

Deep learning for image processing

STAT 471

November 30, 2021

Where we are

- ✓ **Unit 1:** Intro to modern data mining
- ✓ **Unit 2:** Tuning predictive models
- ✓ **Unit 3:** Regression-based methods
- ✓ **Unit 4:** Tree-based methods
- Unit 5:** Deep learning

Lecture 1: Deep learning preliminaries

Lecture 2: Neural networks

Lecture 3: Deep learning for images

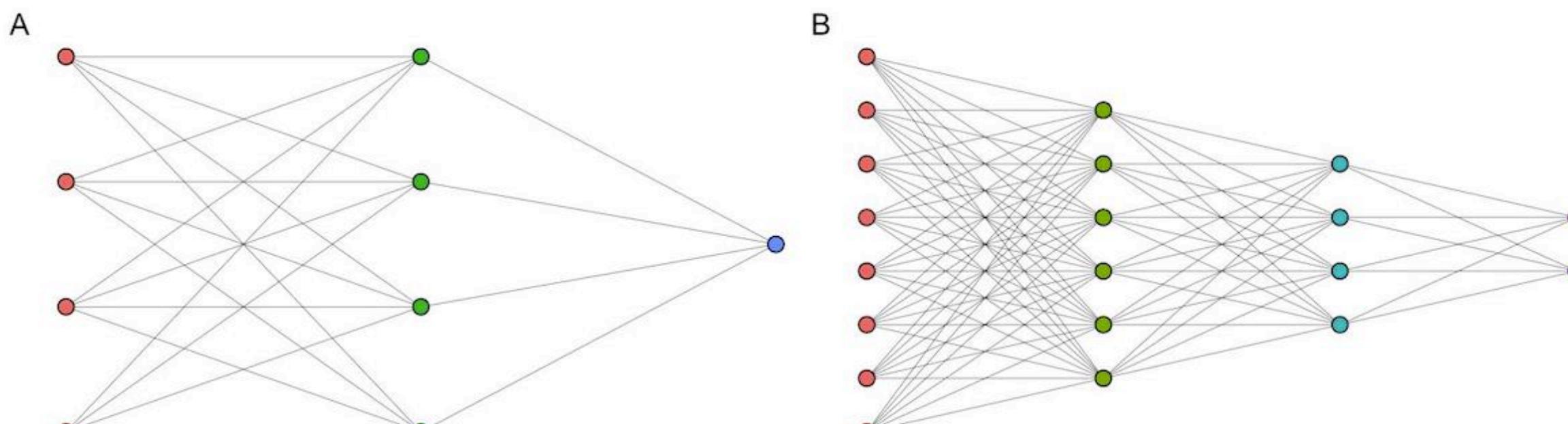
Lecture 4: Deep learning for text

Lecture 5: Unit review and quiz in class

Homework 5 due the following Wednesday.

Network architectures

Fully connected architectures

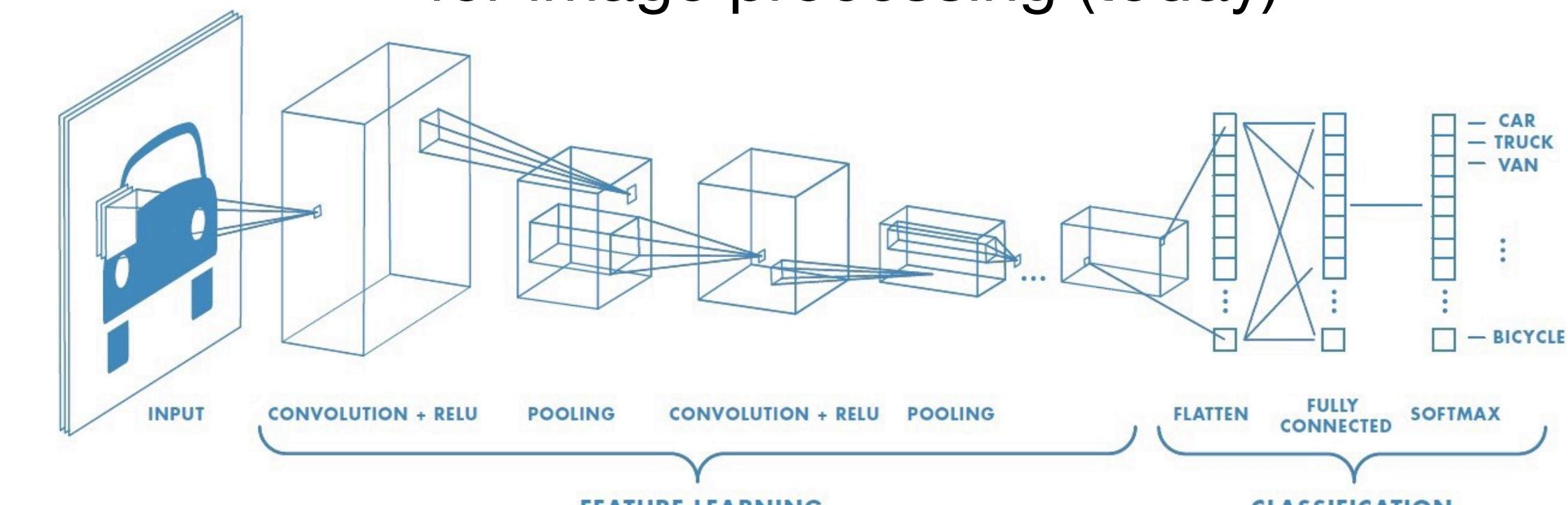


<https://community.rstudio.com/t/visualising-neural-network-architectures/41723>

Have been focused on fully connected architectures

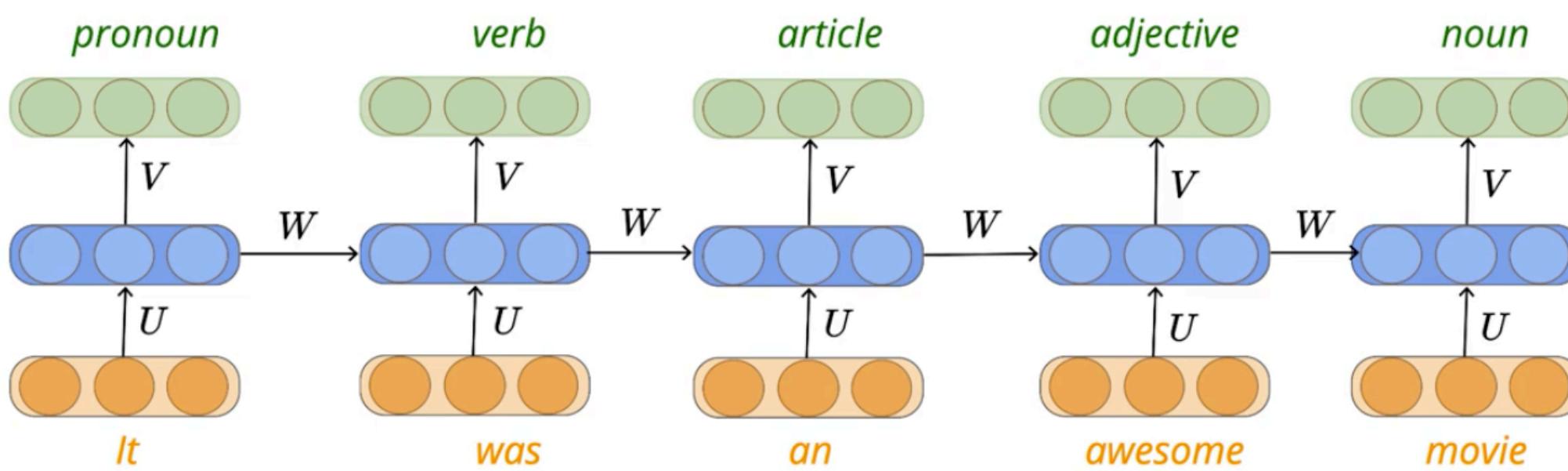
Now, we are going to look at tailored neural networks

Convolutional neural network (CNN) architectures for image processing (today)



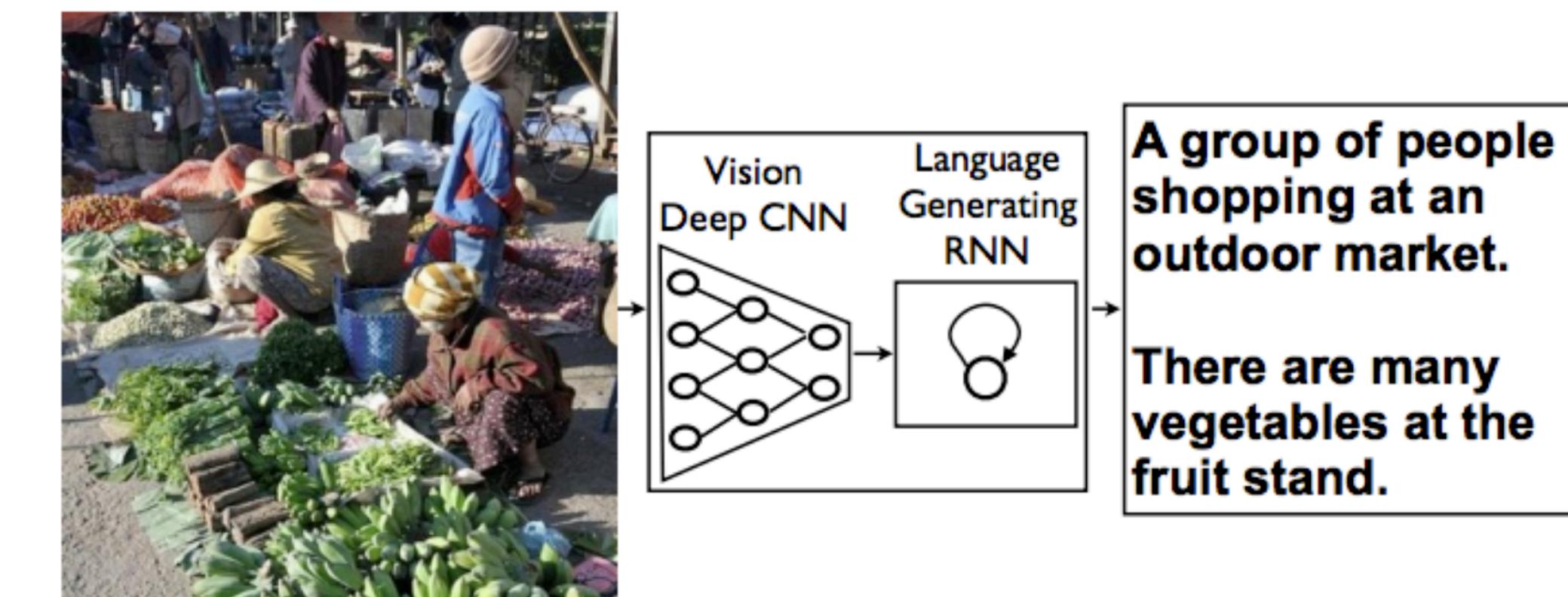
<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Recurrent neural network architectures for language processing (Thursday)



<https://towardsdatascience.com/recurrent-neural-networks-rnn-explained-the-eli5-way-3956887e8b75>

Architecture components are modular and can be composed, e.g. image captioning



https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788398060/3/ch03lvl1sec22/what-is-caption-generation

Case study: Image classification

Prototypical computer vision task:

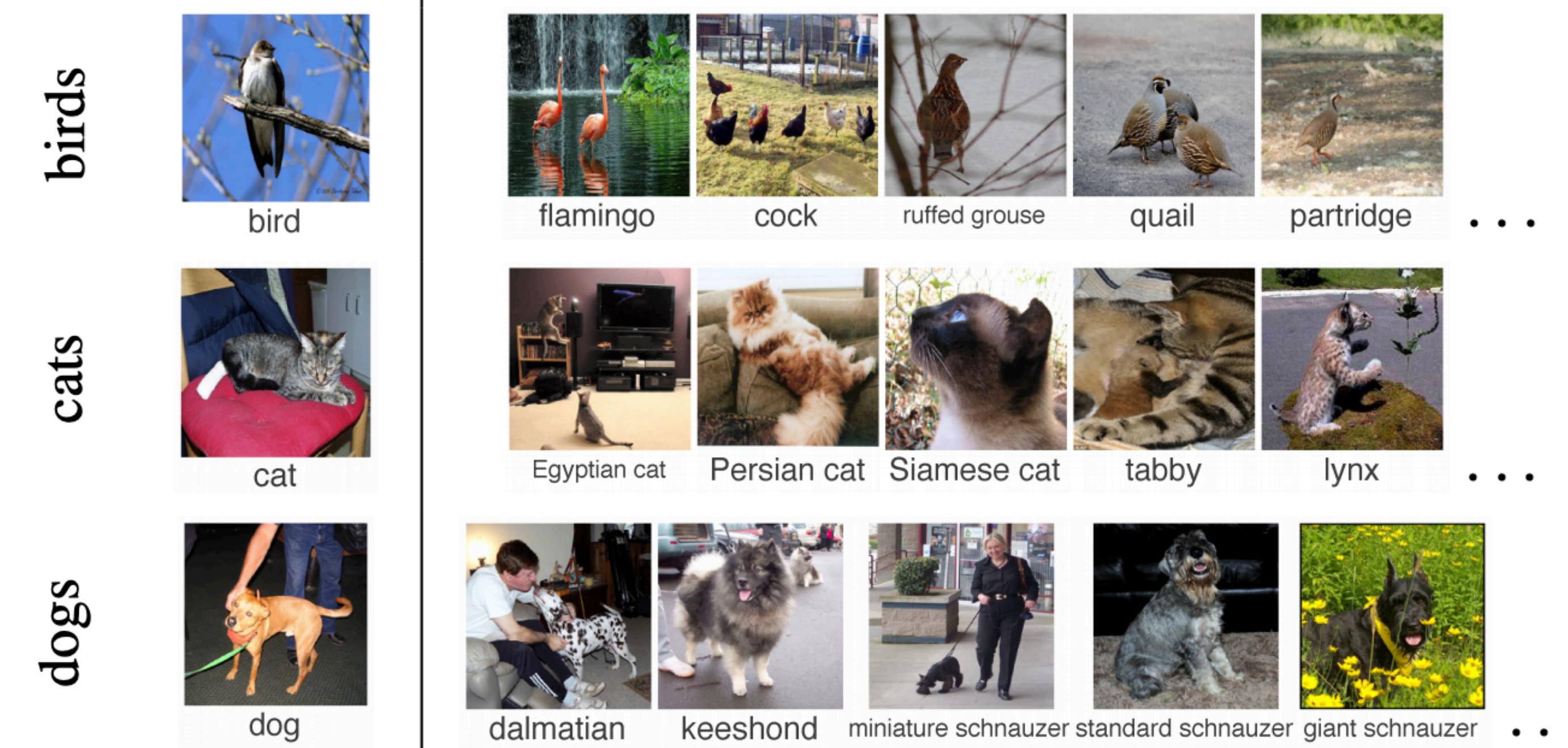
Given an image, classify according to what object it depicts.

Challenges:

- Viewpoint variation
- Illumination
- Deformation
- Occlusion
- Background clutter
- Intraclass variation

Labeled classes

Not even humans are at 100%



<http://ai.stanford.edu/~olga/papers/iccv13-ILSVRCanalysis.pdf>

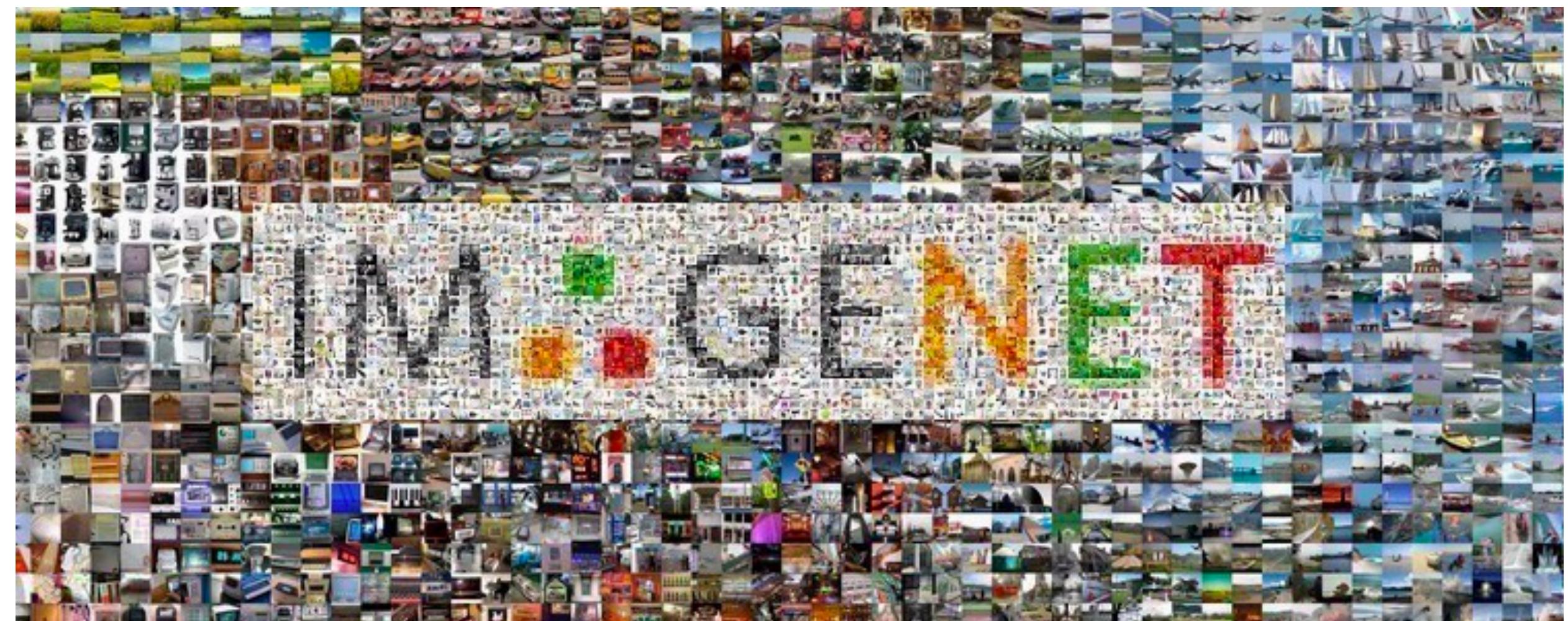
ImageNet

A large dataset for image classification

Assembled in 2009 by downloading lots of images from the web and crowdsourcing their labels.

- Training set: 1.2 million images
- Test set: 100,000 images
- 1000 classes

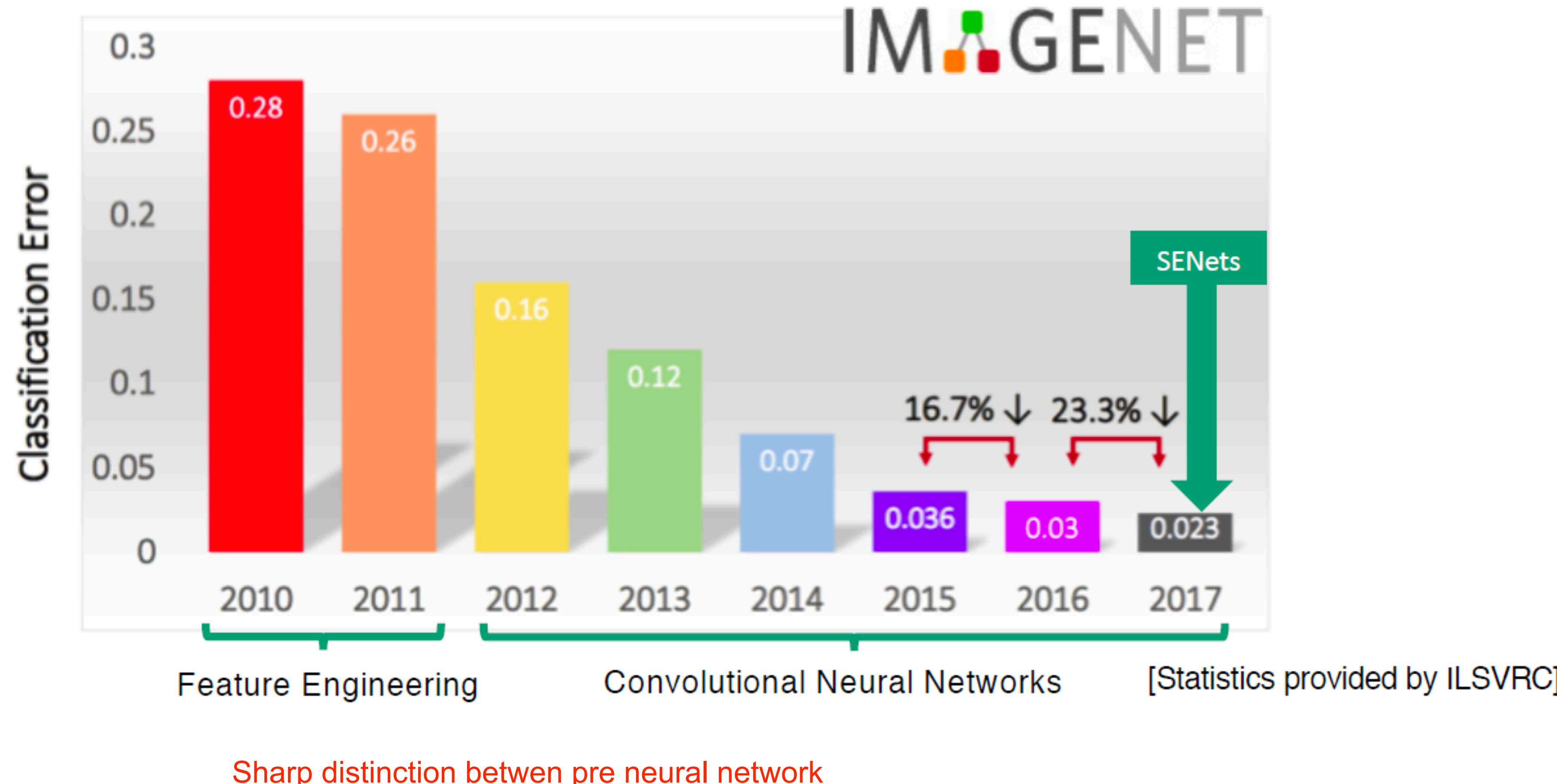
ImageNet Large Scale Visual Recognition Challenge (ILSVRC) held annually between 2010-2017.



<https://medium.com/syncedreview/sensetime-trains-imagenet-alexnet-in-record-1-5-minutes-e944ab049b2c>

ILSVRC results over the years

Base line is 1/# of classes



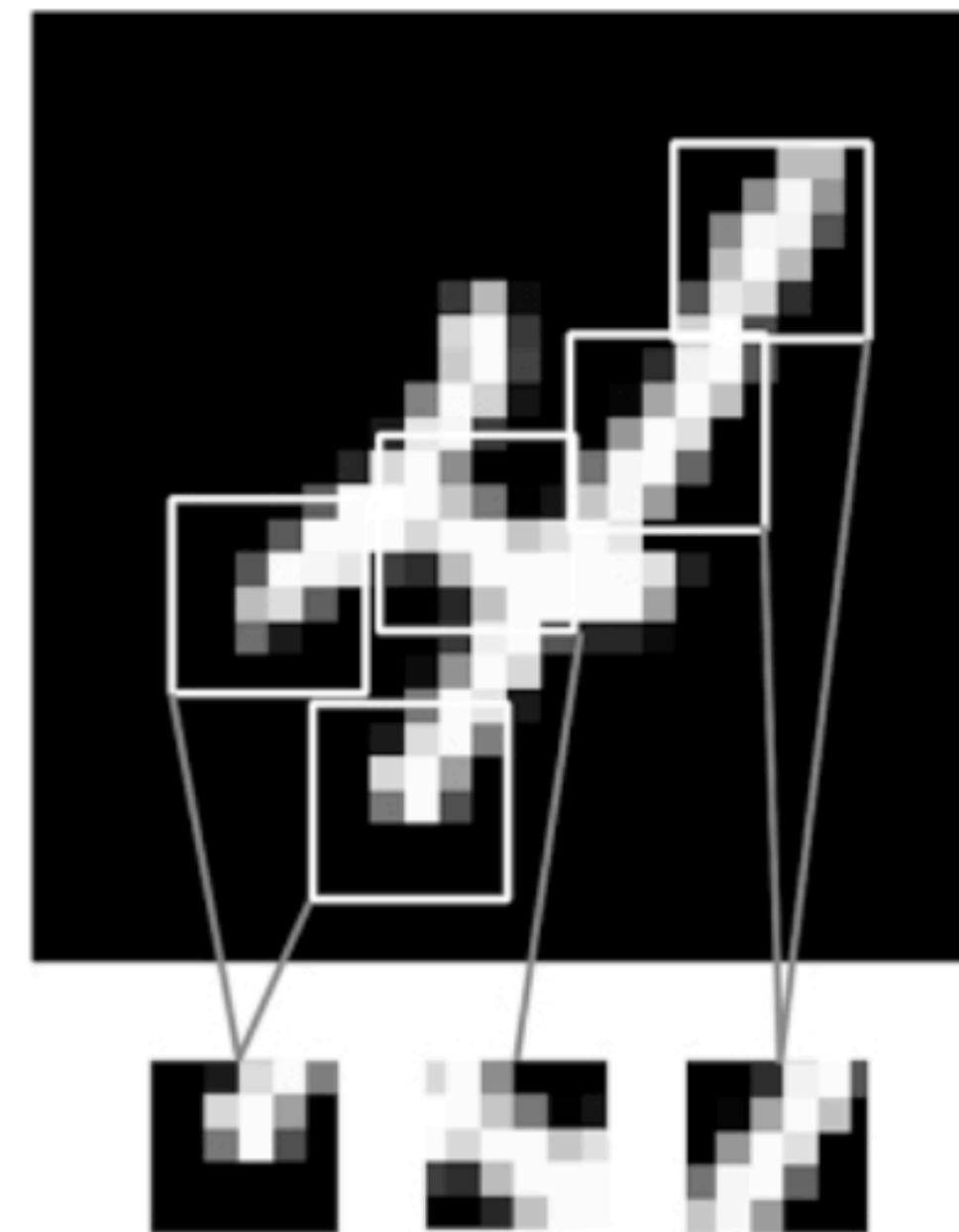
Convolutional neural networks (CNNs) have dominated since 2012.

CNNs are built on image-specific properties

CNNs are built on image-specific properties

Images can be broken into local patterns

Figure 5.1. Images can be broken into local patterns such as edges, textures, and so on.



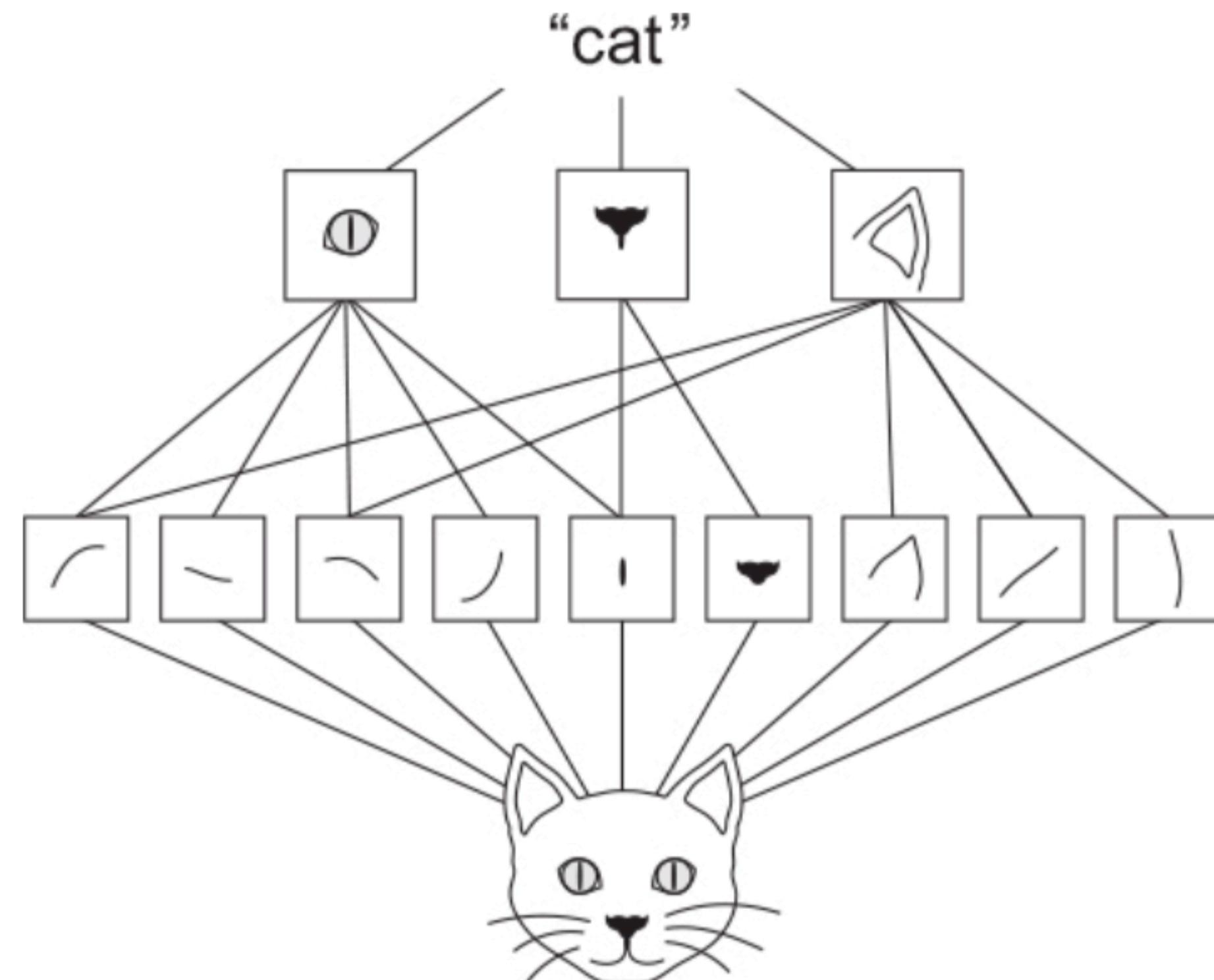
CNNs are built on image-specific properties

Patterns have hierachal structure

Figure 5.1. Images can be broken into local patterns such as edges, textures, and so on.

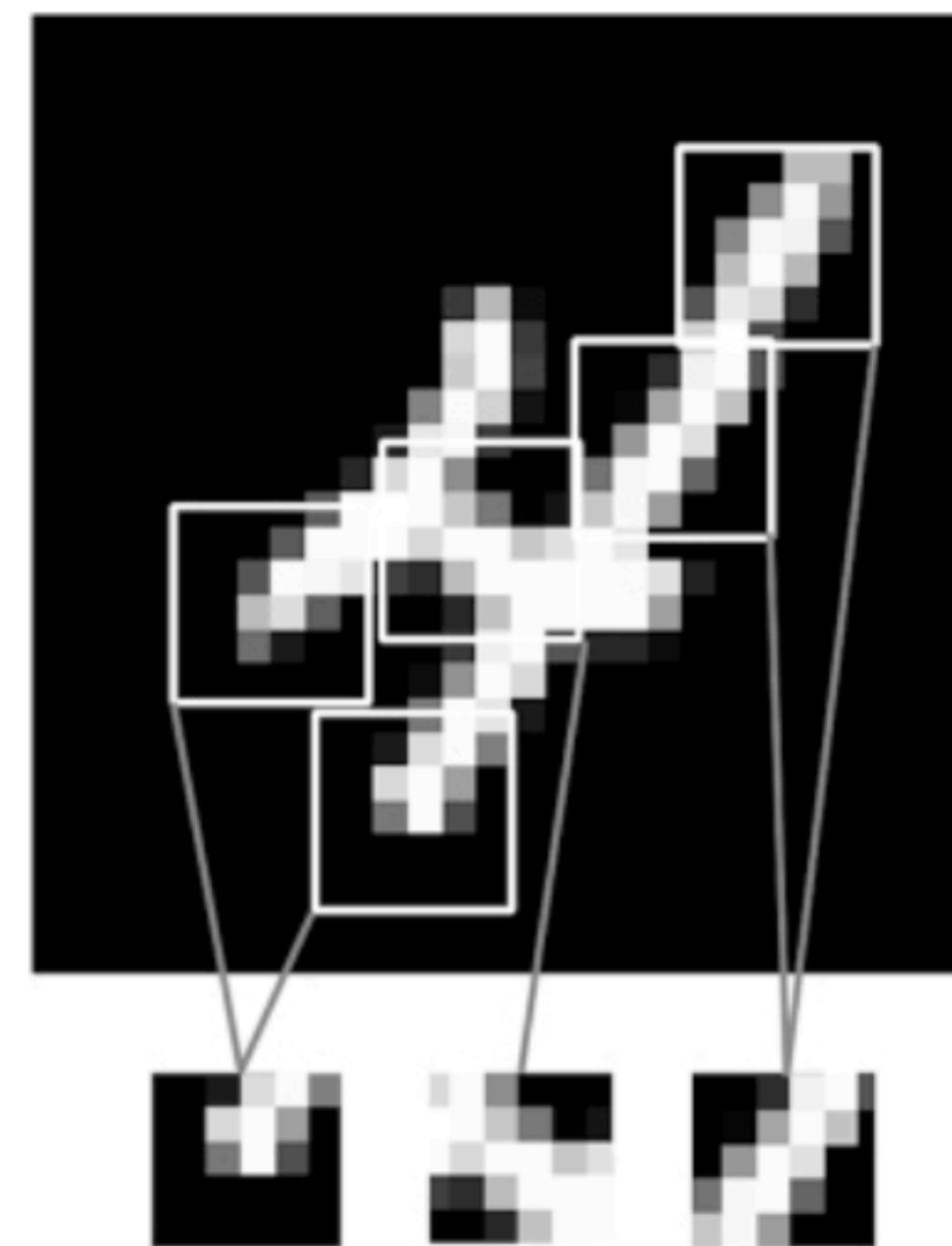


Figure 5.2. The visual world forms a spatial hierarchy of visual modules: hyperlocal edges combine into local objects such as eyes or ears, which combine into high-level concepts such as “cat.”



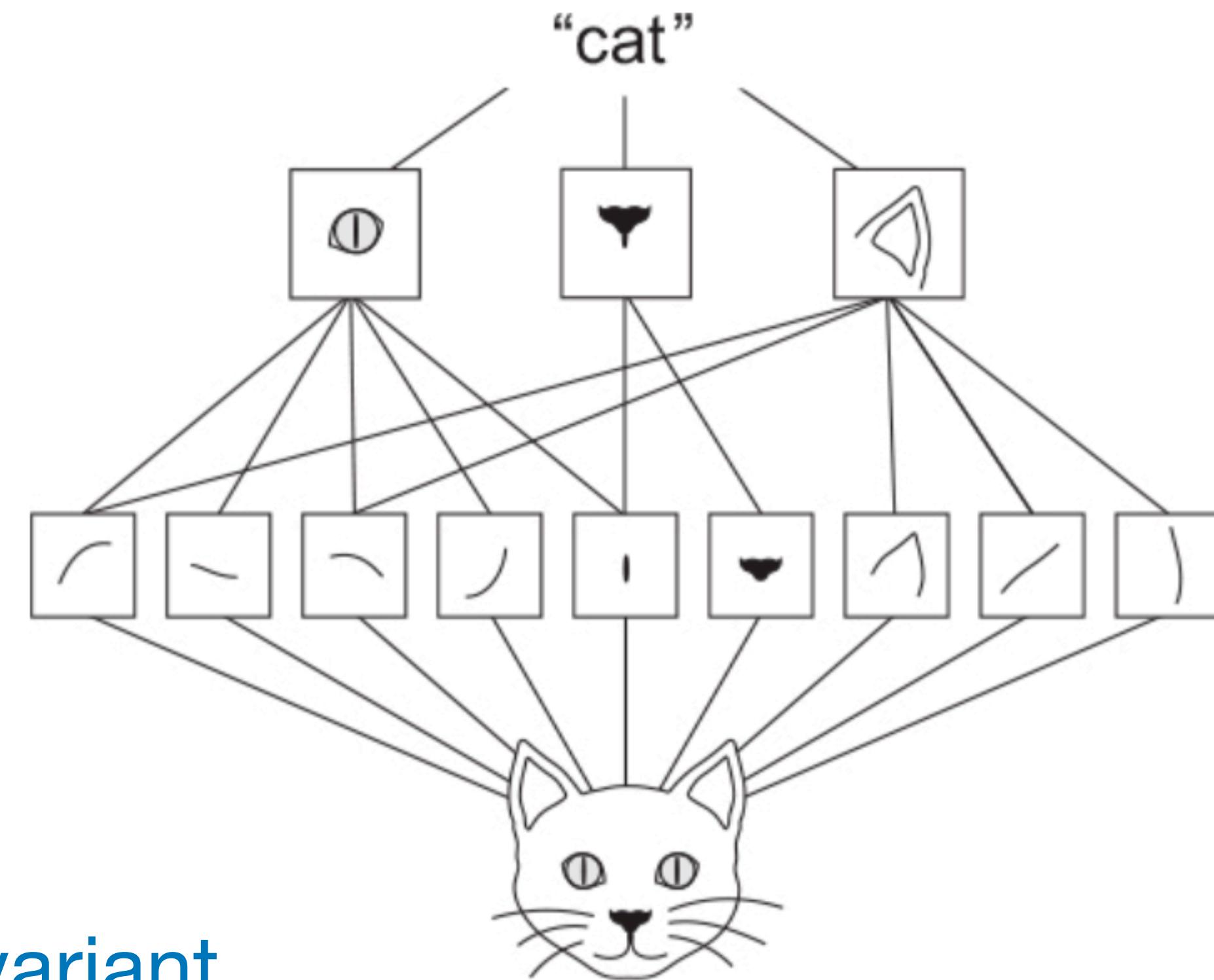
CNNs are built on image-specific properties

Figure 5.1. Images can be broken into local patterns such as edges, textures, and so on.



- Patterns are
- Local
 - Translation-invariant
 - Hierarchical

Figure 5.2. The visual world forms a spatial hierarchy of visual modules: hyperlocal edges combine into local objects such as eyes or ears, which combine into high-level concepts such as “cat.”



Convolution: Searching for patterns

Filter (3x3)



(pattern)

Operation that you do to an input image but also to a filter

A filter is basically a pattern you're looking for

E.g. corner image

Convolution asks where in the image these patterns occur

0	1	2
2	2	0
0	1	2

Input image



Activation map



(presence of pattern)

3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Notice -> this is sum of navy square

Convolution: Searching for patterns

Filter (3x3)



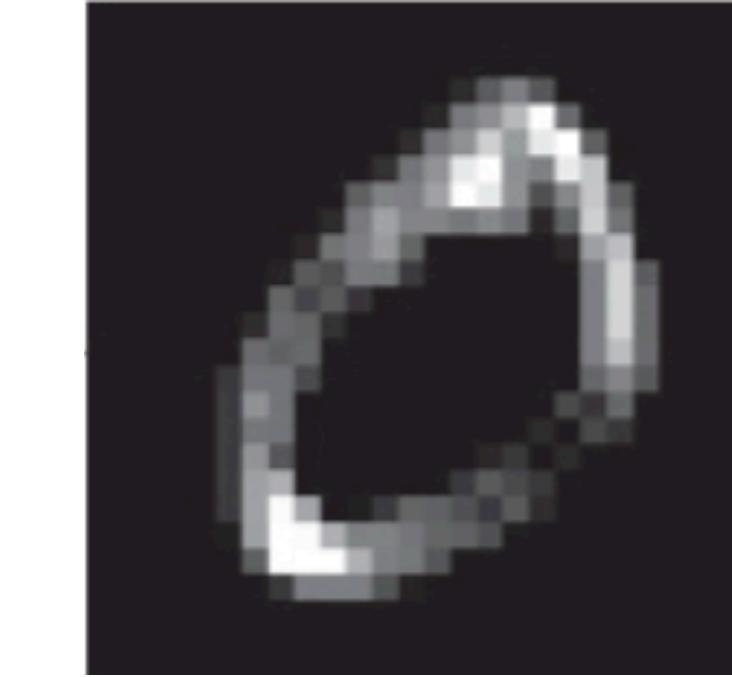
(pattern)

0	1	2
2	2	0
0	1	2

Input image



Activation map



(presence of
pattern)

3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

Convolution: Searching for patterns

Filter (3x3)



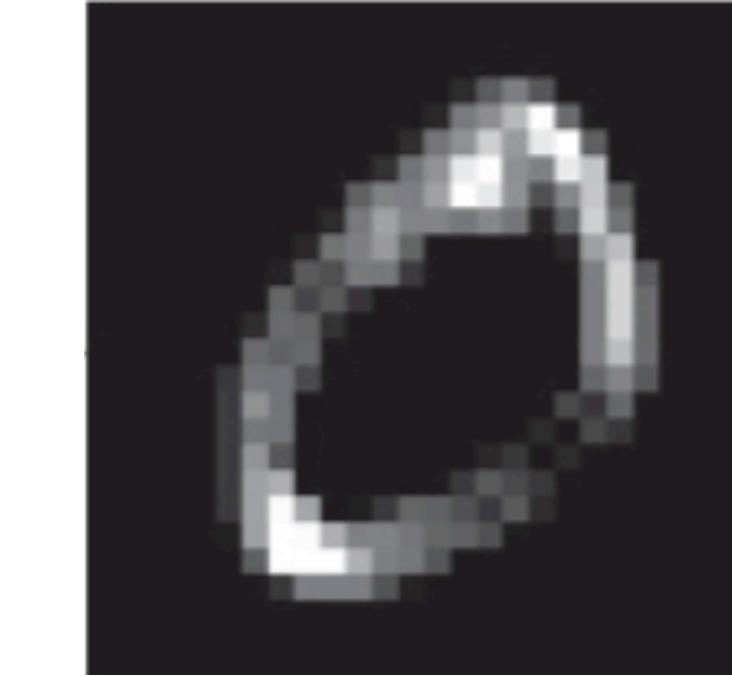
(pattern)

0	1	2
2	2	0
0	1	2

Input image



Activation map



(presence of
pattern)

3_0	3_1	2_2	1	0
0_2	0_2	1_0	3	1
3_0	1_1	2_2	2	3
2	0	0	2	2
2	0	0	0	1

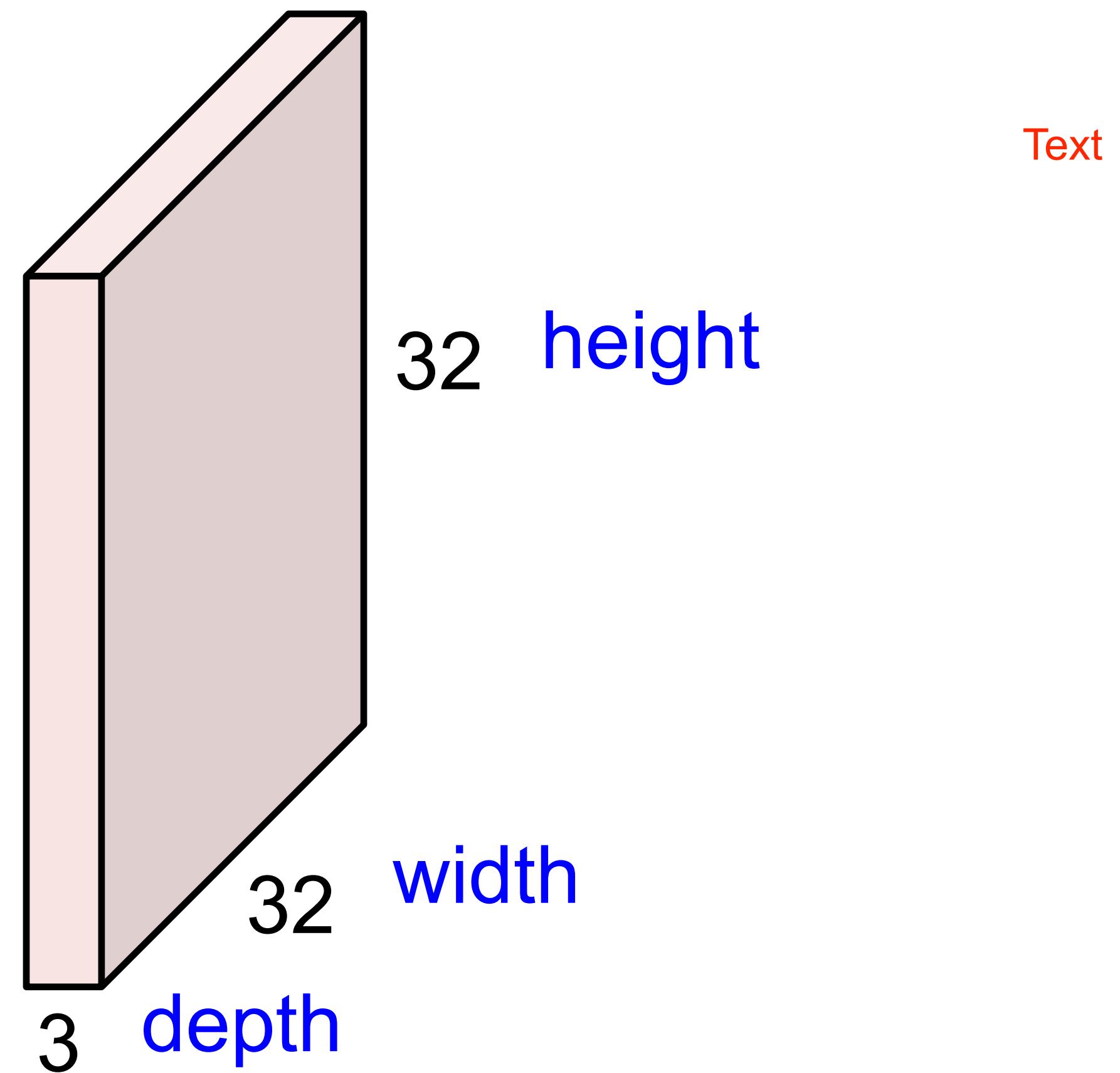
12.0	12.0	17.0
10.0	17.0	19.0
9.0	6.0	14.0

We want to use many filters, each sensitive to a different kind of pattern.

Convolution Layer

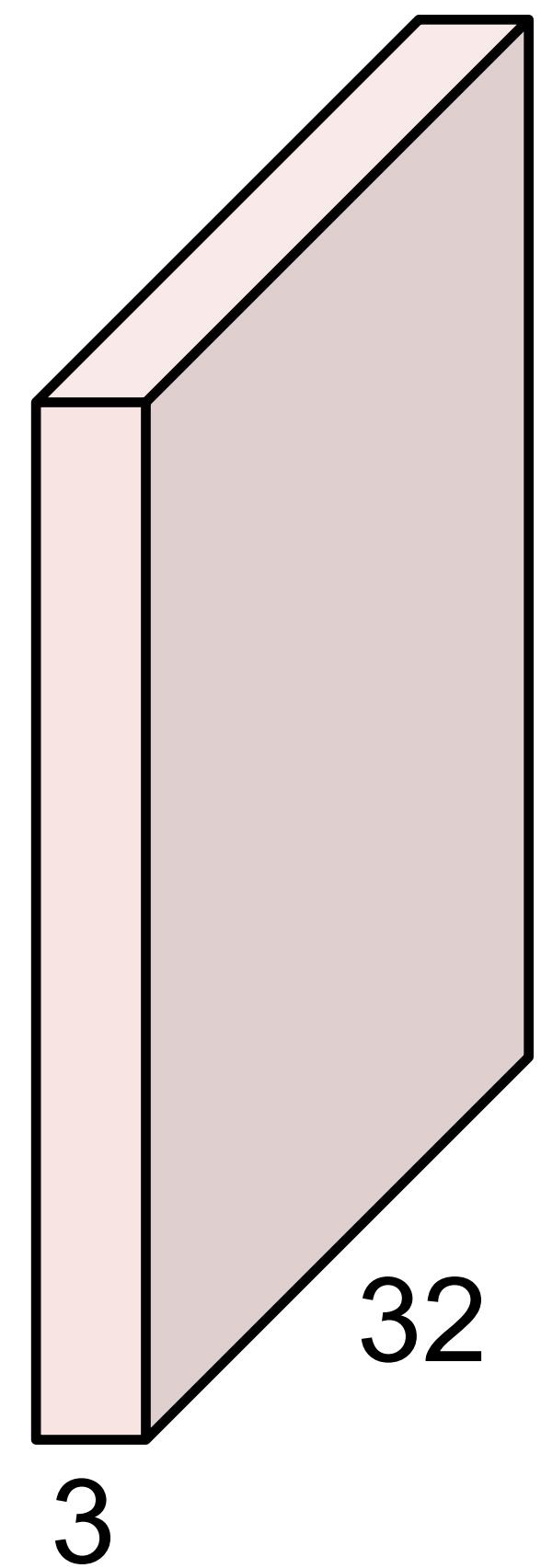
Note: This slide and several following ones are borrowed from Stanford's [CS231n](#).

32x32x3 image (images typically have red, green, and blue **channels**.)

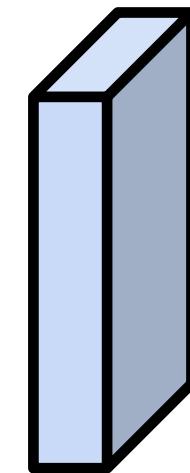


Convolution Layer

32x32x3 image

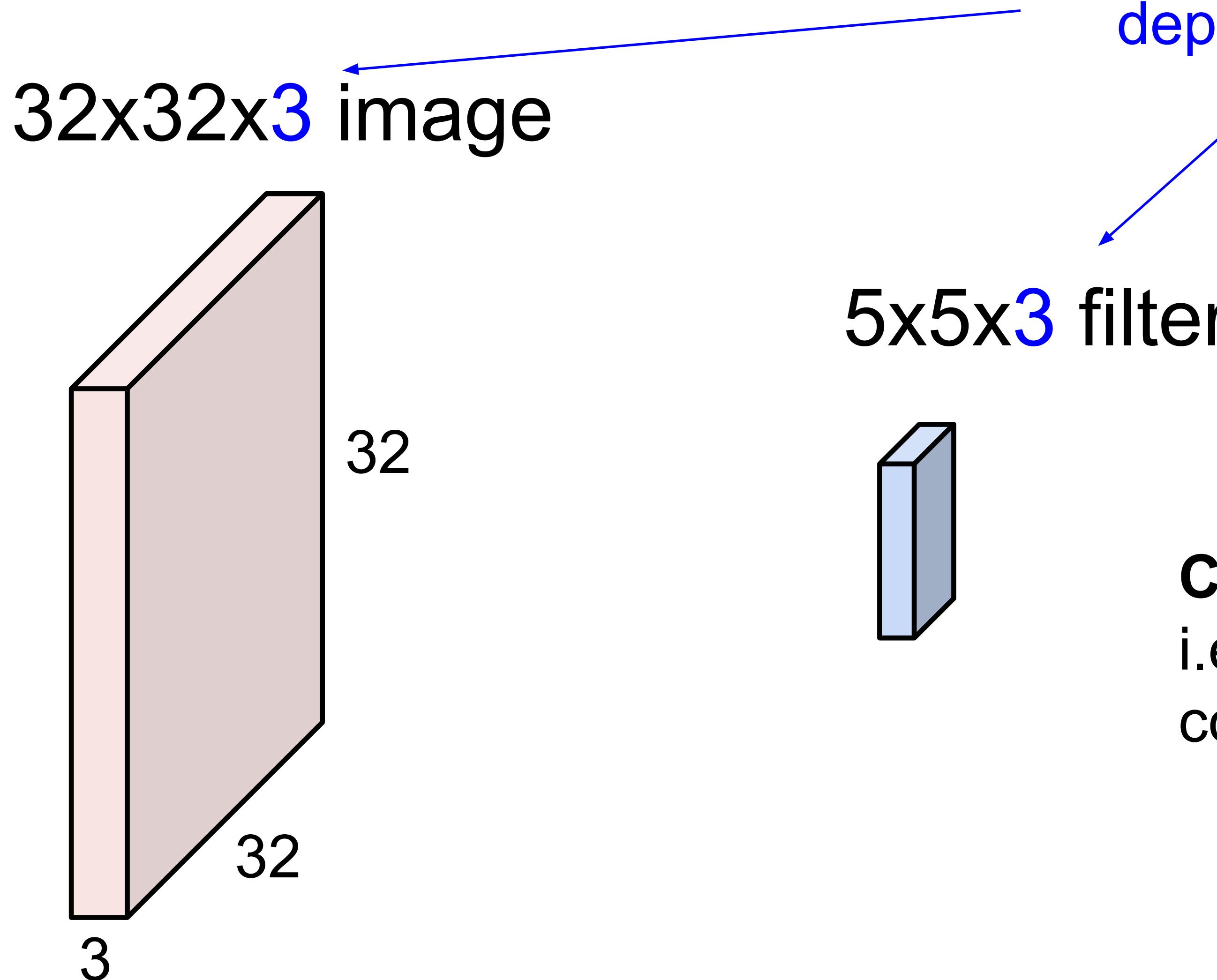


5x5x3 filter



Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

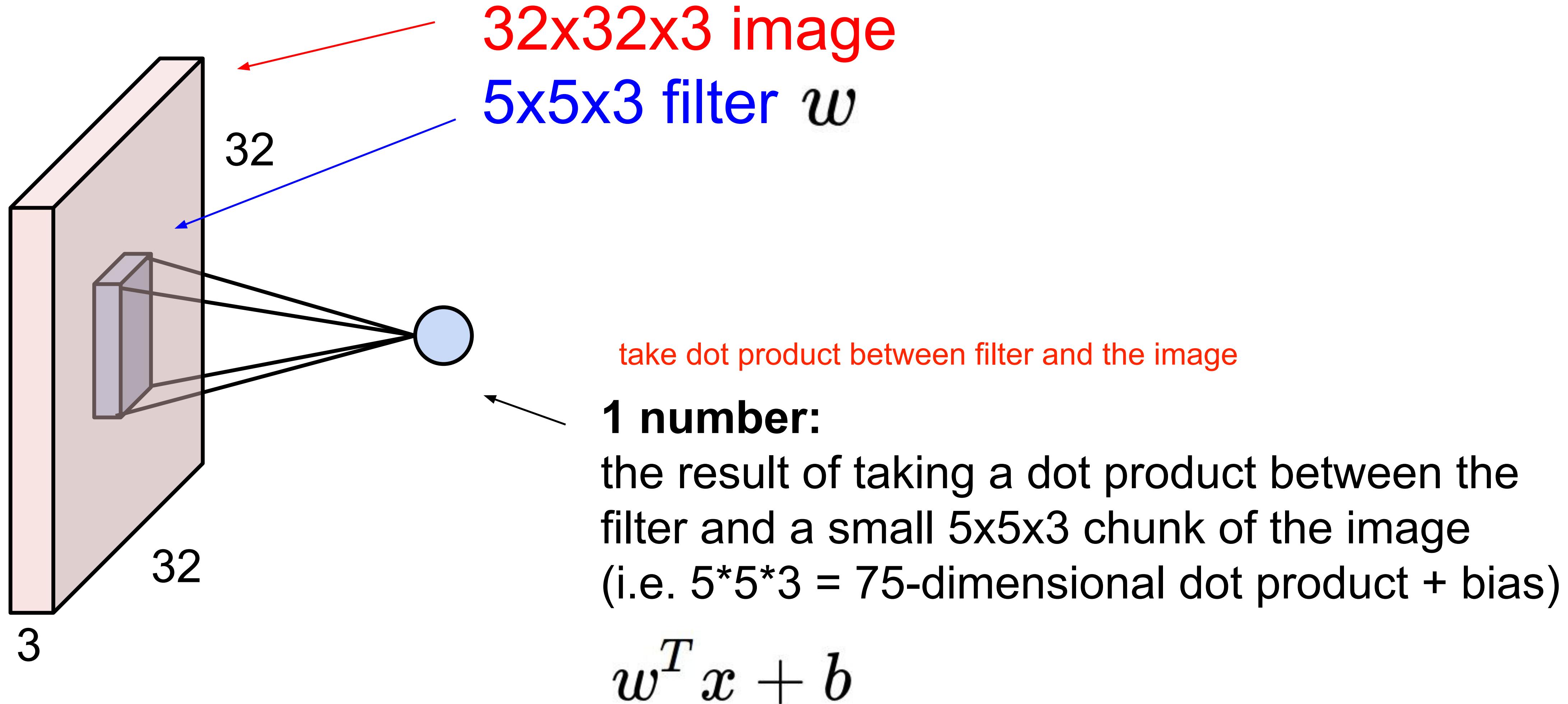
Convolution Layer



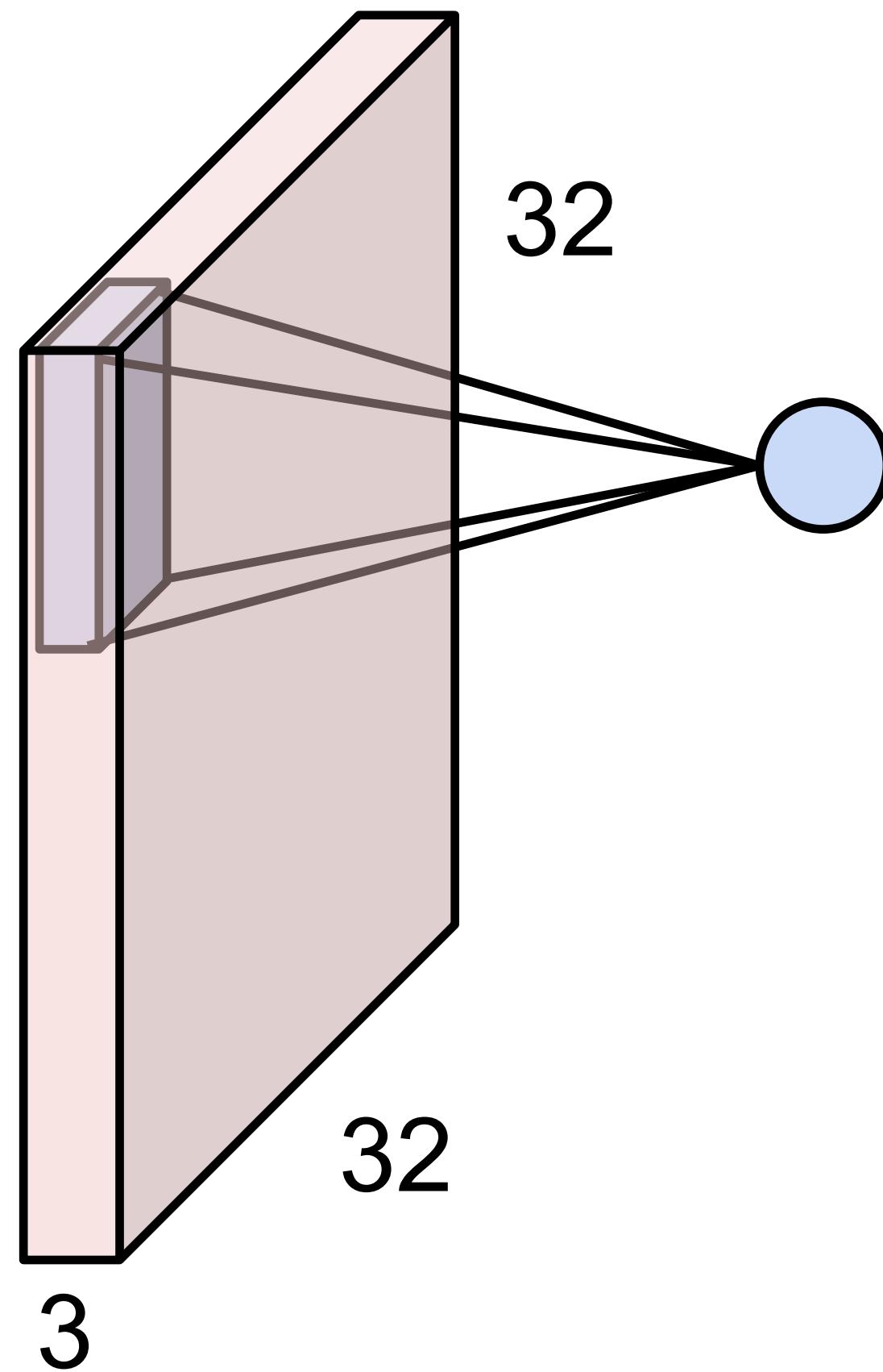
Filters always extend the full depth of the input volume

Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

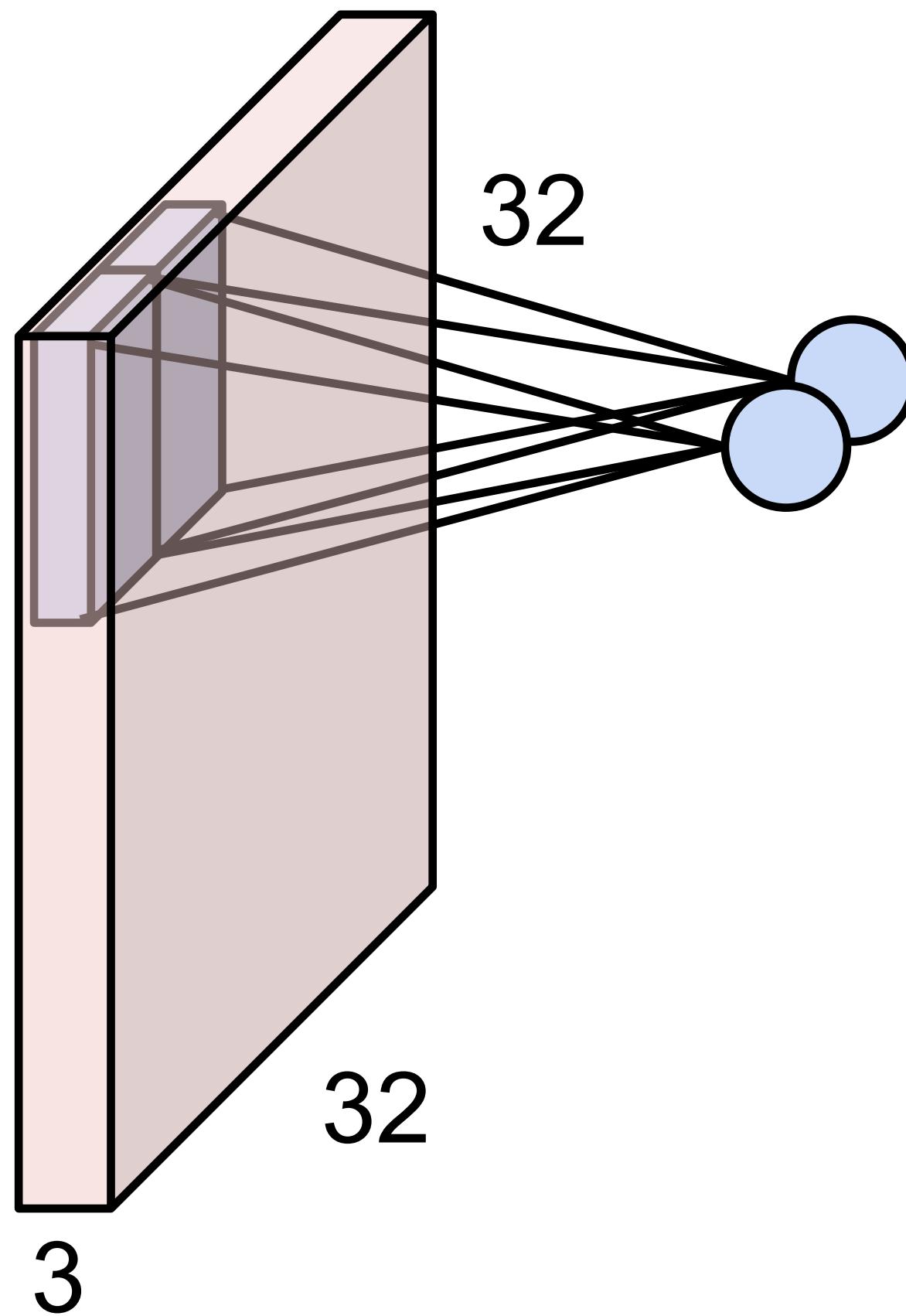
Convolution Layer



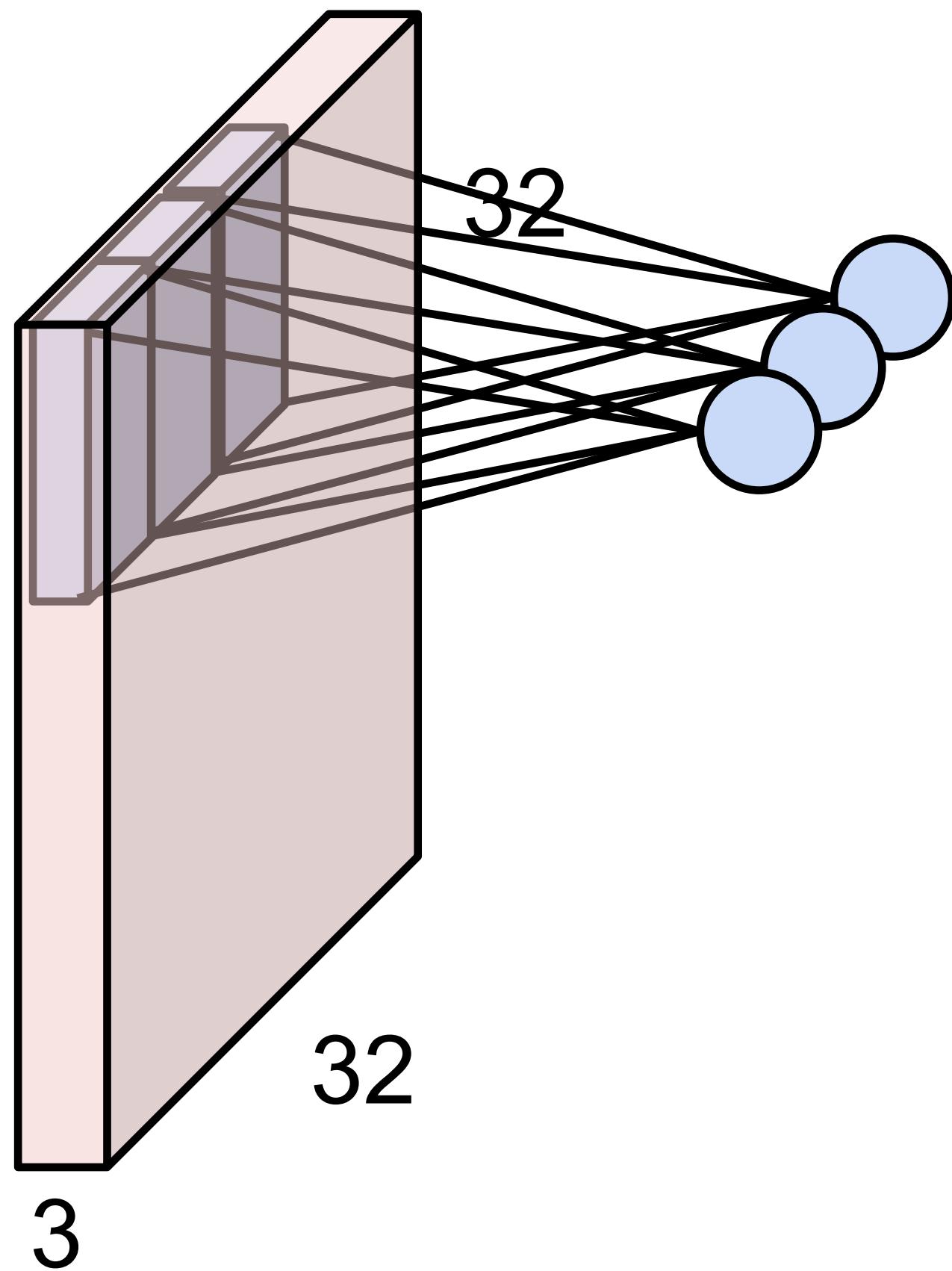
Convolution Layer



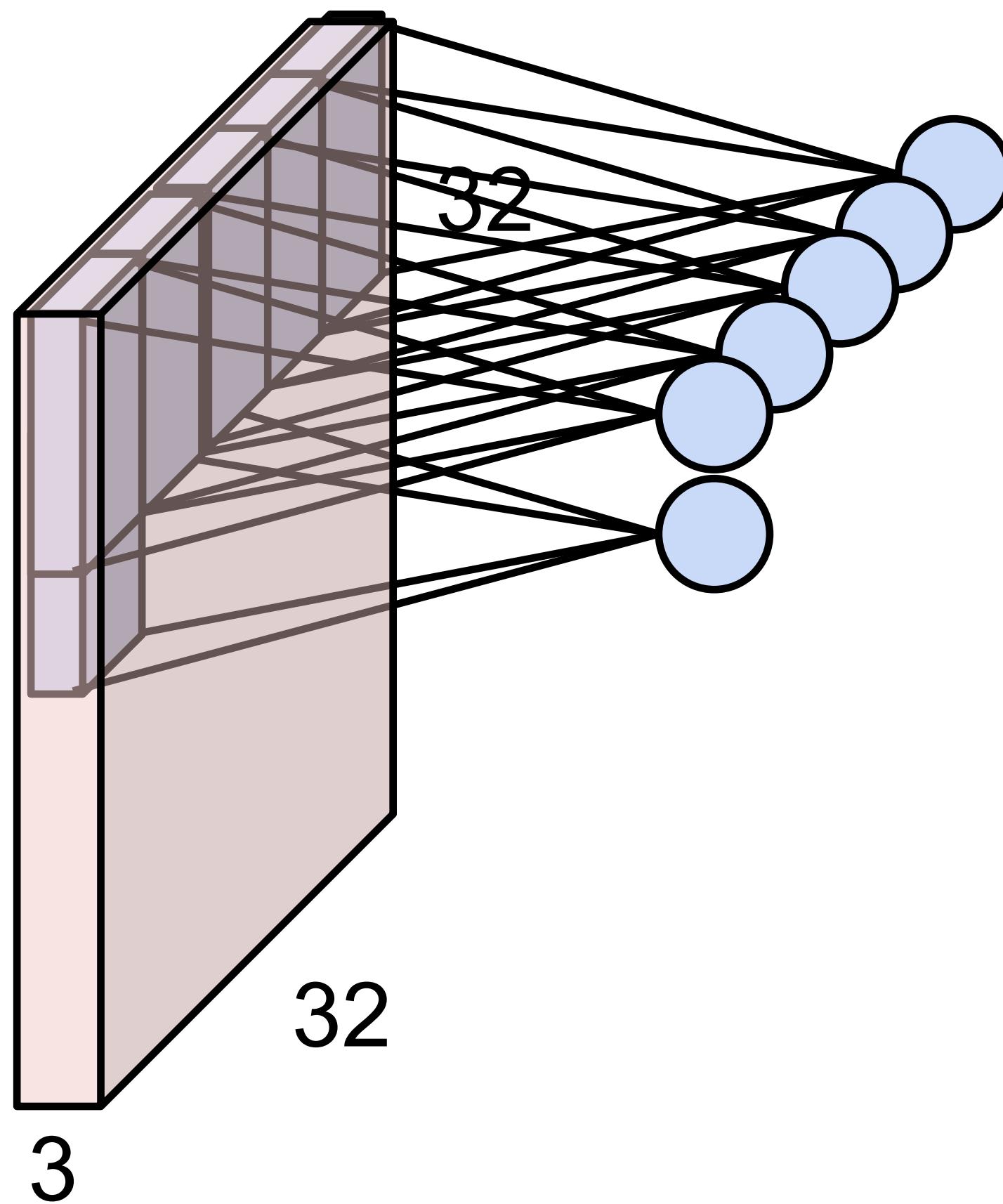
Convolution Layer



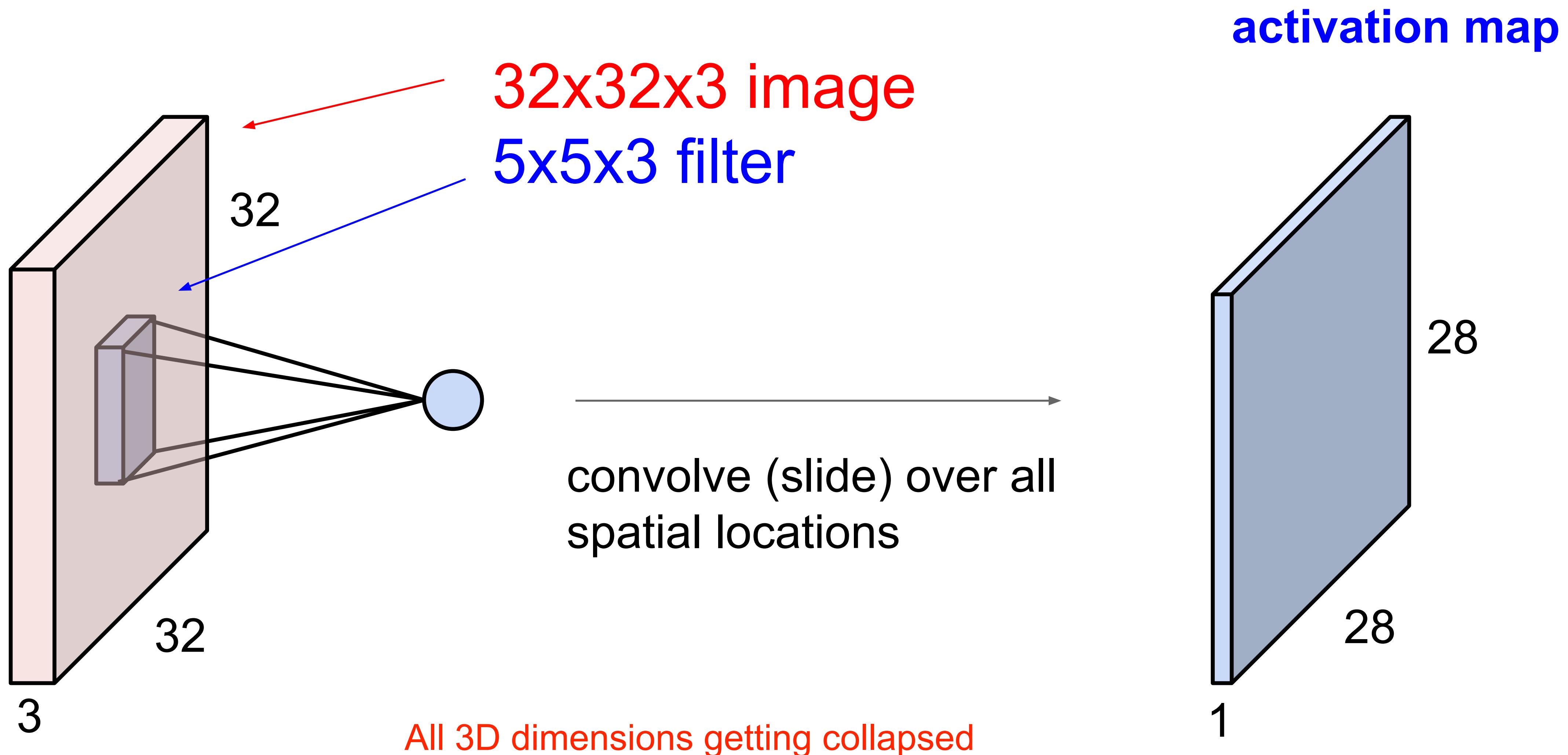
Convolution Layer



Convolution Layer



Convolution Layer

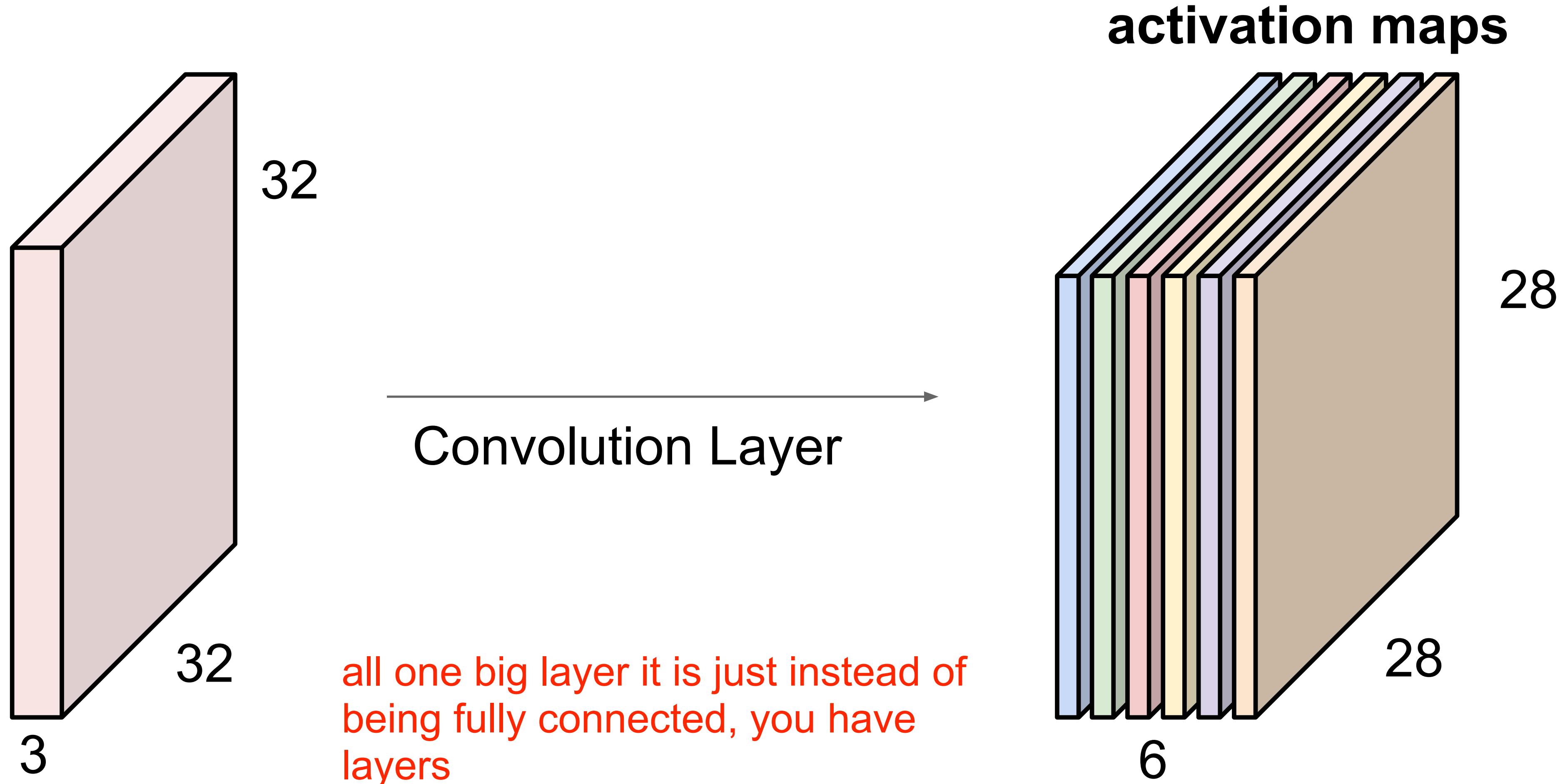


Convolution Layer

consider a second, green filter

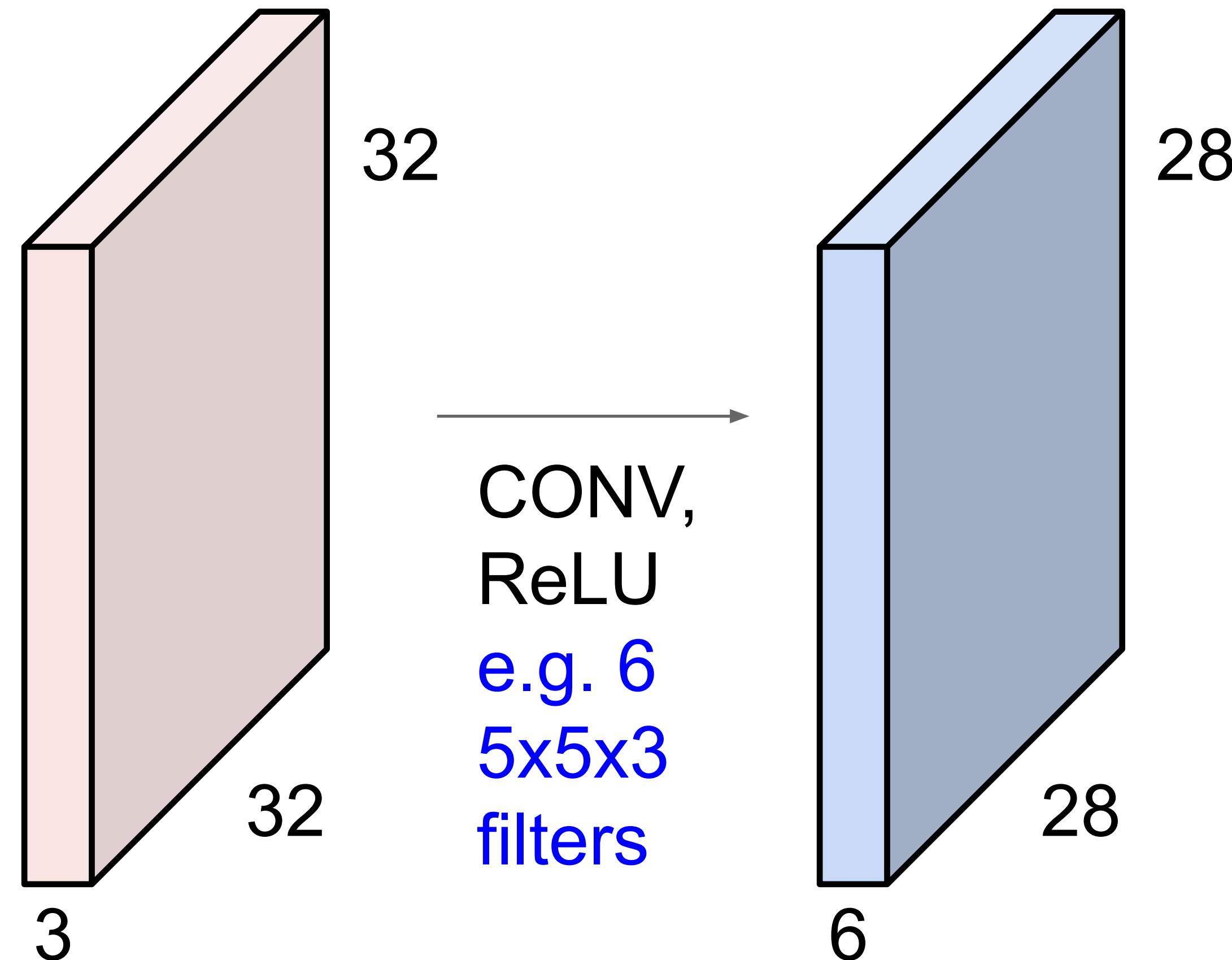


For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

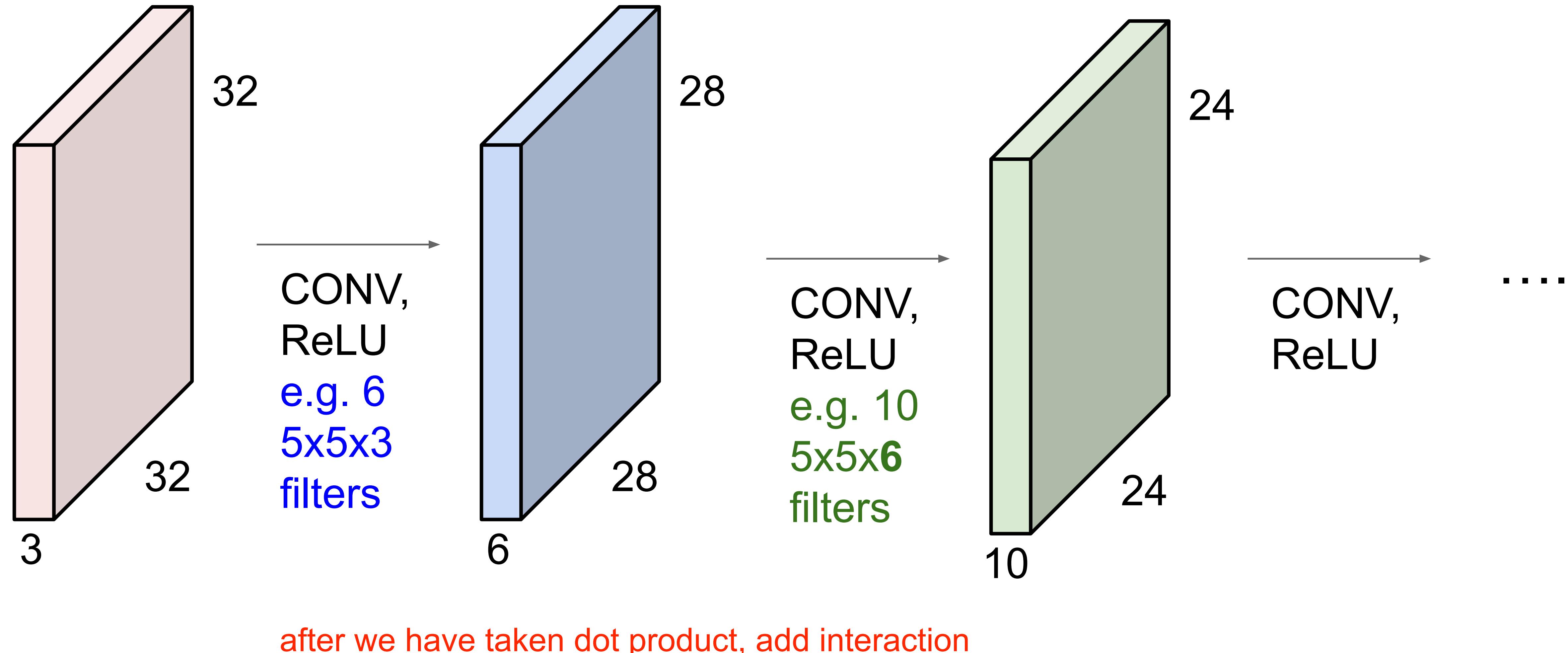


We stack these up to get a “new image” of size 28x28x6!

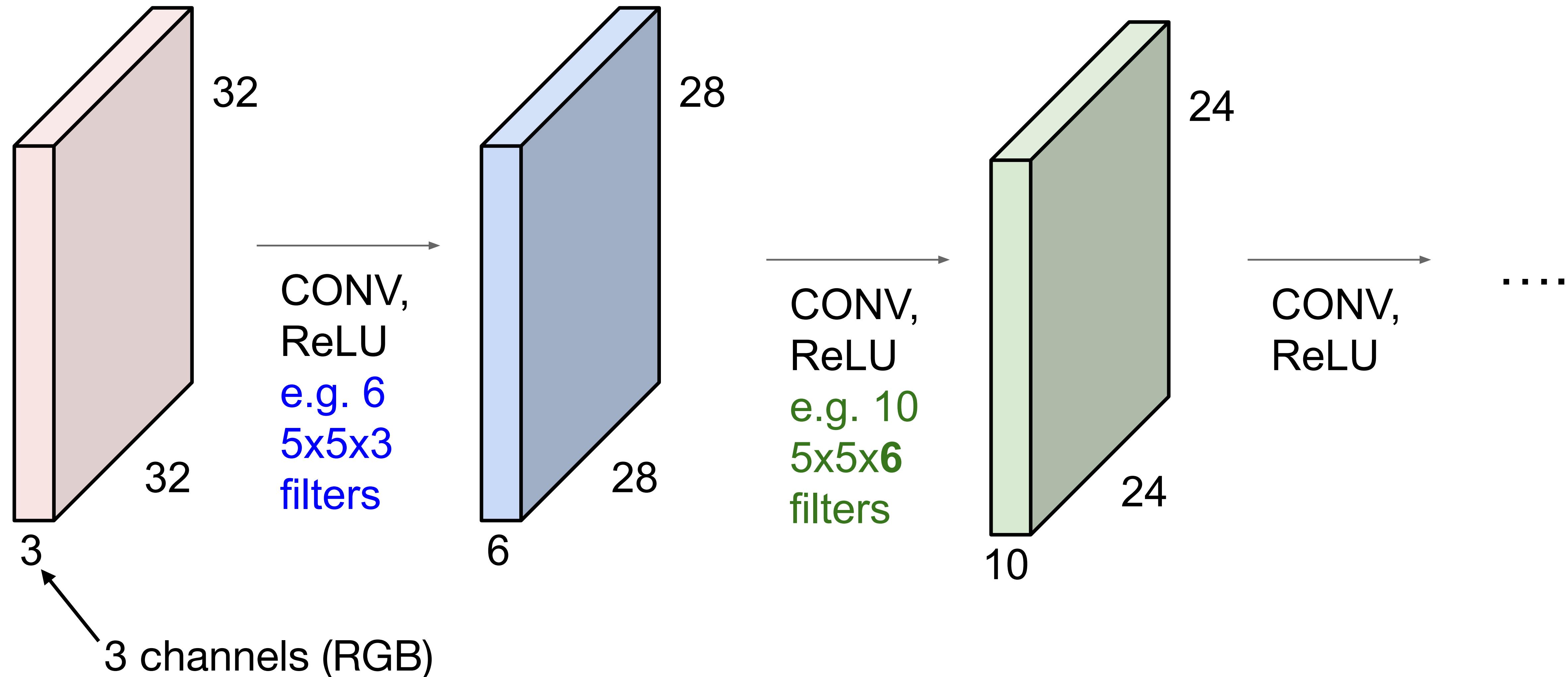
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



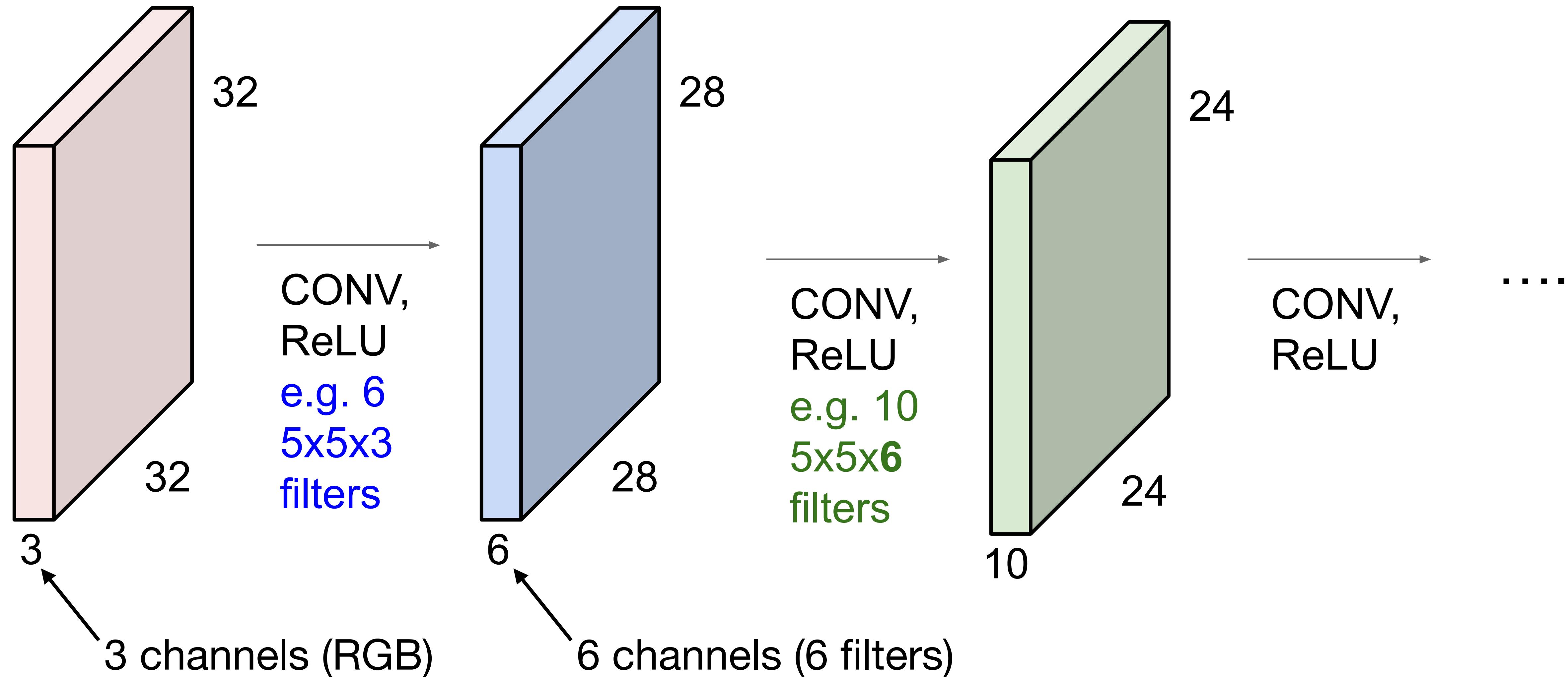
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



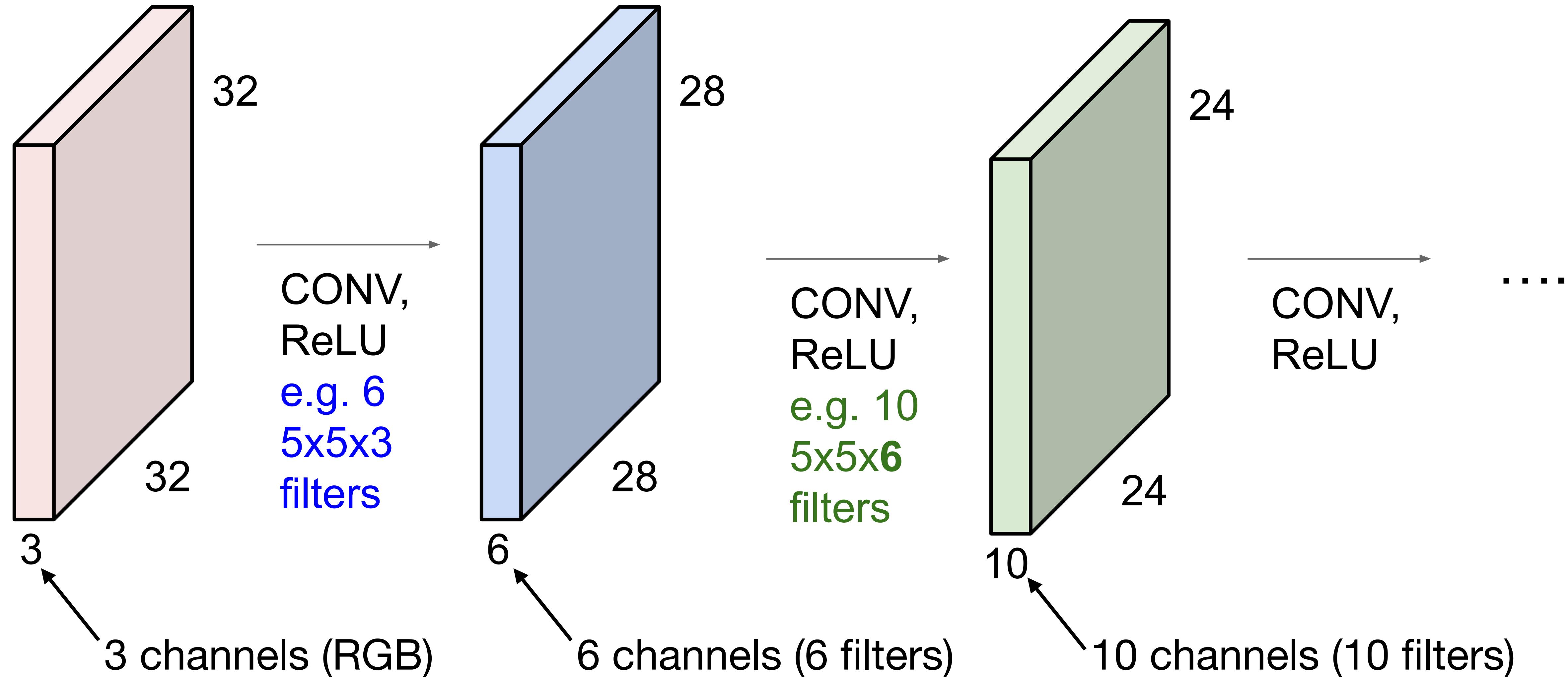
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions

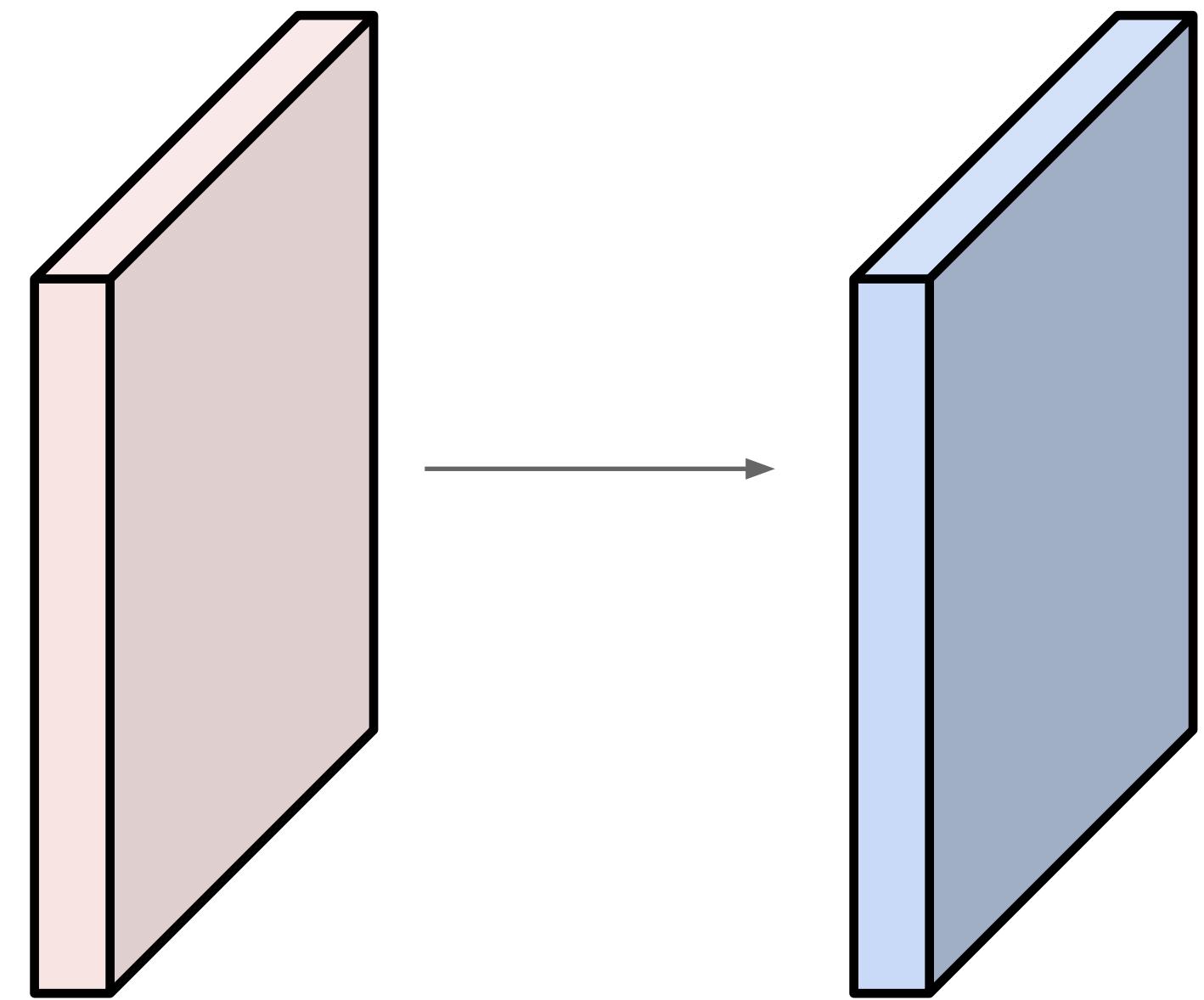


Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



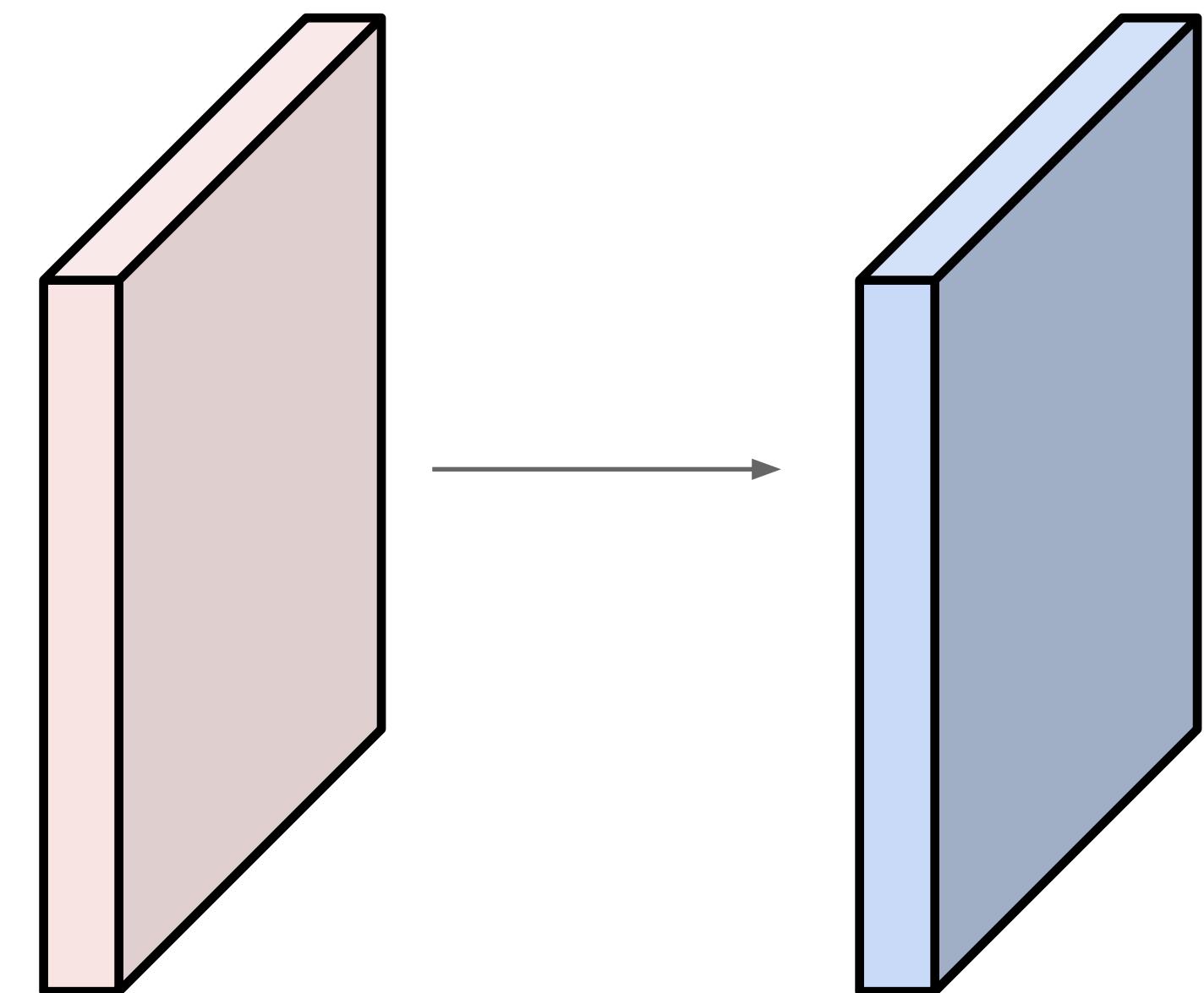
Input volume: 32x32x3

10 5x5 filters In reality, 5x5x3 filters + 1 bc bias term



Number of parameters in this layer?

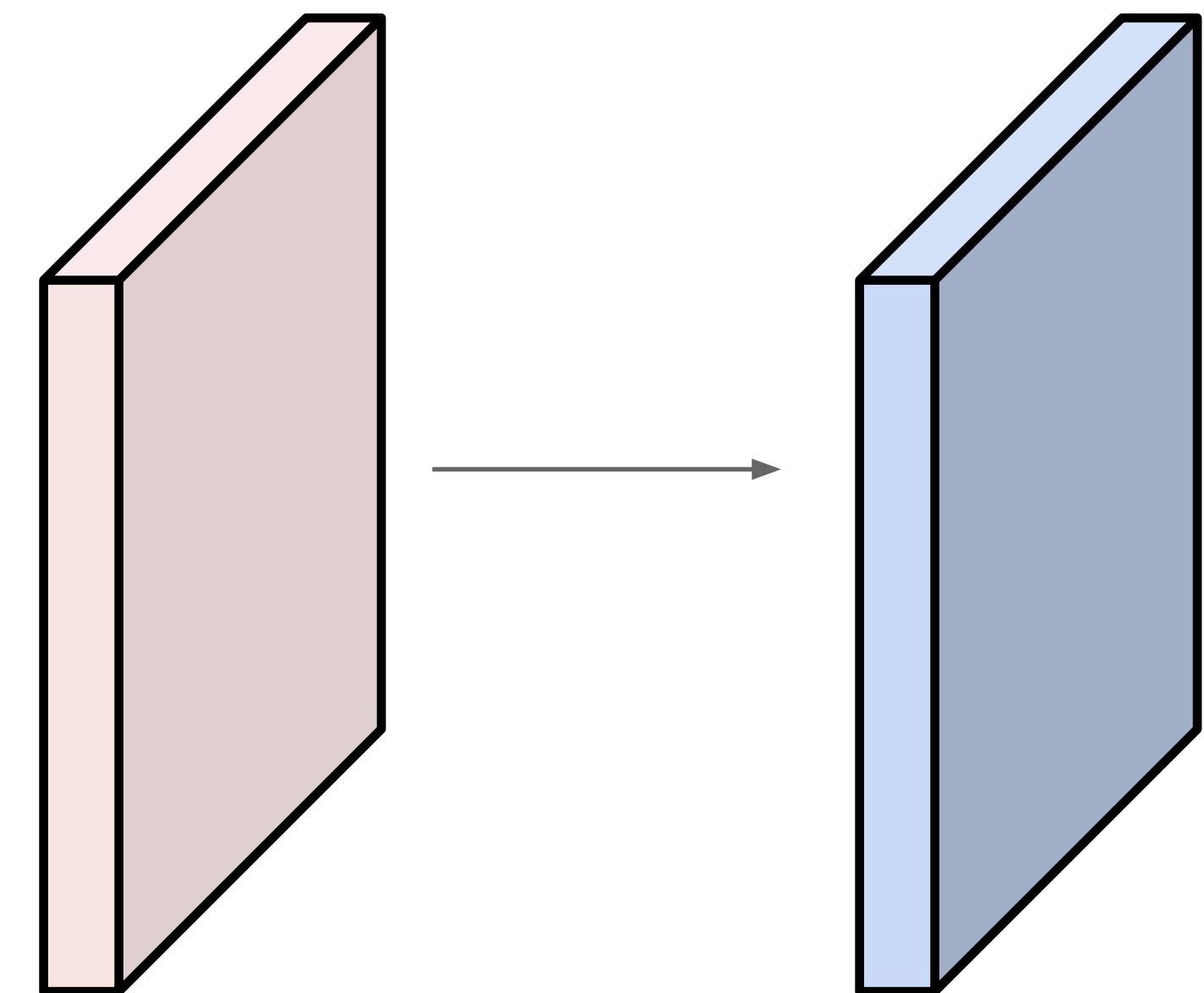
This is different than size



Input volume: **32x32x3**
10 5x5 filters

Number of parameters in this layer?

each filter has $5*5*3 + 1 = 76$ params (+1 for bias)
 $\Rightarrow 76*10 = 760$



Input volume: **32x32x3**
10 5x5 filters

Number of parameters in this layer?

each filter has $5*5*3 + 1 = 76$ params (+1 for bias)
 $\Rightarrow 76*10 = 760$

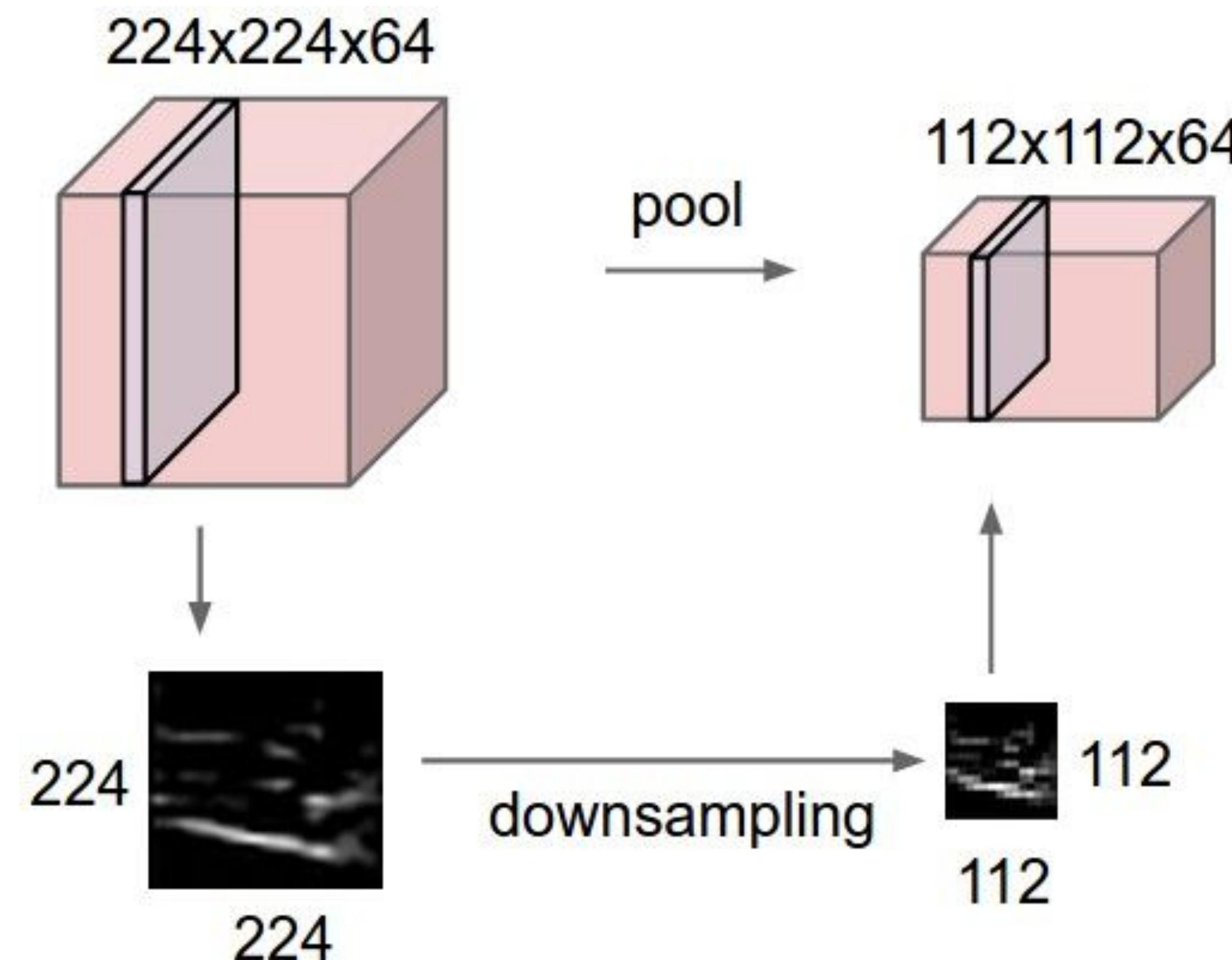
In general, parameters in conv layer =
(filter width x filter height x input channels + 1) x number of filters.

Pooling layer

- makes the representations smaller and more manageable
- operates over each activation map independently:

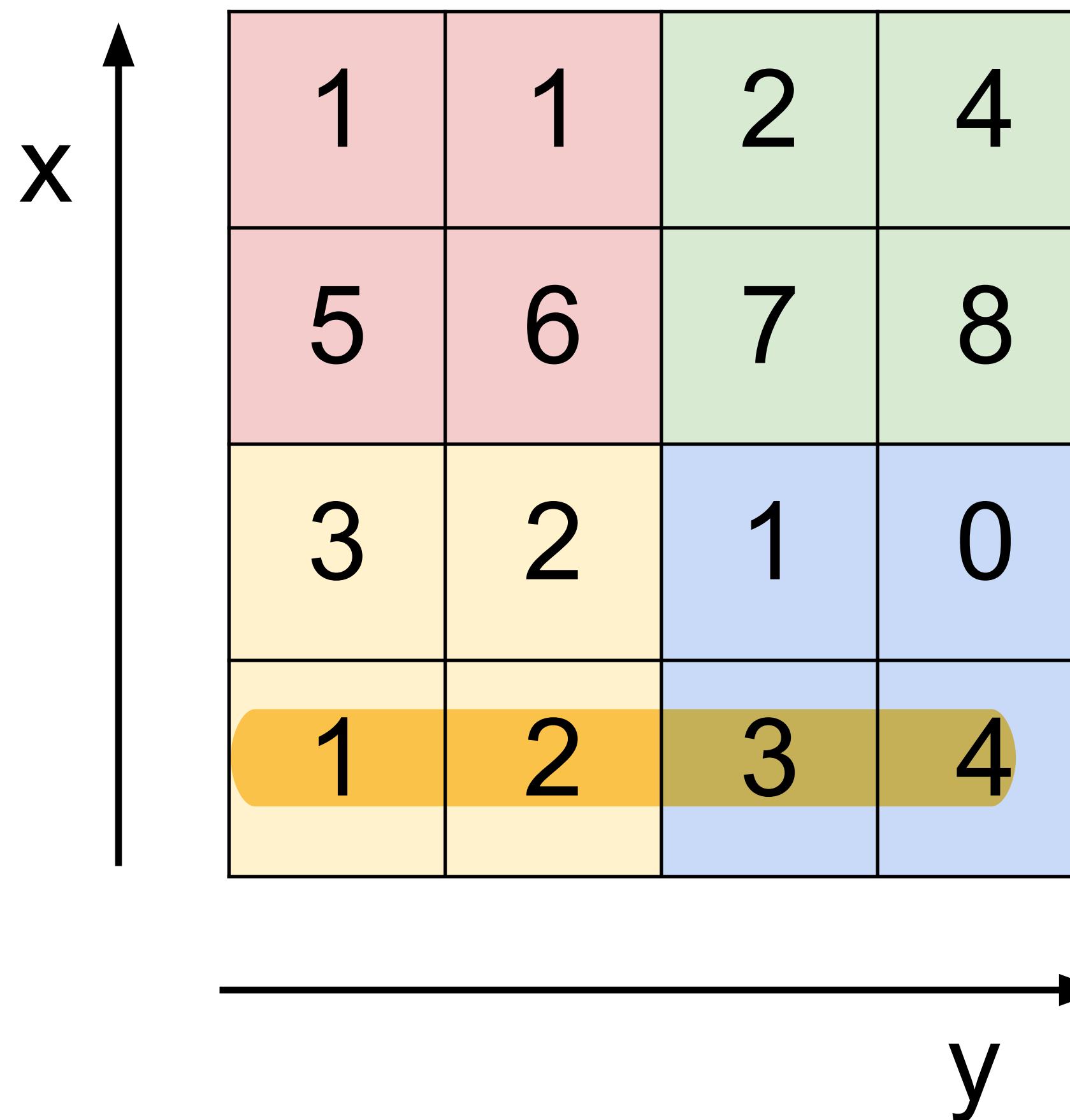
while convolutions take
the entire depth of the
image and they sort of
do a dot product

pooling layers does this
for each slice

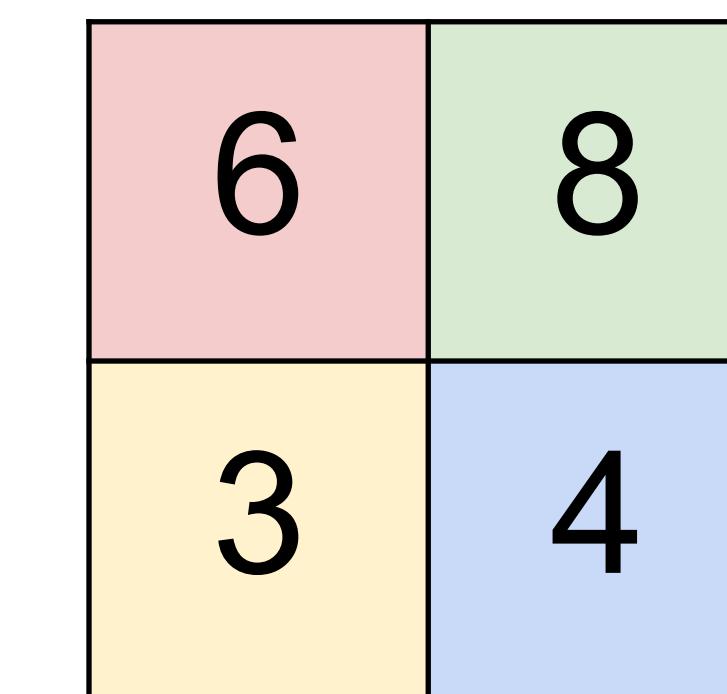


MAX POOLING

Single depth slice



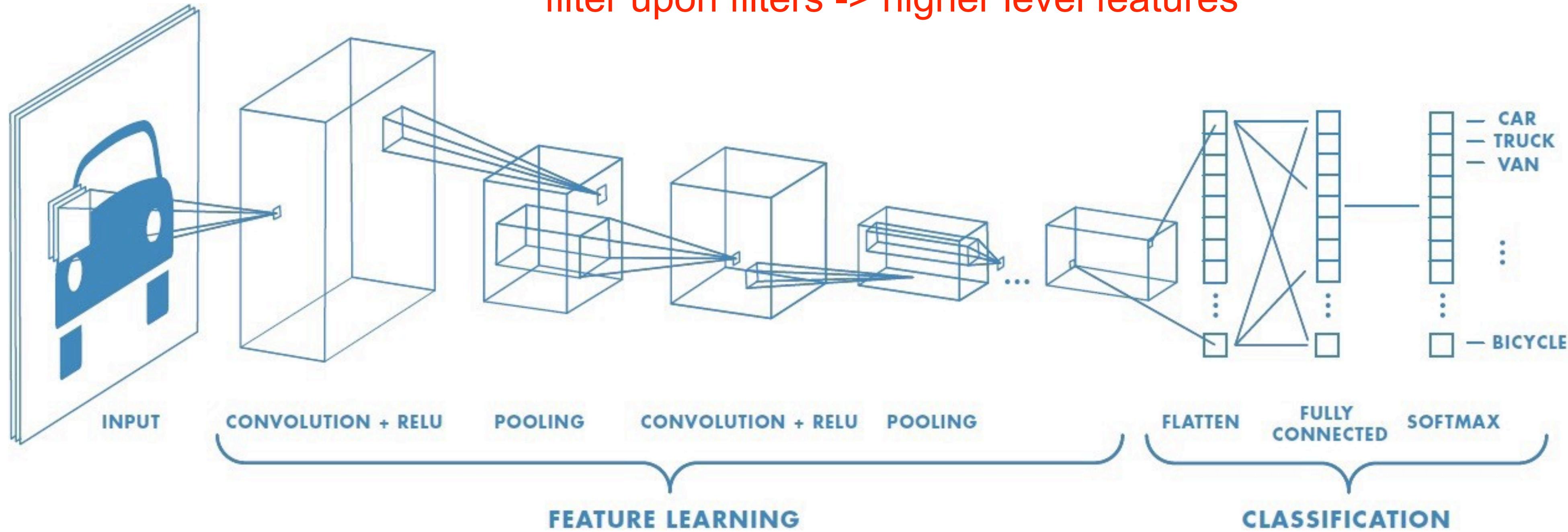
max pool with 2x2 filters
and stride 2



activation by activation instead of
collapsing across activation loops

Convolutional neural networks

filter upon filters -> higher level features



<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

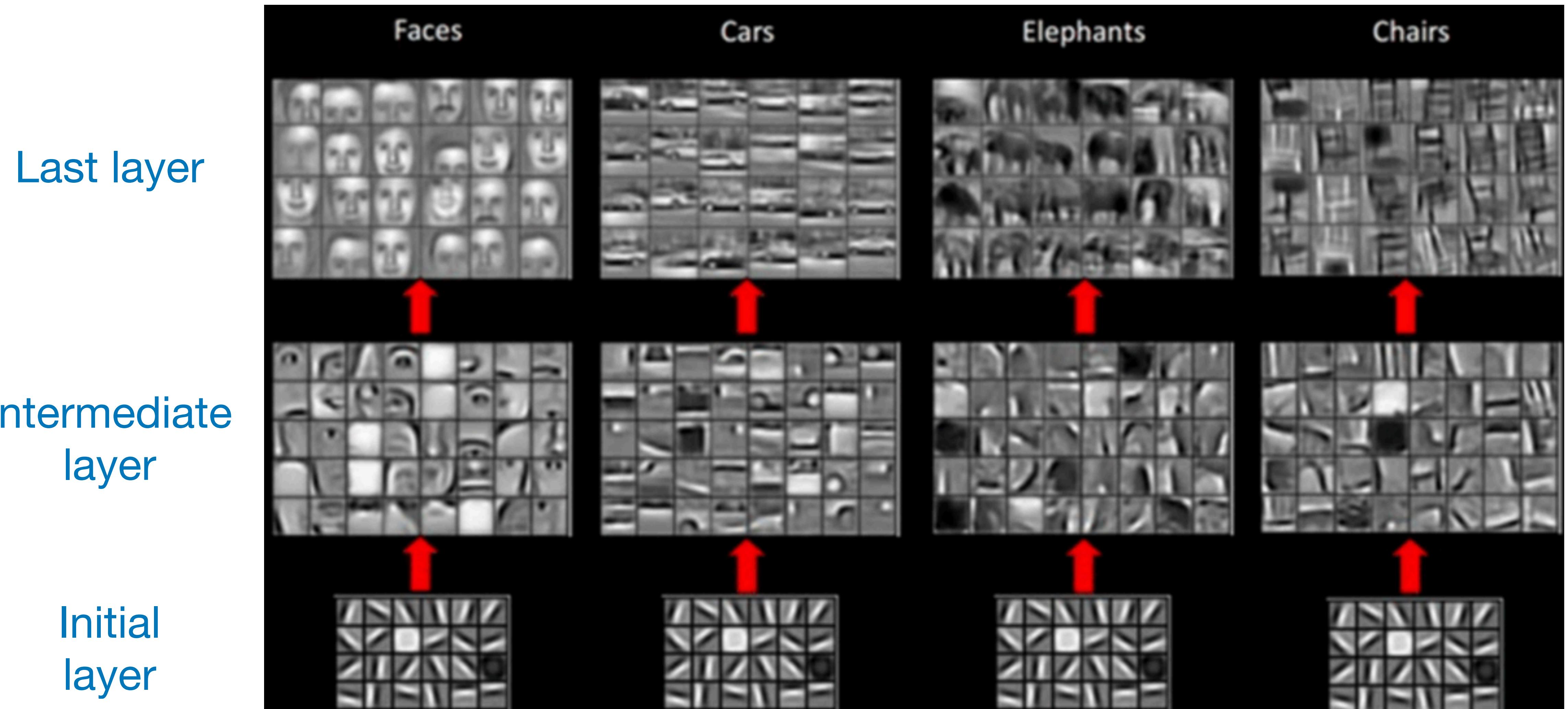
A CNN stacks together several alternating convolution and pooling layers, followed by a fully connected layer and a softmax output.

Filters, weights in fully connected layer, and biases learned by optimizing cross-entropy loss via stochastic gradient descent.

filters are very similar to weights

Interpreting the filters learned by a CNN

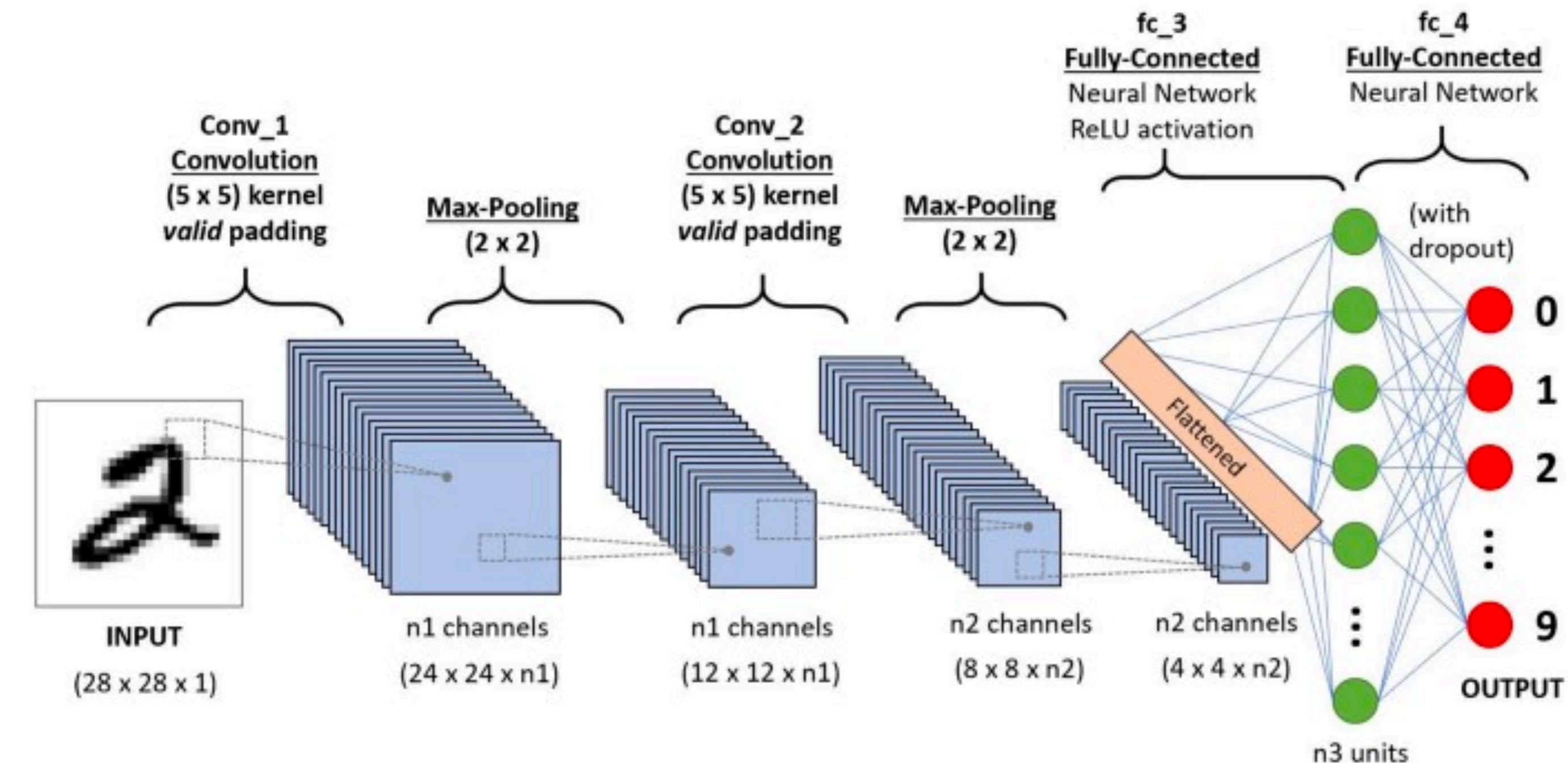
as you get deeper into neural network get into more complicated things



The original CNN architecture

LeNet architecture for hand-written digit recognition (1989).

Idea existed decades ago but data and computing power only became available in 2010s.



<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

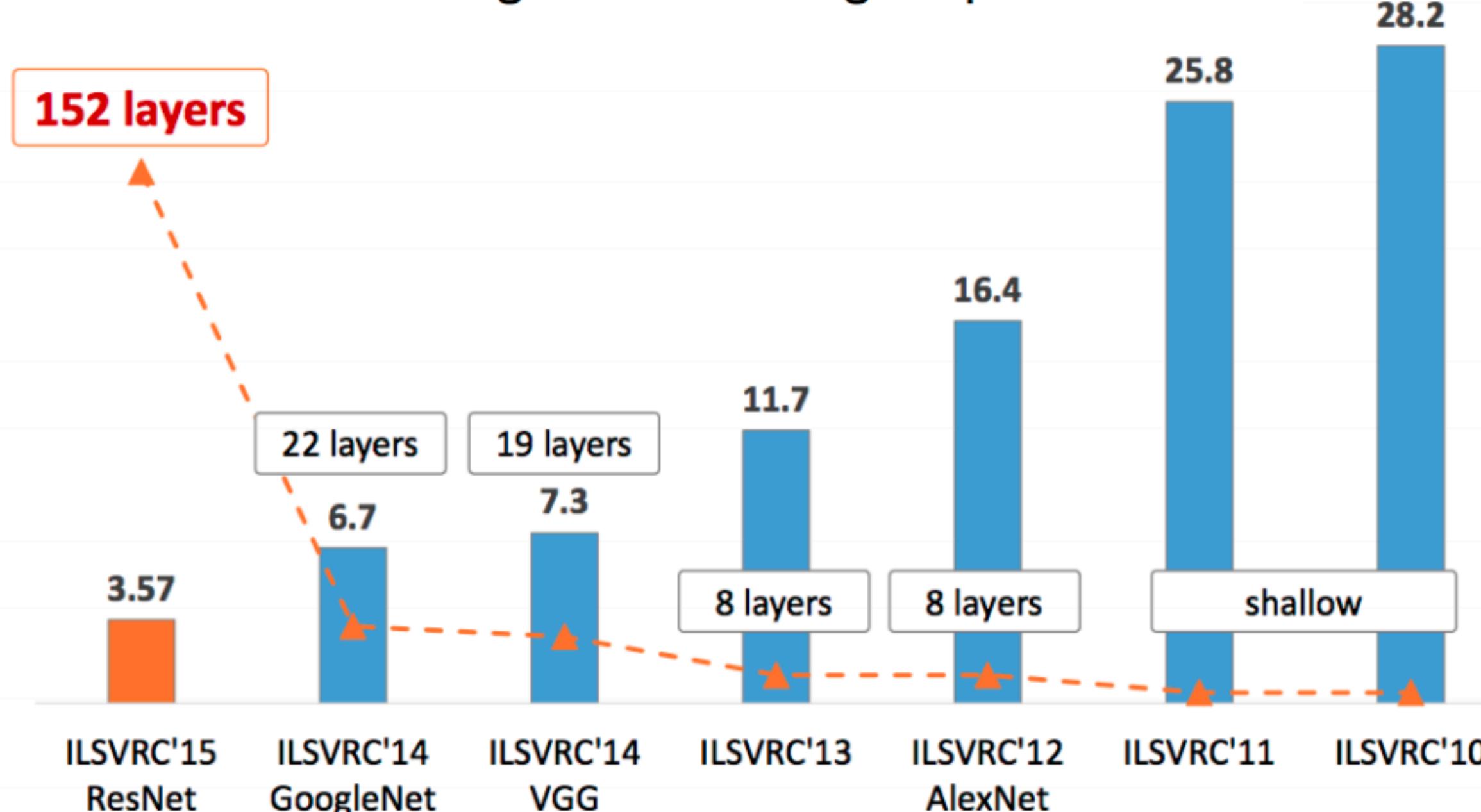


Yann LeCun

Modern CNN architectures

by adding depth, computation goes through the roof but performance also increases

Classification: ImageNet Challenge top-5 error



Model	Size (M)	Top-1/top-5 error (%)	# layers	Model description
AlexNet	238	41.00/18.00	8	5 conv + 3 fc layers
VGG-16	540	28.07/9.33	16	13 conv + 3 fc layers
VGG-19	560	27.30/9.00	19	16 conv + 3 fc layers
GoogleNet	40	29.81/10.04	22	21 conv + 1 fc layers
ResNet-50	100	22.85/6.71	50	49 conv + 1 fc layers
ResNet-152	235	21.43/3.57	152	151 conv + 1 fc layers

https://www.researchgate.net/figure/The-comparison-of-different-CNN-architectures-on-model-size-classification-error-rate_tbl1_320199404

<https://medium.com/@RaghavPrabhu/cnn-architectures-lenet-alexnet-vgg-googlenet-and-resnet-7c81c017b848>

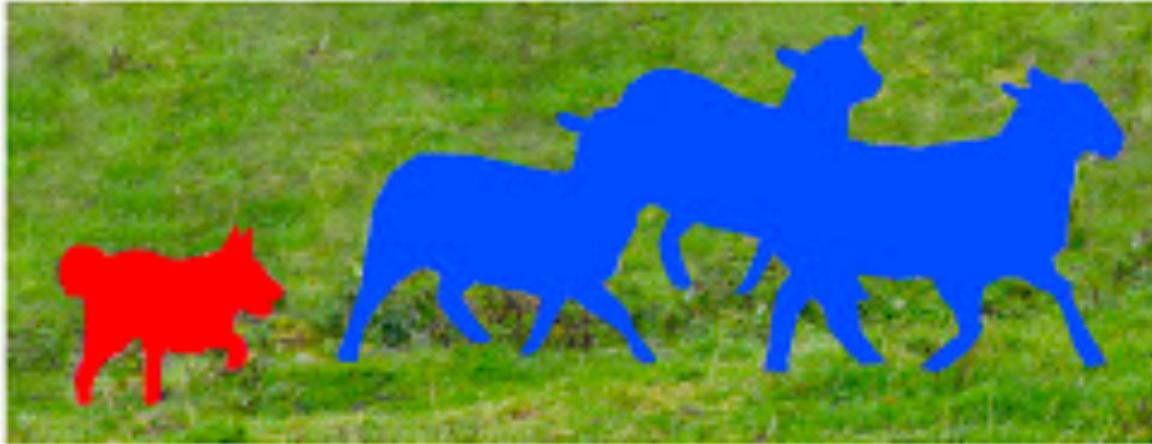
CNNs are getting progressively deeper with time.

Other applications of CNNs

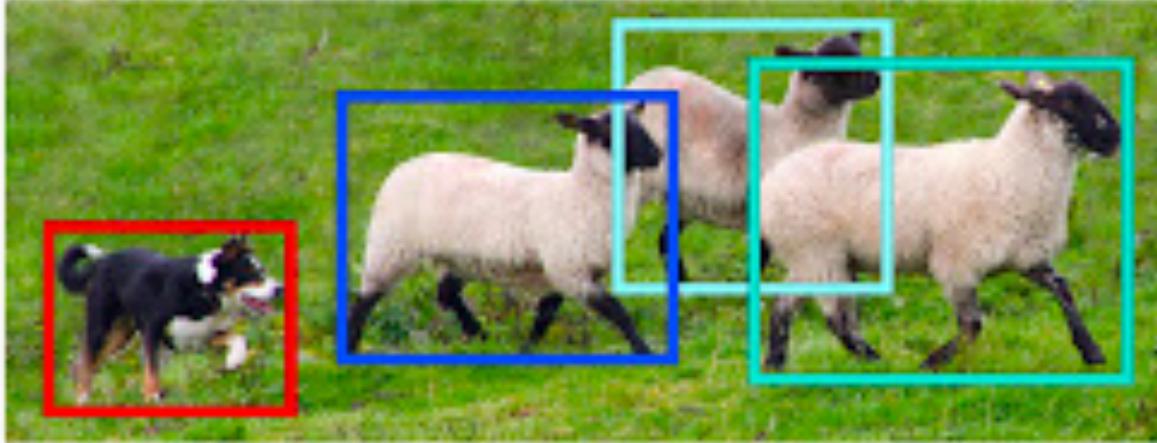
Other applications of CNNs



Image Recognition



Semantic Segmentation



Object Detection



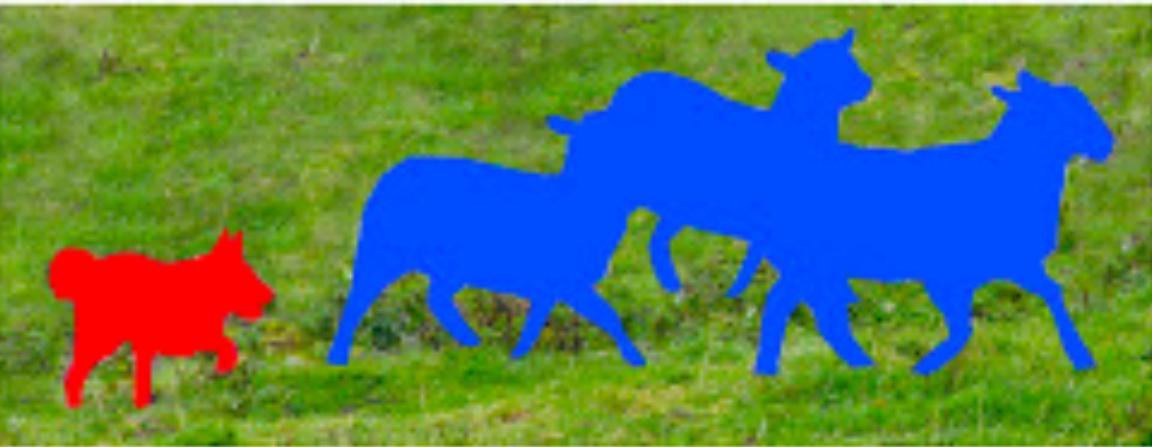
Instance Segmentation

<http://manipulation.csail.mit.edu/segmentation.html>

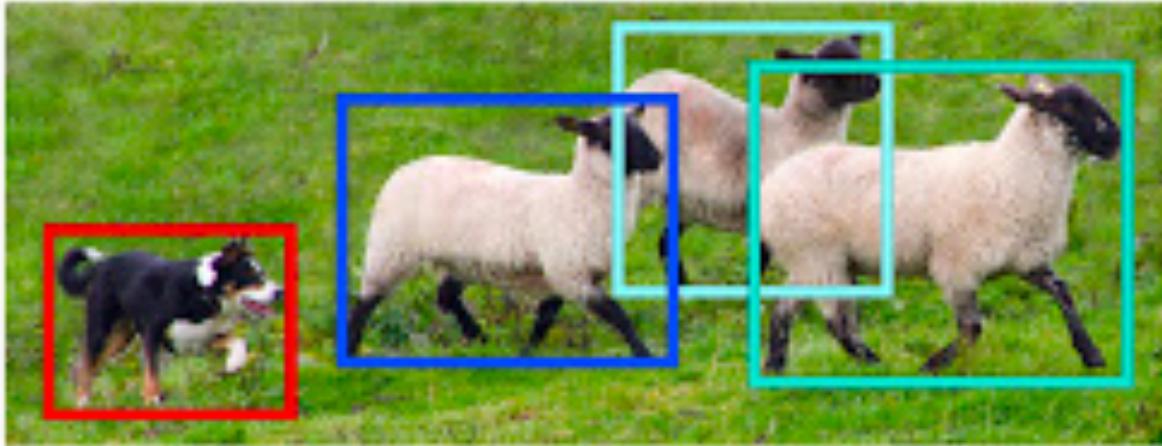
Other applications of CNNs



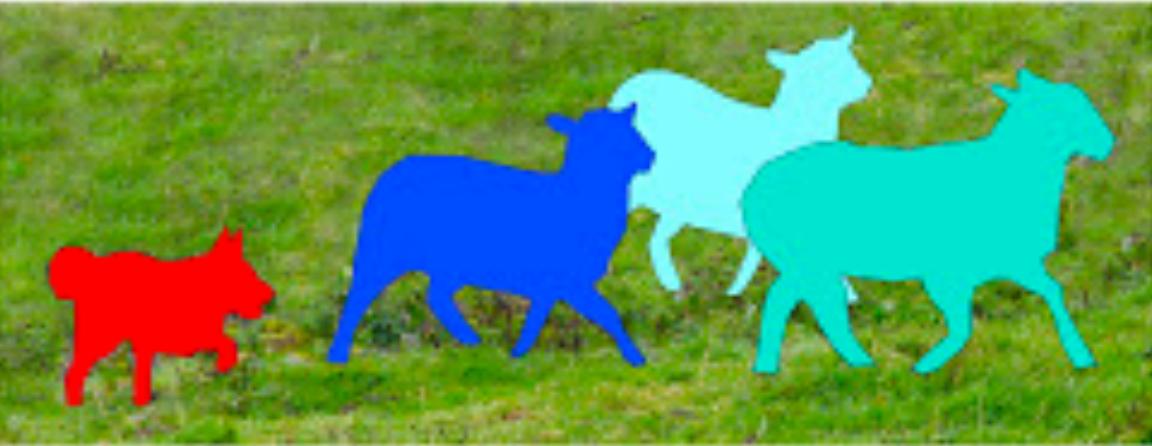
Image Recognition



Semantic Segmentation



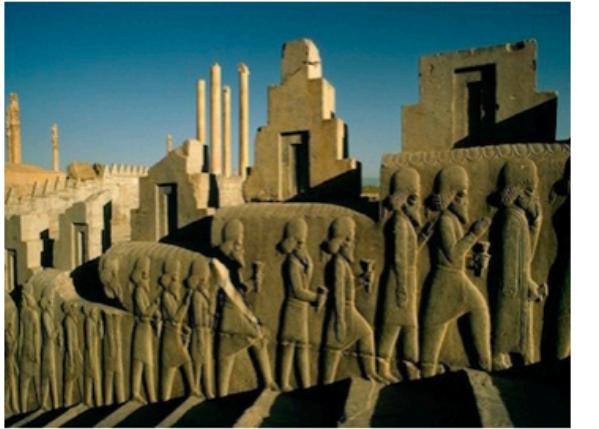
Object Detection



Instance Segmentation

<http://manipulation.csail.mit.edu/segmentation.html>

content image



style image



generated image



Ancient city of Persepolis

The Starry Night (Van Gogh)

Persepolis
in Van Gogh style

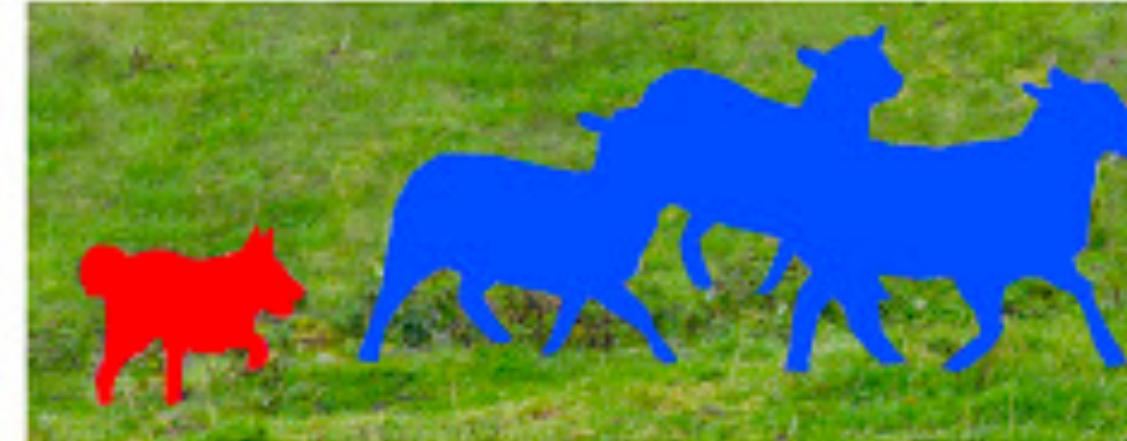
<https://towardsdatascience.com/neural-style-transfer-on-real-time-video-with-full-implementable-code-ac2dbc0e9822>

Style Transfer

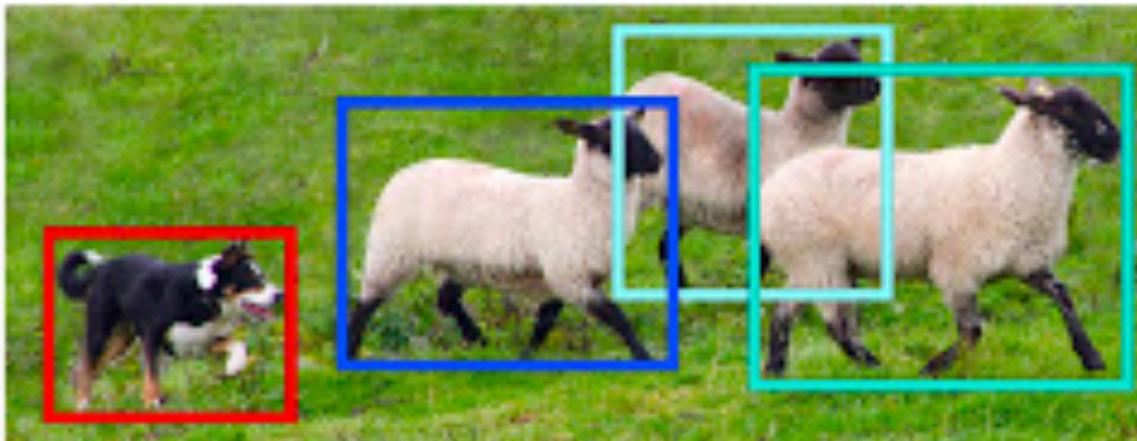
Other applications of CNNs



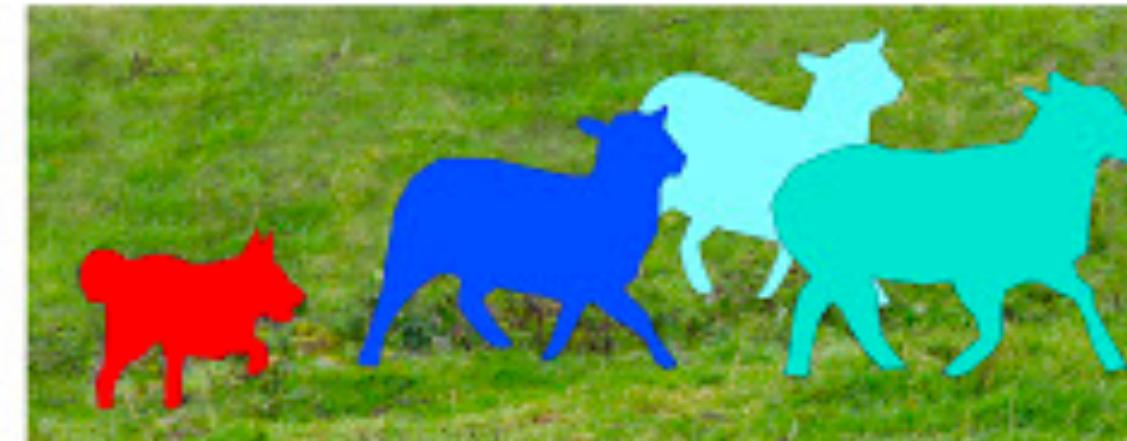
Image Recognition



Semantic Segmentation



Object Detection



Instance Segmentation

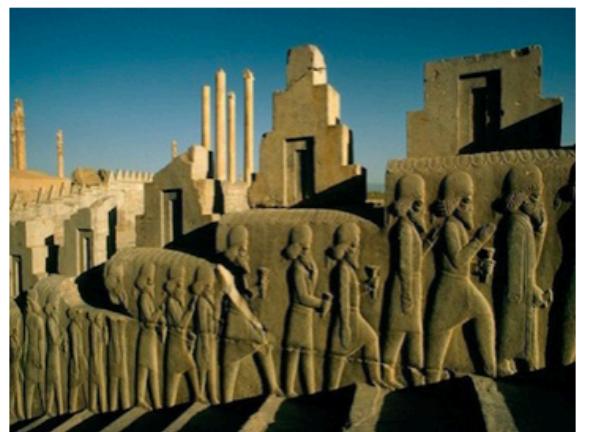


<https://medium.com/@nabilibin14/chain-colorization-using-cdcgans-and-cnn-from-learned-deep-prior-1405dab48df3>

Image colorization

<http://manipulation.csail.mit.edu/segmentation.html>

content image



style image



generated image



Ancient city of Persepolis

The Starry Night (Van Gogh)

Persepolis
in Van Gogh style

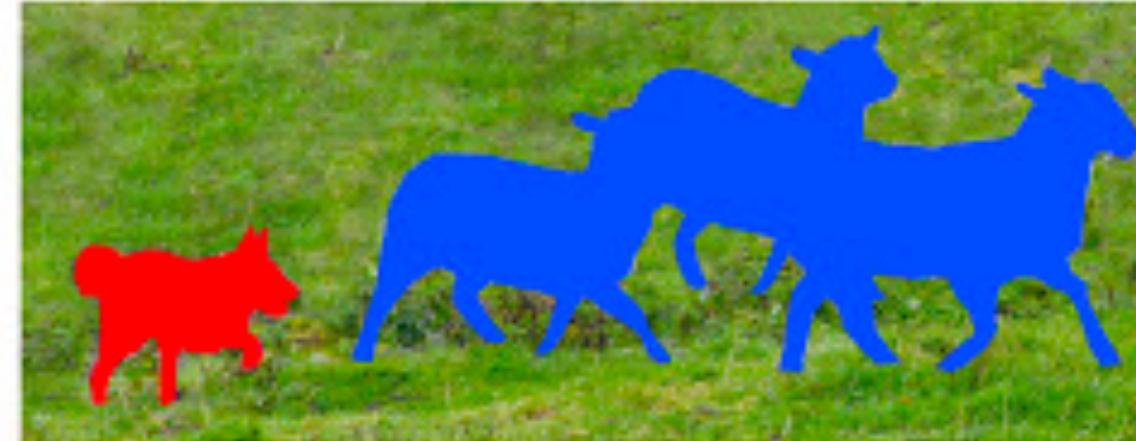
<https://towardsdatascience.com/neural-style-transfer-on-real-time-video-with-full-implementable-code-ac2dbc0e9822>

Style Transfer

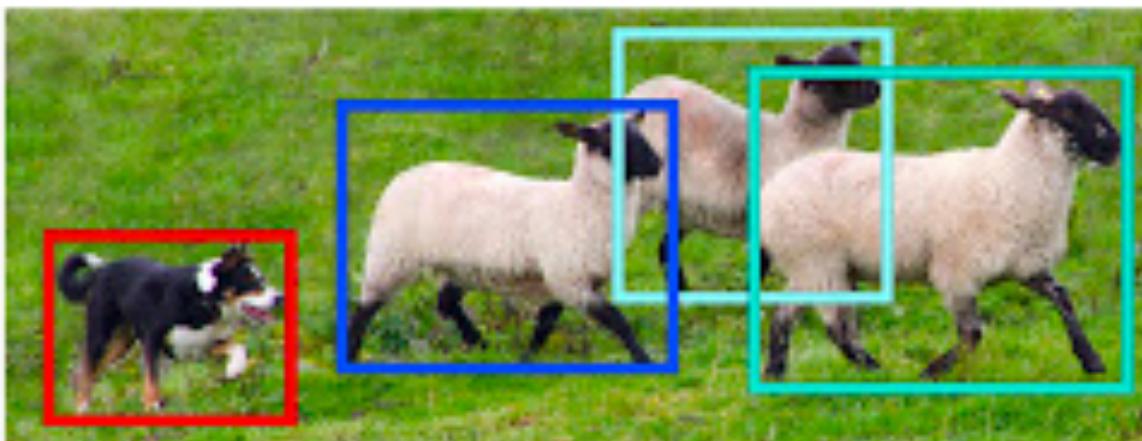
Other applications of CNNs



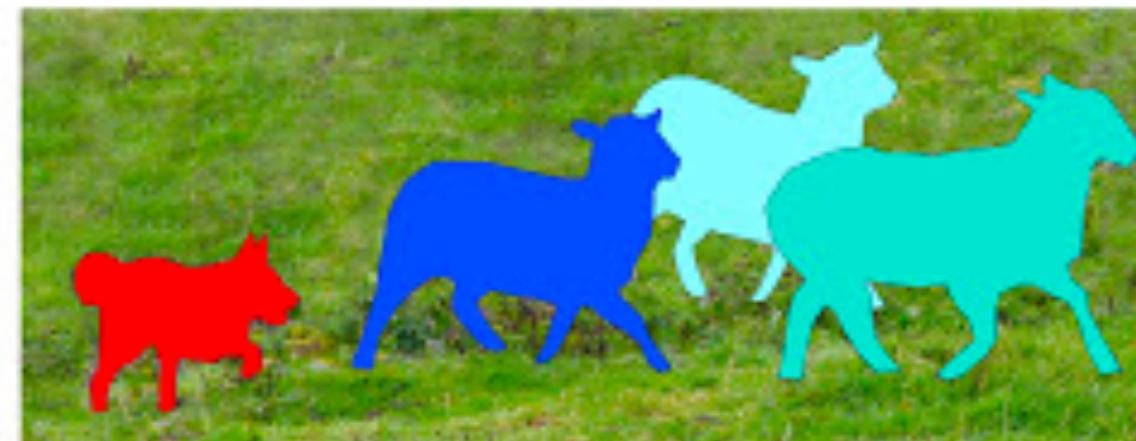
Image Recognition



Semantic Segmentation

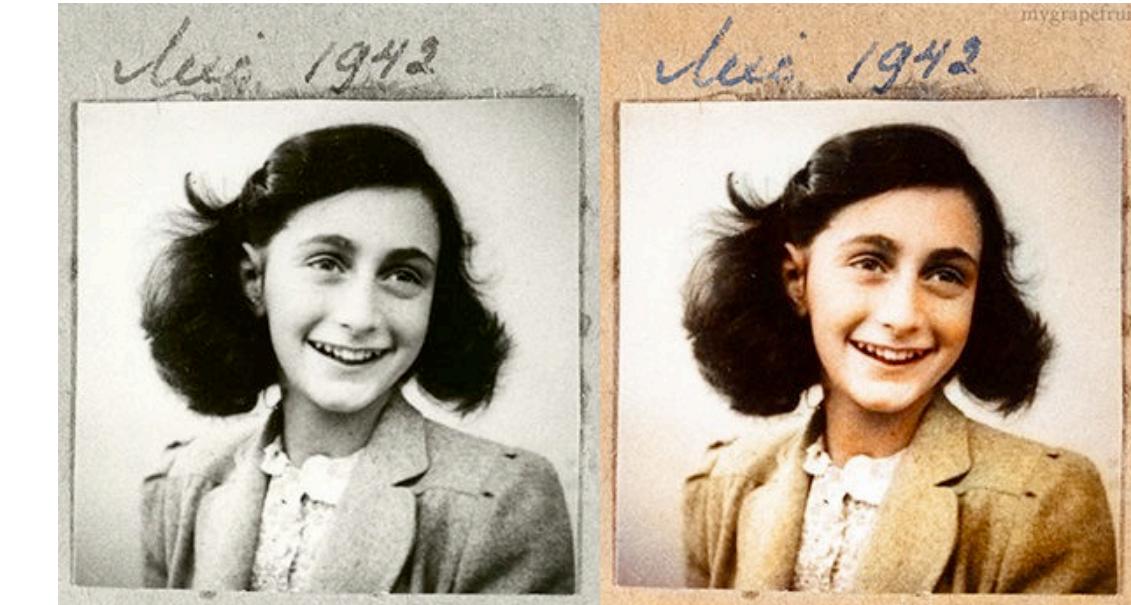


Object Detection



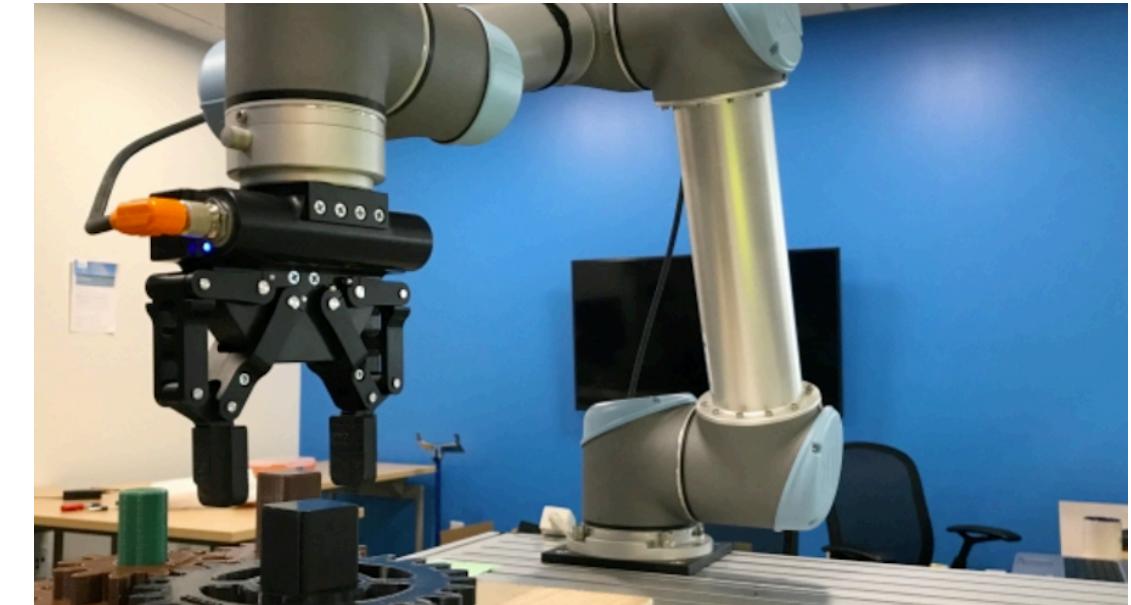
Instance Segmentation

<http://manipulation.csail.mit.edu/segmentation.html>



<https://medium.com/@nabilibin14/chain-colorization-using-cdcgans-and-cnn-from-learned-deep-prior-1405dab48df3>

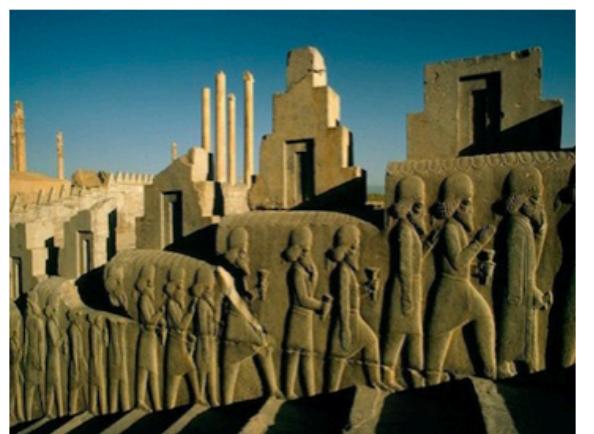
Image colorization



<https://www.therobotreport.com/reinforcement-learning-industrial-robotics/>

Reinforcement learning

content image



style image



generated image



Ancient city of Persepolis

The Starry Night (Van Gogh)

Persepolis
in Van Gogh style

<https://towardsdatascience.com/neural-style-transfer-on-real-time-video-with-full-implementable-code-ac2dbc0e9822>

Style Transfer

Other applications of CNNs

training: existing images



Image Recognition

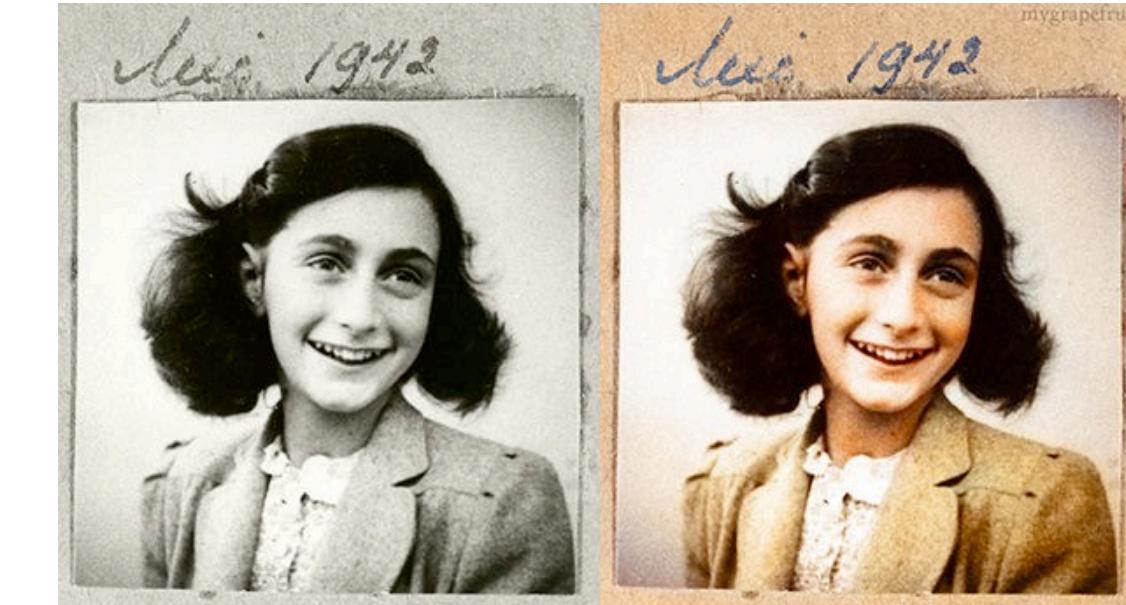
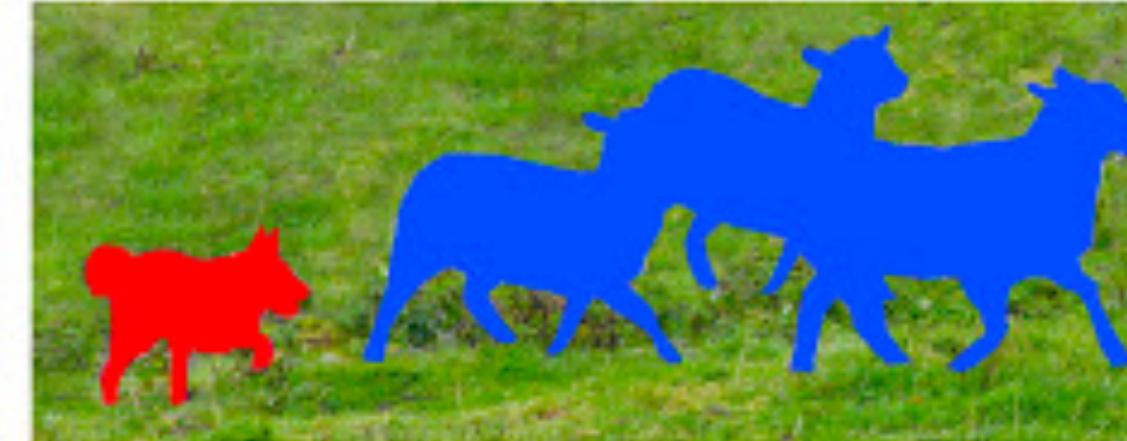
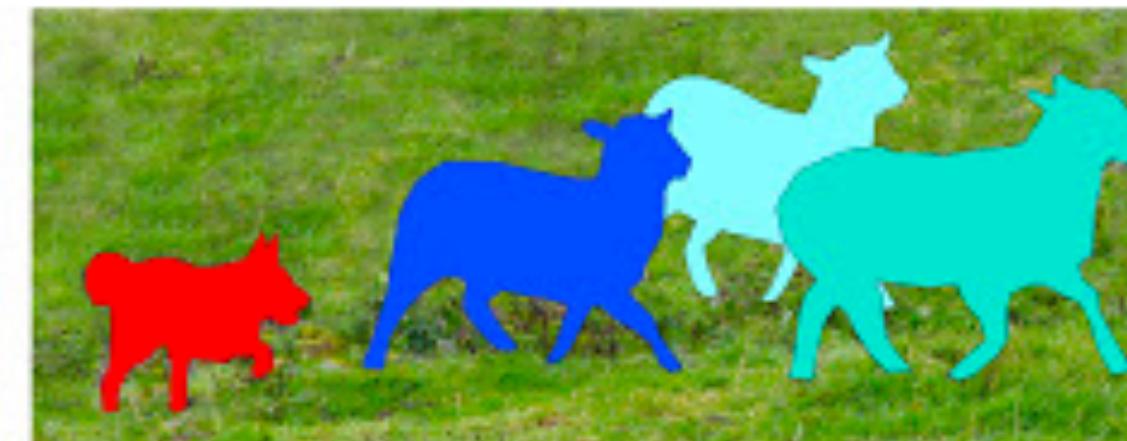
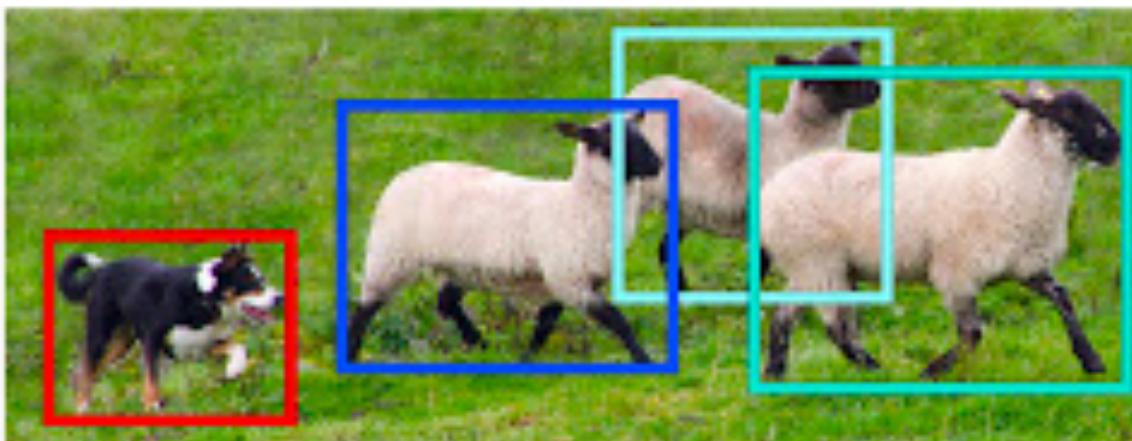
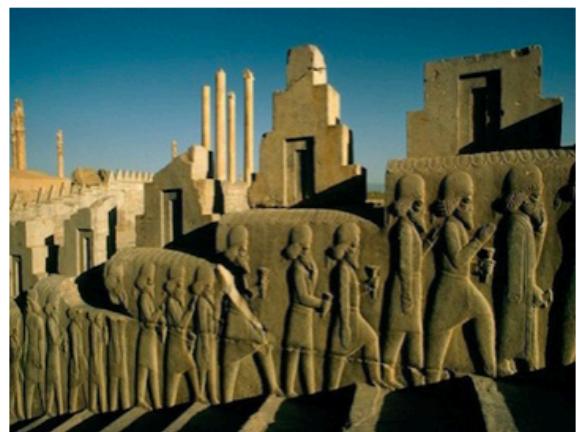


Image colorization

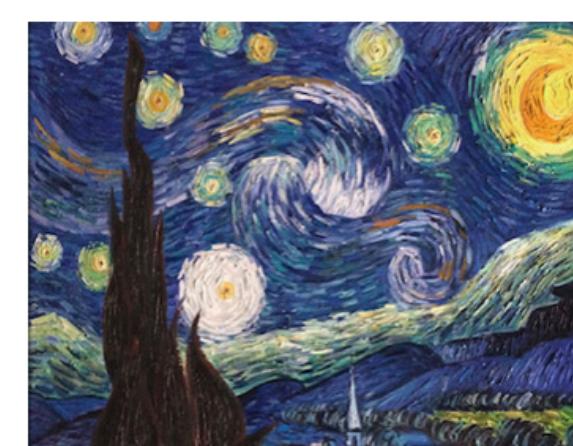


<http://manipulation.csail.mit.edu/segmentation.html>

content image



+



=

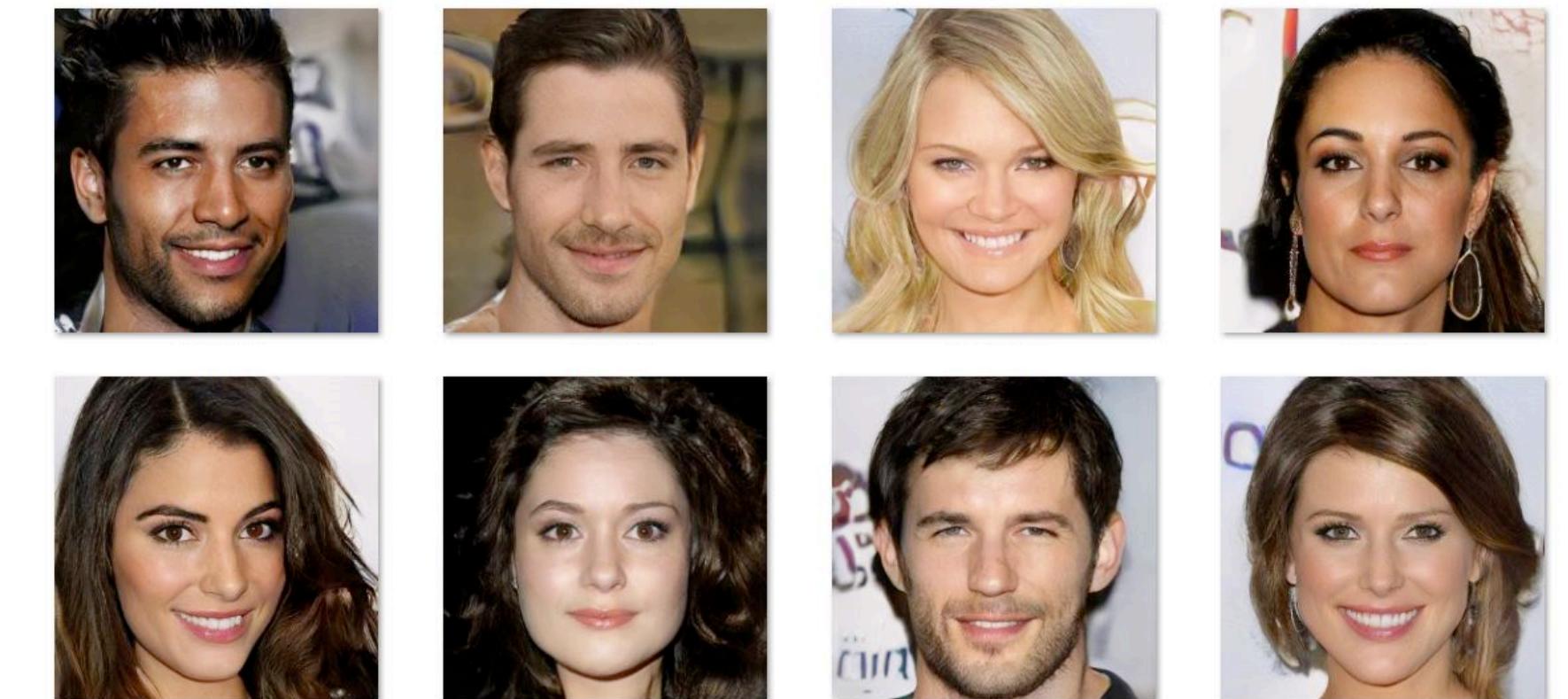


<https://towardsdatascience.com/neural-style-transfer-on-real-time-video-with-full-implementable-code-ac2dbc0e9822>

Style Transfer



Reinforcement learning



Generative models