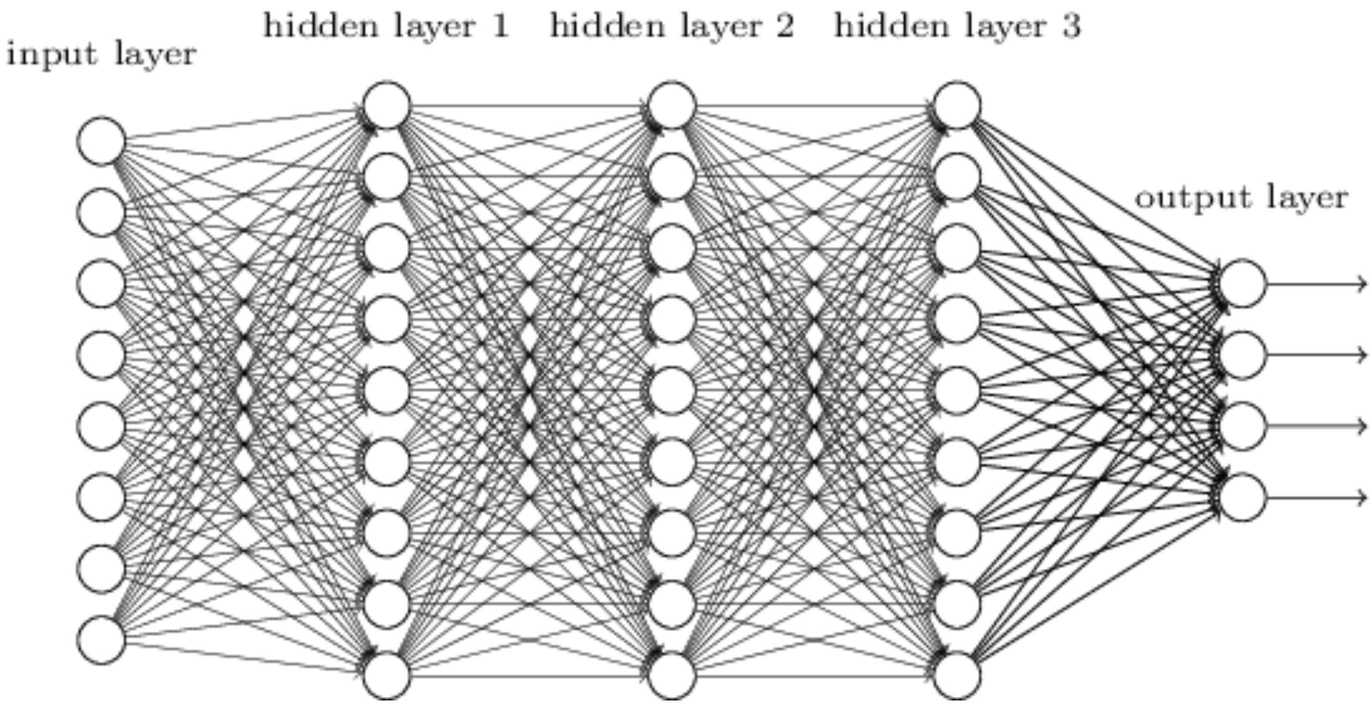
COMP4434 Big Data Analytics

Lecture 9

Convolutional Neural Networks

HUANG Xiao
xiaohuang@comp.polyu.edu.hk

Smaller Network?



- From this fully connected model, do we really need all the edges?
- Can some of these be shared?

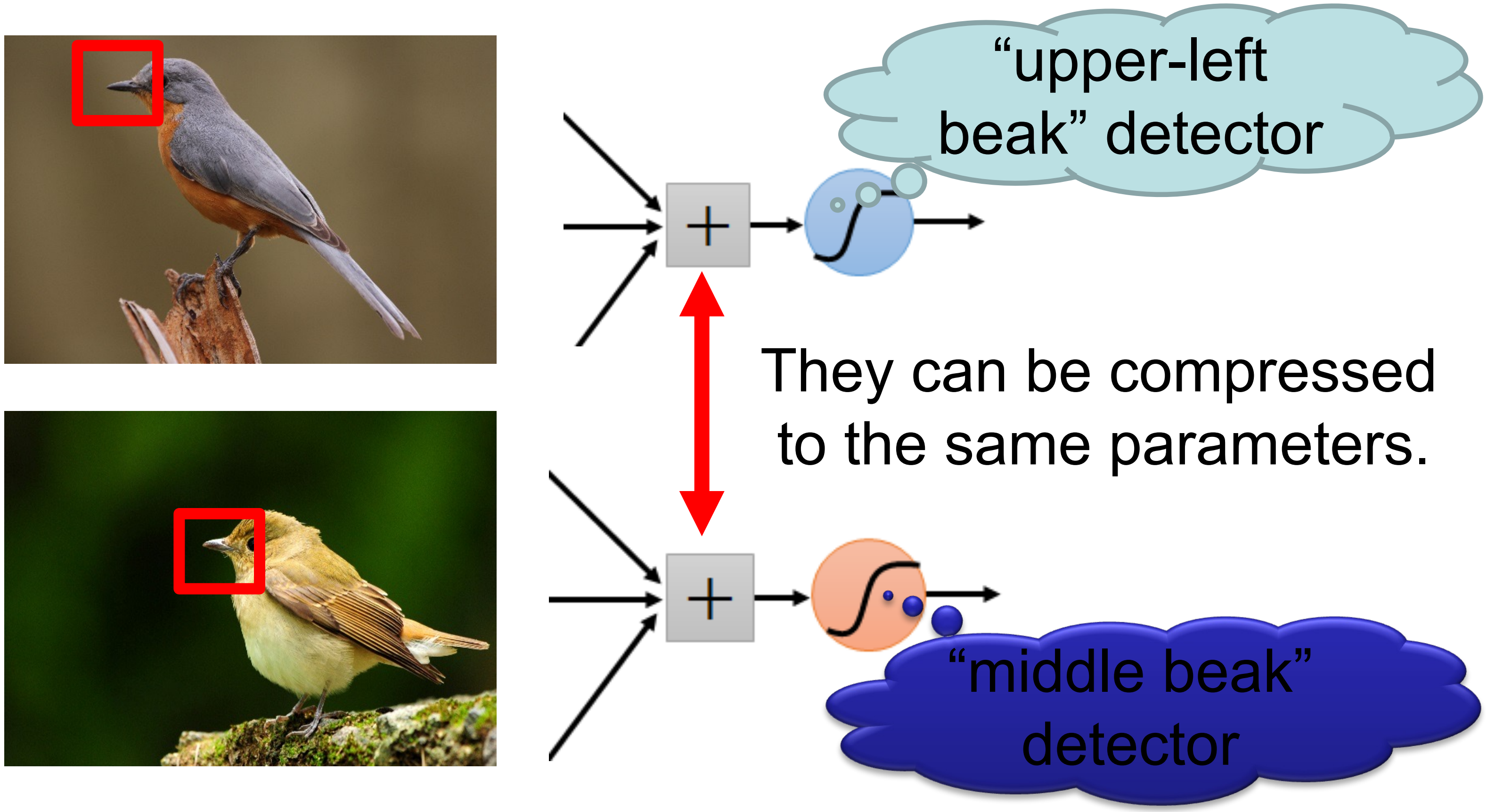
Consider learning an image:

- Some patterns are much smaller than the whole image

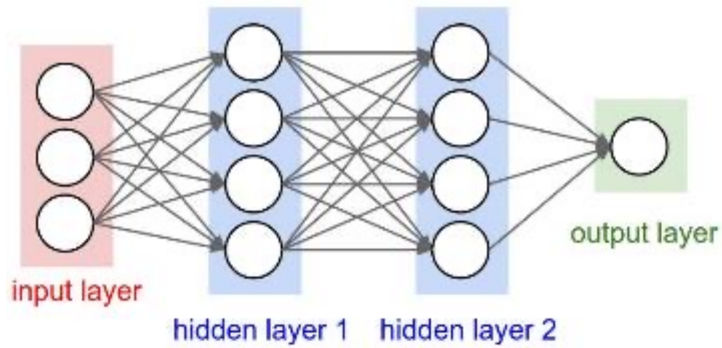
Can represent a small region with fewer parameters

Same pattern appears in different places

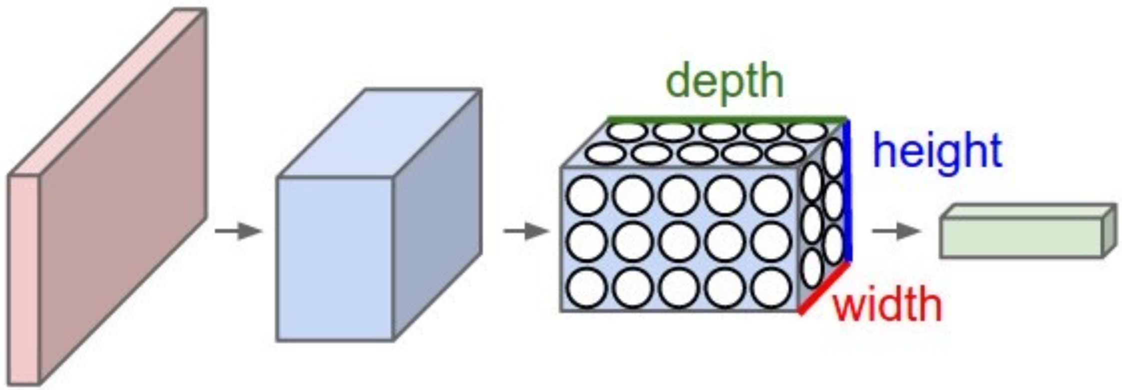
- They can be compressed!
  What about training a lot of such "small" detectors and each detector must "move around".



MLP vs convolutional neural network



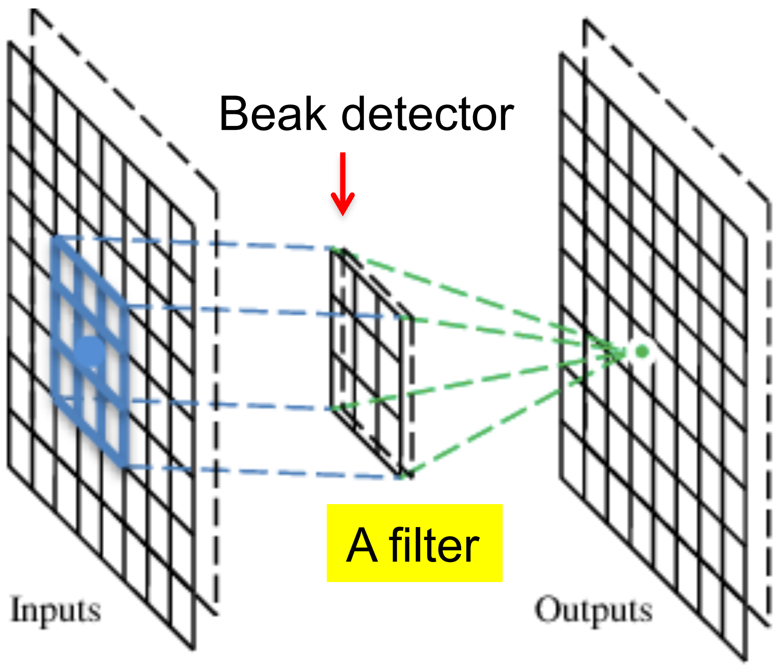
A regular 3-layer Neural Network.



A CNN arranges its neurons in three dimensions (width, height, depth). Every layer of a CNN transforms the 3D input volume to a 3D output volume. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels)

A convolutional layer

A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.



Convolution

These are the network parameters to be learned.

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3)

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Dot product

3 -1

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3 -3

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

$\boxed{3}$	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
$\boxed{3}$	-2	-2	-1

Convolution

-1	1	-1
-1	1	-1
-1	1	-1

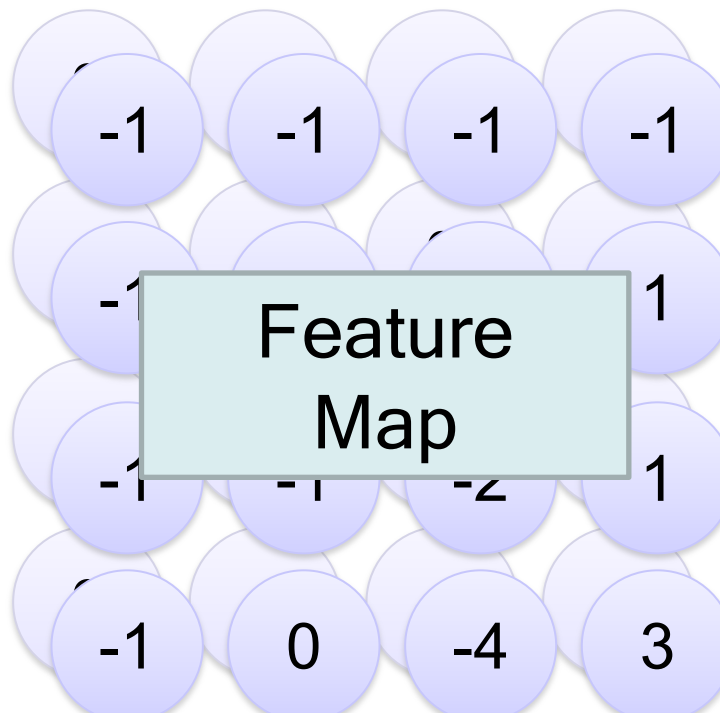
Filter 2

stride=1

Repeat this for each filter

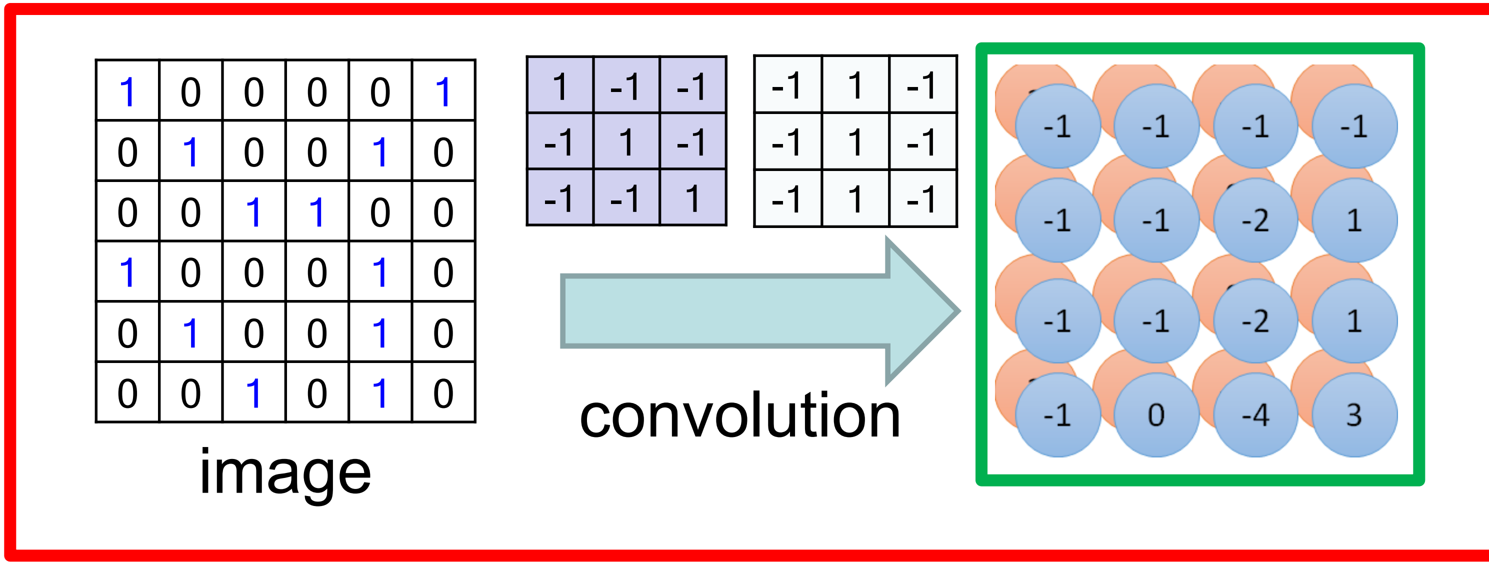
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

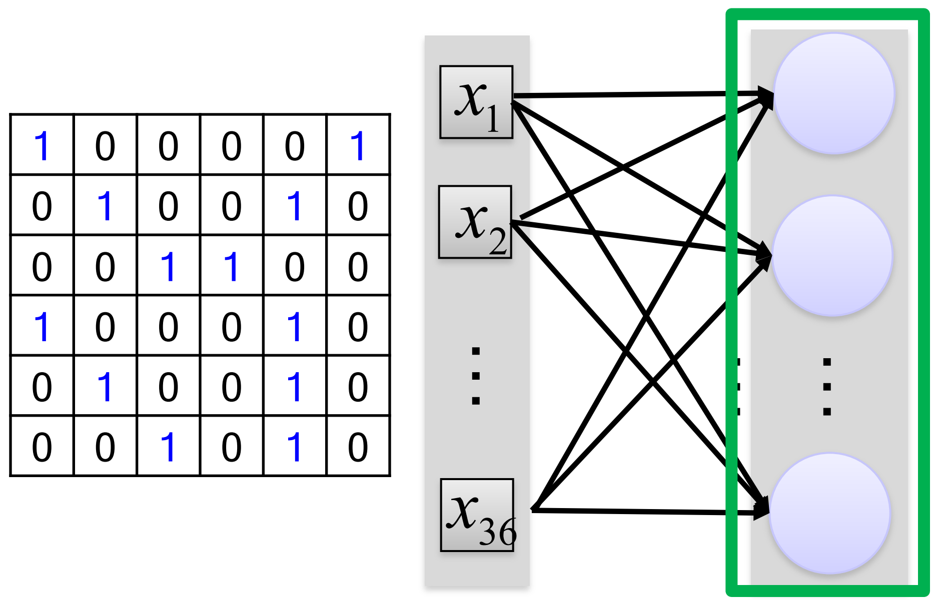


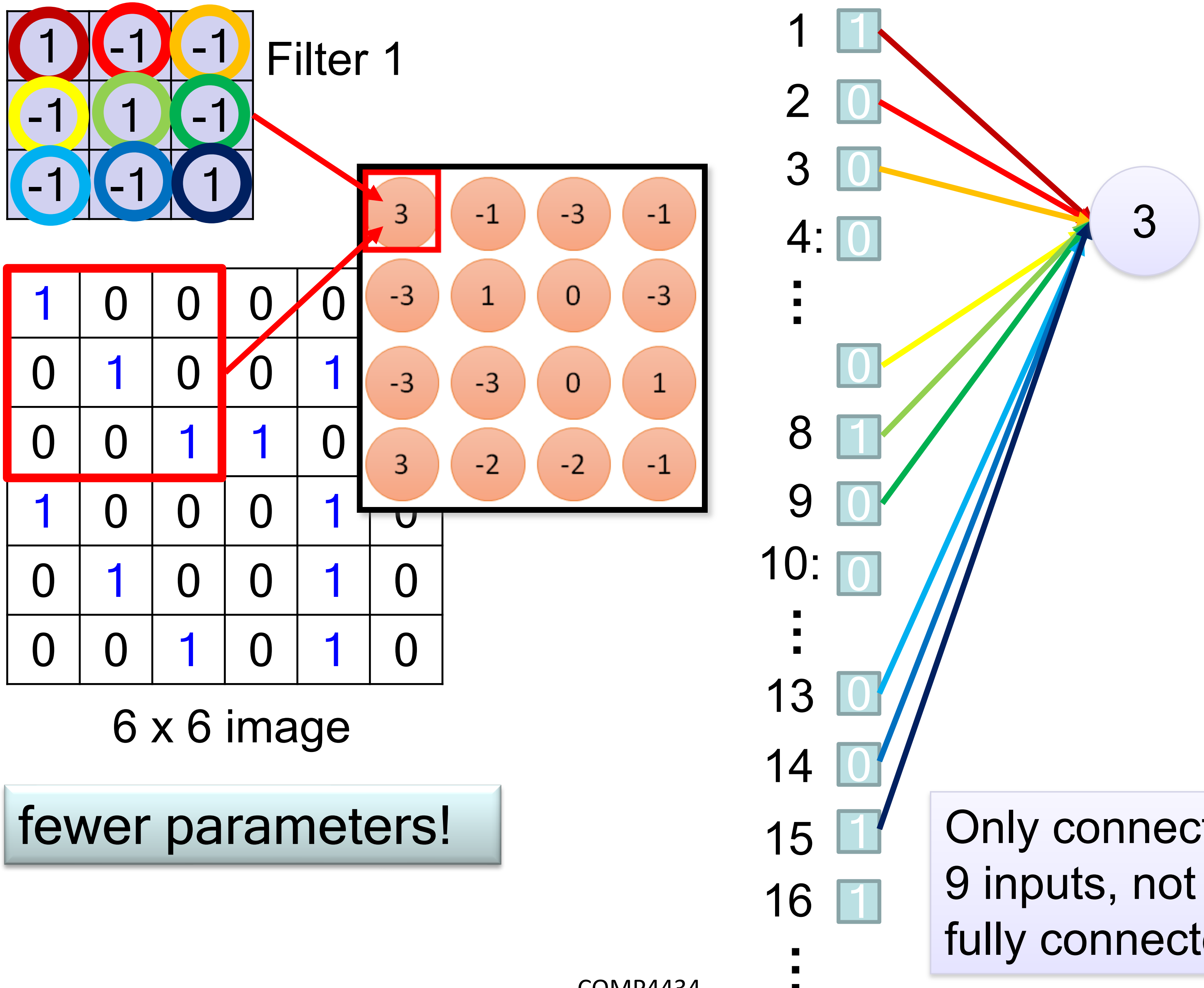
Two 4 x 4 images
Forming 2 x 4 x 4 matrix

Convolution vs Fully Connected



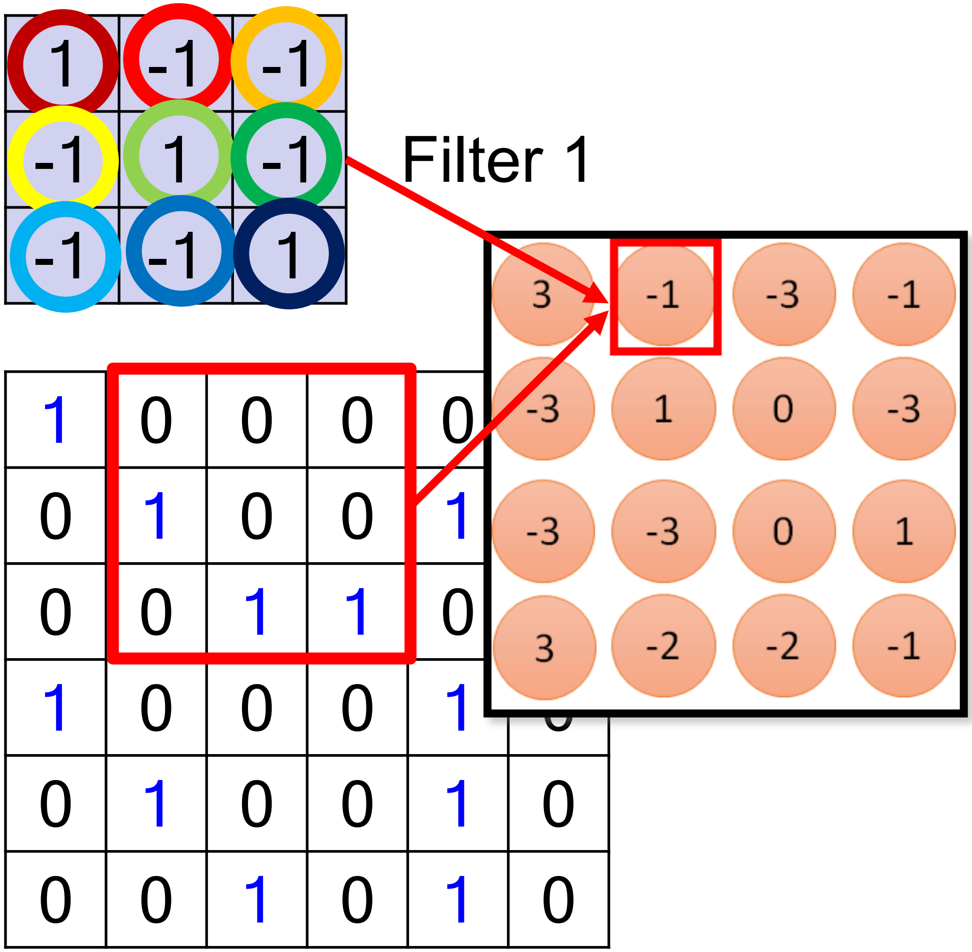
Fully-connected





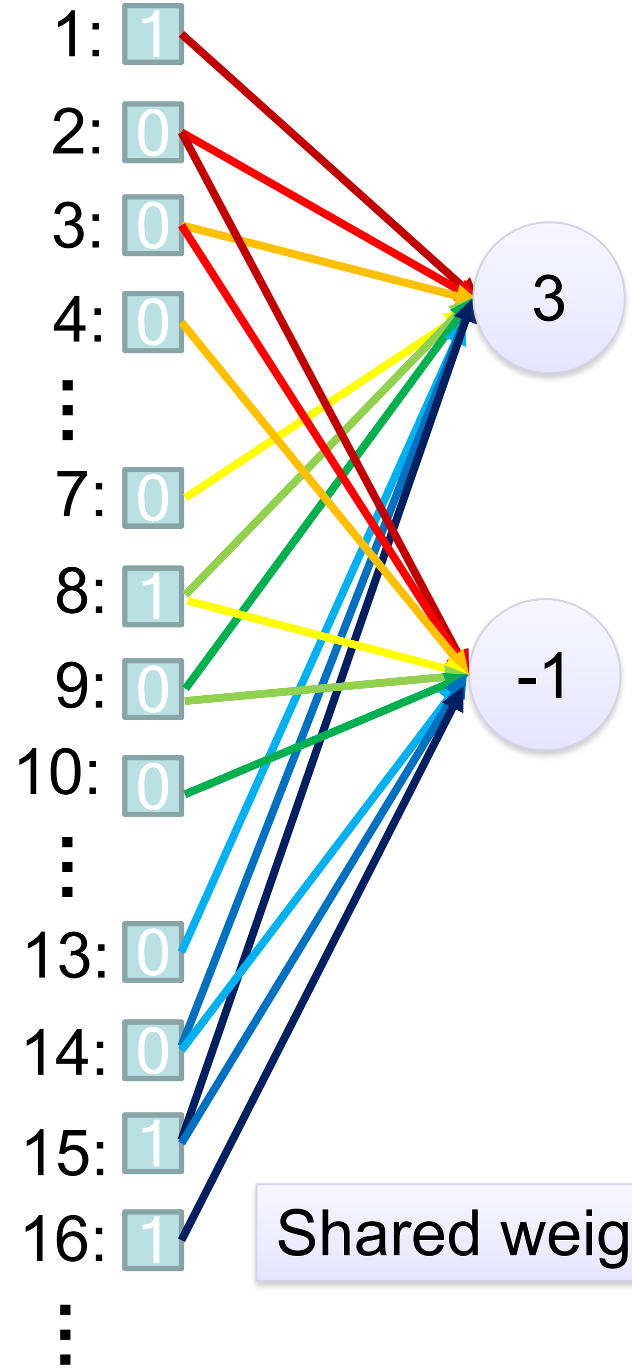
fewer parameters!

Only connect to 9 inputs, not fully connected



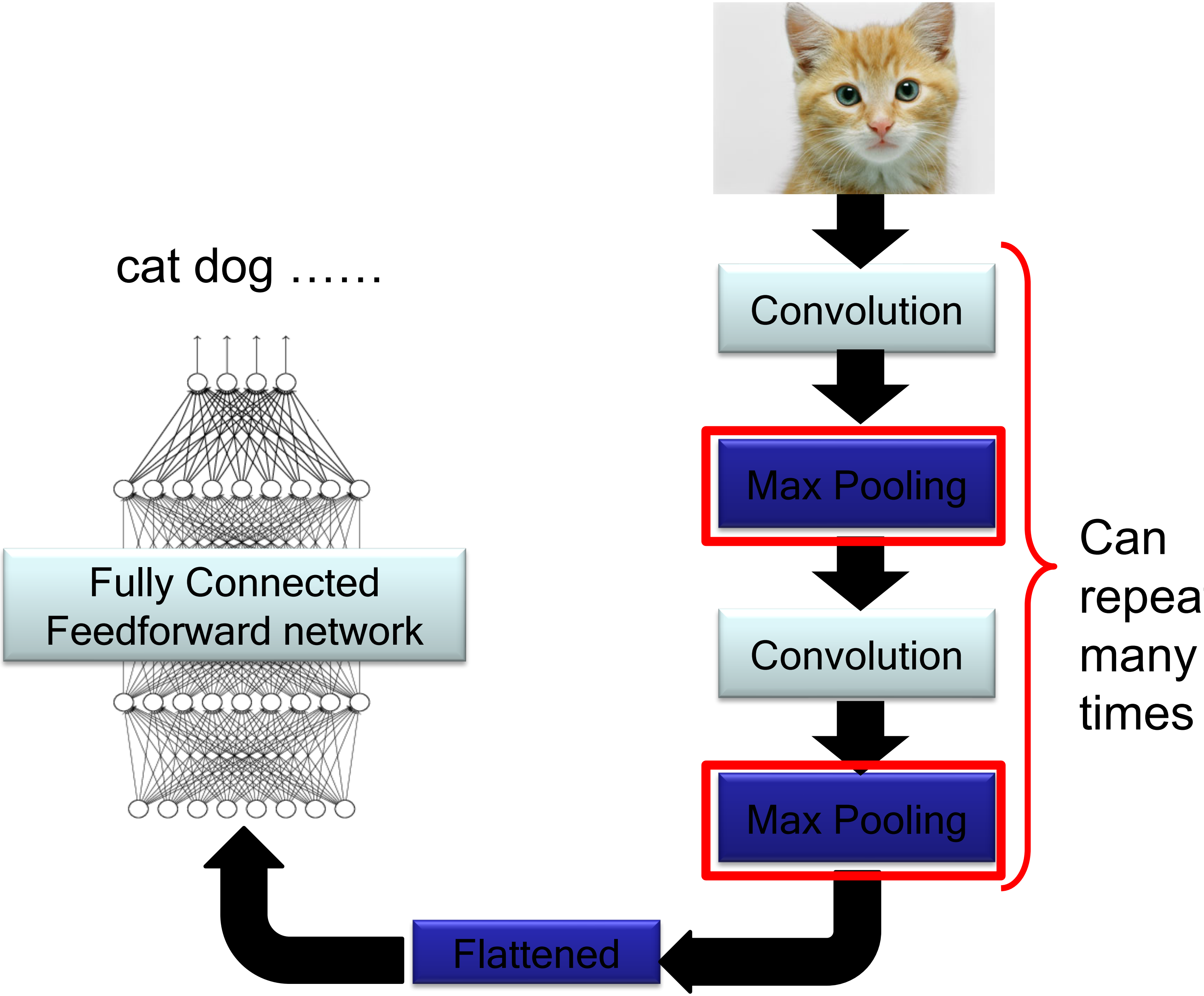
Fewer parameters

Even fewer parameters



Shared weights

The whole CNN



Max Pooling

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

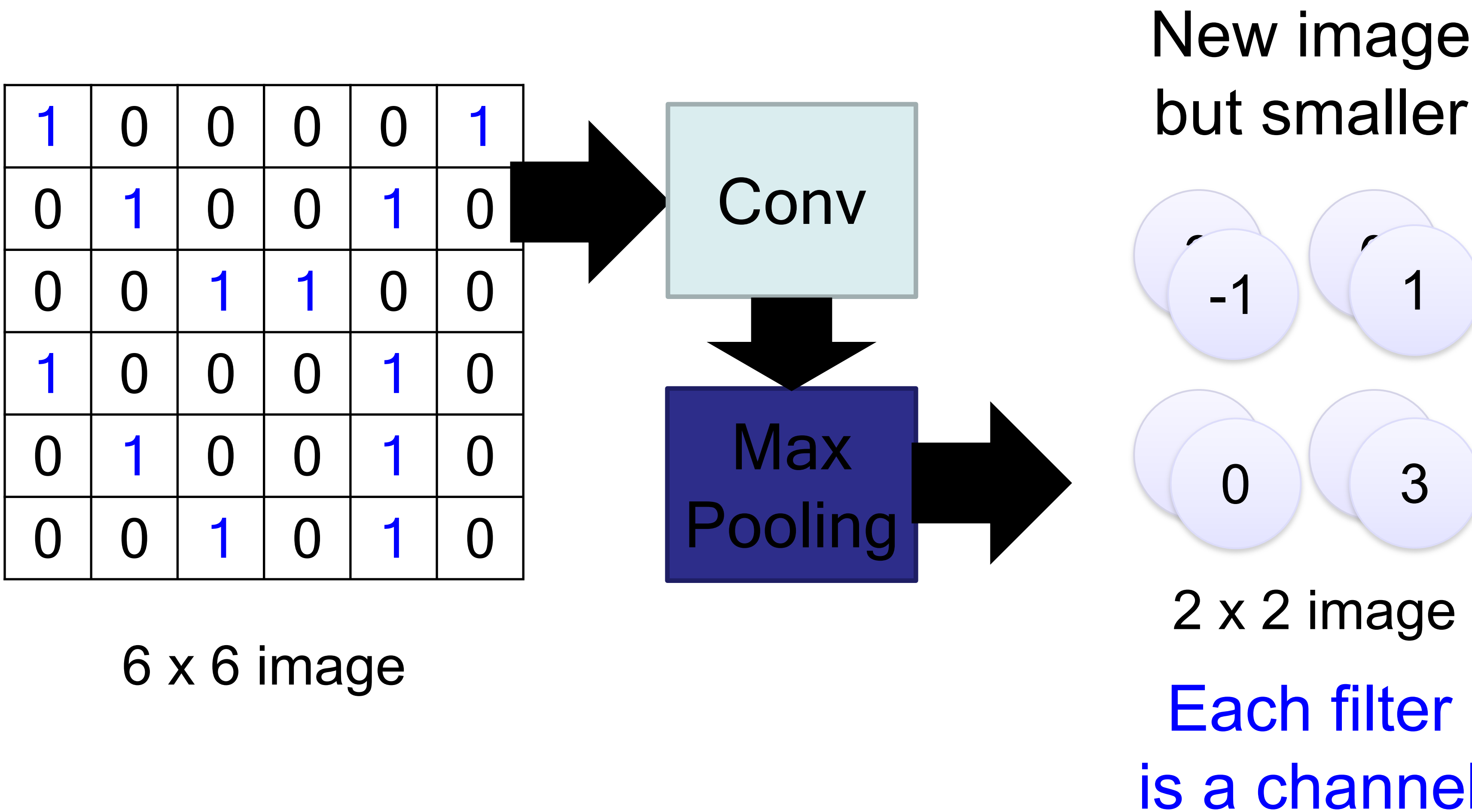
-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

-1	-1	-1	-1
-1	-1	-2	1
-1	-1	-2	1
-1	0	-4	3

Max Pooling


1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image
Conv
Max Pooling
New image but smaller
-1
1
0
3
2 x 2 image
Each filter is a channel

Why Pooling

- Subsampling pixels will not change the object

bird

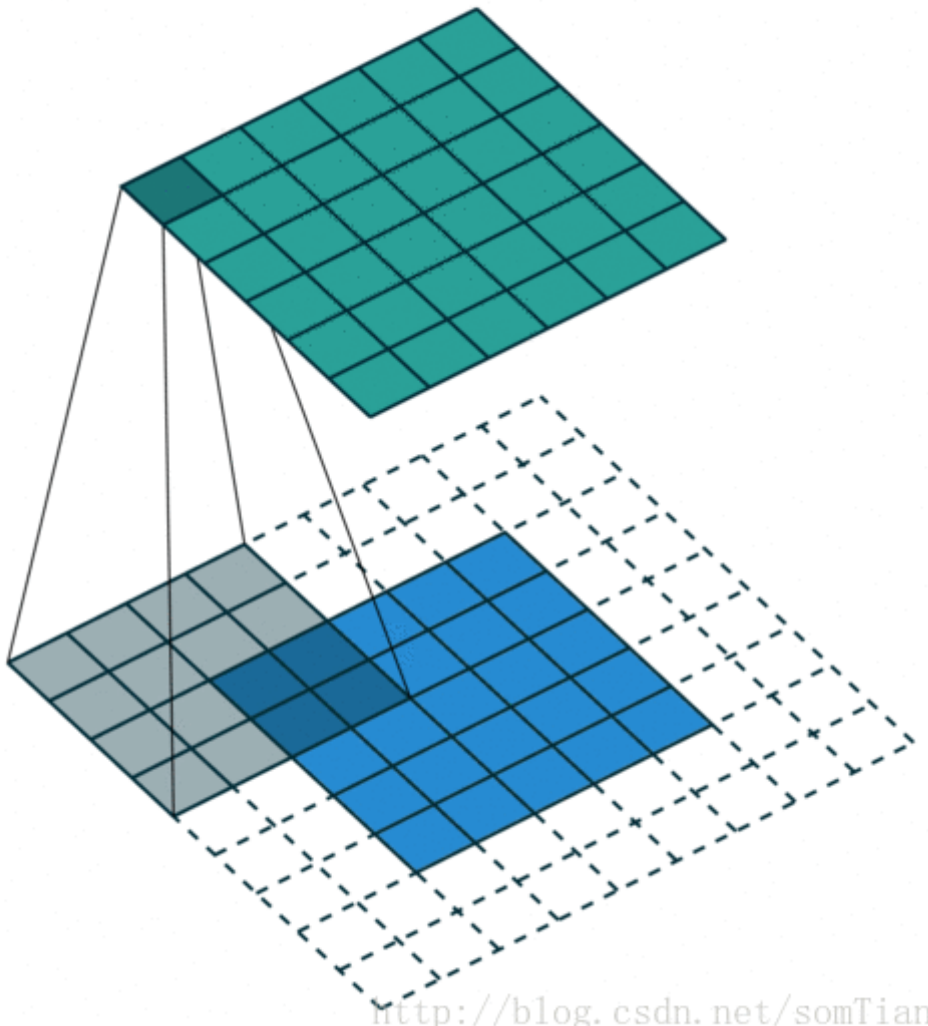
Subsampling

bird

We can subsample the pixels to make image smaller

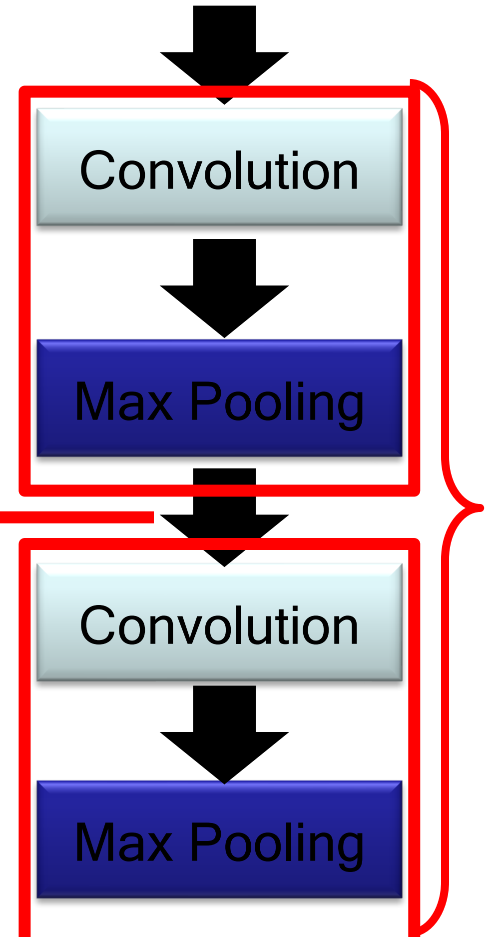
➡ fewer parameters to characterize the image

Convolutional kernel

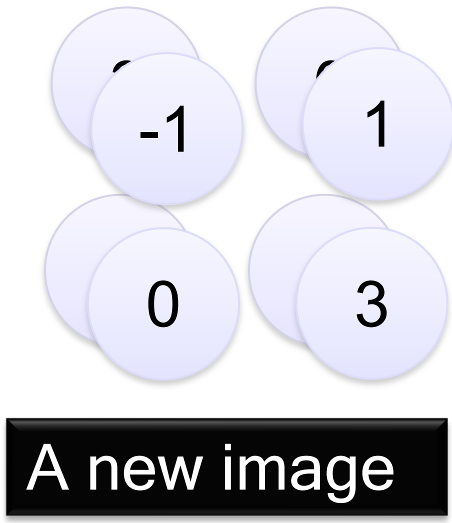


- A convolutional layer has a number of filters that does convolutional operation
- This image show the convolutional operation for one filter
- Each filter detects a small pattern and learns its parameter

The whole CNN



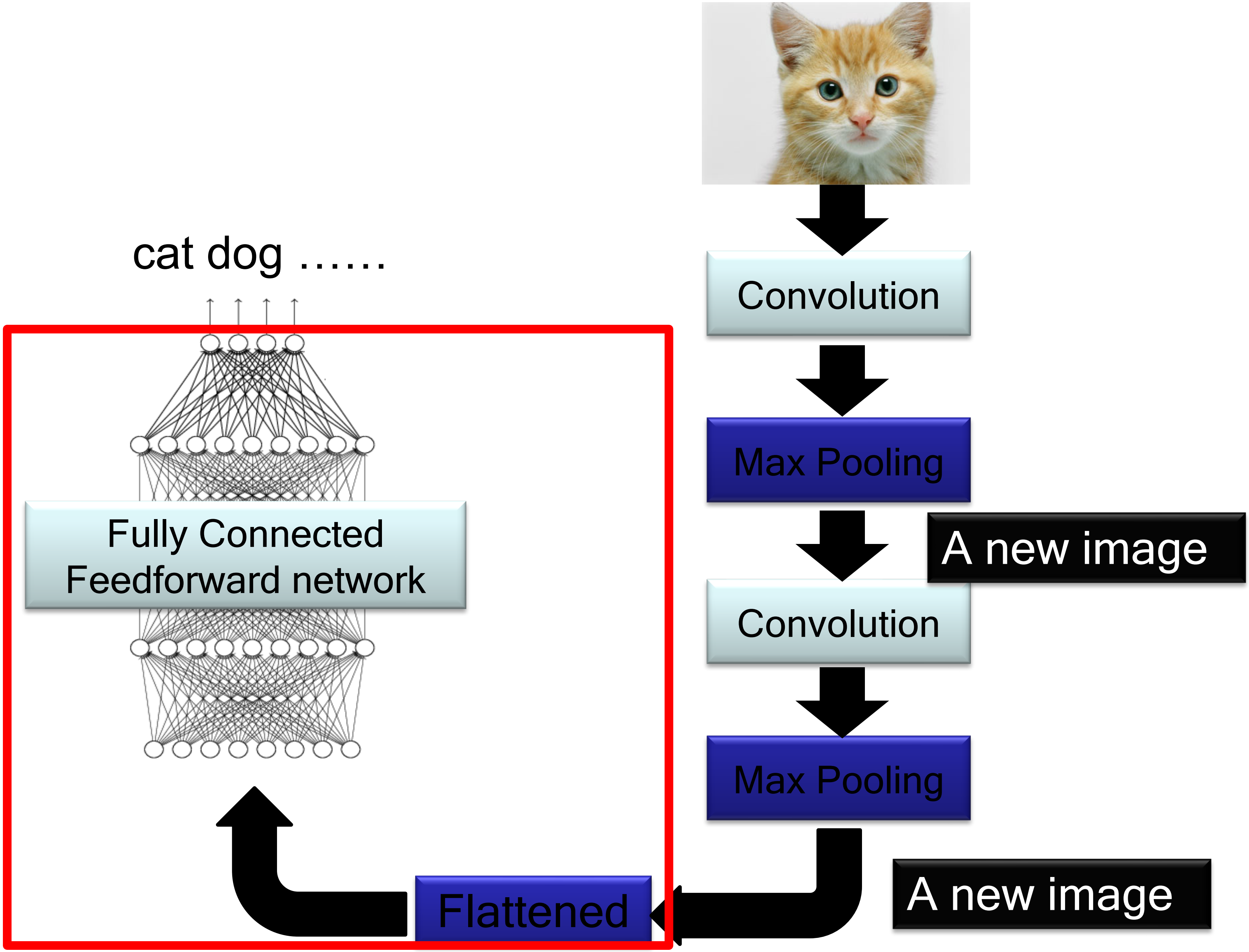
Can repeat many times

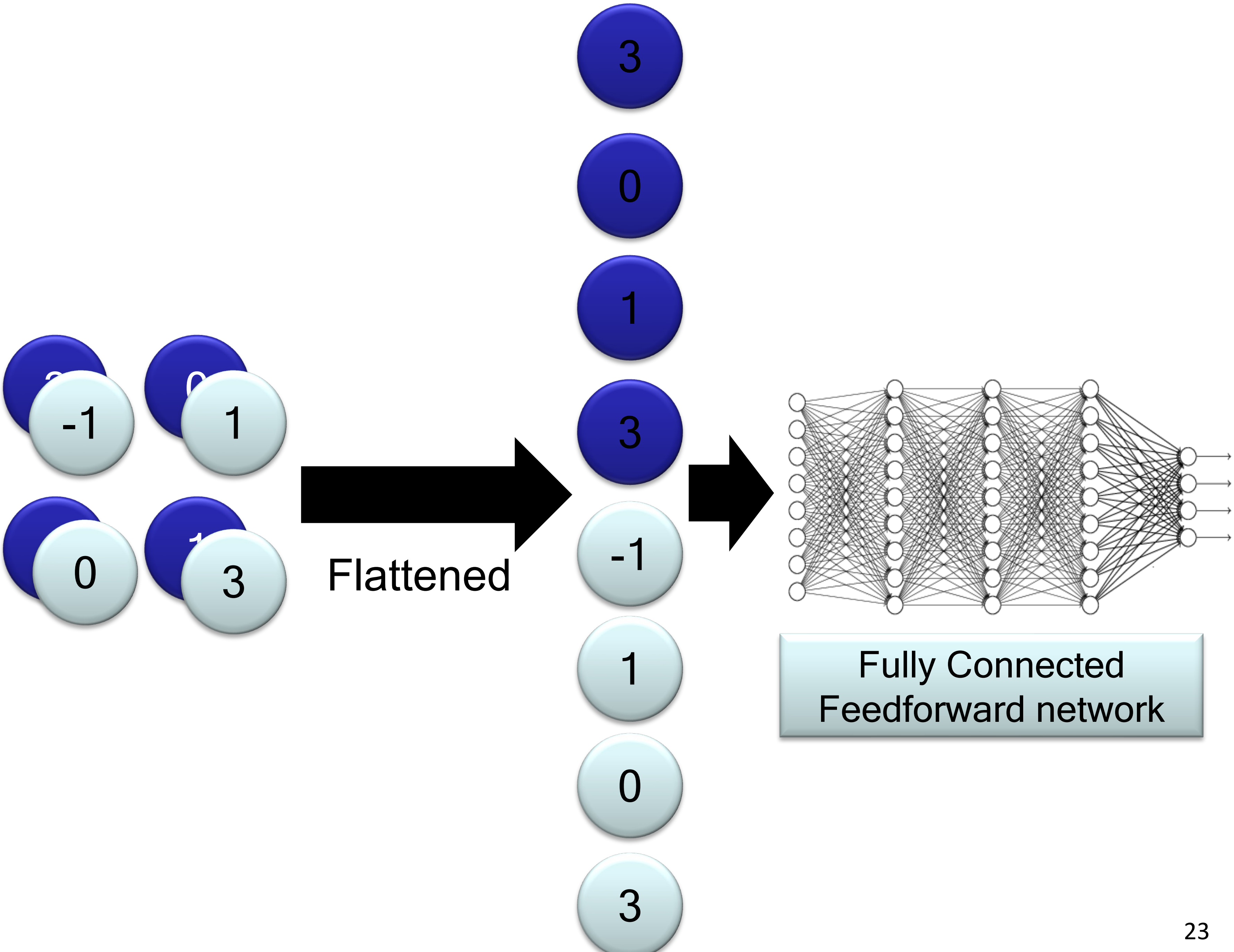


Smaller than the original image

The number of channels is the number of filters

The whole CNN

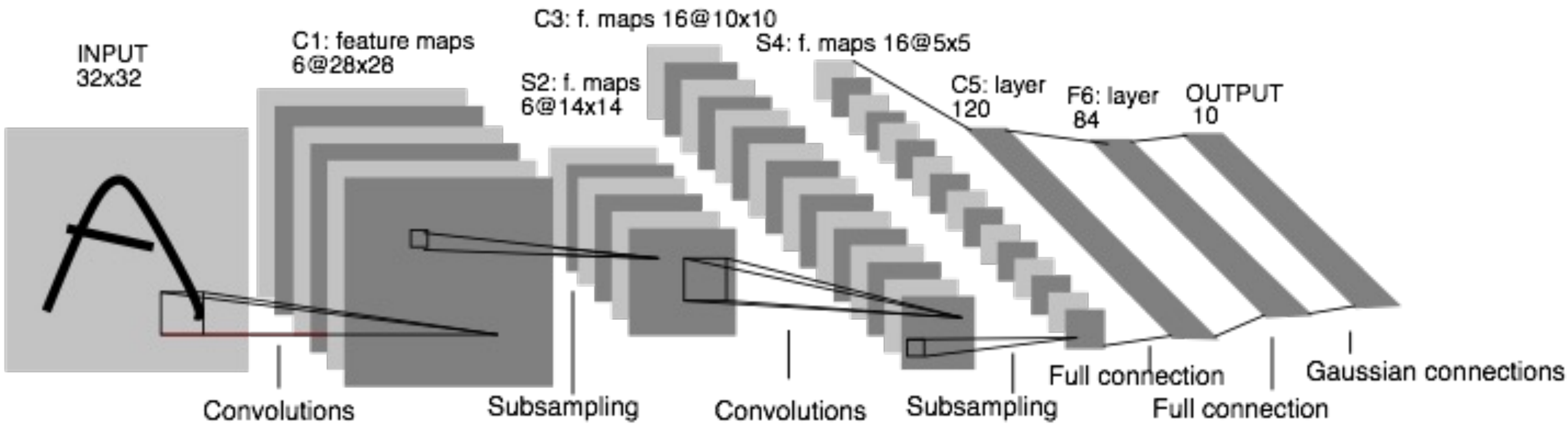



-1
1
0
3
Flattened
3
0
1
3
-1
1
0
3
Fully Connected
Feedforward network

A CNN compresses a fully connected network

- Reducing number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity

Convolutional Neural Networks in 1998



- LeNet: a layered model composed of convolution and subsampling operations followed by a holistic representation and ultimately a classifier for handwritten digits
- CPU

Color image: RGB 3 channels

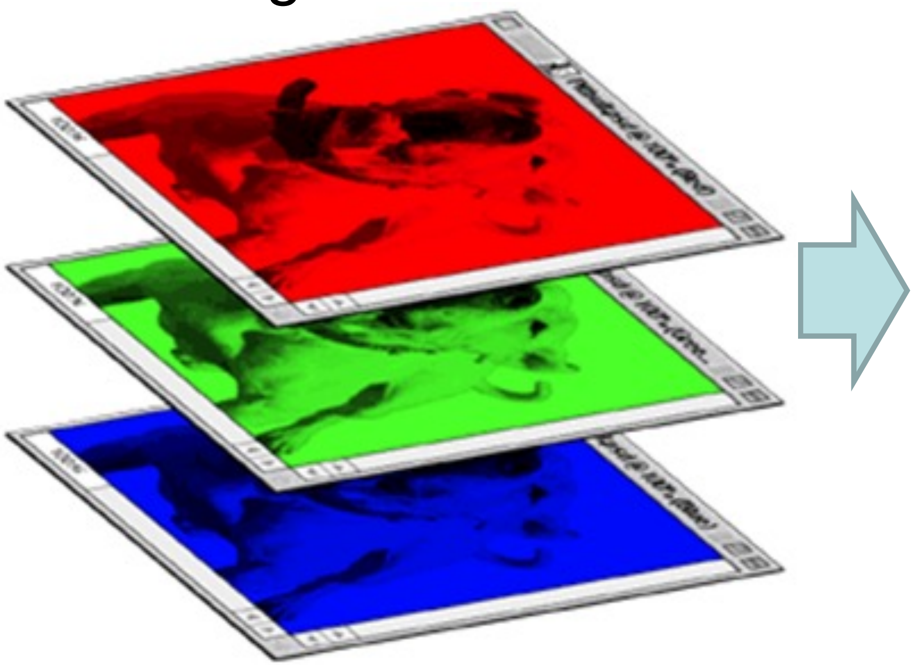
Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Filter 2

-1	1	-1
-1	1	-1
-1	1	-1

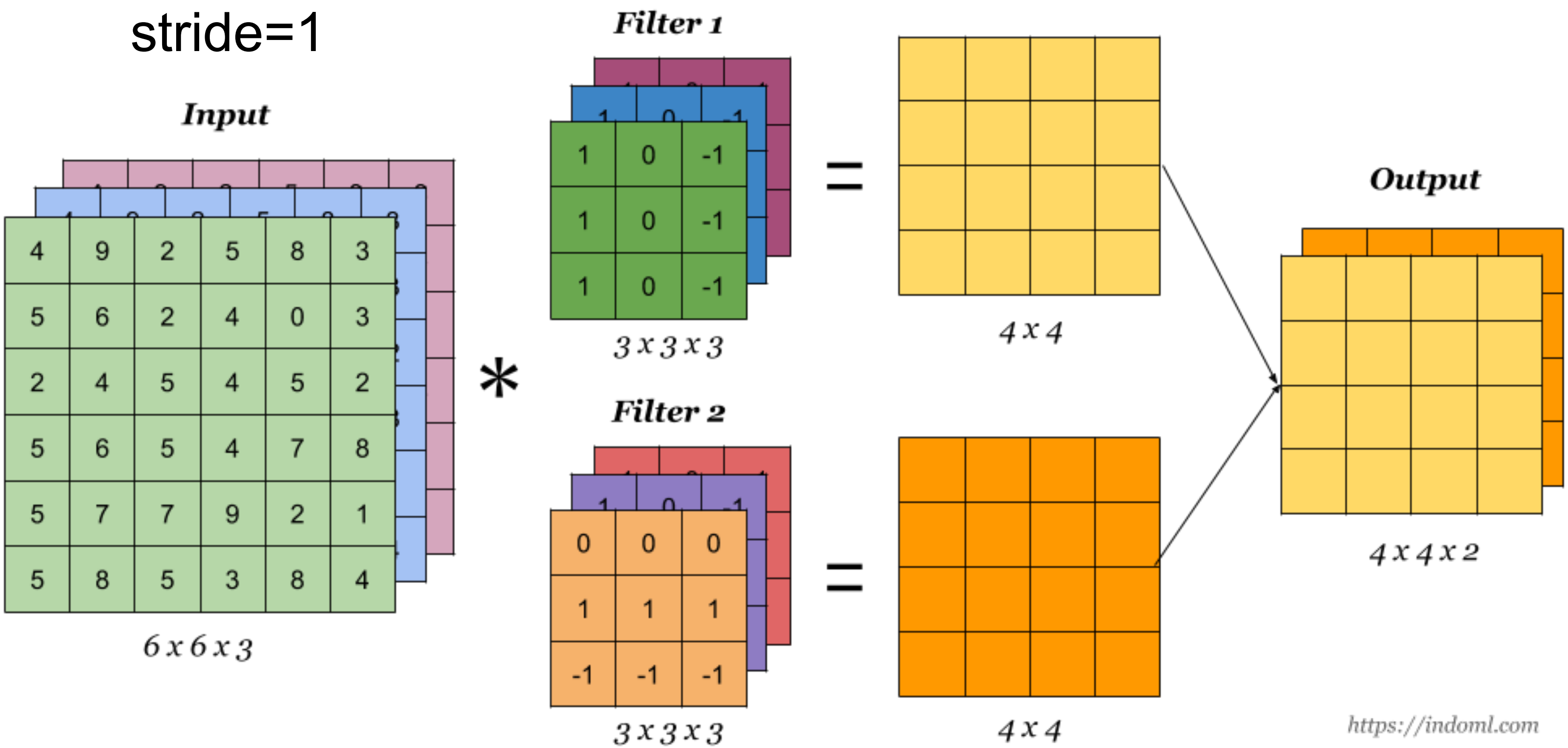
Color image



1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

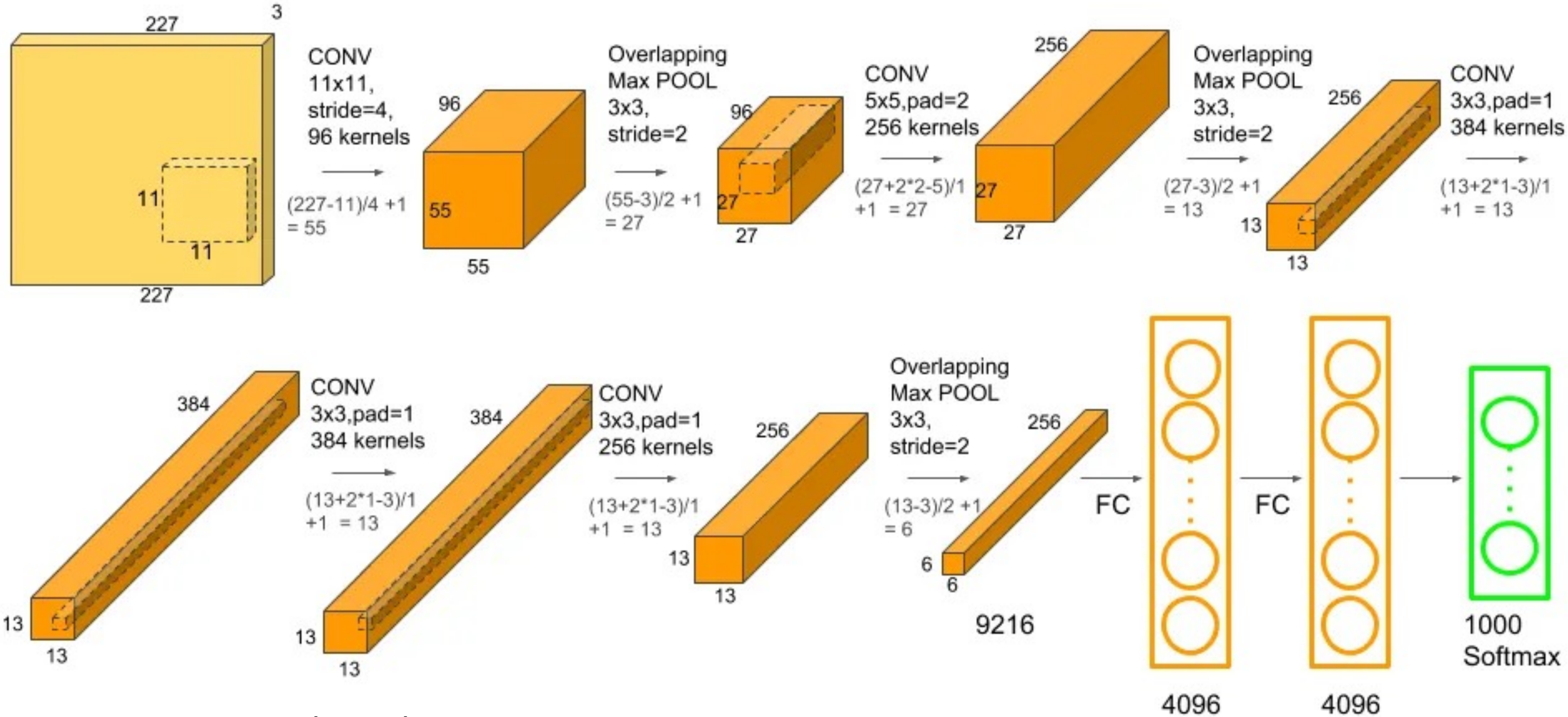
Each image can store discrete pixels with conventional brightness intensities between 0 and 255

3 channels -> depth of filters = 3



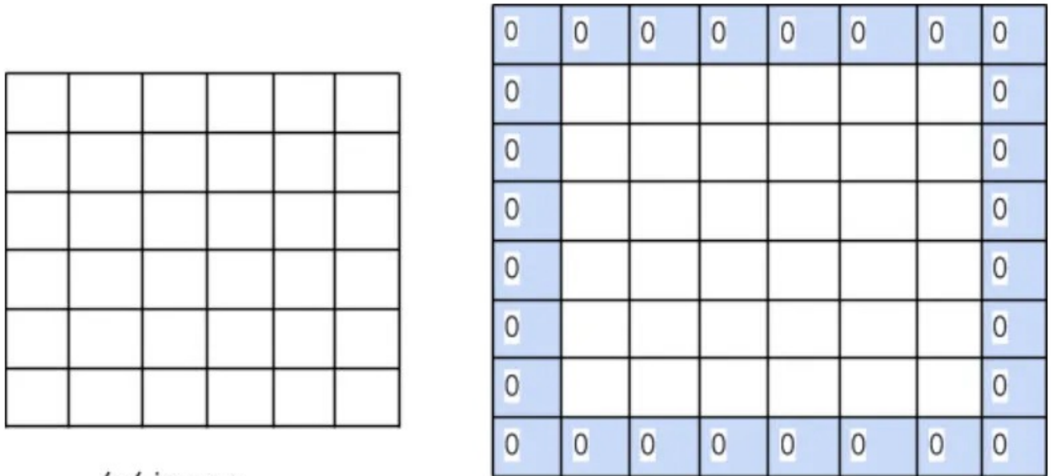
- A filter must always have the same number of channels as the input, often referred to as "depth"
- Weighted sum from 3 channels

Convolutional Neural Networks in 2012



- Input 227*227*3. GPU.
- AlexNet: a layered model composed of convolution, subsampling, and further operations followed by a holistic representation and all-in-all a landmark classifier on ImageNet Large Scale Visual Recognition Challenge 2012
- + data; + gpu; + non-saturating nonlinearity; + regularization

Padding



6x6 image

6x6 image with 1 layer of zero padding

Filter

1	0
0	0.5

Stride X

Input

0	0	0	0	0	0
0	1	0	0.5	0.5	0
0	0	0.5	1	0	0
0	0	1	0.5	1	0
0	1	0.5	0.5	1	0
0	0	0	0	0	0

Stride Y

Output

0.5	0	0.25	0.25
0	1.25	0.5	0.5
0	0.5	0.75	1.5
0.5	0.25	1.25	1

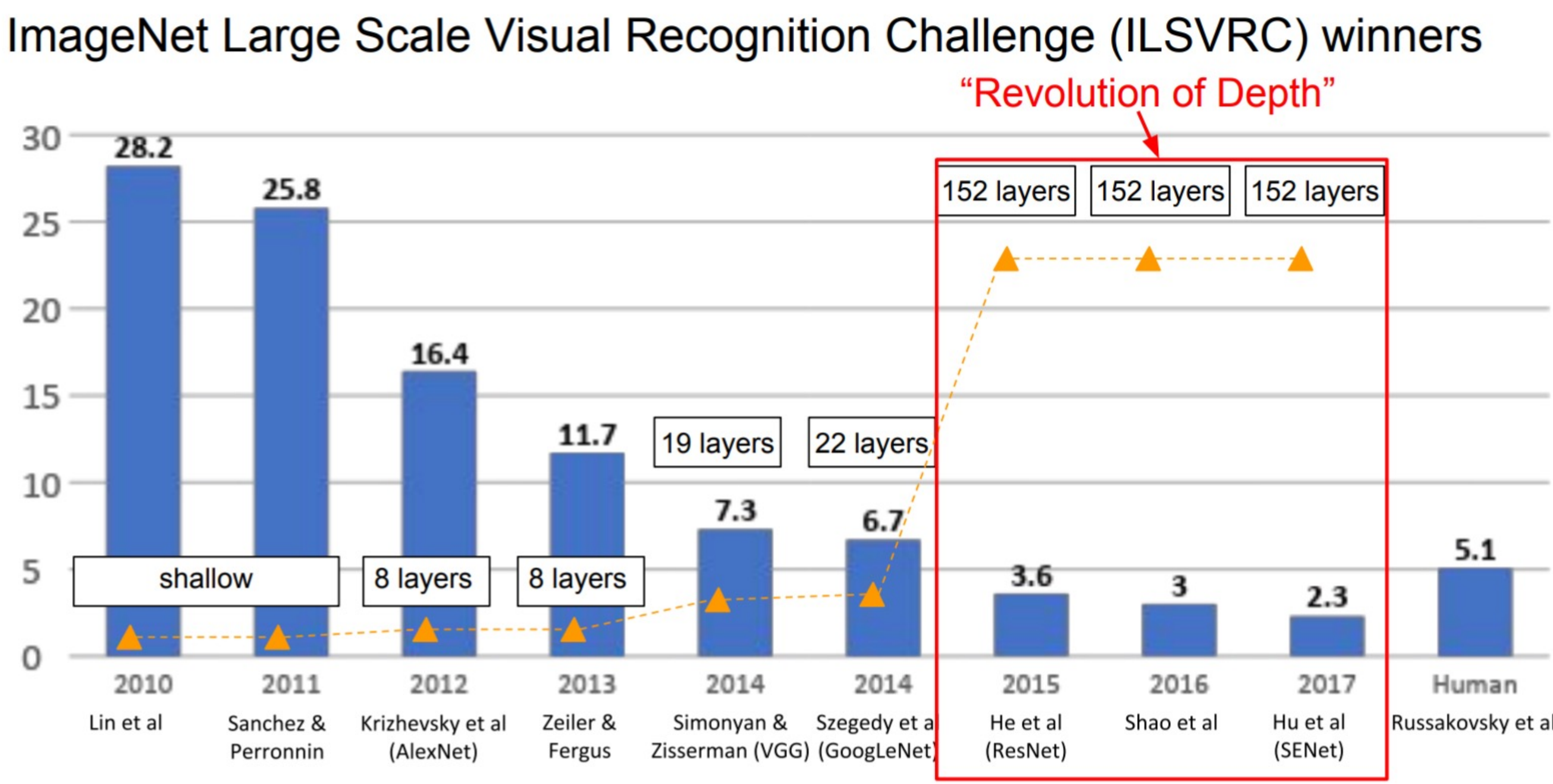
outDim = (inpDim)/strideDim

Exercise

- Suppose your input size is 64x64x16. You use a convolutional layer with 32 filters that are each 6x6, and a stride of 2 and padding of 1. What is the output size of this convolutional layer?

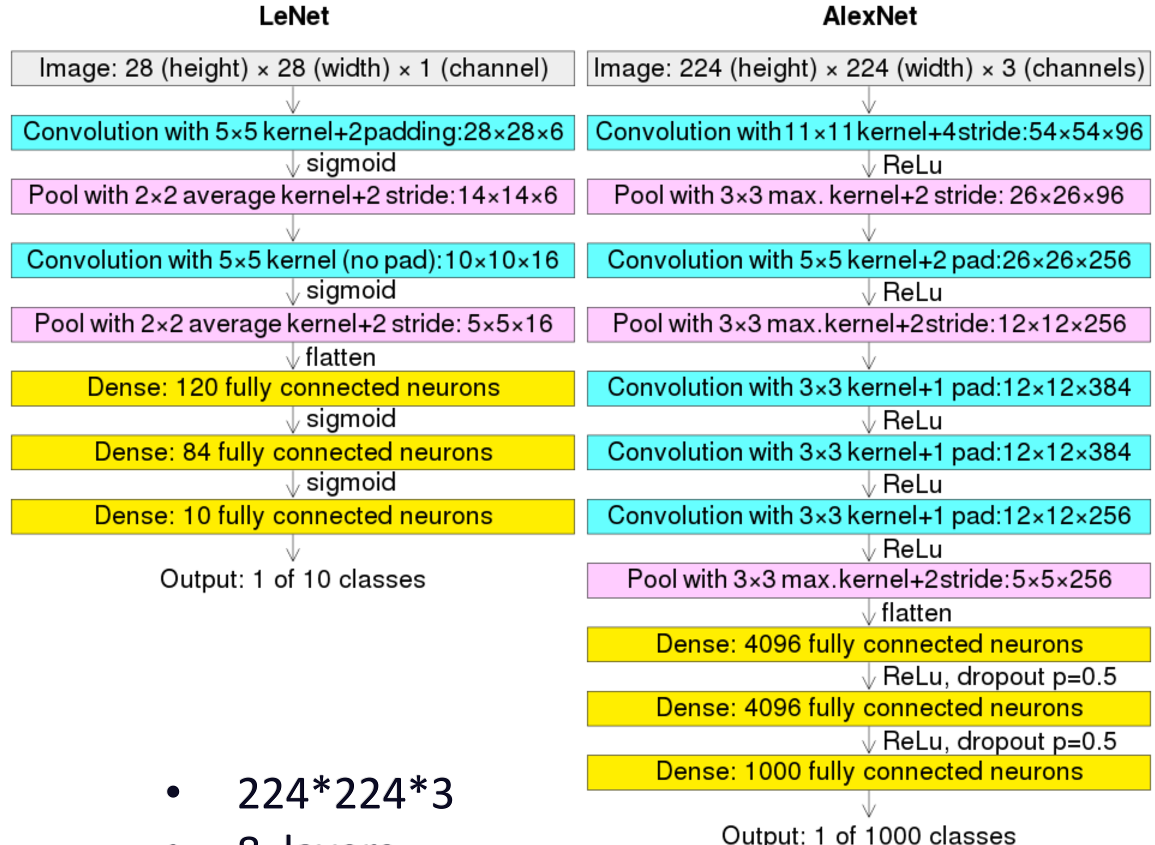
- $(64 + 2 * 1 - 6)/2 + 1 = 31$
- The output size is 31x31x32

The popular CNNs



- LeNet, 1998
- AlexNet, 2012
- VGGNet, 2014
- ResNet, 2015

LeNet vs AlexNet



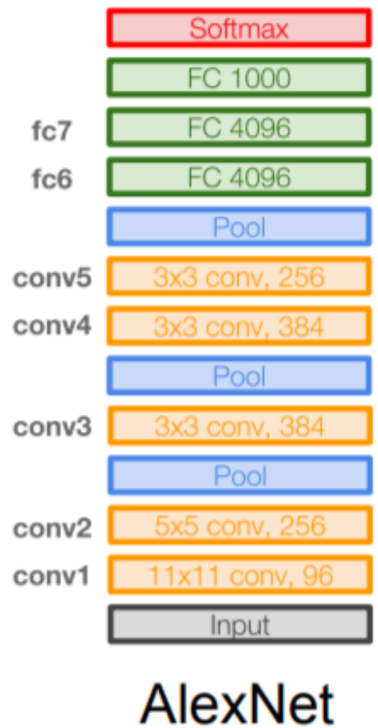
- Input: 32*32*1
- 7 layers
- 2 conv and 4 fully connected layers for classification
- 60 thousand parameters
- Only two complete convolutional layers (Conv, nonlinearities, and pooling as one complete layer)

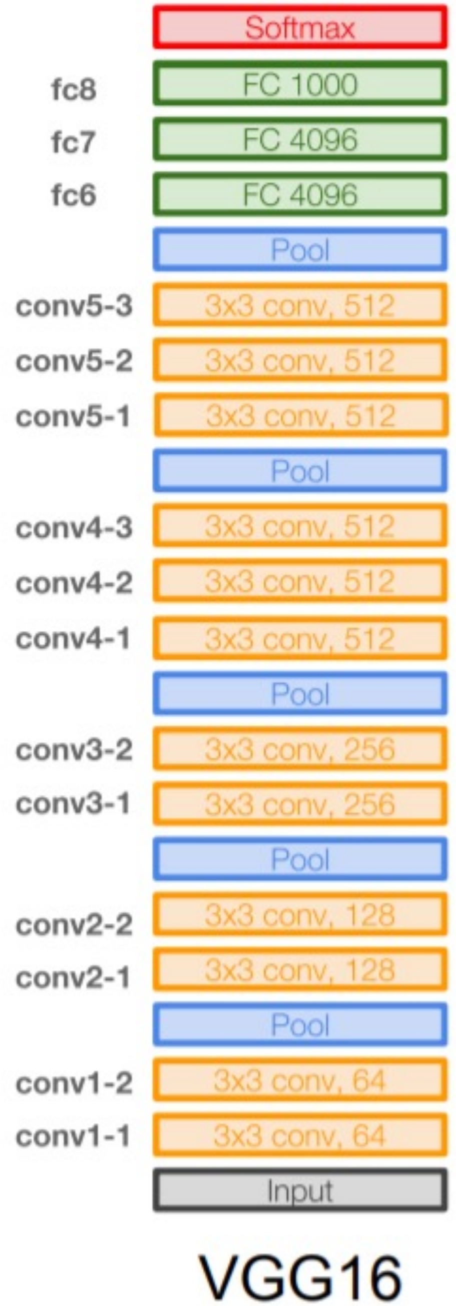
- 224*224*3
- 8 layers
- 5 conv and 3 fully classification
- 5 convolutional layers, and 3,4,5 stacked on top of each other
- Three complete conv layers

- 60 million parameters, insufficient data
- Data augmentation:
  - Patches (224 from 256 input), translations, reflections
  - PCA, simulate changes in intensity and colors

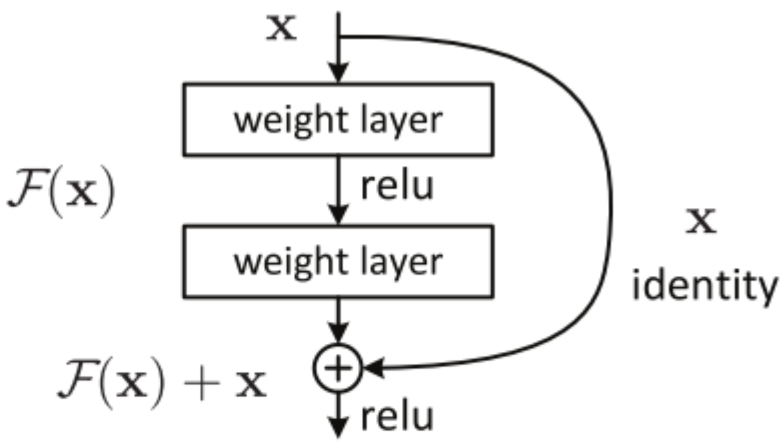
VGGNet

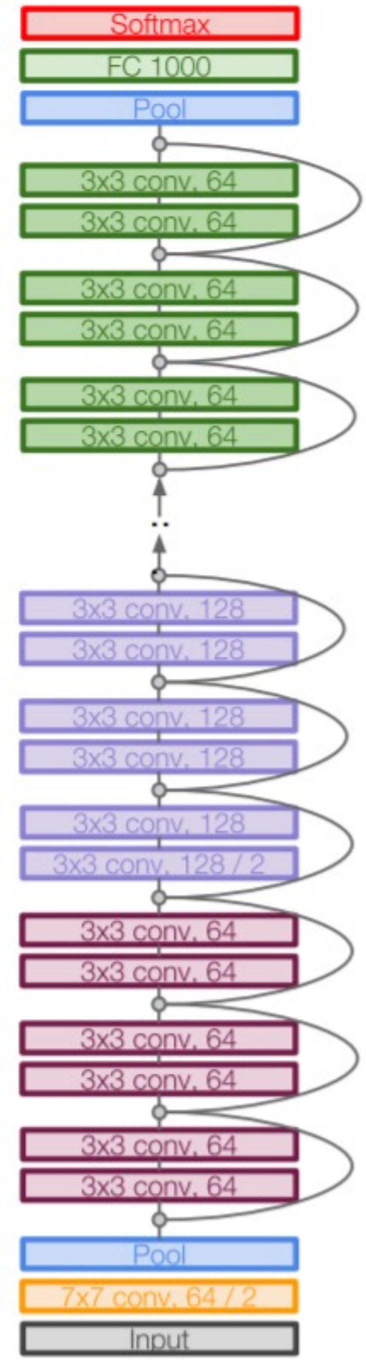
- 16 layers
- Only 3*3 convolutions
- 138 million parameters





ResNet

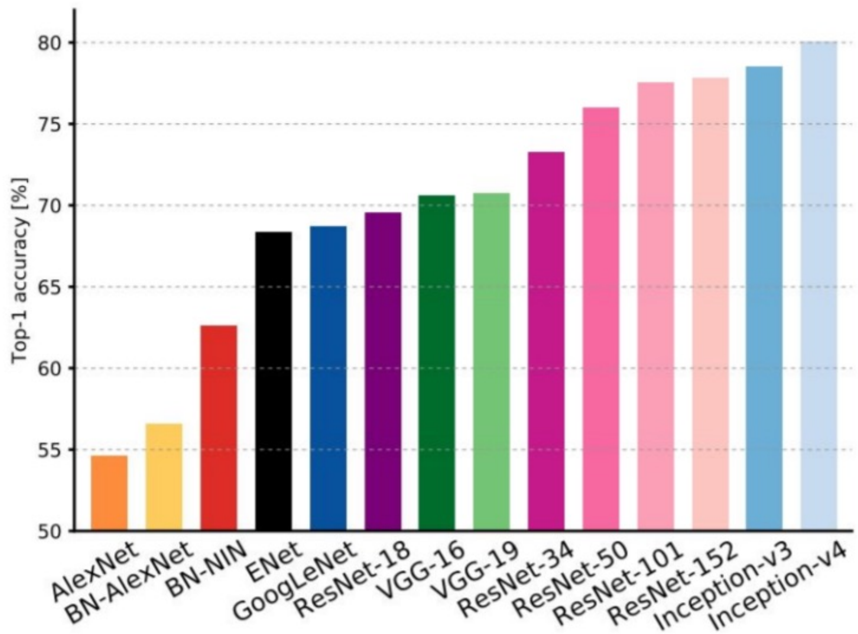


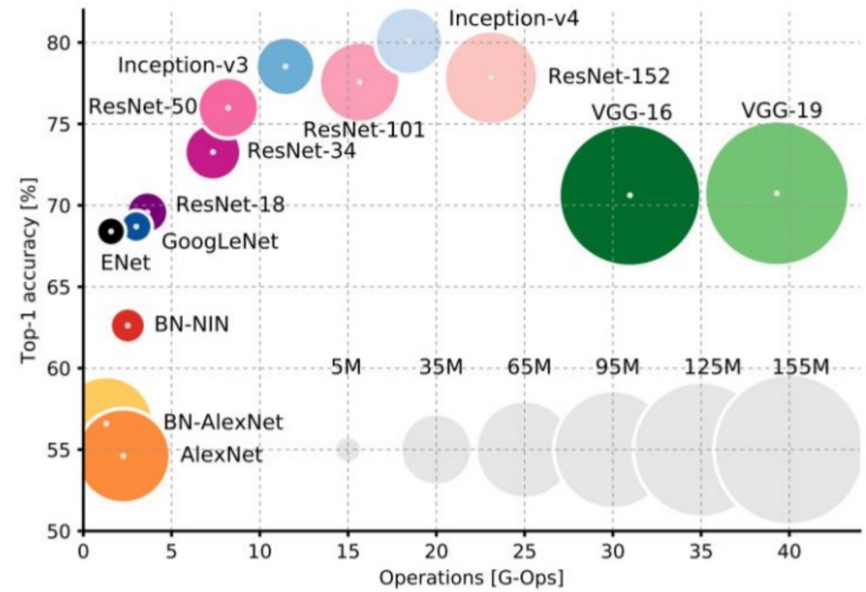


- 152 layers
- skip connections
- ResNet50

Computational complexity

Comparing complexity...

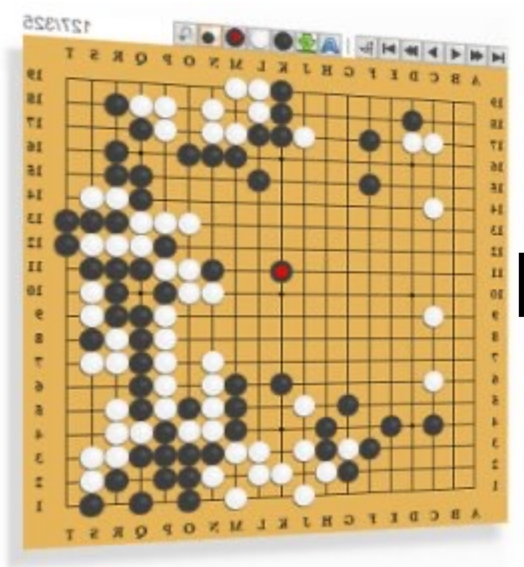




An Analysis of Deep Neural Network Models for Practical Applications, 2017.

- The memory bottleneck
- GPU, a few GB

CNN Application 1: AlphaGo



19 x 19 matrix

Black: 1

white: -1

none: 0

Neural Network

Next move

(19 x 19 positions)

Fully-connected feedforward network can be used

But CNN performs much better

AlphaGo's policy network

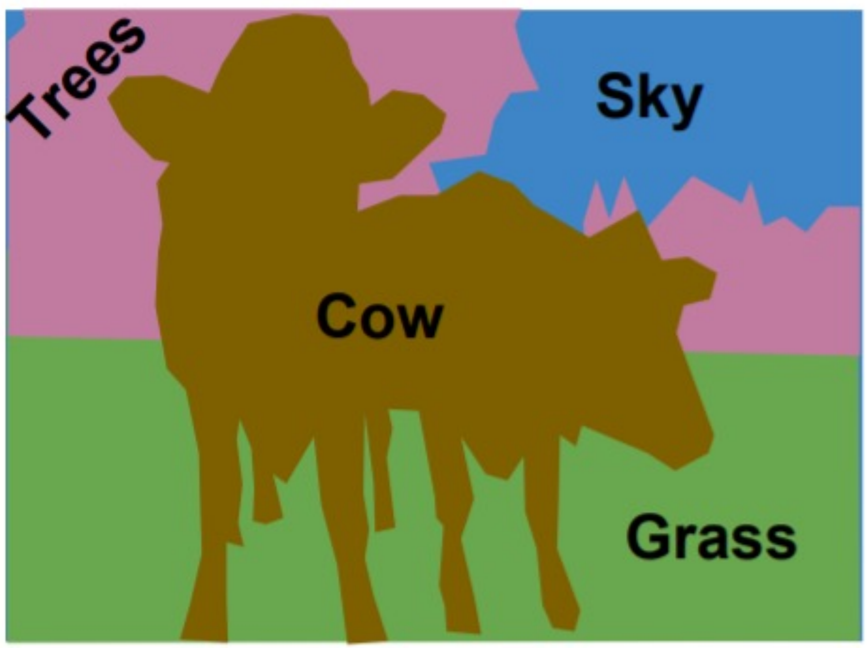
The following is quotation from their Nature article:

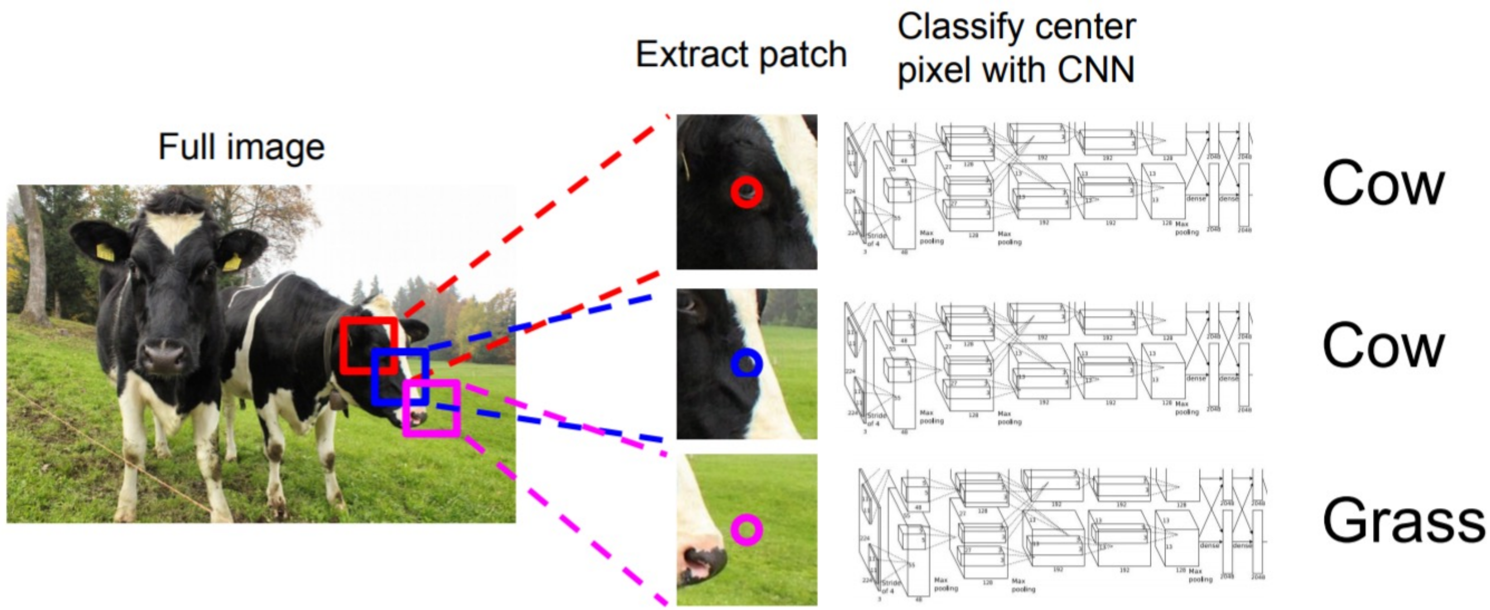
Note: AlphaGo does not use Max Pooling.

Neural network architecture. The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21×21 image, then convolves k filters of kernel size 3×3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1×1 with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Table 3 additionally show the results of training with $k = 128$, 256 and 384 filters.

CNN application 2: Semantic segmentation

This image is CC0 public domain





Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

He et al, “Mask R-CNN”, arXiv 2017