POLITECNICO MILANO 1863

Cognitive Robotics

2018/2019

Convolutional Neural Networks

Matteo Matteucci

matteo.matteucci@polimi.it

Artificial Intelligence and Robotics Lab - Politecnico di Milano

Neural Networks for Image Recognition

Too hard to train!

Prior Art

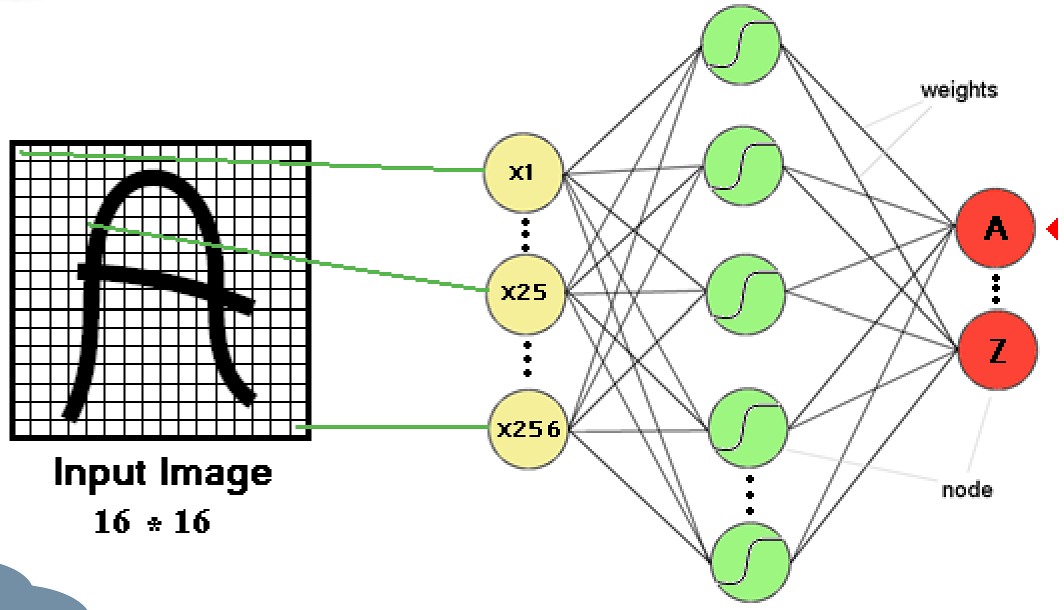
Layer of Perceptrons

Layer of Perceptrons

Layer of Perceptrons

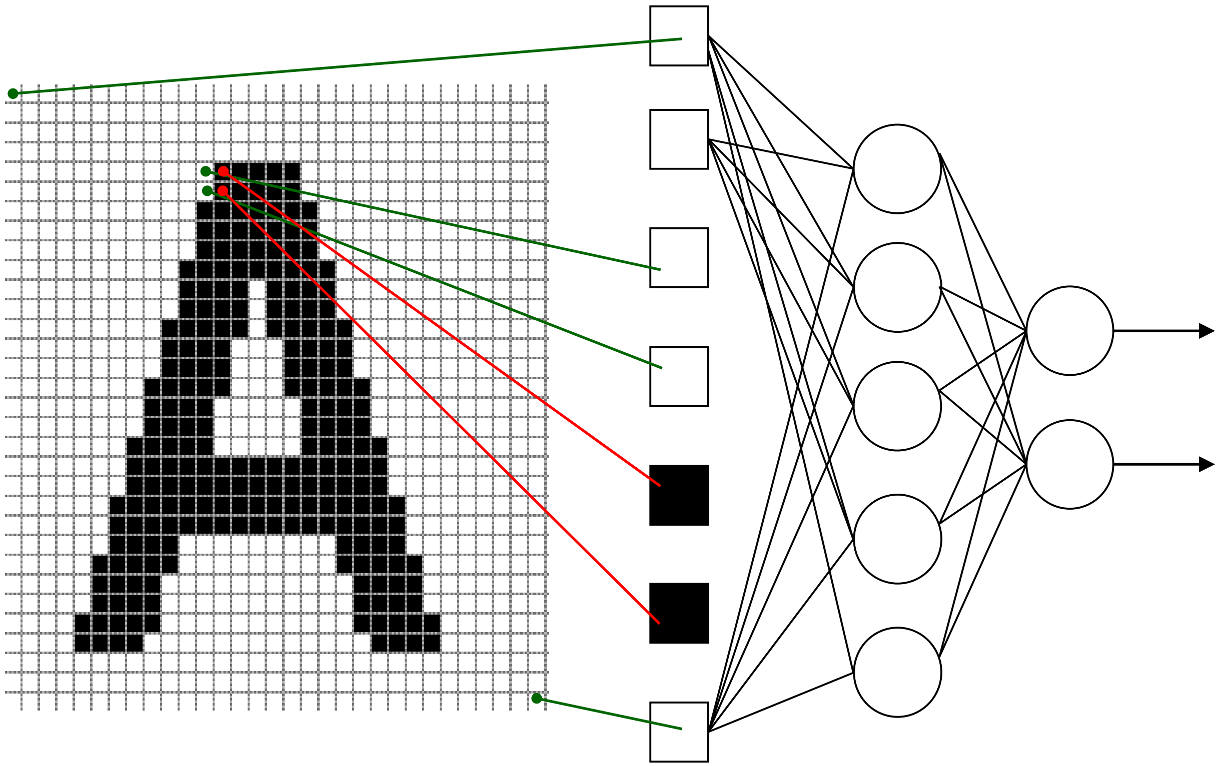
Image

Too sensitive to noise!



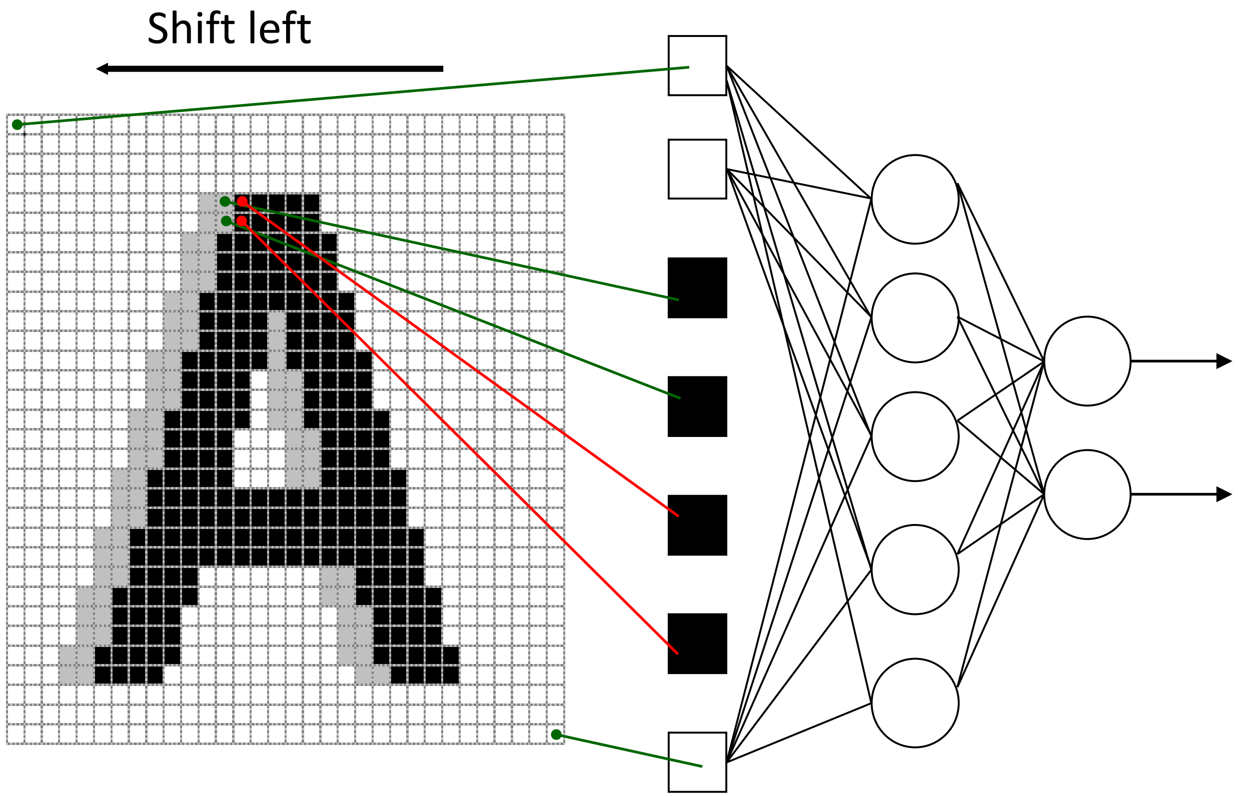
Feed Forward Networks Drawbacks

FFNN have little, if any, invariance to shifting, scaling, and other forms of distortion



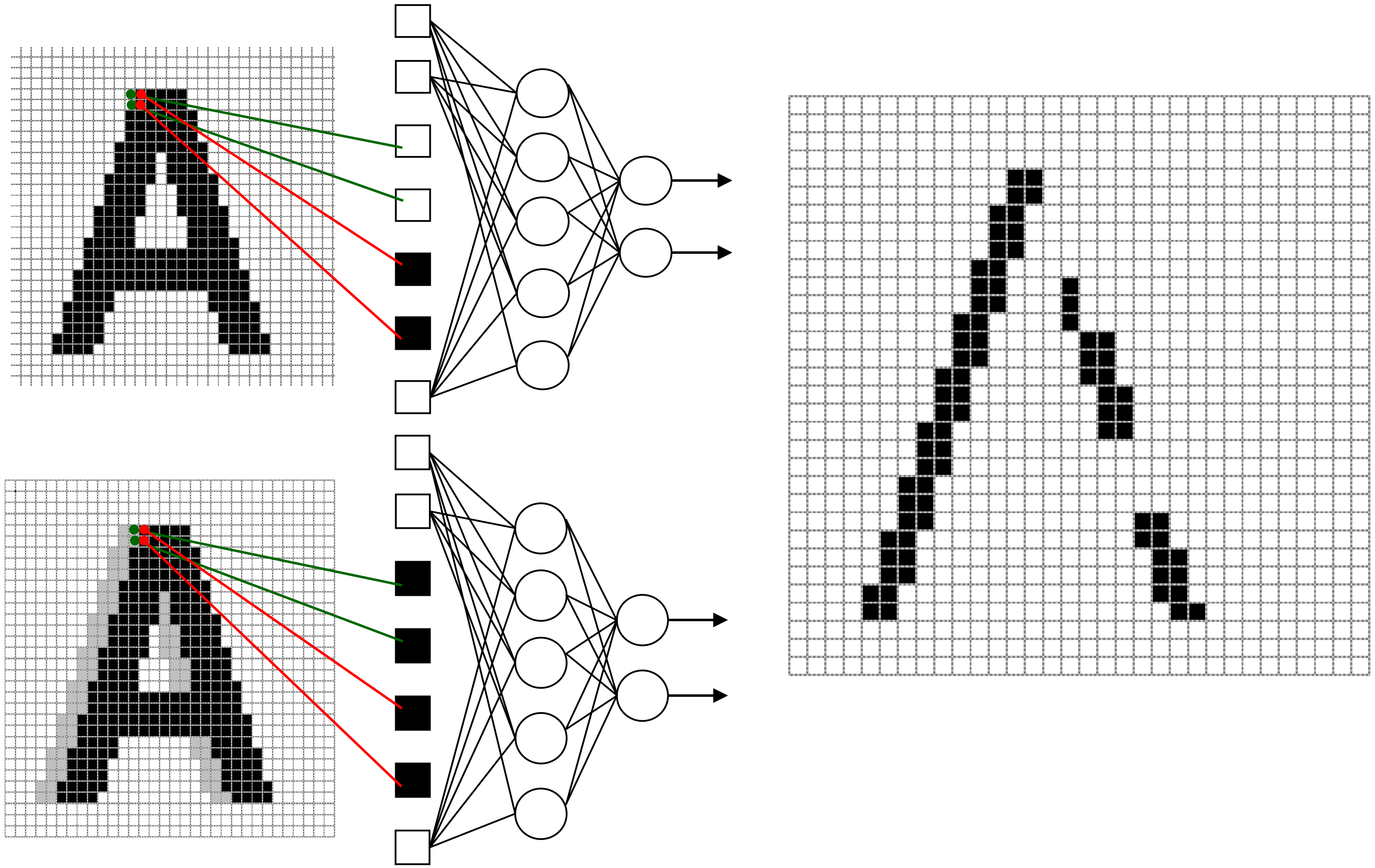
Feed Forward Networks Drawbacks

FFNN have little, if any, invariance to shifting, scaling, and other forms of distortion



Feed Forward Networks Drawbacks

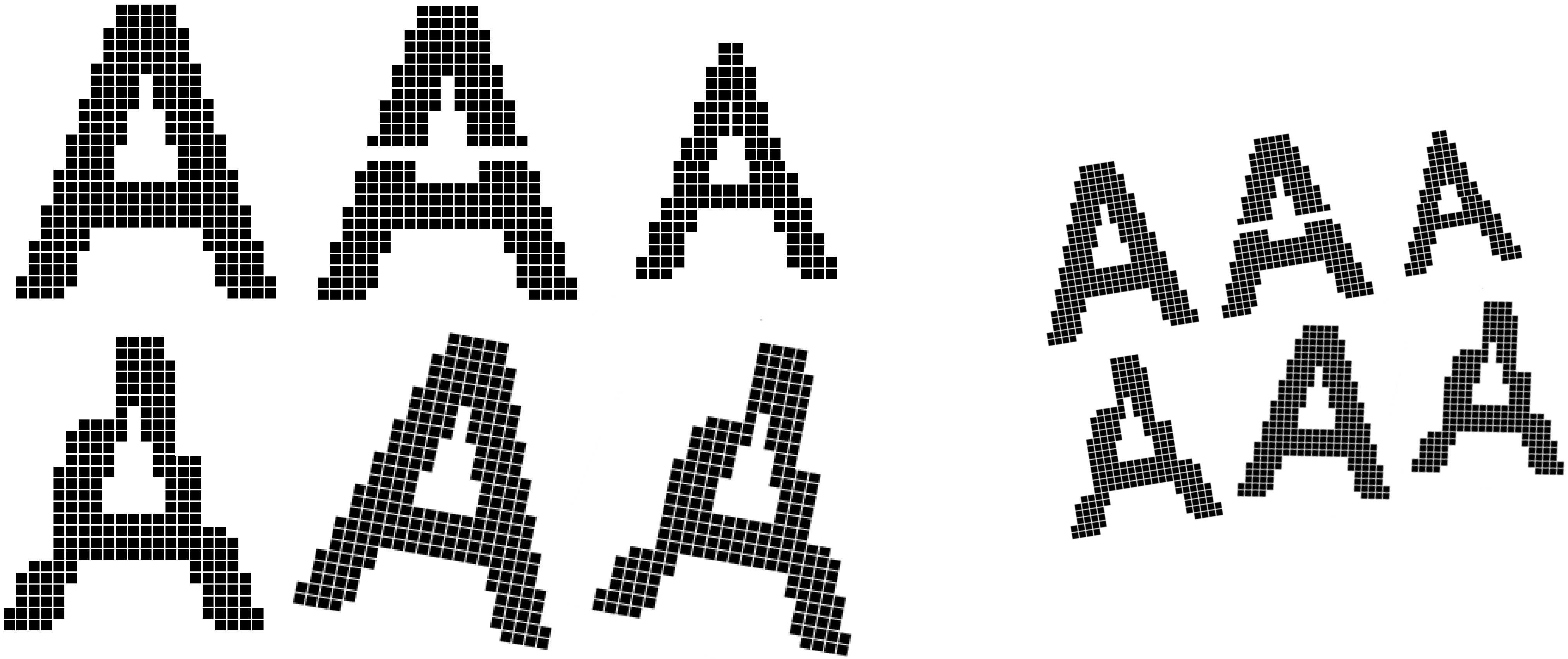
FFNN have little, if any, invariance to shifting, scaling, and other forms of distortion



Almost 150 input change from just a 2px shift left

Feed Forward Networks Drawbacks

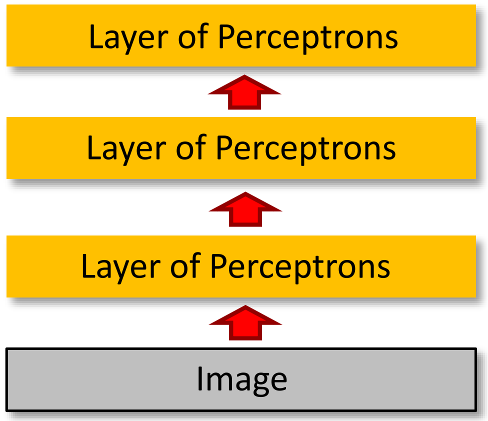
FFNN have little, if any, invariance to shifting, scaling, and other forms of distortion



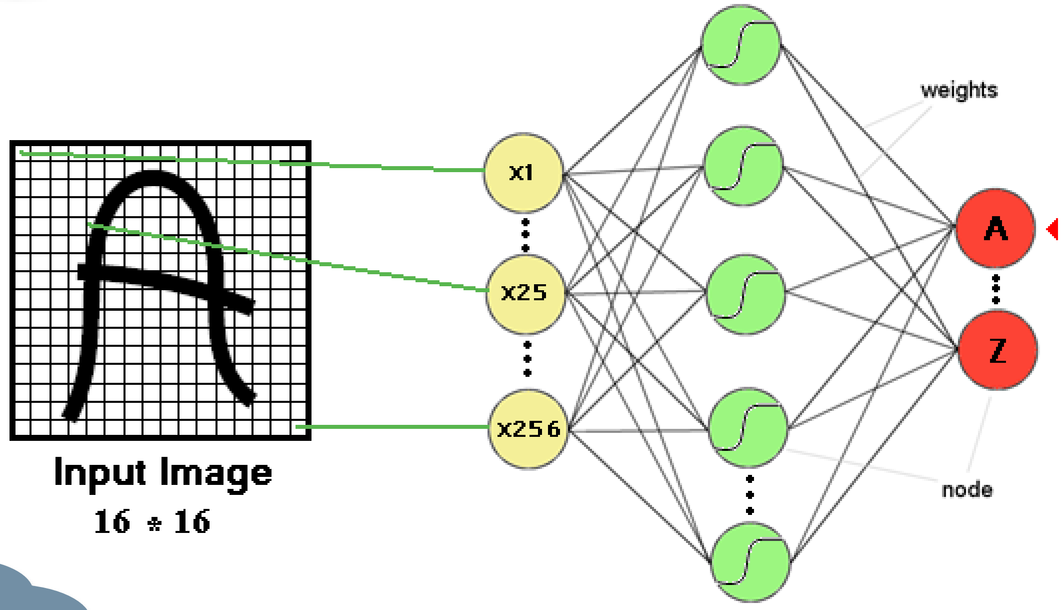
State of the Art in Image Recognition

Too hard to train!

Prior Art



Too sensitive to noise!



State of the Art in Image Recognition

Prior Art

Layer of Perceptrons

↑

Layer of Perceptrons

↑

Layer of Perceptrons

↑

Image

⇒

Previous State of Art

SVM

↑

Pooling

↑

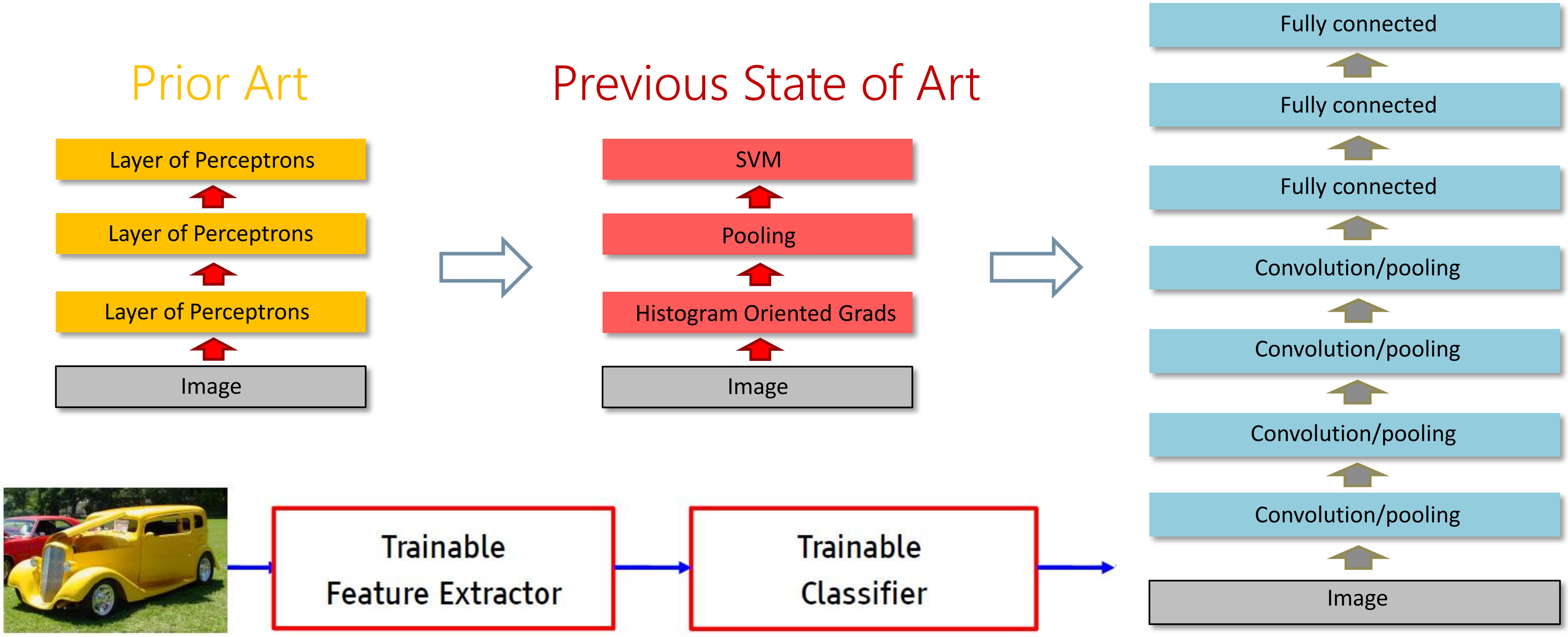
Histogram Oriented Grads

↑

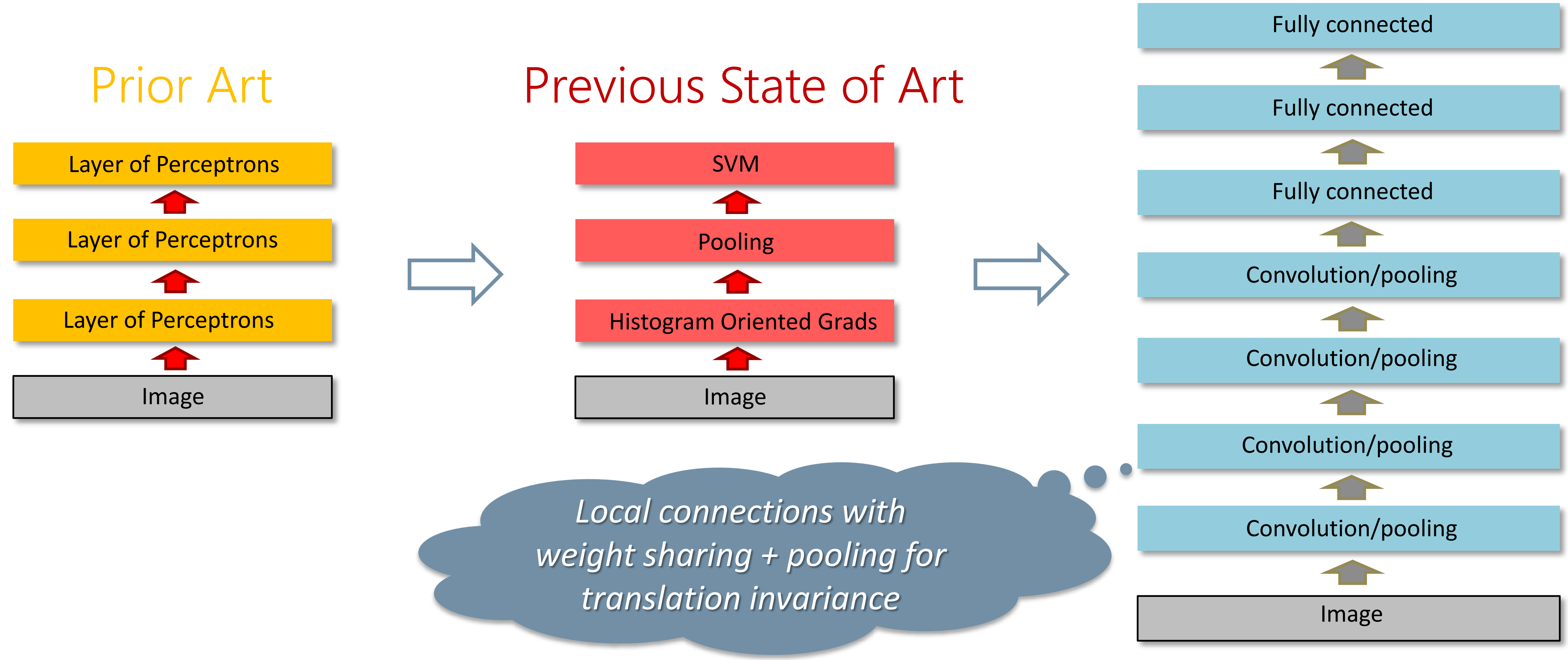
Image

→ hand-crafted Feature Extractor → "Simple" Trainable Classifier →

State of the Art in Image Recognition



State of the Art in Image Recognition



Spatial Convolutions

A spatial convolution implement a spatial filtering

$$(a \star b)[i,j] = \sum_{i',j'} a[i',j']\,b[i-i',j-j']$$

Different filters (weights) reveal a different characteristics of the input

b

$$* \; \frac{1}{8} \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & 4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

a

$a * b$

Spatial Convolutions

A spatial convolution implement a spatial filtering

$$(a \star b)[i,j] = \sum_{i',j'} a[i',j']\, b[i-i', j-j']$$

Different filters (weights) reveal a different characteristics of the input

b

$*$

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

a

$a * b$

Spatial Convolutions

A spatial convolution implement a spatial filtering

$$(a \star b)[i,j] = \sum_{i',j'} a[i',j']b[i-i',j-j']$$

Different filters (weights) reveal a different characteristics of the input

b

$*$

0	-1	0
-1	4	-1
0	-1	0

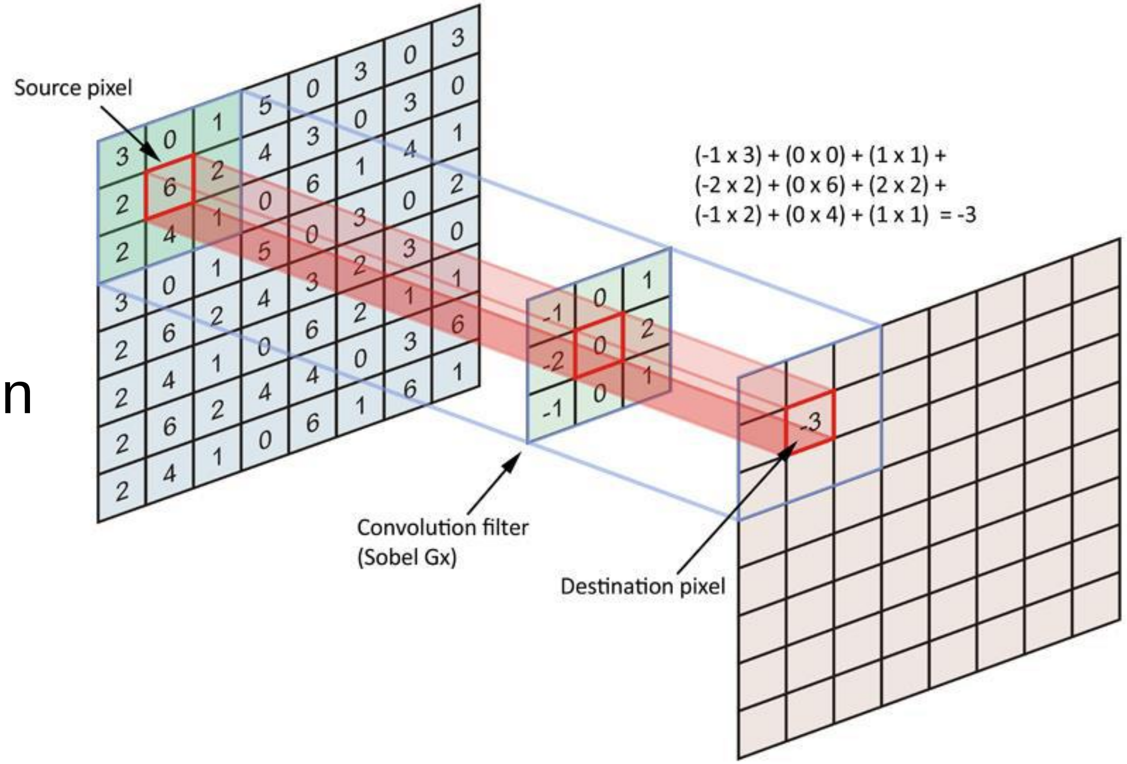
a

$a * b$

Discrete Convolutions

The convolution layer performs the following

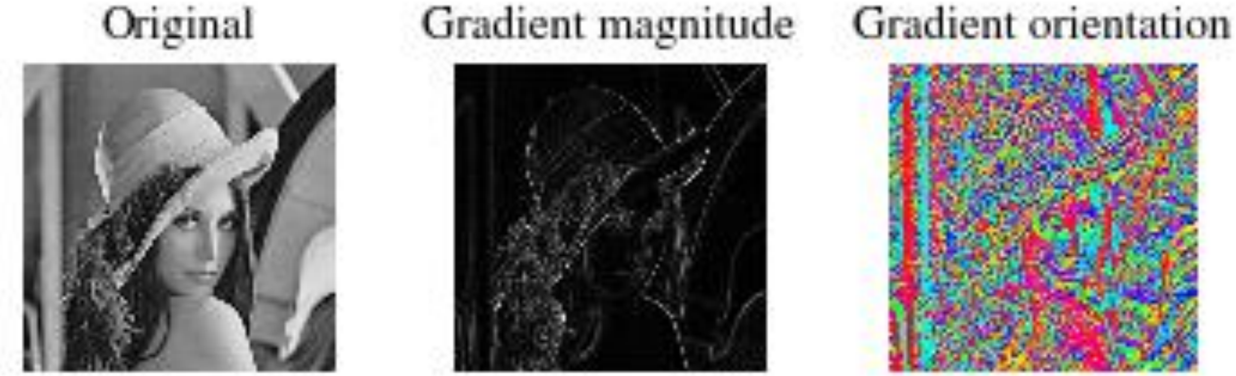
- A Kernel (shaded area) slides over input feature map (blue)
- Elementwise products computed between the kernel and the overlapped input
- Result is summed up and constitute the output feature map (cyan)



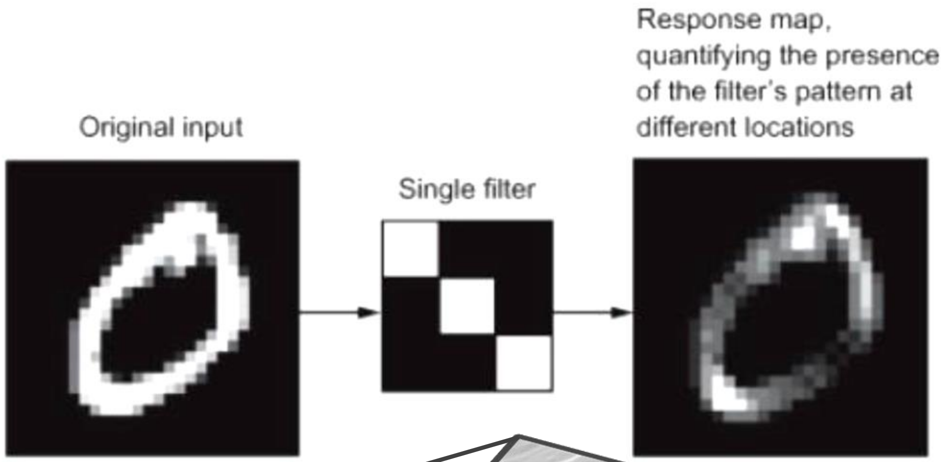
Well know tool in image processing

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1



Deep Convolutional Neural Networks for Image Recognition



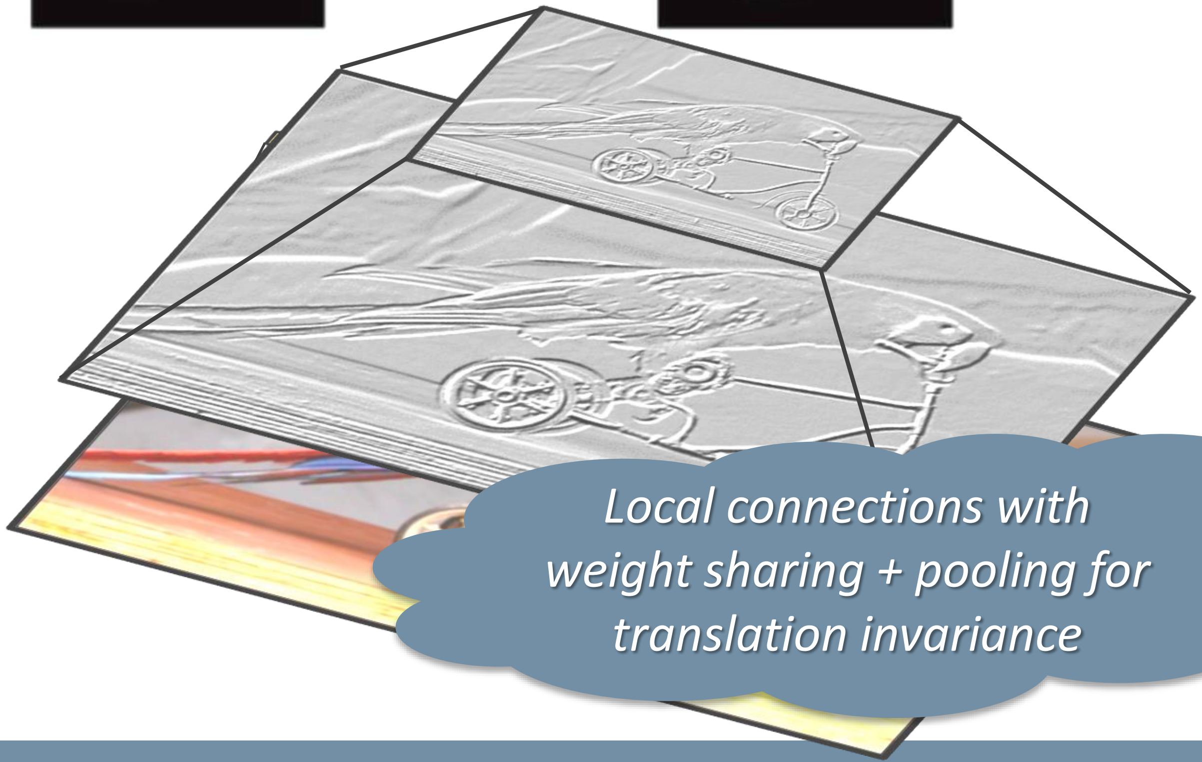
Pooling

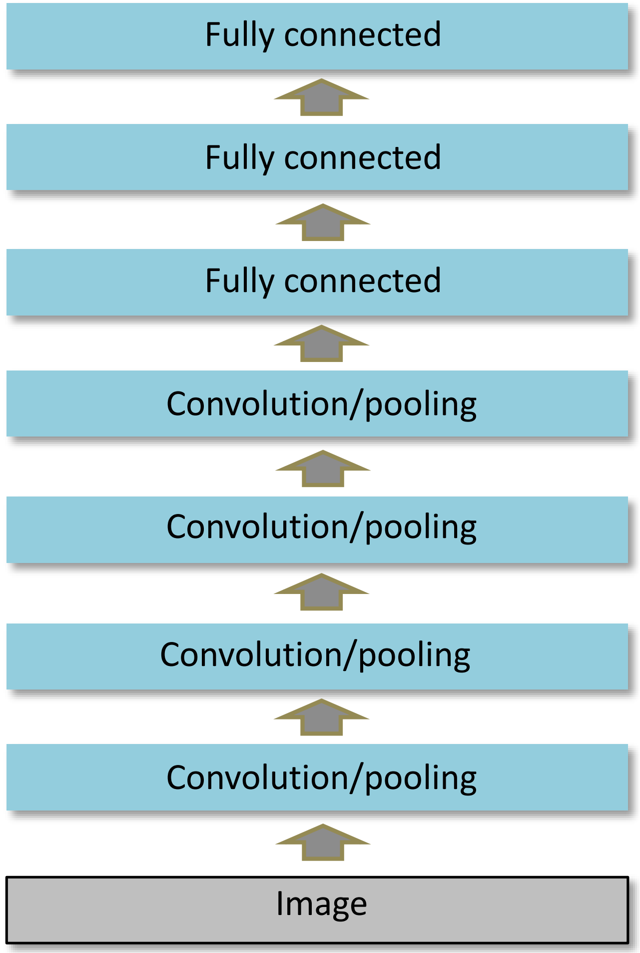
⇧

Convolution

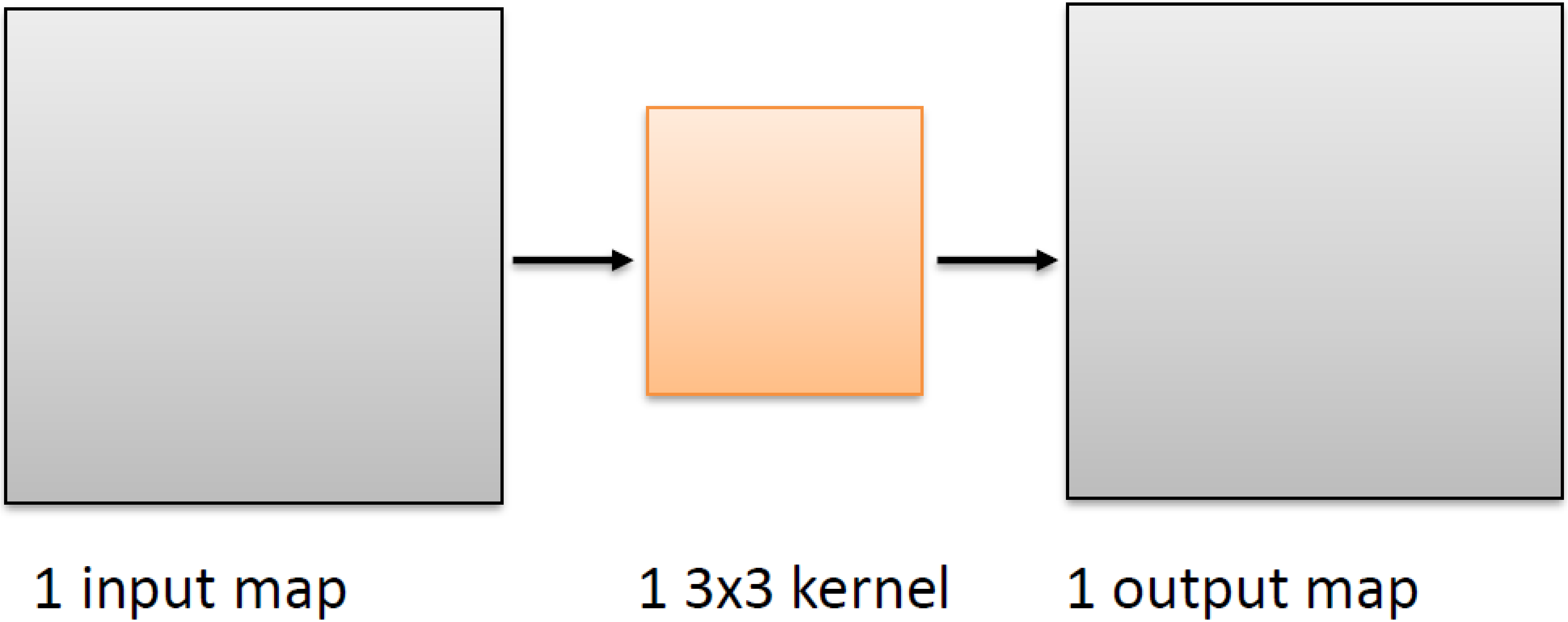
⇧

Image

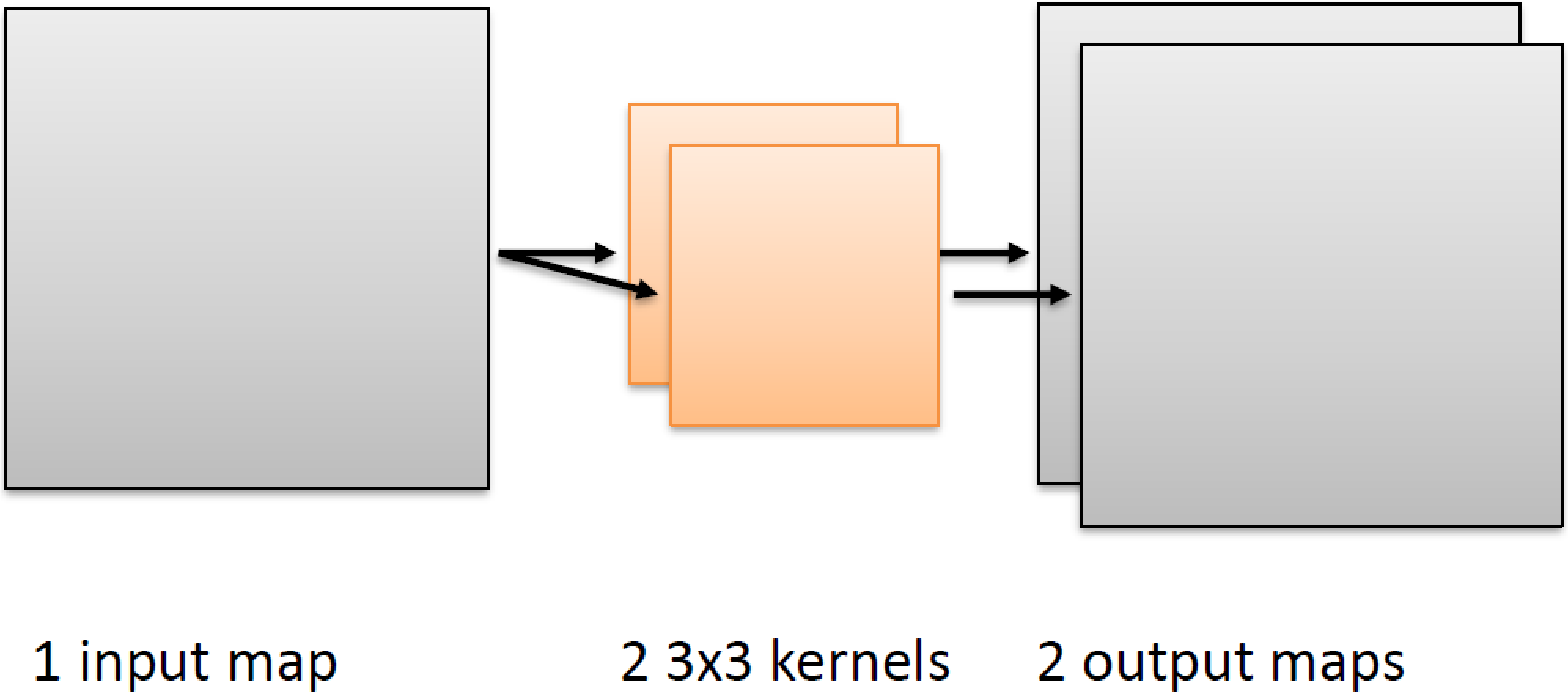




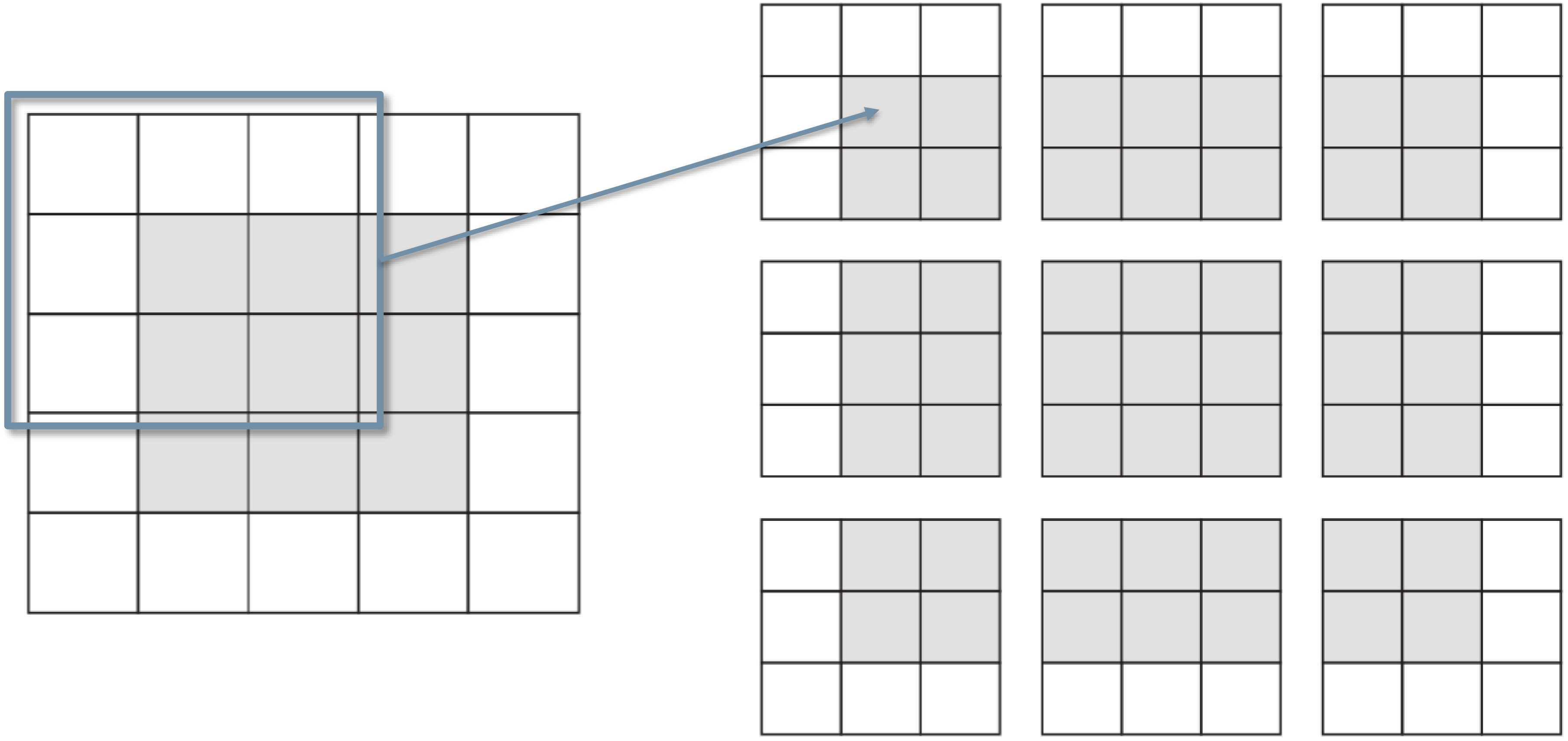
Dealing with multiple maps



Dealing with multiple maps



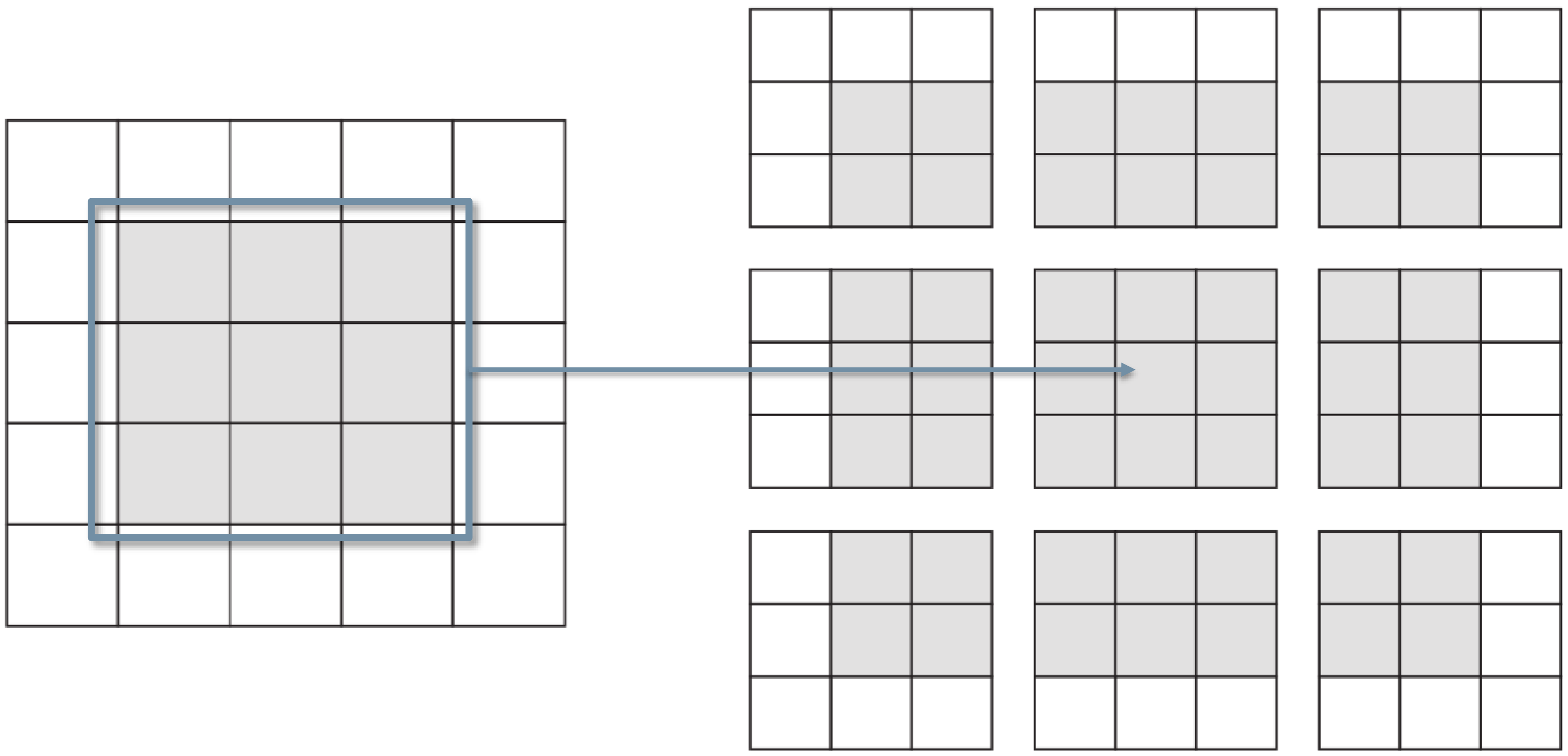
Convolutions details: Padding



Input map: 5x5

Output map: 3x3

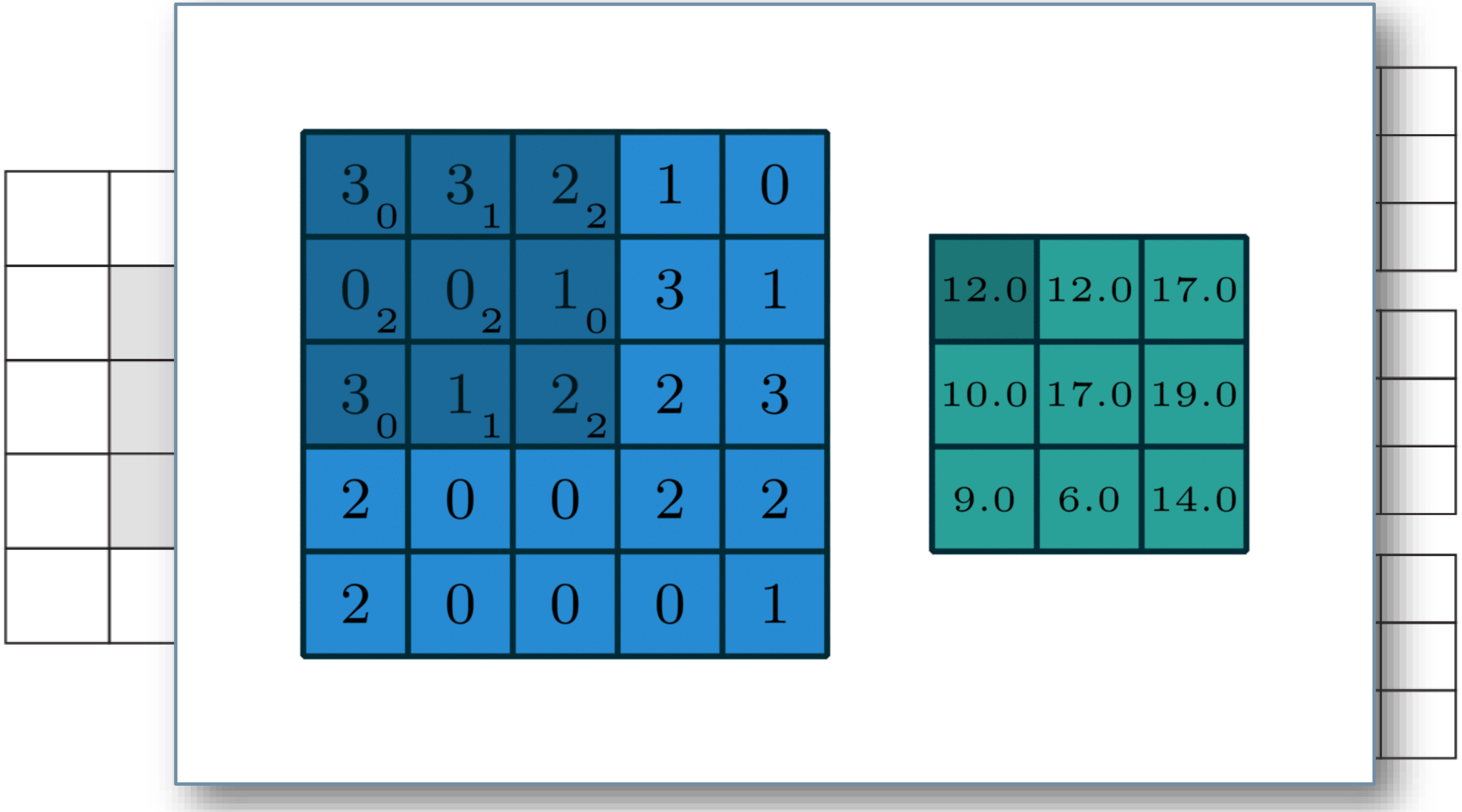
Convolutions details: Padding



Input map: 5x5

Output map: 3x3

Convolutions details: Padding

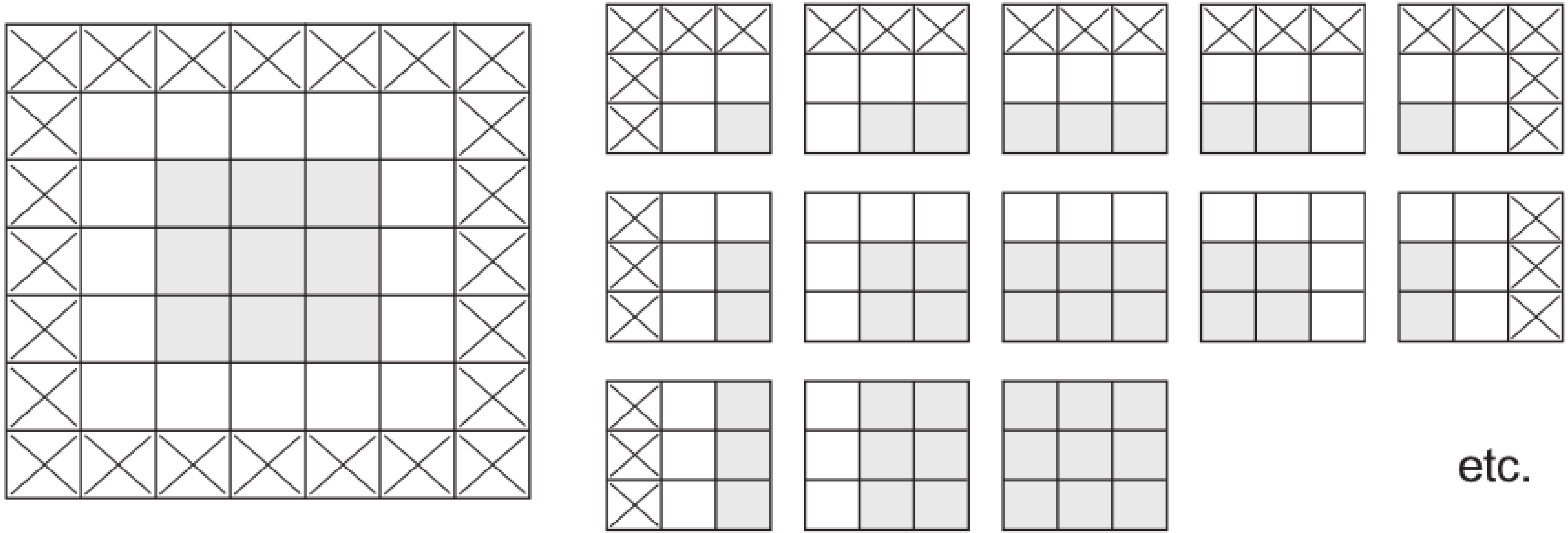


Input map: 5x5

Output map: 3x3

Convolutions details: Padding

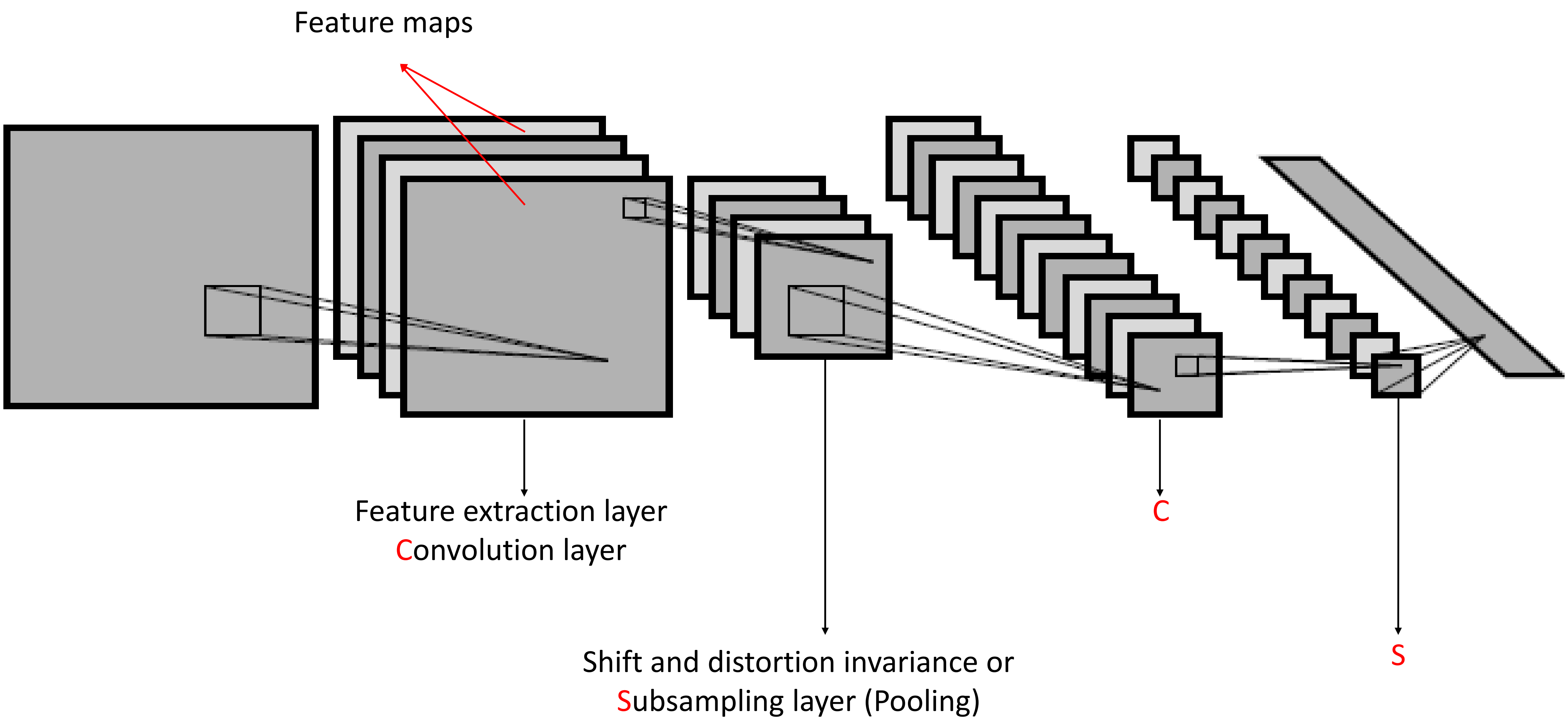
You can have same size onvolutions by zero padding:



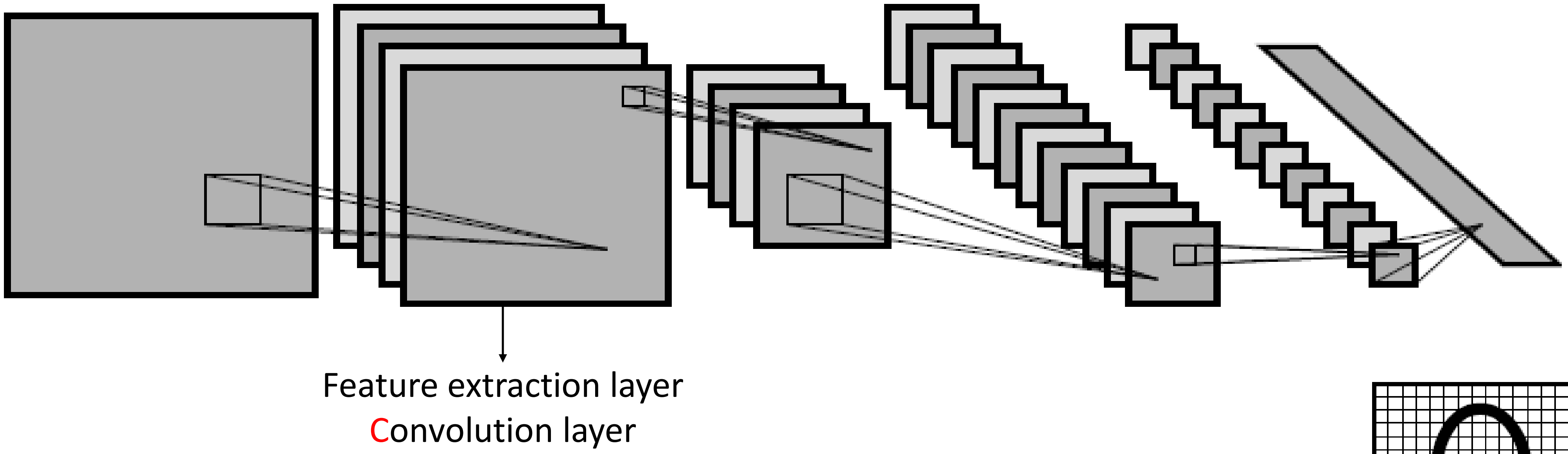
Input map: 5x5

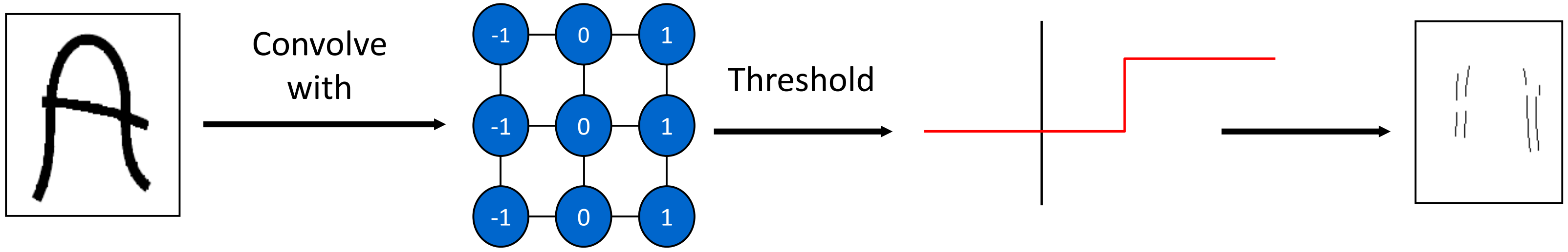
Output map: 5x5

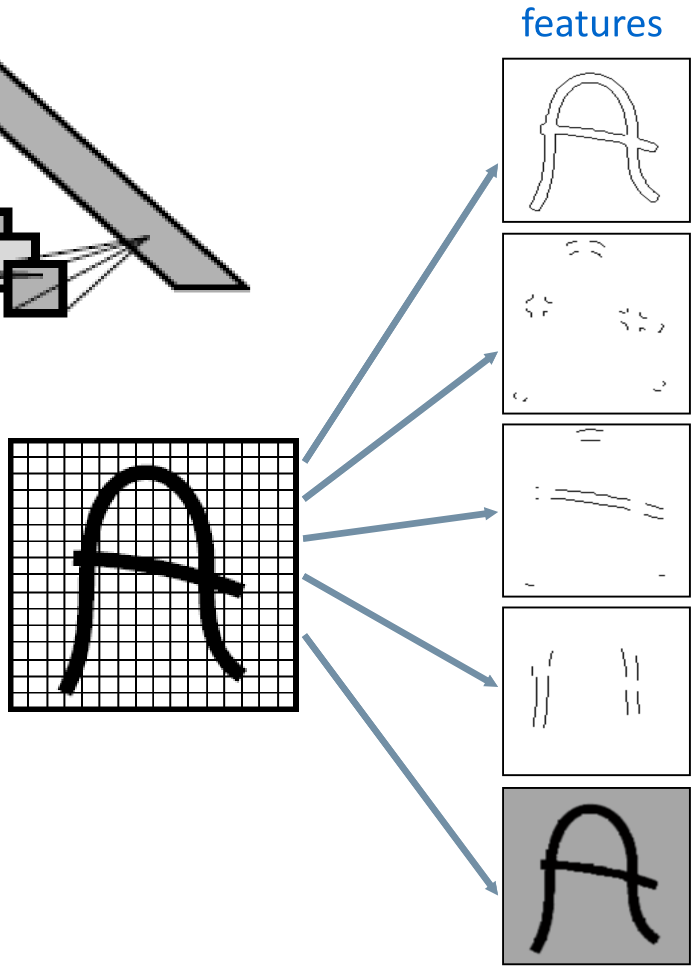
CNN Topology



Convolutional Layer



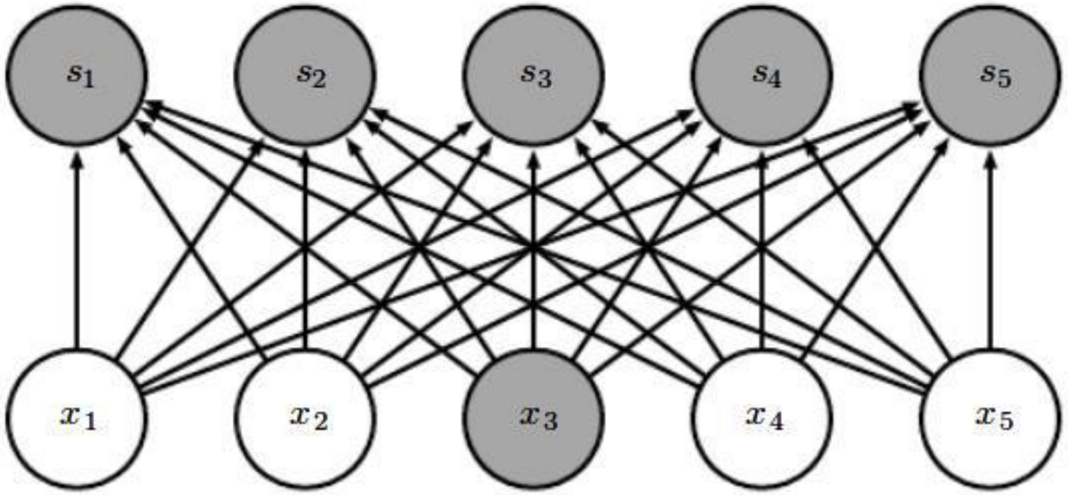




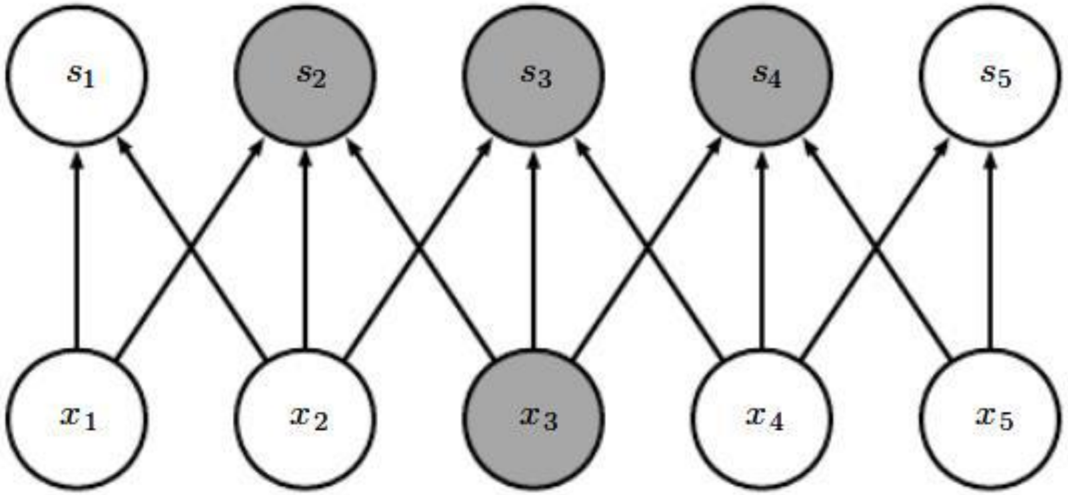
Why Convolutional Layers?

3x1 conv = 3 weights
(+ 1 bias)

Sparse connectivity + Parameter Sharing



Fully Connected



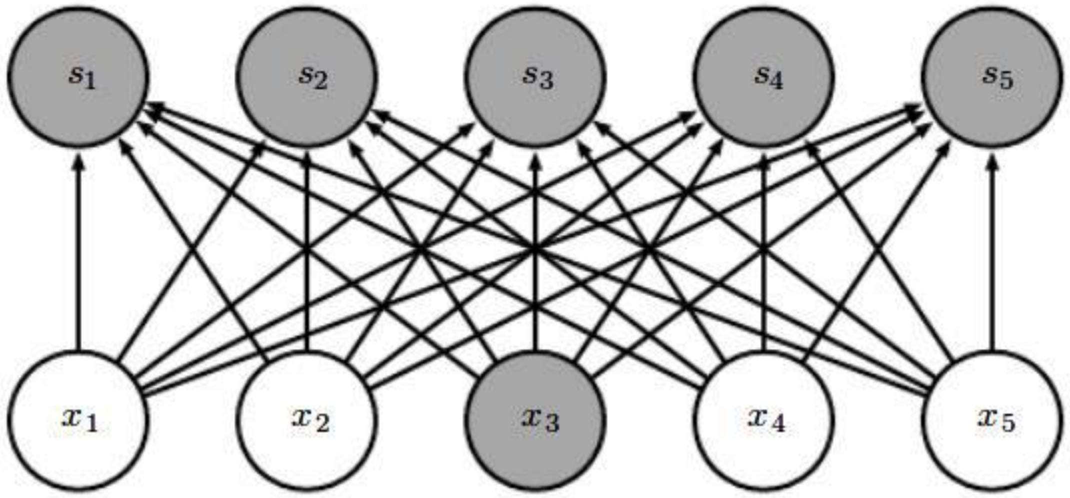
3x1 Convolutional

5x5 = 25 weights
(+ 5 biases)

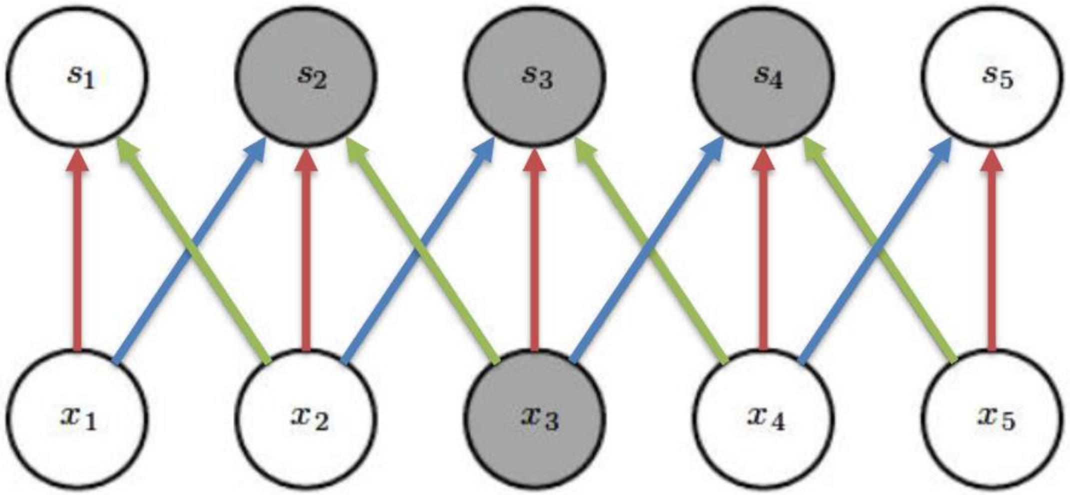
Why Convolutional Layers?

3x1 conv = 3 weights
(+ 1 bias)

Sparse connectivity + Parameter Sharing + Translational Invariance



Fully Connected



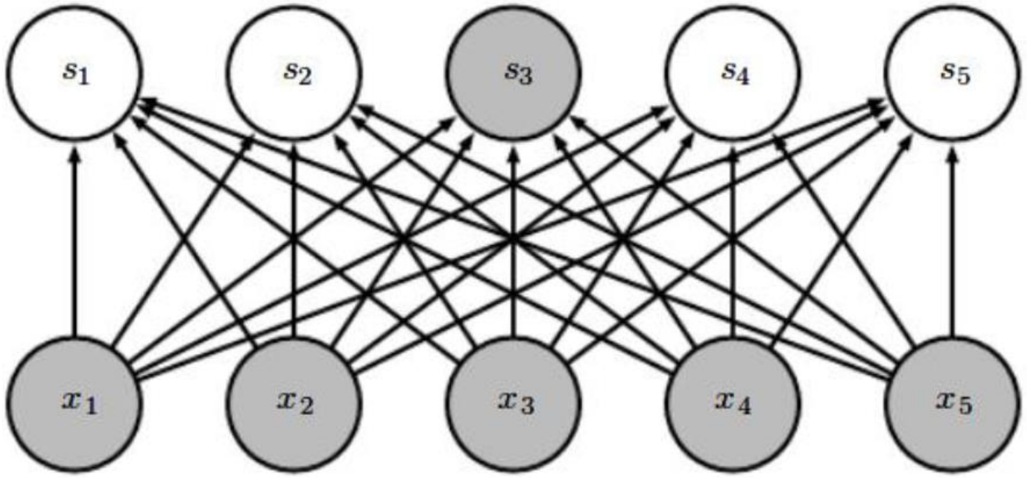
3x1 Convolutional

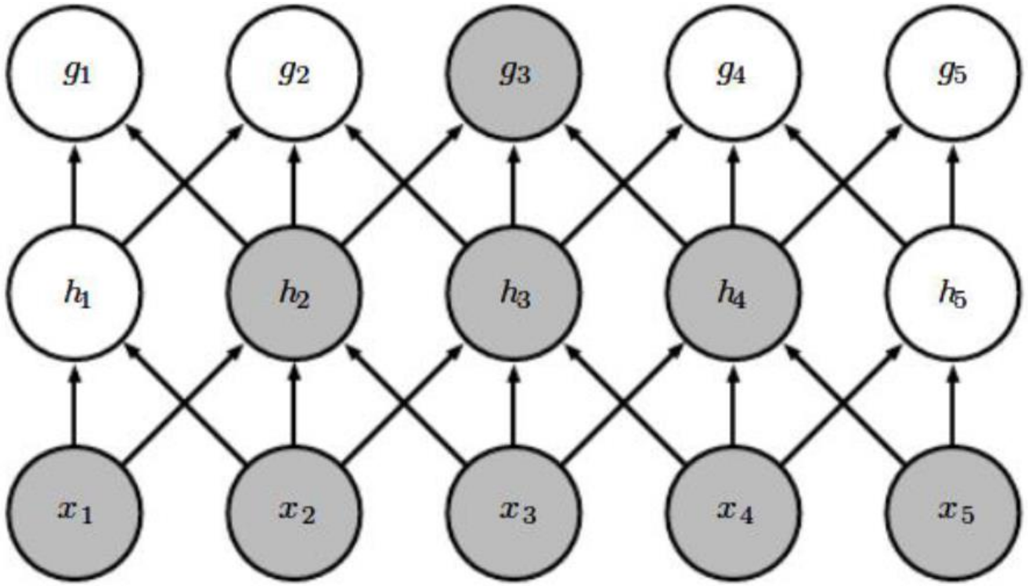
5x5 = 25 weights
(+ 5 biases)

Receptive fields

3x1+3x1 = 6 weights
(+ 2 biases)

Deeper networks depend on wider patches of the input



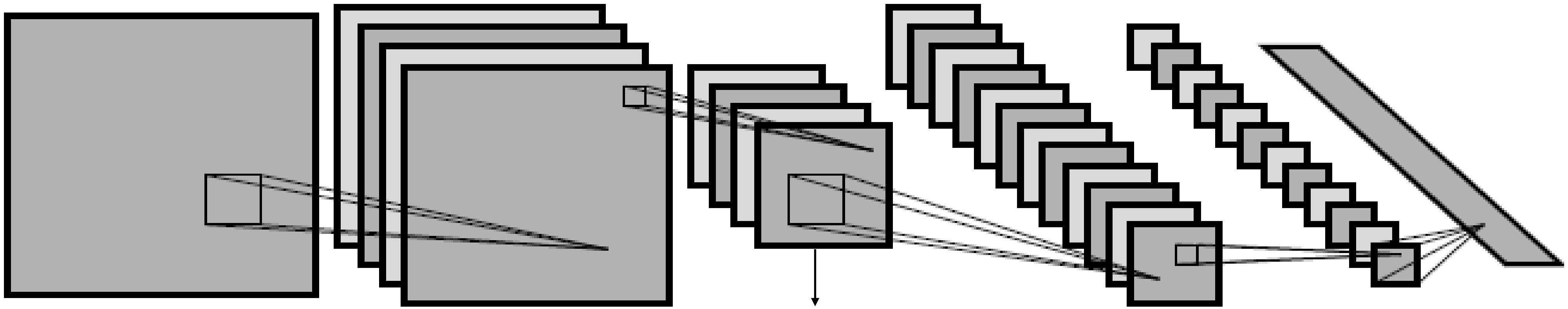


Fully Connected

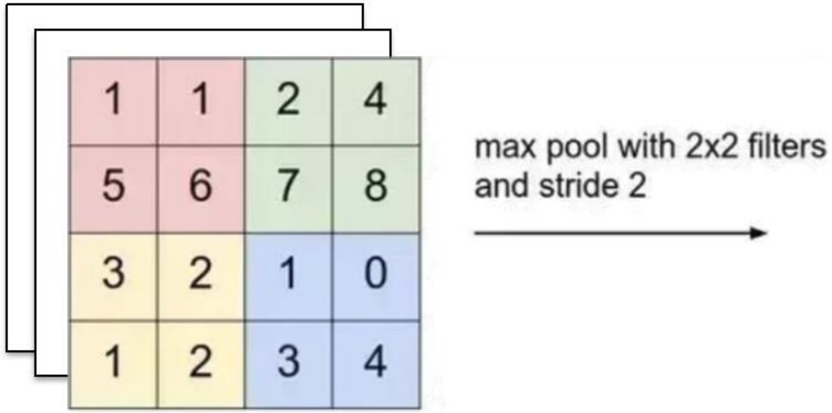
3x1 + 3x1 Convolutional

5x5 = 25 weights
(+ 5 biases)

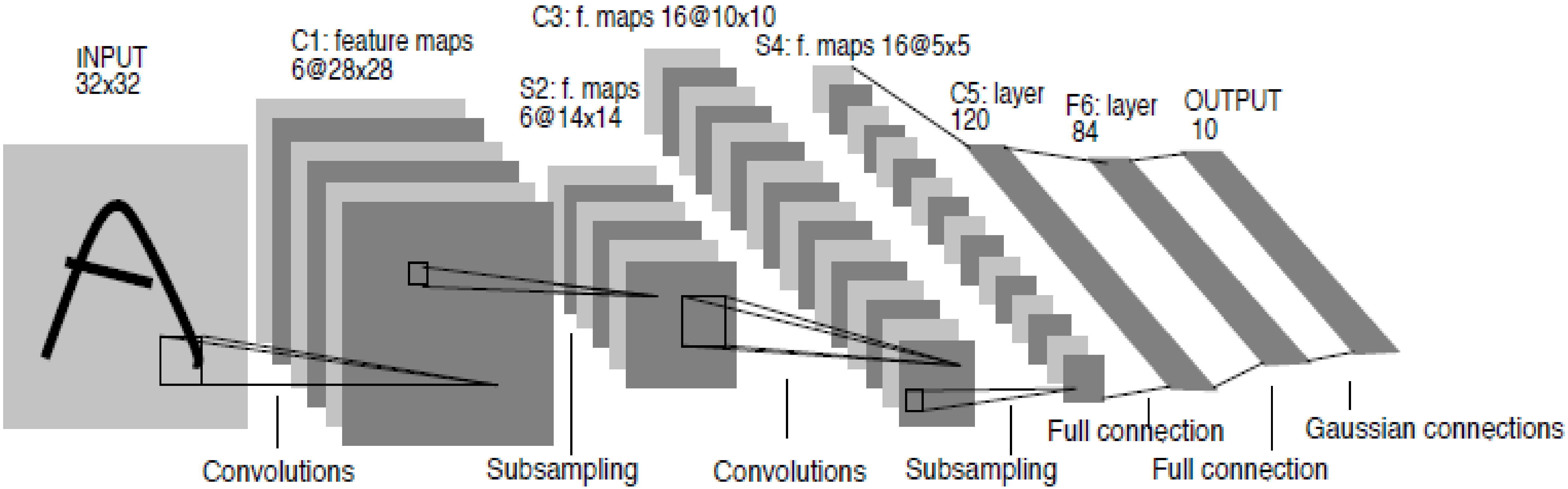
Pooling Layer



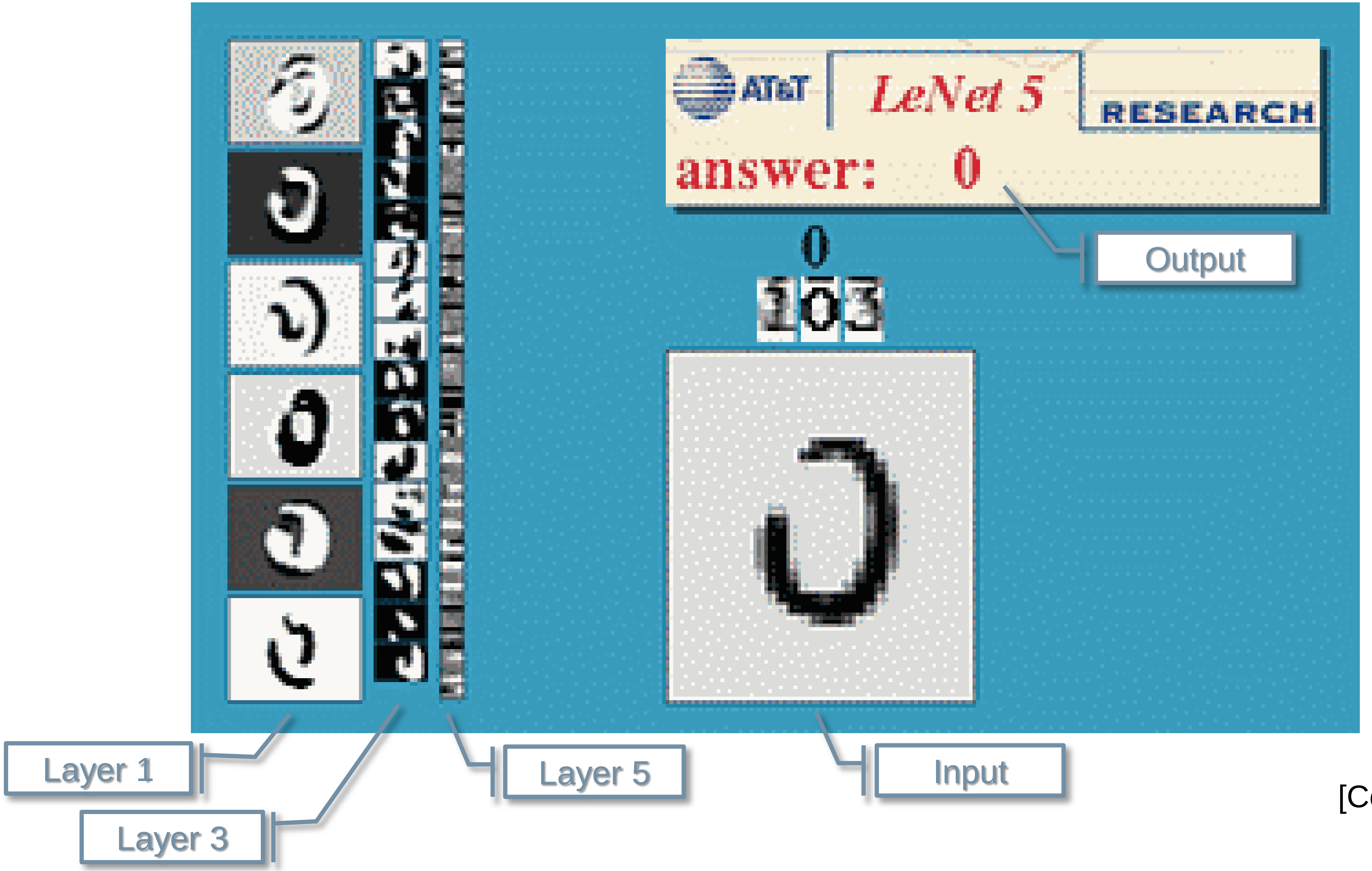
Shift and distortion invariance or
Subsampling layer (Pooling)



LeNet (LeCun, 1998)

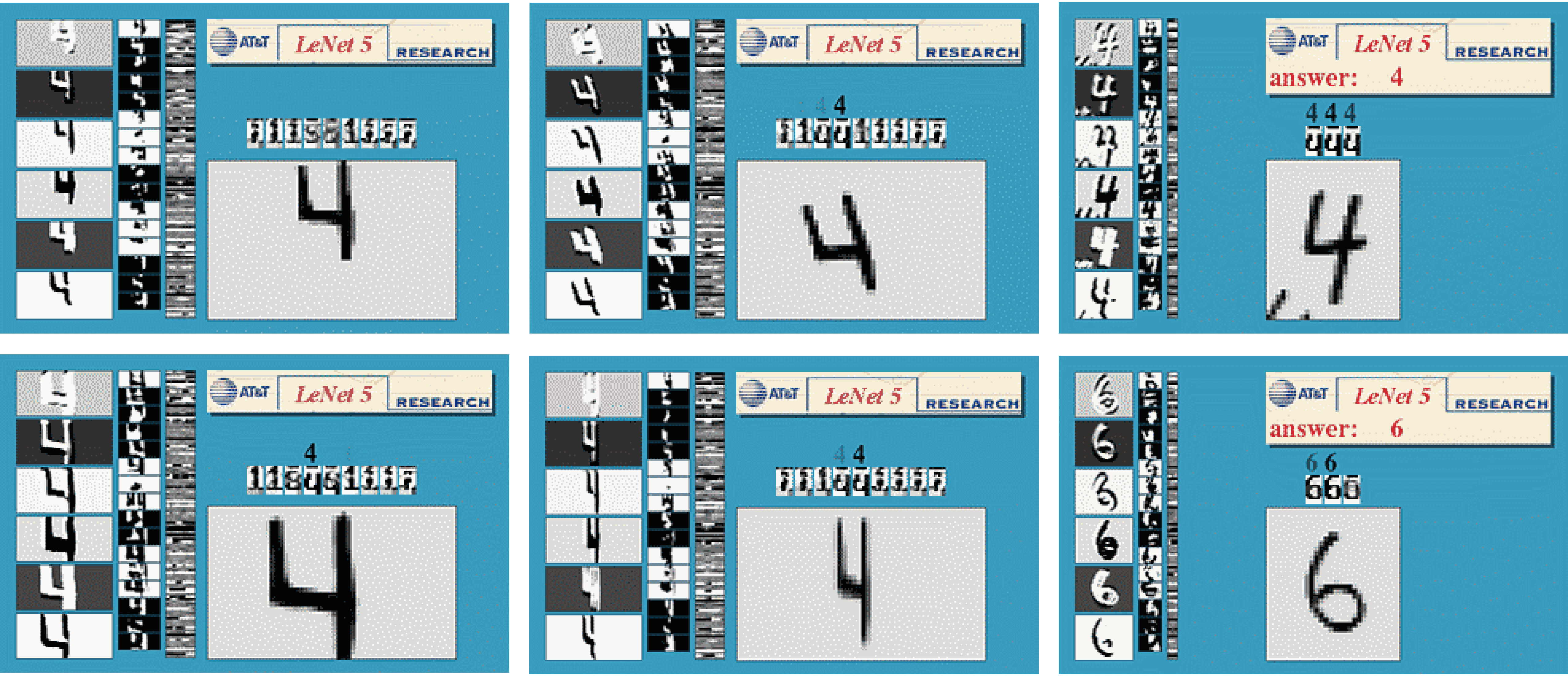


LeNet-5



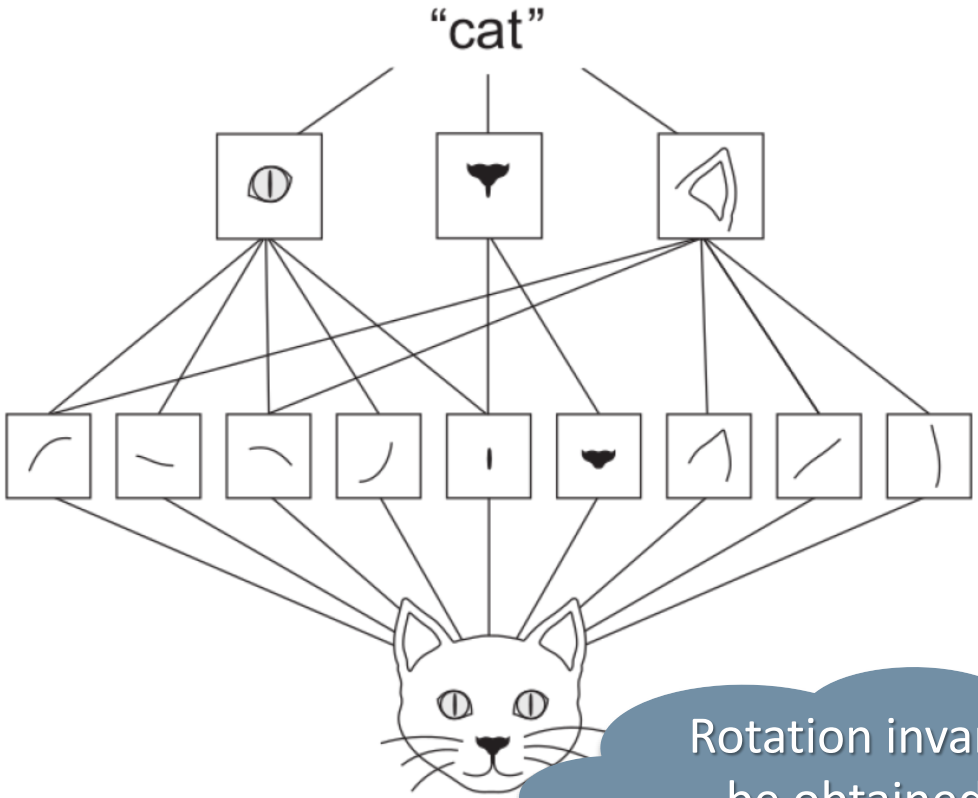
[Courtesy of Yan LeCun]

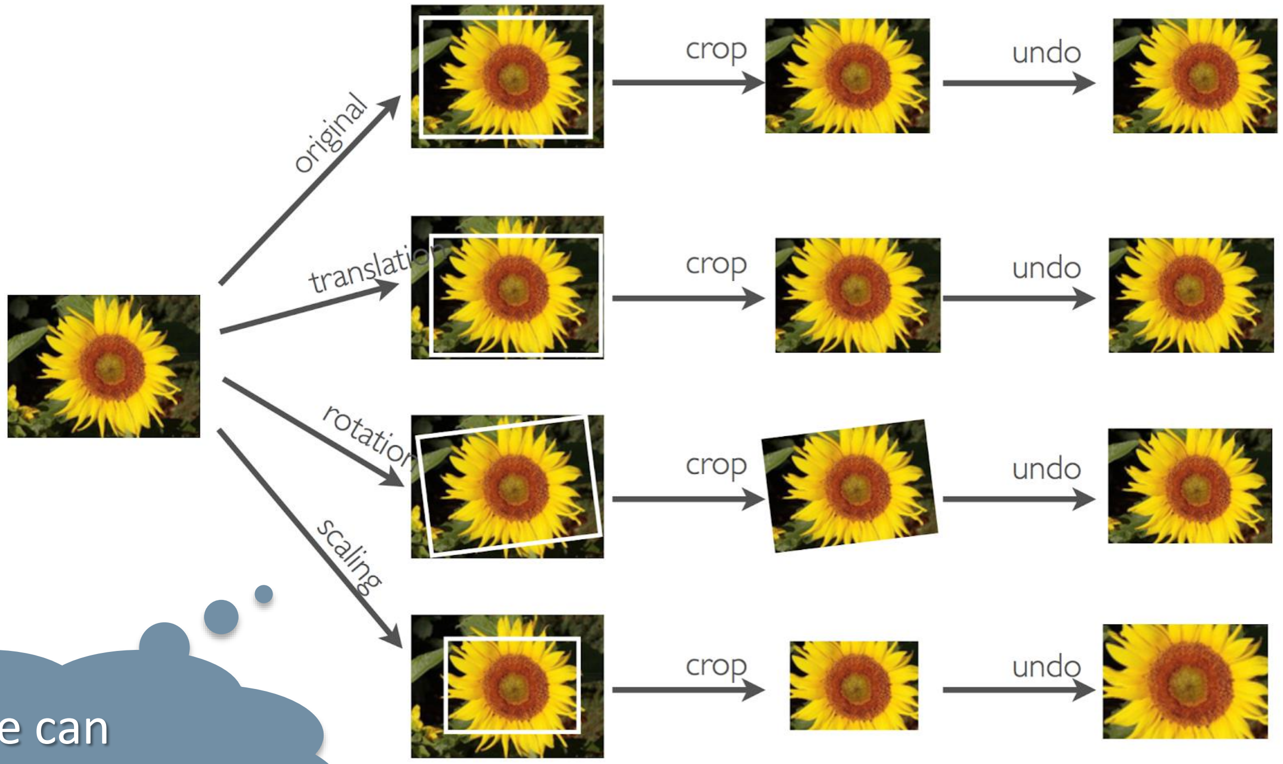
LeNet Invariance



Why Convolutional Networks Work?

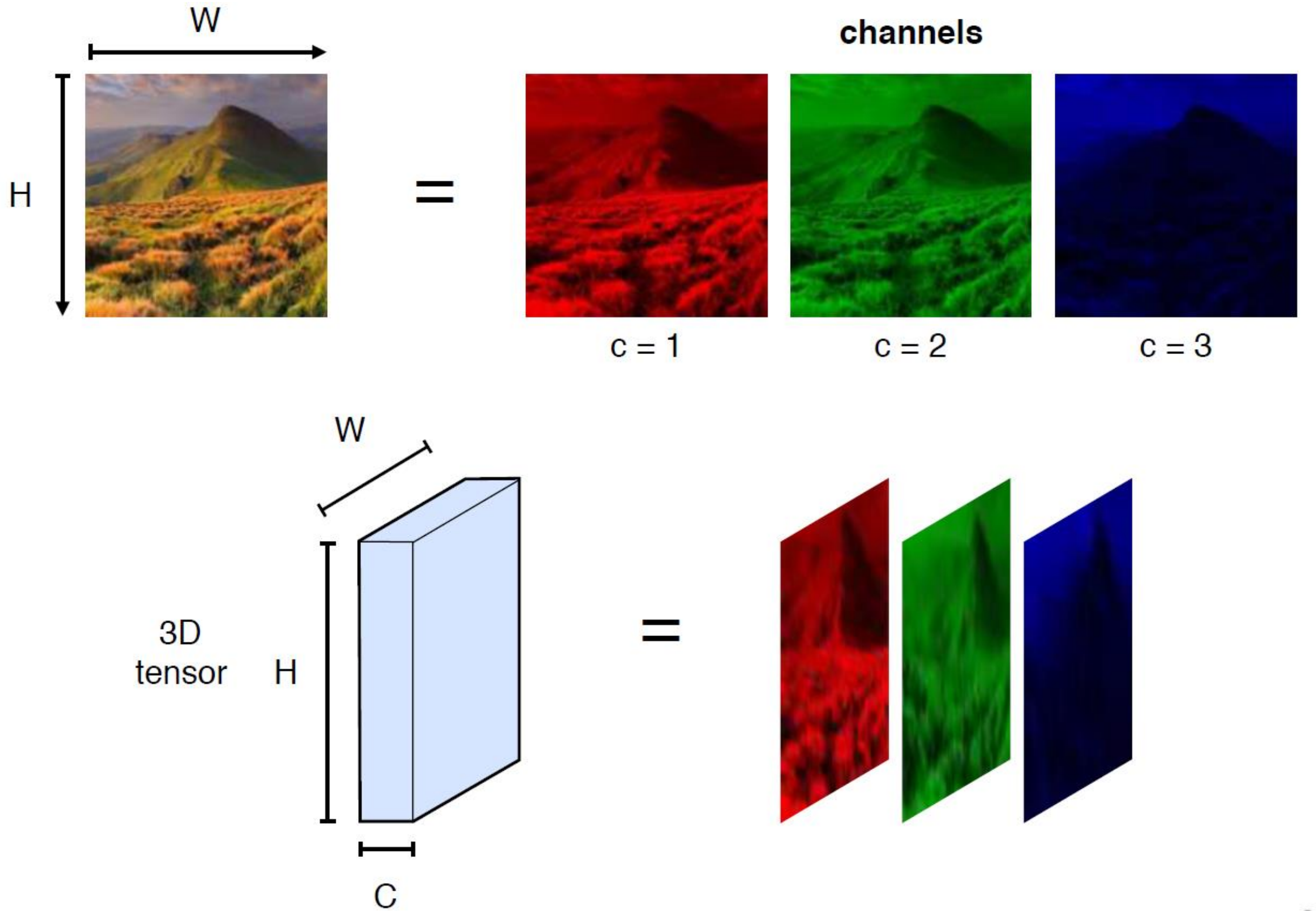
Convolutional neural networks learn a hierarchy of translation invariant features

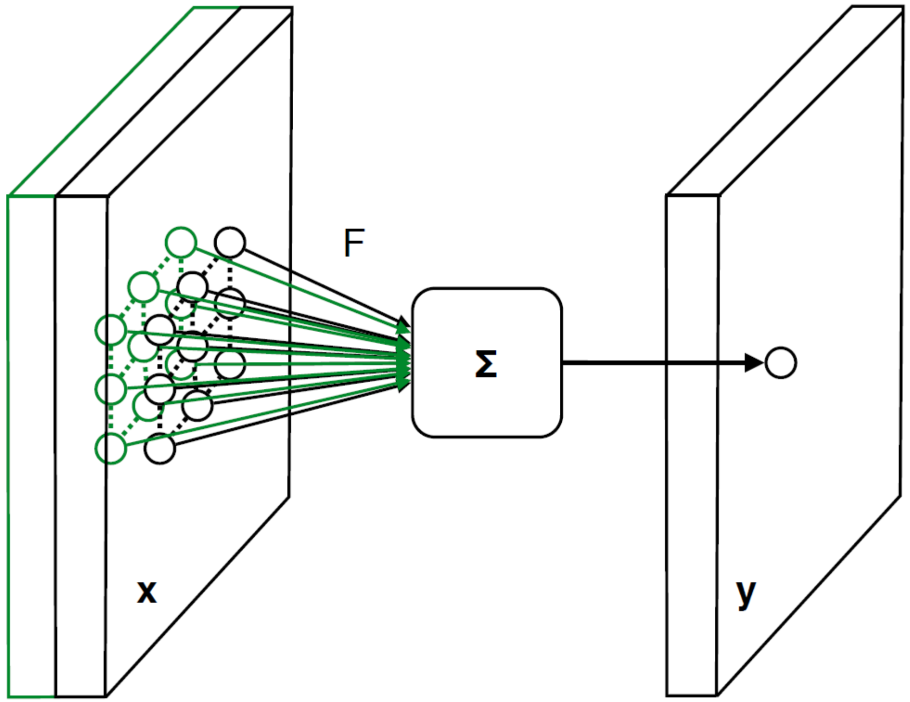




Rotation invariance can be obtained by data augmentation …

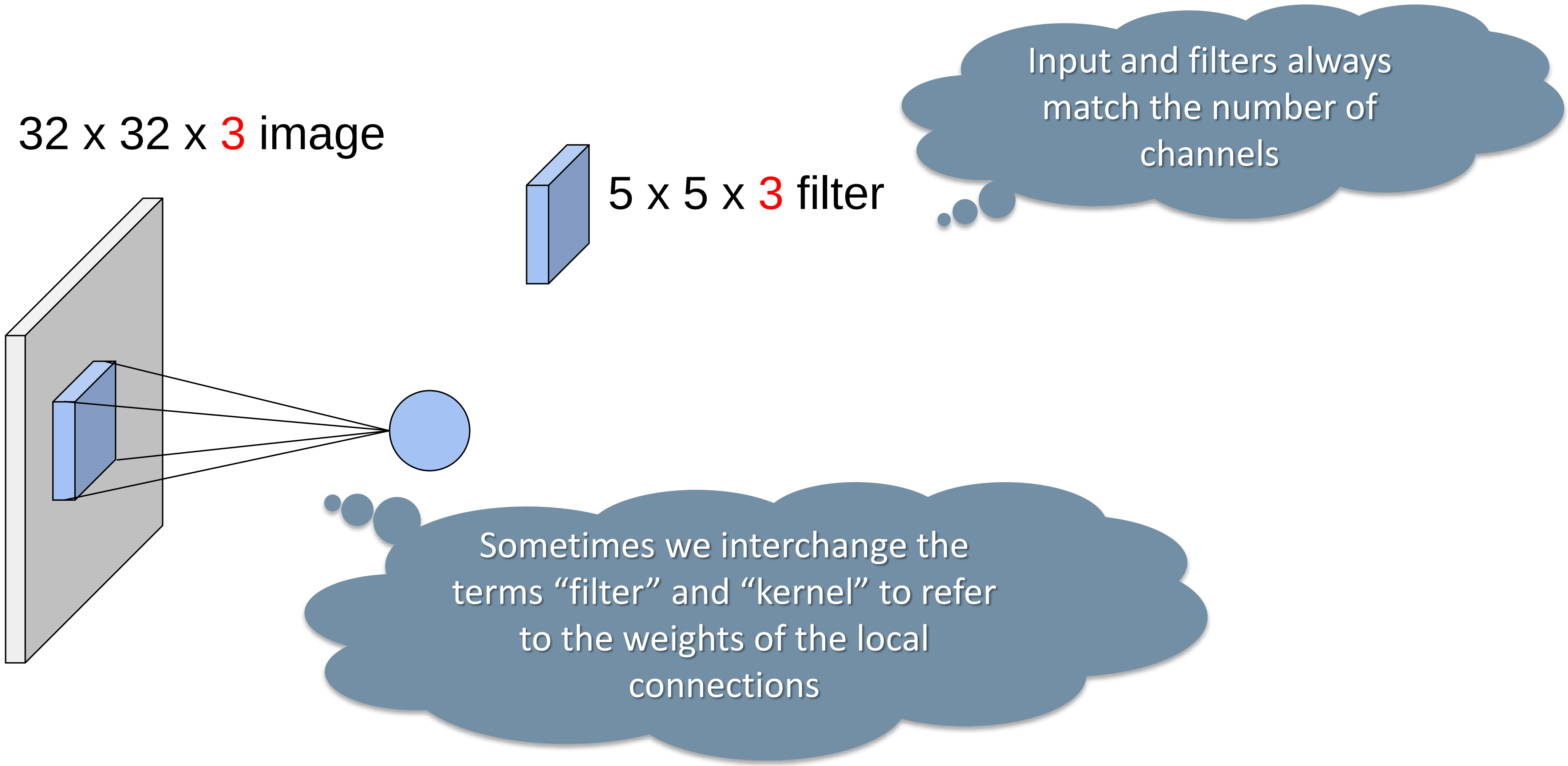
Tensors and 3D Convolutions



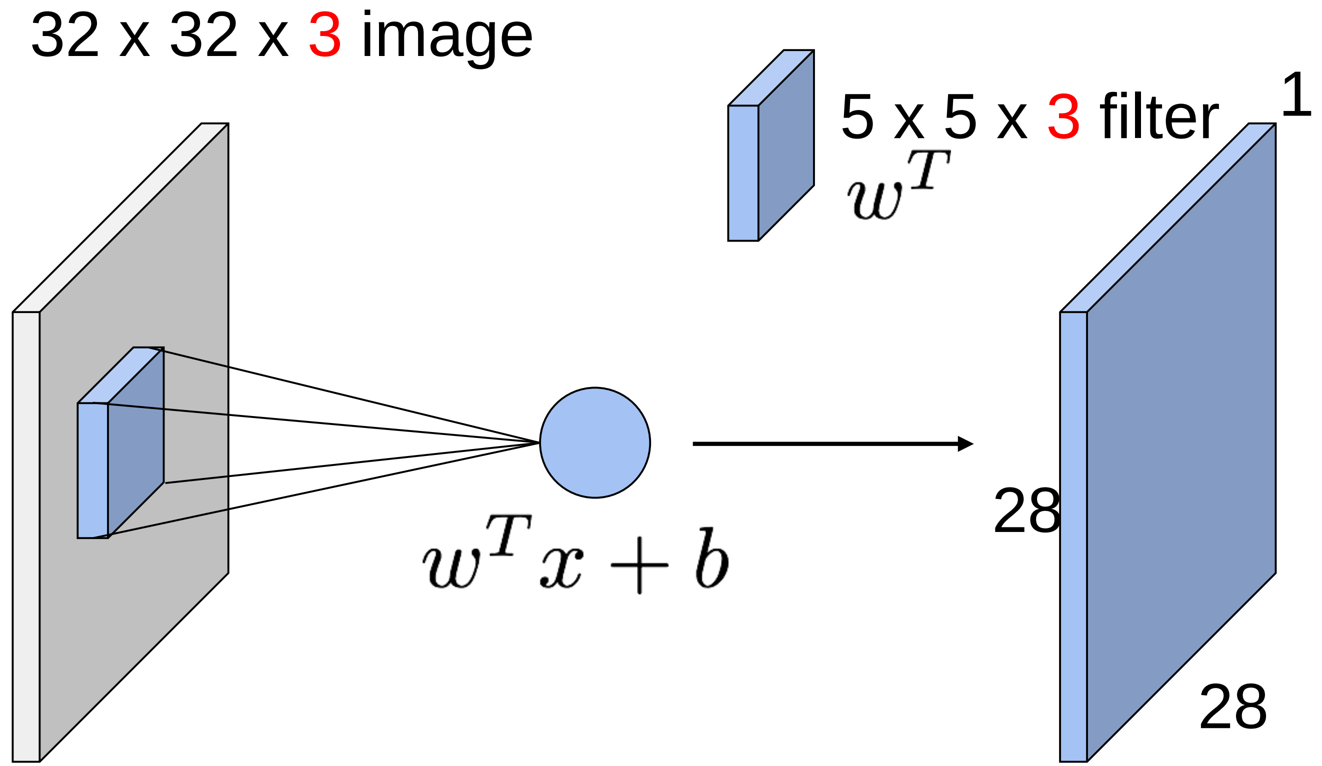


Slide credit: Andrea Vedaldi

Convolutional Neural Networks in a Nutshell

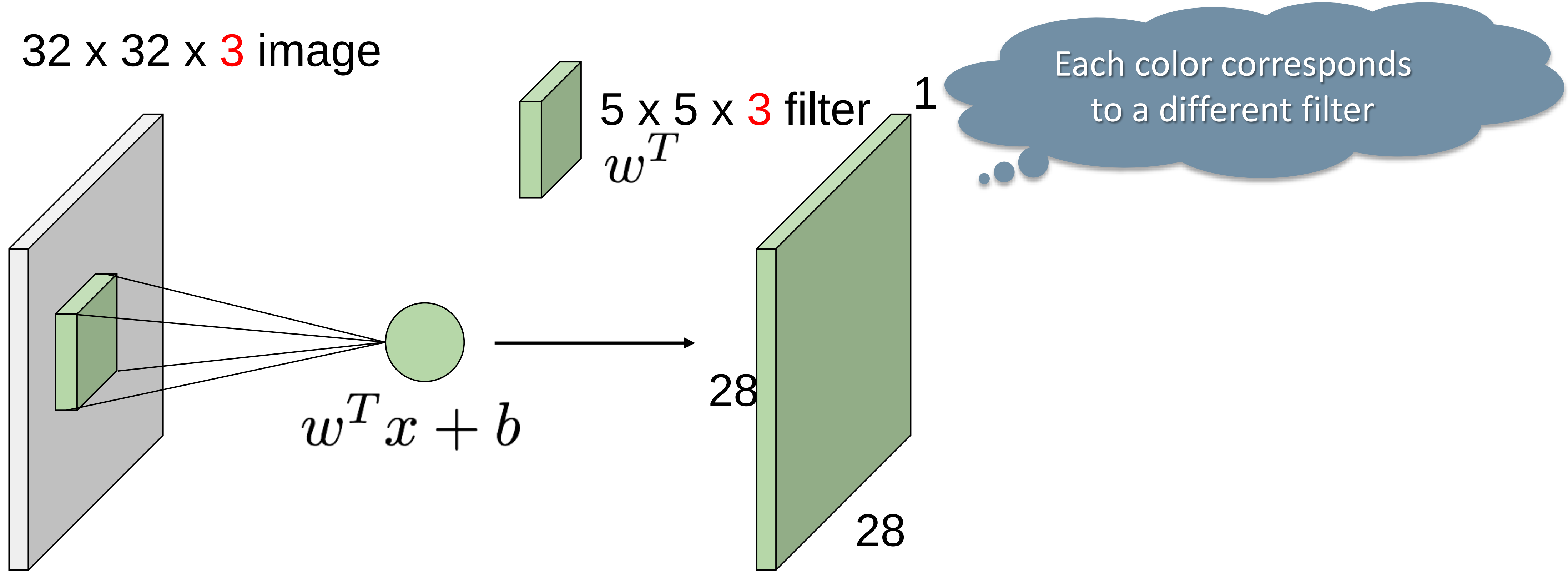


Convolutional Neural Networks in a Nutshell



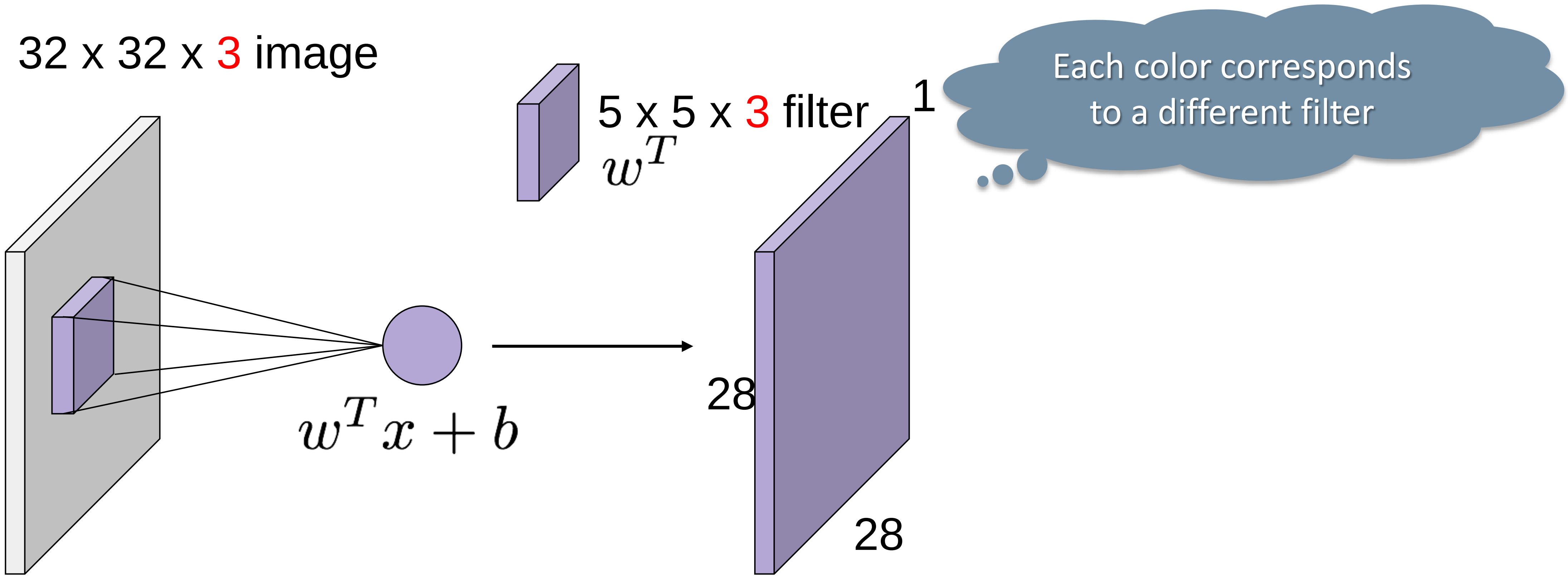
37

Convolutional Neural Networks in a Nutshell



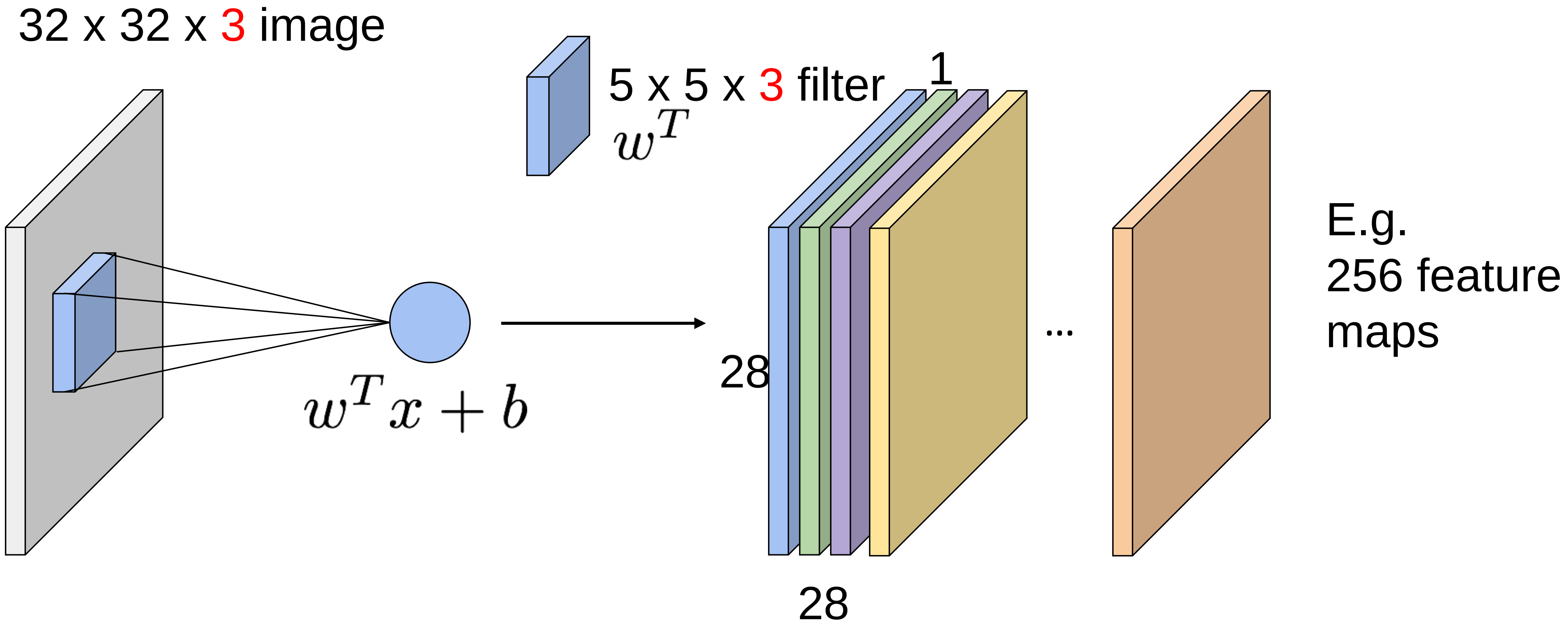
38

Convolutional Neural Networks in a Nutshell



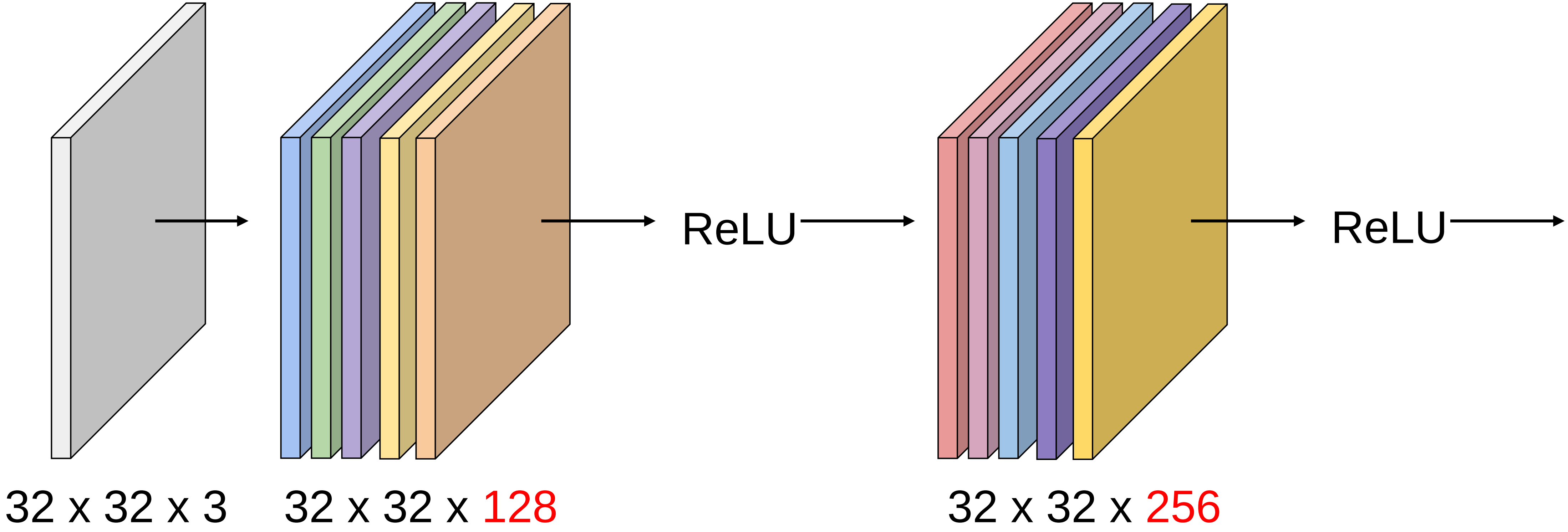
39

Convolutional Neural Networks in a Nutshell



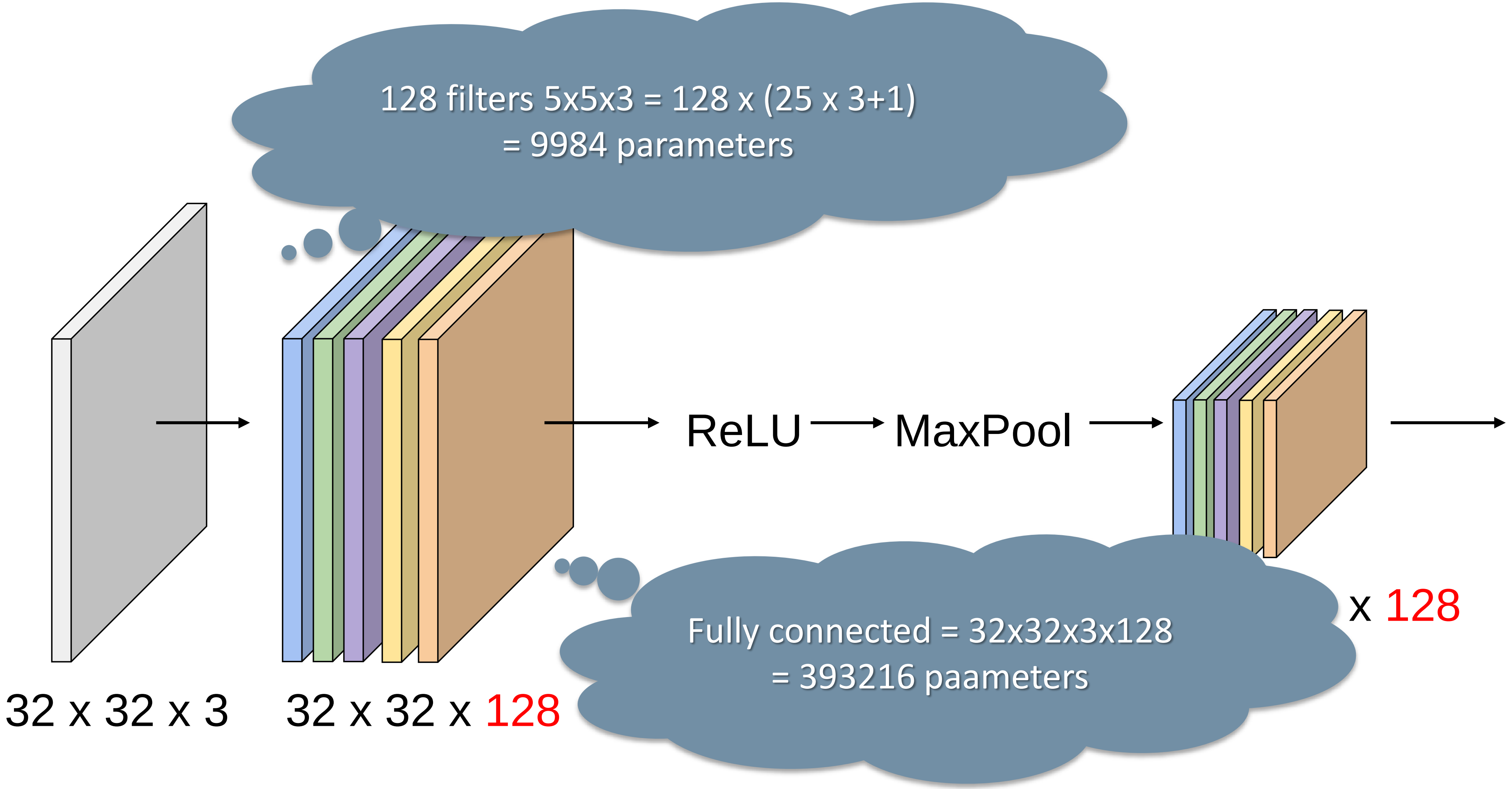
40

Convolutional Neural Networks in a Nutshell



41

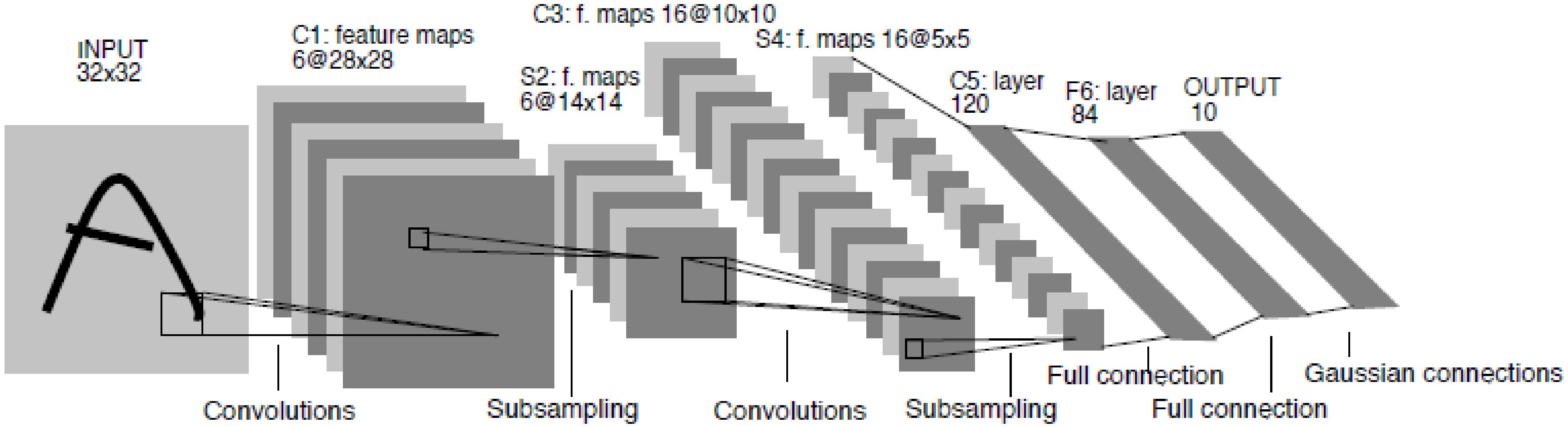
Convolutional Neural Networks in a Nutshell



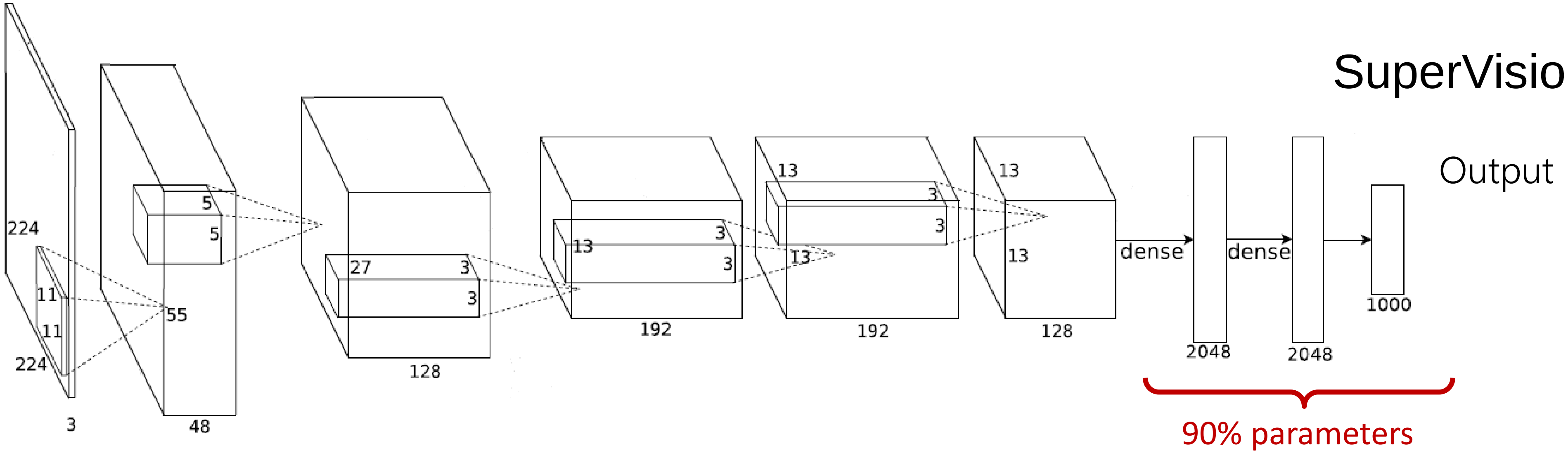
42

Deep CNN for image recognition

LeCun et al. 1998



SuperVision, 2012

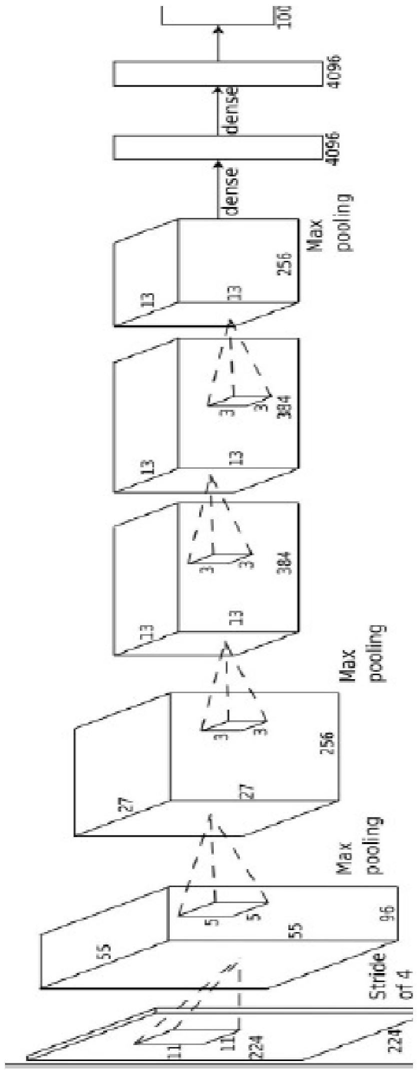


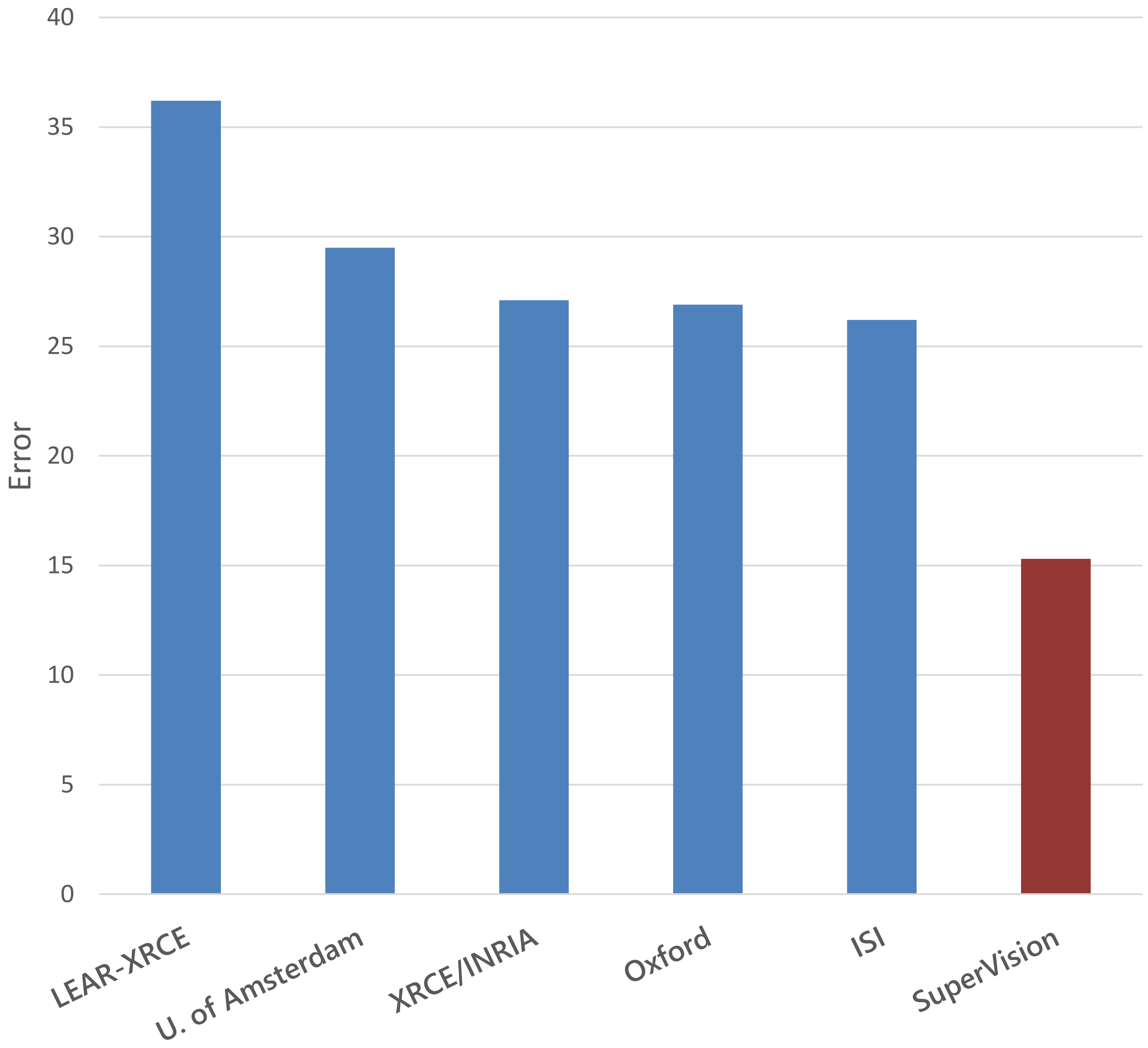
IMAGENET

1000 classes, 1.5 Million labeled images (2012)

And the winner is ...

4M	FULL CONNECT	4Mflop
16M	FULL 4096/ReLU	16M
37M	FULL 4096/ReLU	37M
	MAX POOLING	
442K	CONV 3x3/ReLU 256fm	74M
1.3M	CONV 3x3ReLU 384fm	224M
884K	CONV 3x3/ReLU 384fm	149M
	MAX POOLING 2x2sub	
	LOCAL CONTRAST NORM	
307K	CONV 11x11/ReLU 256fm	223M
	MAX POOL 2x2sub	
	LOCAL CONTRAST NORM	
35K	CONV 11x11/ReLU 96fm	105M





Krizhevsky, Sutskever, Hinton (2012)

Large convolutional net

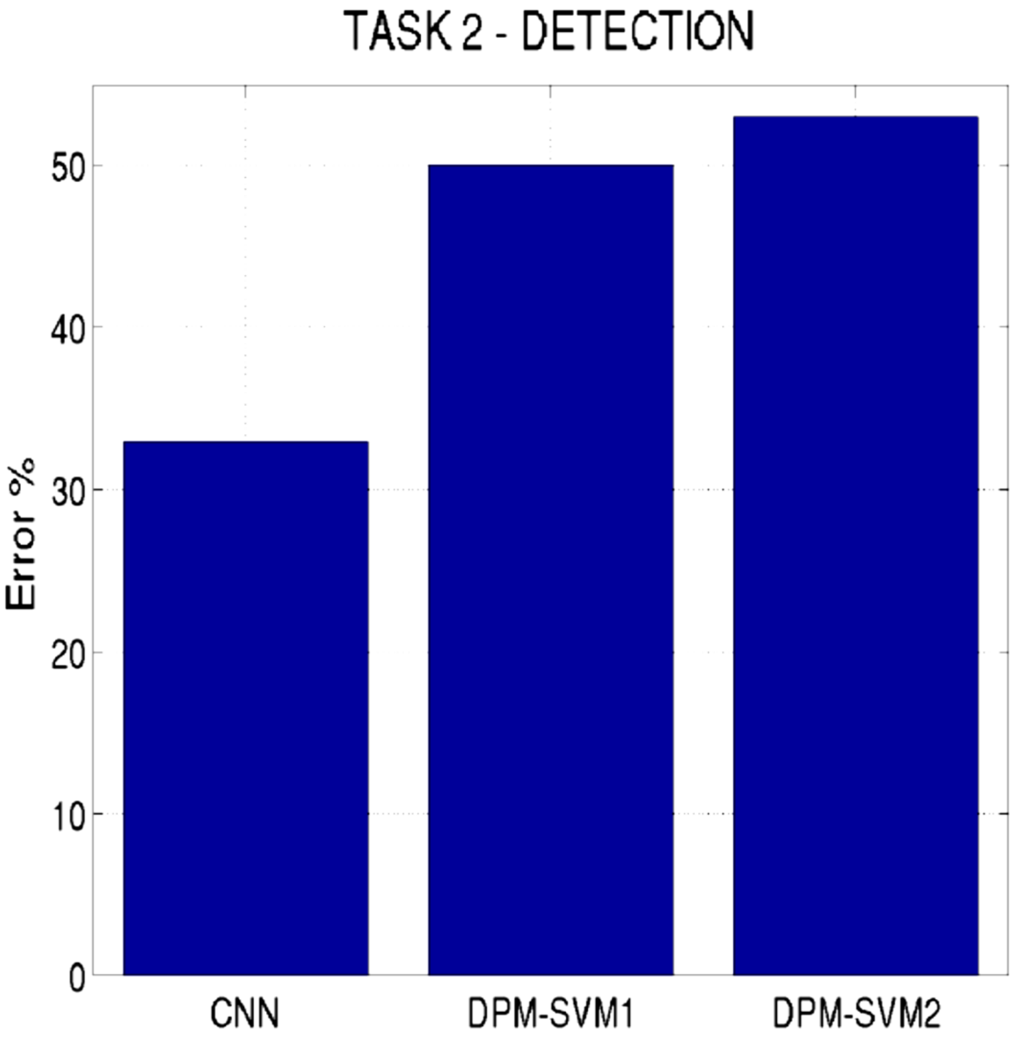
- 650K neurons, 832M synapses, 60M parameters
- Trained with backpropagation on GPU
- Trained «with all the tricks Yann came up with in the last 20 years, plus dropout» (Hinton NIPS'10)
- Image preprocessing: contrast normalization, rectification, etc.

Error rate: 15% (whenever the correct class isn't in top 5)

Previous state of the art: 25% error

A revolution in Computer Vision

- Acquired by Google in Jan 2013
- Deployed in Google+ Photo Tagging in May 2013



Zeiler and Fergus (2013)

Convolutional network

- 8 layers, input 224x224 pixels
- Conv – pool – conv – pool – conv conv – conv – full – full – full
- Rectified-linear Units (ReLU)
- Divisive contrast normalization across features [Jarret et al. 2009]

Trained on ImageNet 2012 training set

- 1.3M images, 1000 classes
- 10 different crops/flips per image

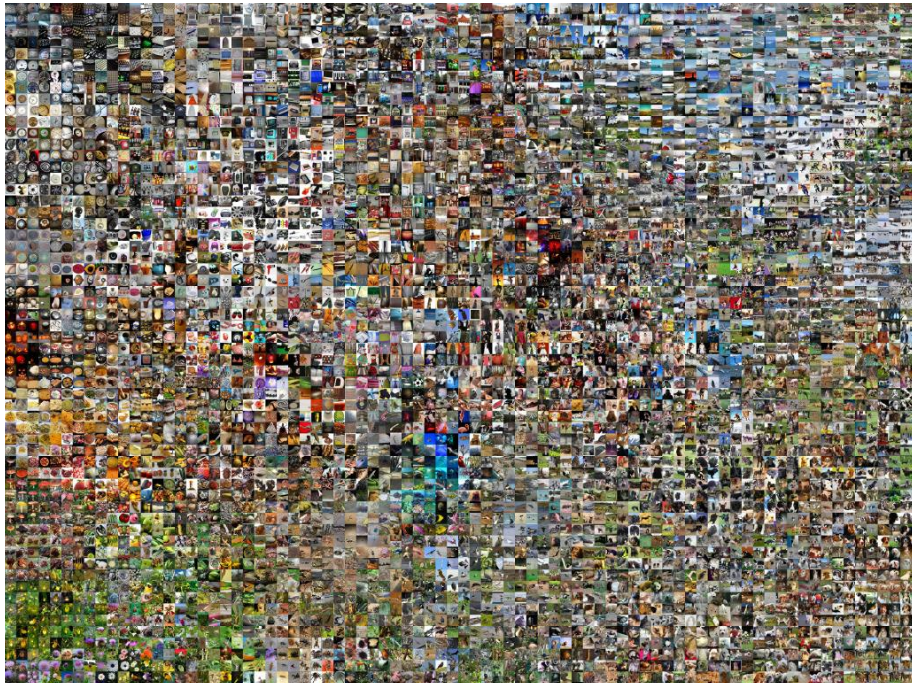
Stochastic gradient descent

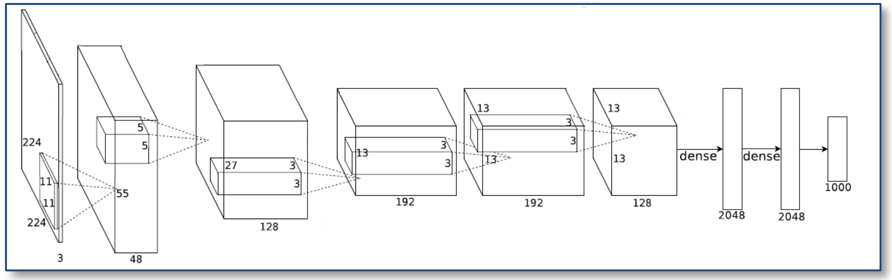
- 70 epochs (7-10 days)
- Learning rate anealing
- Regularization with dropout

Clarifai	11.7%	Deep CNN
NUS	13.0%	SVM based + Deep CNN
ZF	13.5%	Deep CNN
Andrew Howard	13.6%	Deep CNN
OverFeat-NYU	14.1%	Deep CNN
UvA-Euvison	14.2%	Deep CNN

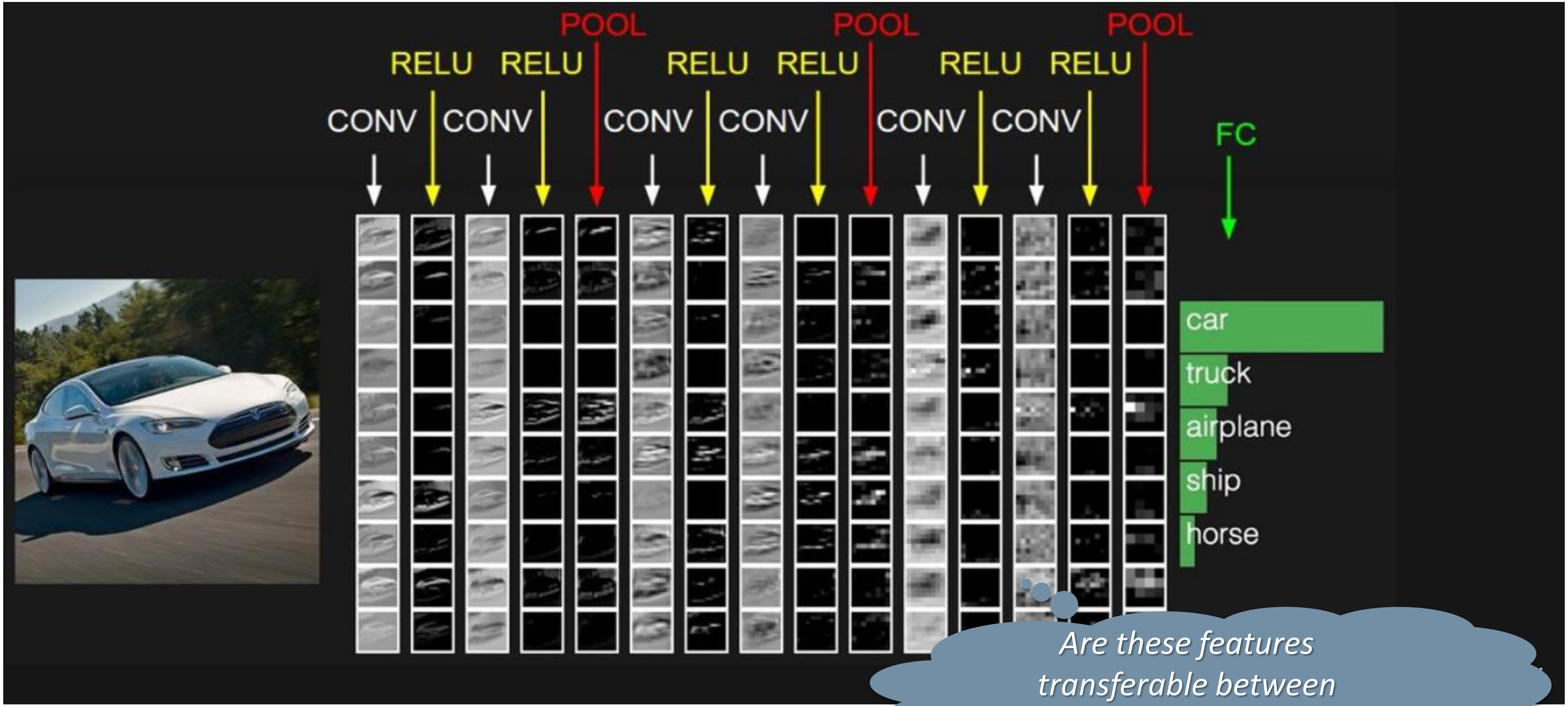
Human level performance!!!

Supervision (ImageNet - 2012)



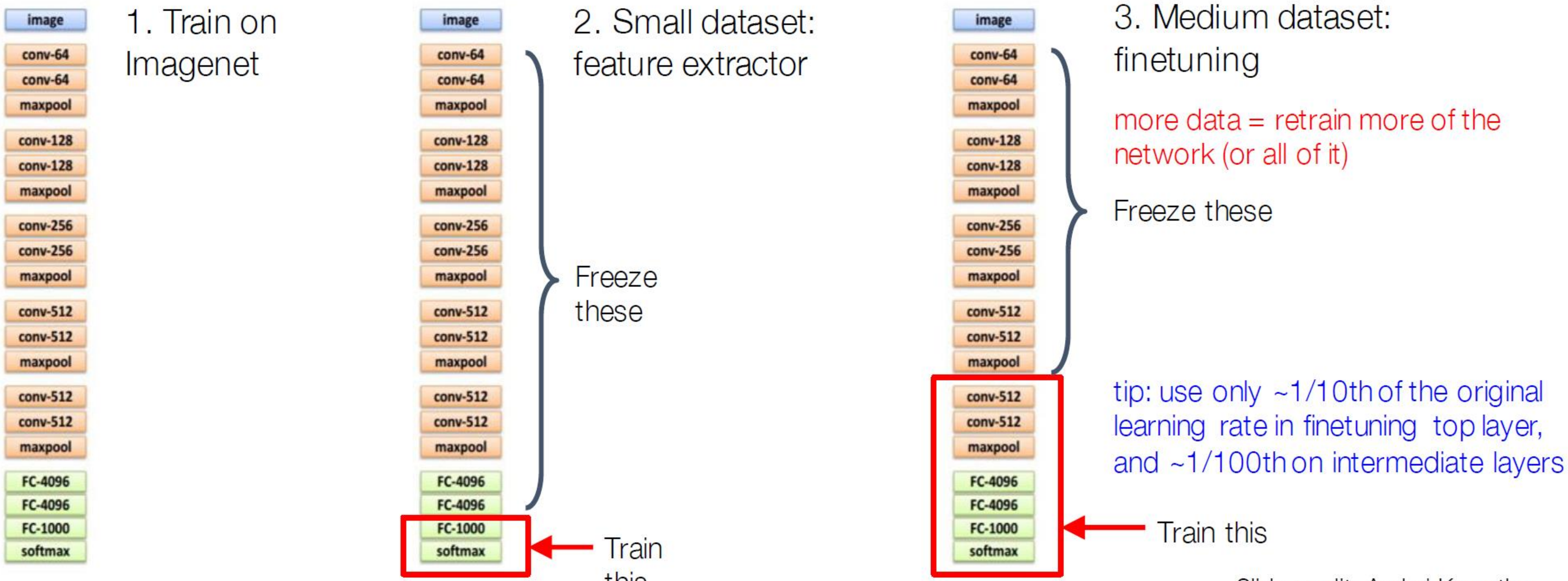


Feature Learning in Convolutional Networks



Are these features transferable between tasks/datasets?

Transfer Learning

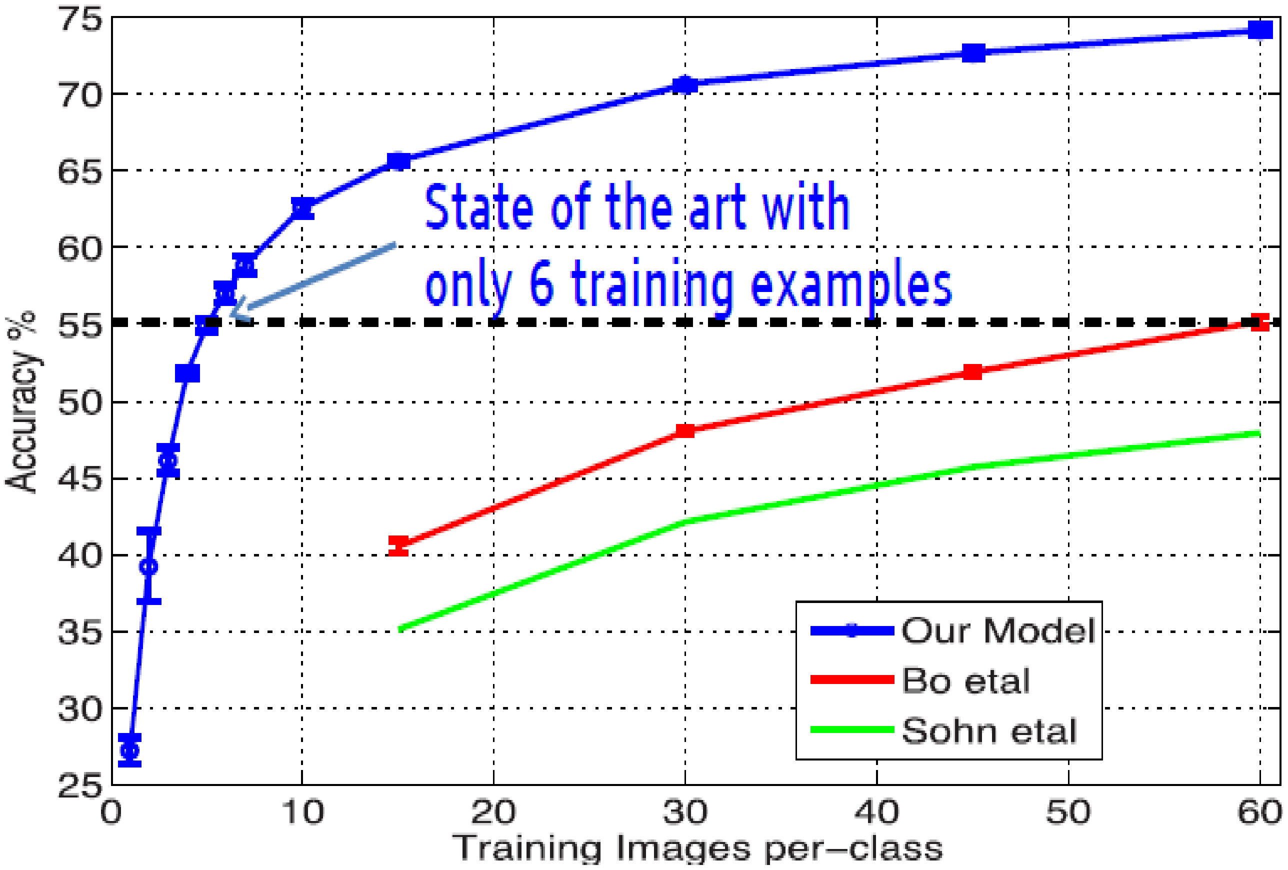


Slide credit: Andrej Karpathy

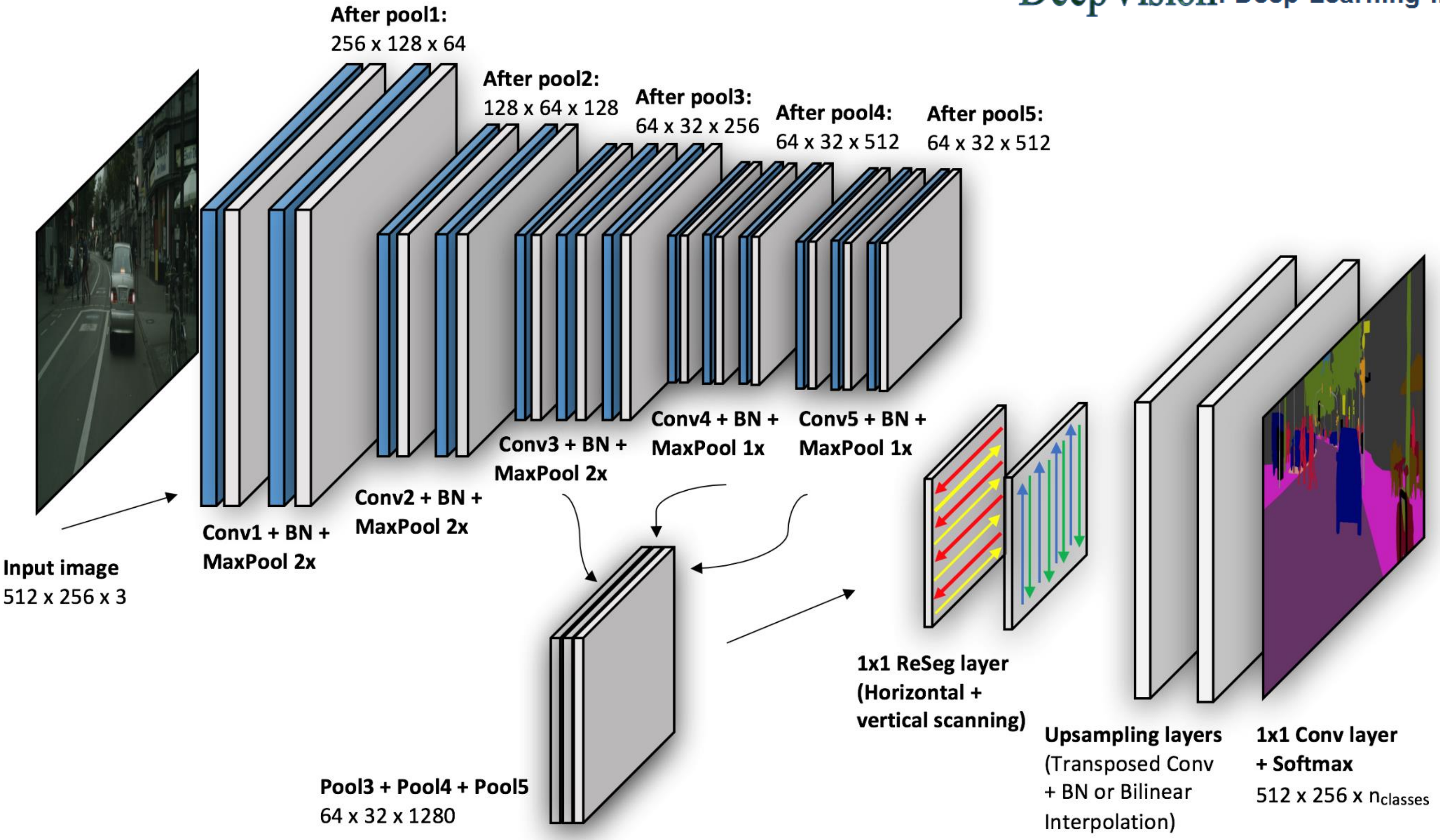
Features are generic

Can we reuse the low level processing from CNN?

- Network trained on ImageNet first
- Last layer chopped off
- Last layer trained on Caltech 256 keeping previous layers fixed
- State of the art accuracy with only 6 samples/class

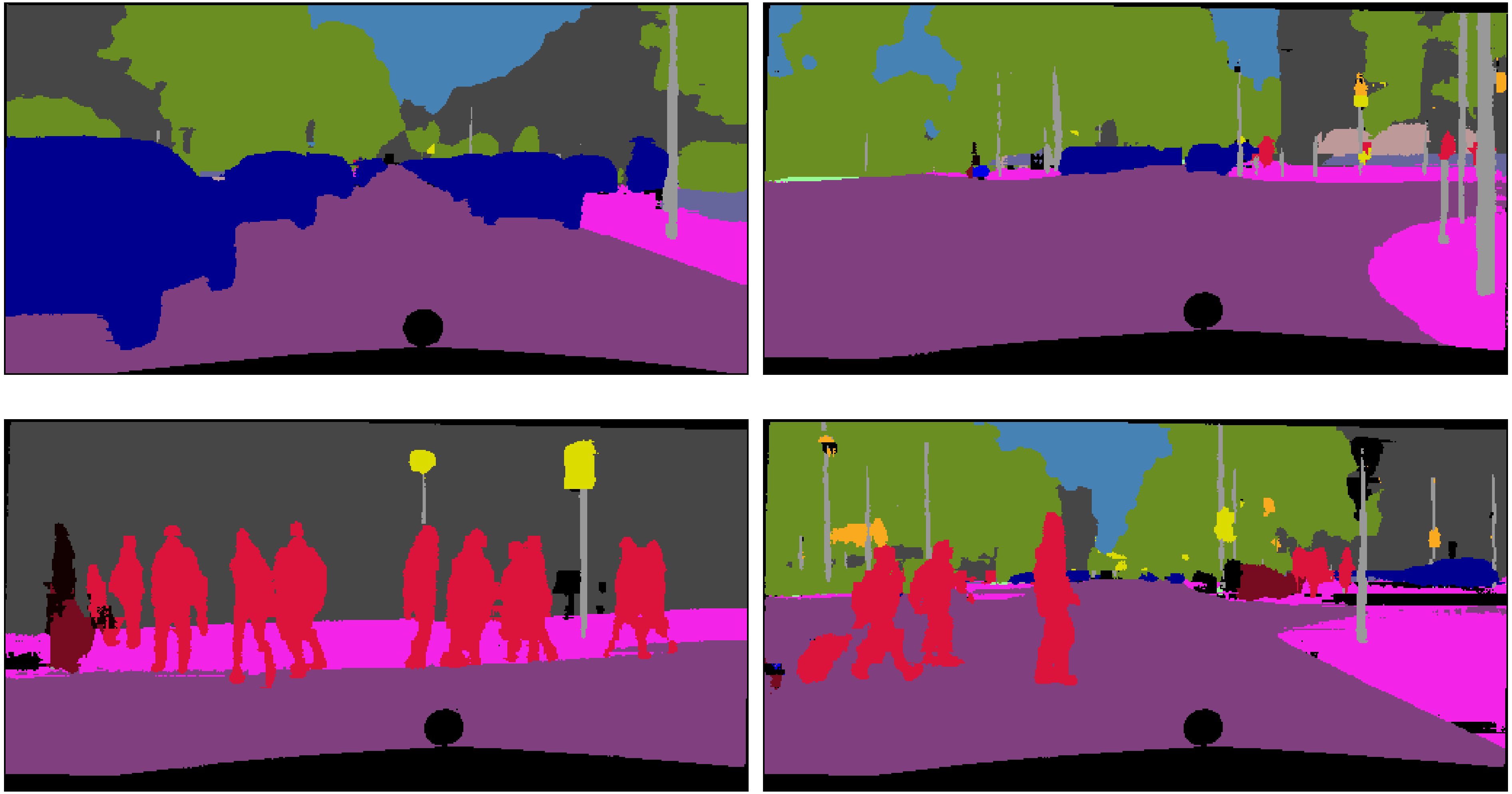


VGG + ReSeg Architecture



Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, Aaron Courville
ReSeg: A Recurrent Neural Network-based Model for Semantic Segmentation. CVPR Workshops 2016

Results on Cityscape

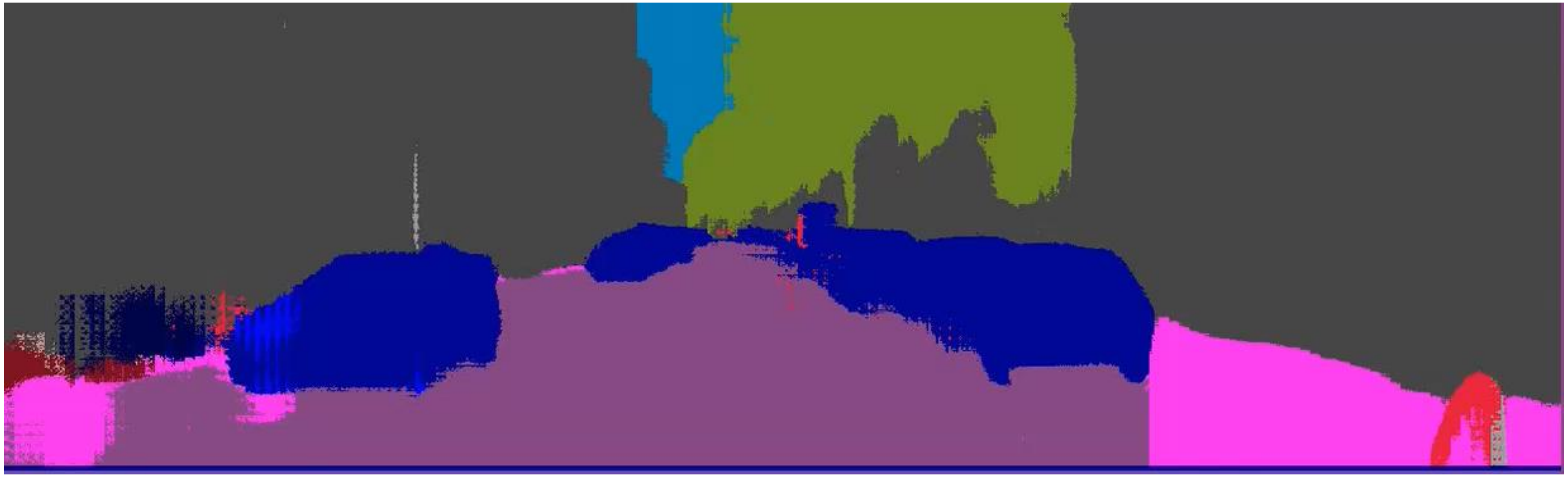


19 semantic classes, 3275 training images, 500, validation, 1525 test images (2048 × 1024 resolution)

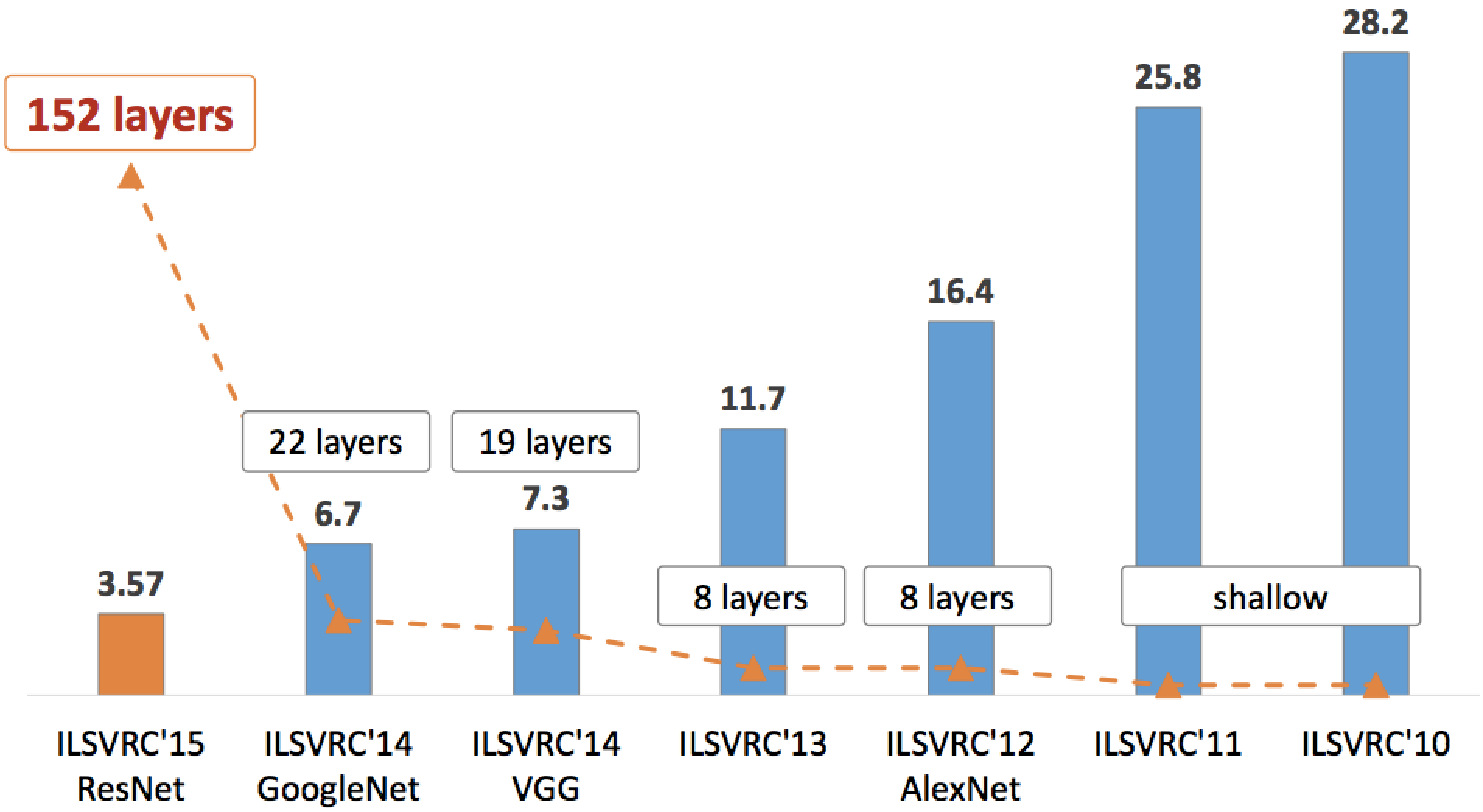
Results on CamVid

11 semantic classes, 367 training images, 101 validation, 233 test images (480 × 360 resolution)

«On every street»

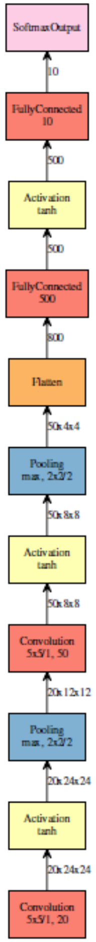


Revolution of Depth



How Deep is Enough?

LeNet (1998)



2 convolutional layers
2 fully connected layers

How Deep is Enough?

LeNet (1998)

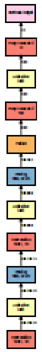
2 convolutional layers
2 fully connected layers

AlexNet (2012)

5 convolutional layers
3 fully connected layers

How Deep is Enough?

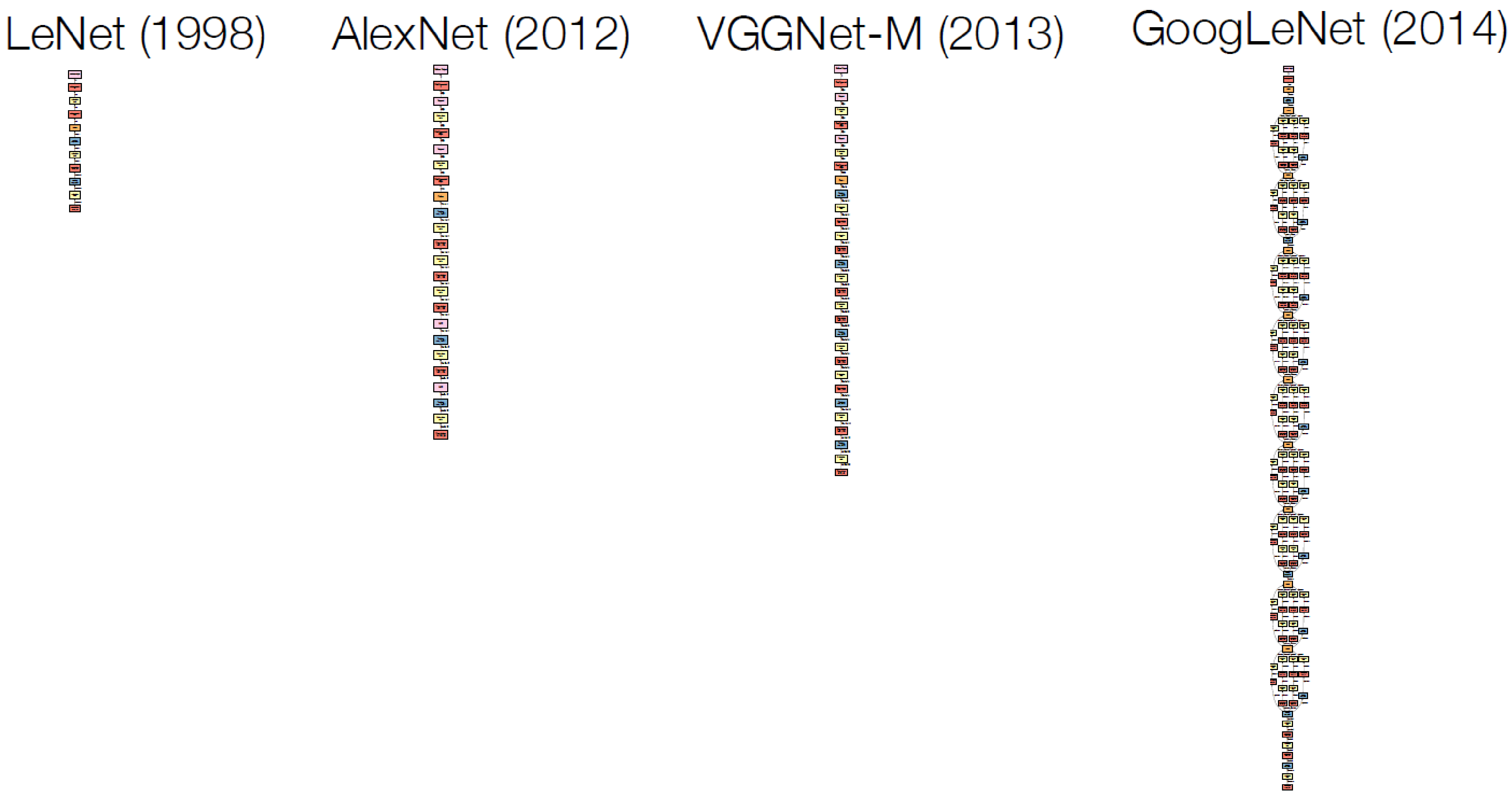
LeNet (1998)



AlexNet (2012)

VGGNet-M (2013)

How Deep is Enough?



How Deep is Enough?