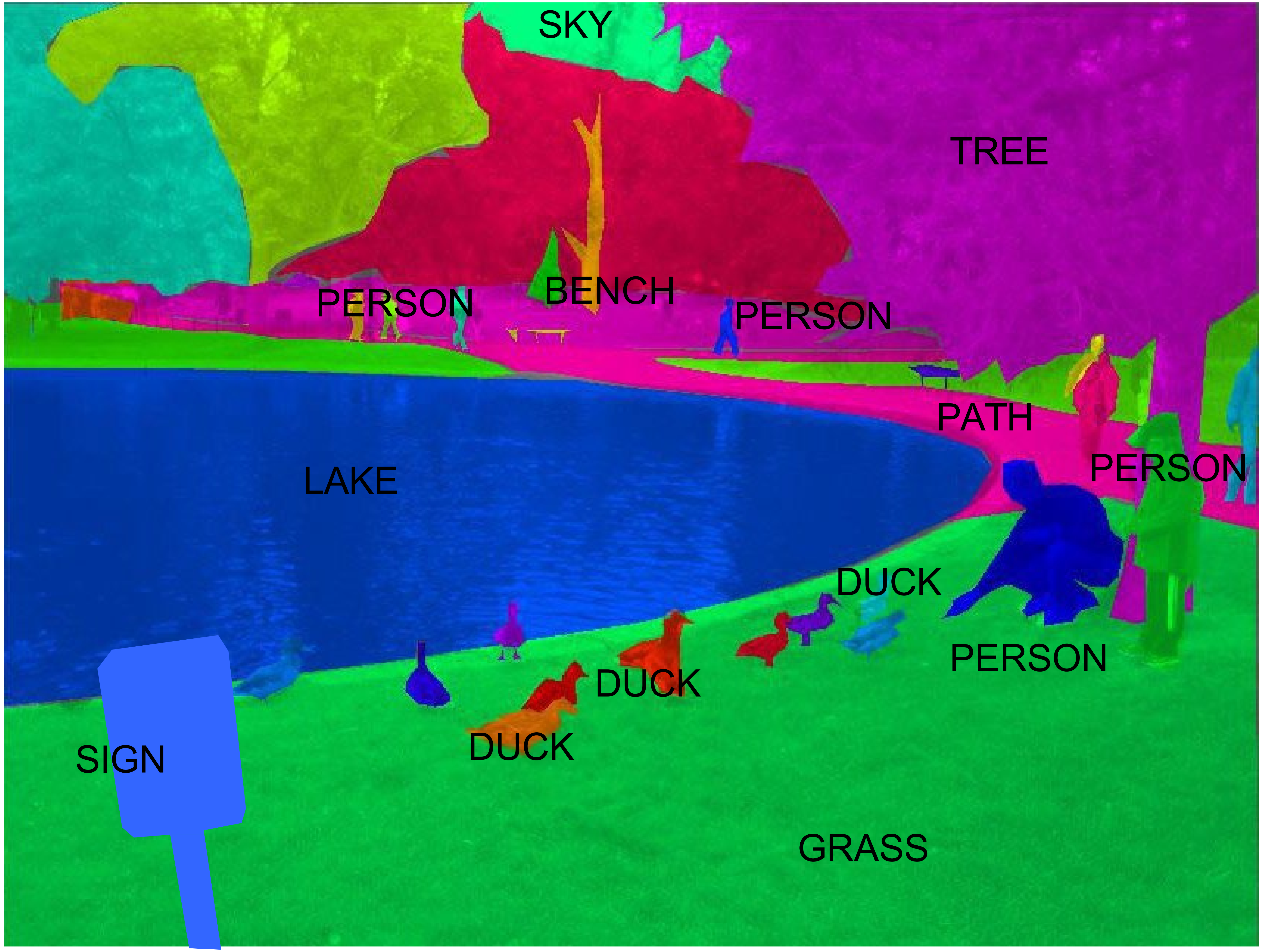
Visual Recognition with Deep Neural Networks

CMSC 491/691 Robust Machine Learning

UMBC

Announcements

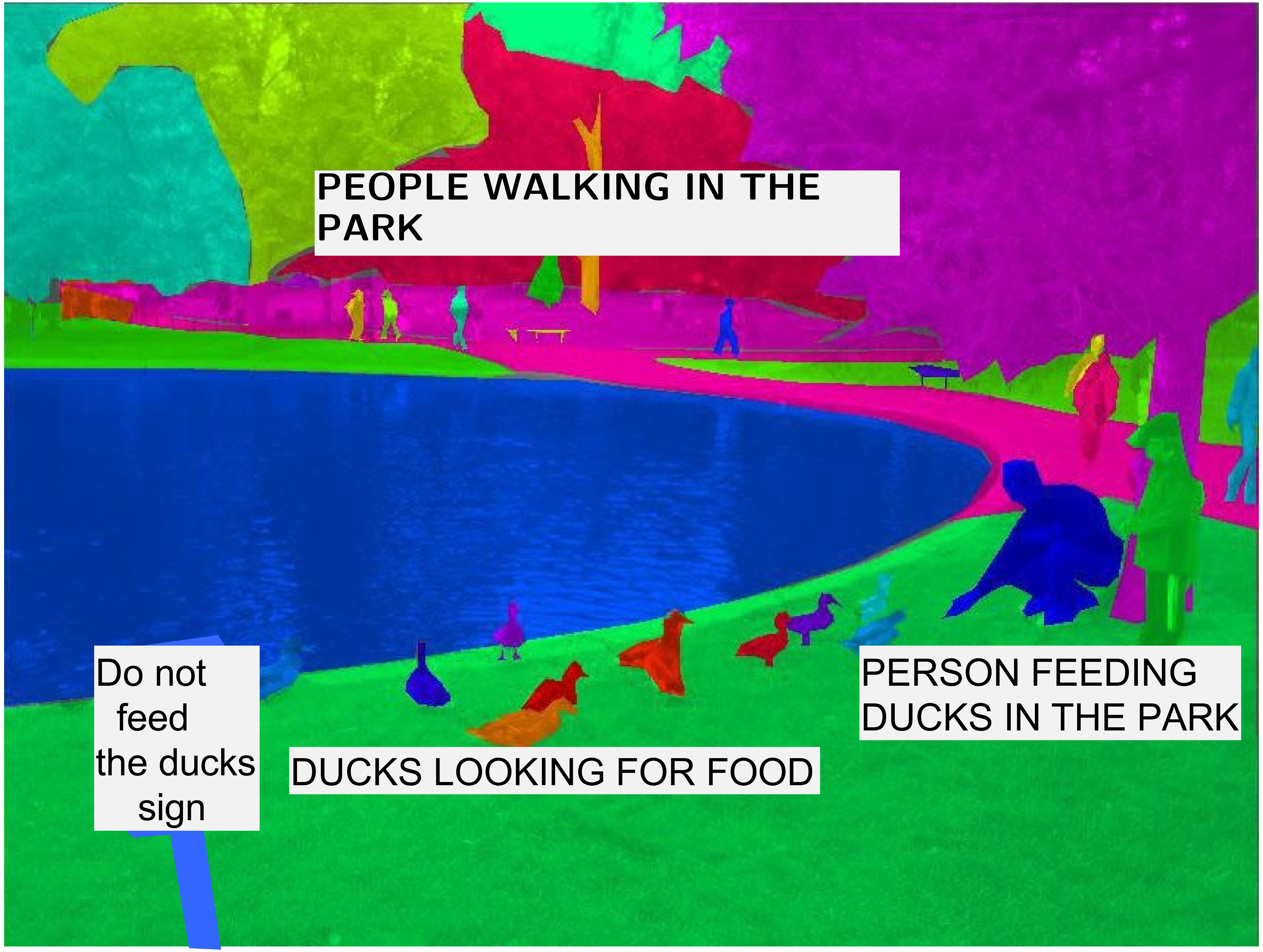
- **Team formation is due by next Monday**
  - Announcement on Blackboard has sign-up link (same link as Presentation/Survey)
- **Project Proposals will be due around Sept 23 (tentative)**
  - Announcement coming soon. 2-page proposal
  - Rough outline:
    - *Title of the project*
    - *Names and emails of group members*
    - *Problem you wish to tackle (and why)*
    - *Proposed approach and methods*
    - *Novelty (how is your approach better or different than previous work)*
    - *Timeline*
    - *Plan for Individual Contributions (What each student in the group will do)*
    - *Expected Outcome and Worst-Case Outcome: tell us the what you ideally want to achieve through this project*



Label each pixel as a category. Each category has a unique color.

Scene-Level Classification: This is a “PARK”

Image Captioning: Describe the image in human language (e.g. English)



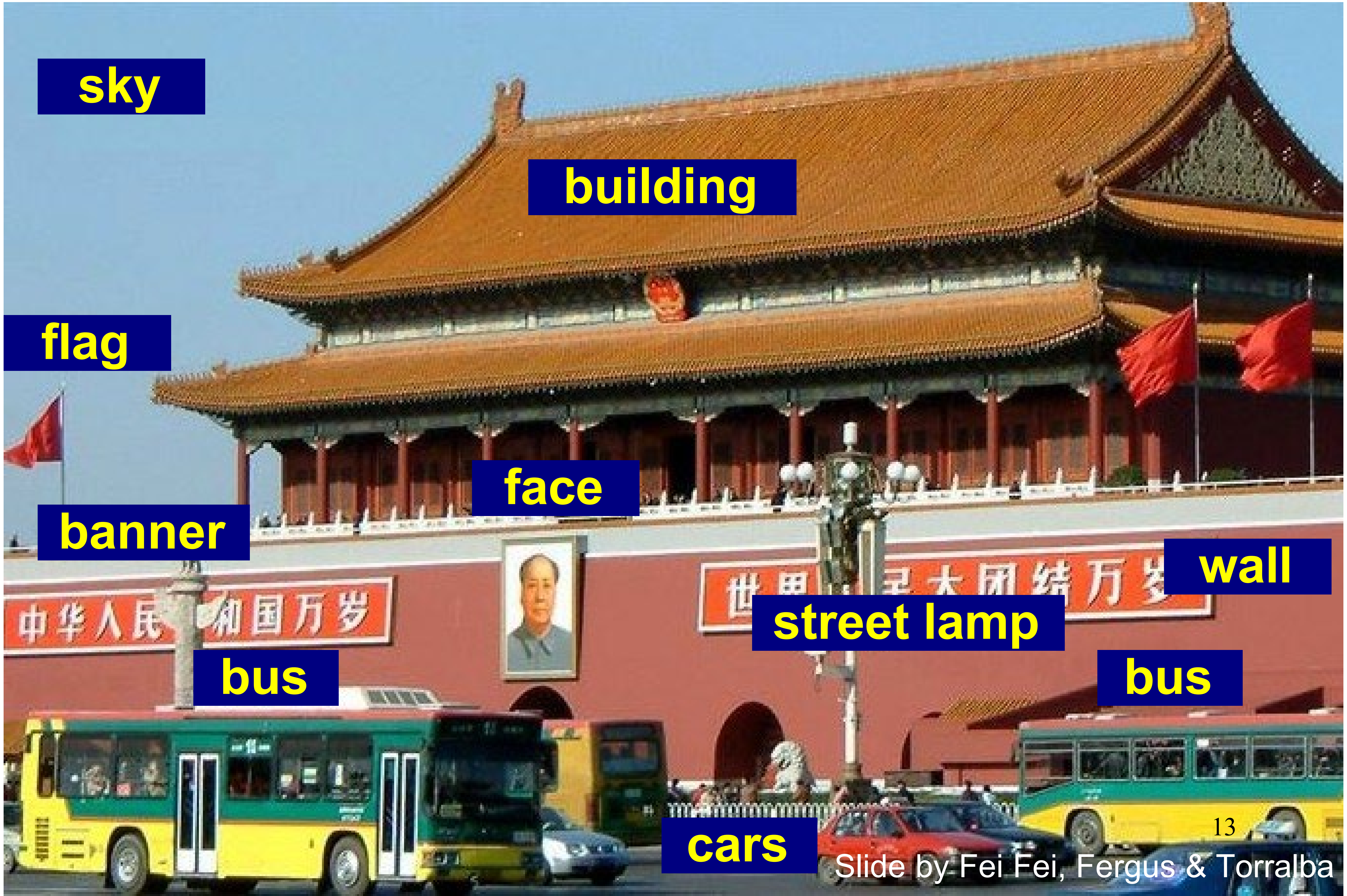
Dense Image Captioning: Describe several parts of the image

What makes this challenging?

Why do we care about recognition?

- The concept of "**categories**" encapsulates semantic information that humans use when communicating with each other.
- Categories are also linked with what can we do with those objects.

Object categories aren’t everything



Object categories aren't everything

A picture is worth a 1000 words…

Or just 10?

sky

building

flag

face

banner

wall

street lamp

bus

bus

cars

Source: A. Efros

How finegrained should categories be ?

A Beijing City Transit Bus #17, serial number 43253?

Source: A. Efros

Need more general (useful) information

What can we say the very first time we see this thing?

Functional:

- A large vehicle that may be moving fast, probably to the right, and will hurt you if you stand in its way.
- However, at specified places, it will allow you to enter it and transport you quickly over large distances.

Communicational:

- bus, autobus, λεωφορείο, ônibus, автобус, 公共汽车, etc.

Source: A. Efros

Visual challenges with categories

- A lot of categories are functional
- Categories are 3D, but images are 2D
- World is highly varied

Chair

car

train

Source: A. Efros

Limits to direct perception

Importance of Context

Source: Antonio Torralba

We might think seeing is believing ...

Video by Antonio Torralba (starring Rob Fergus)

But is it ?

Video by Antonio Torralba (starring Rob Fergus)

Image classification

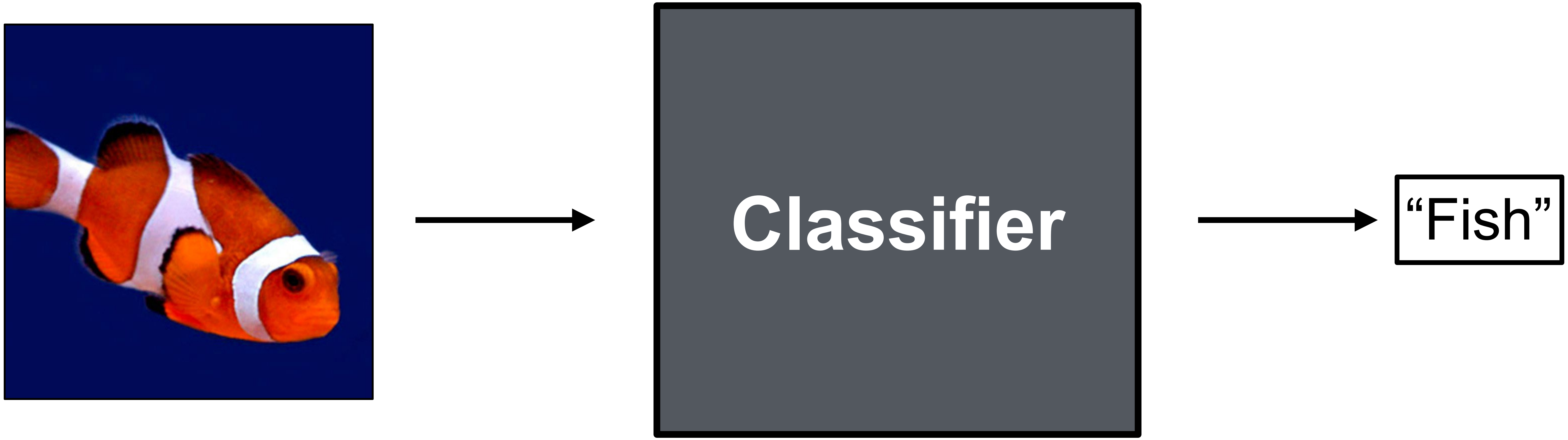


image **x**

label y

Source: Isola, Torralba, Freeman

Image classification

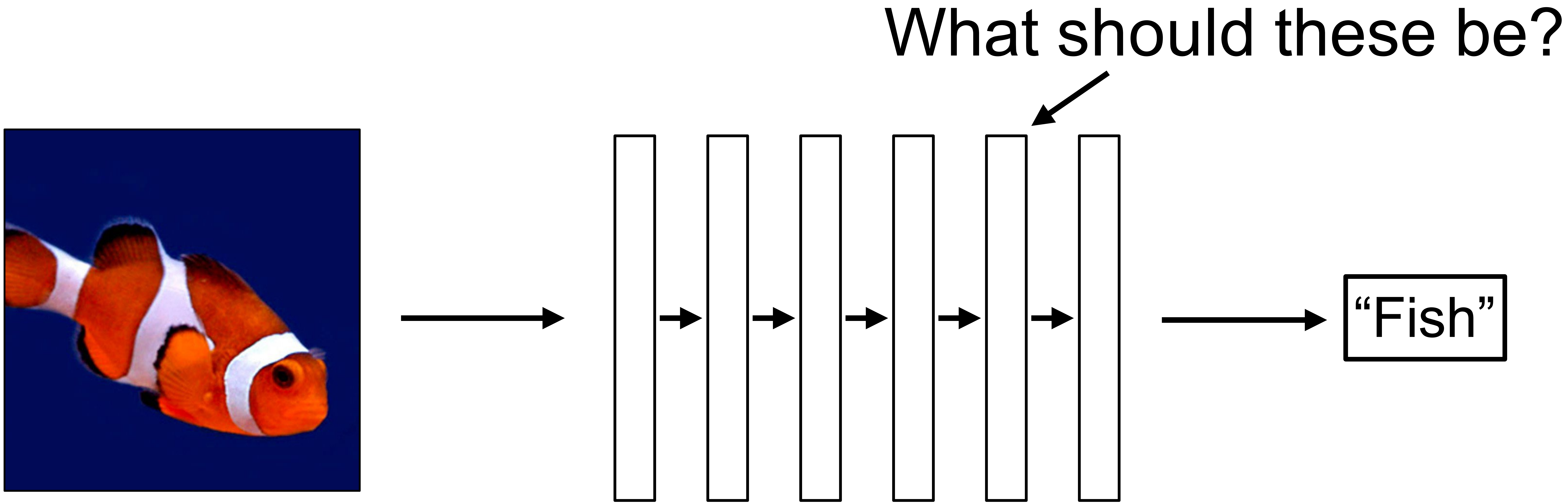
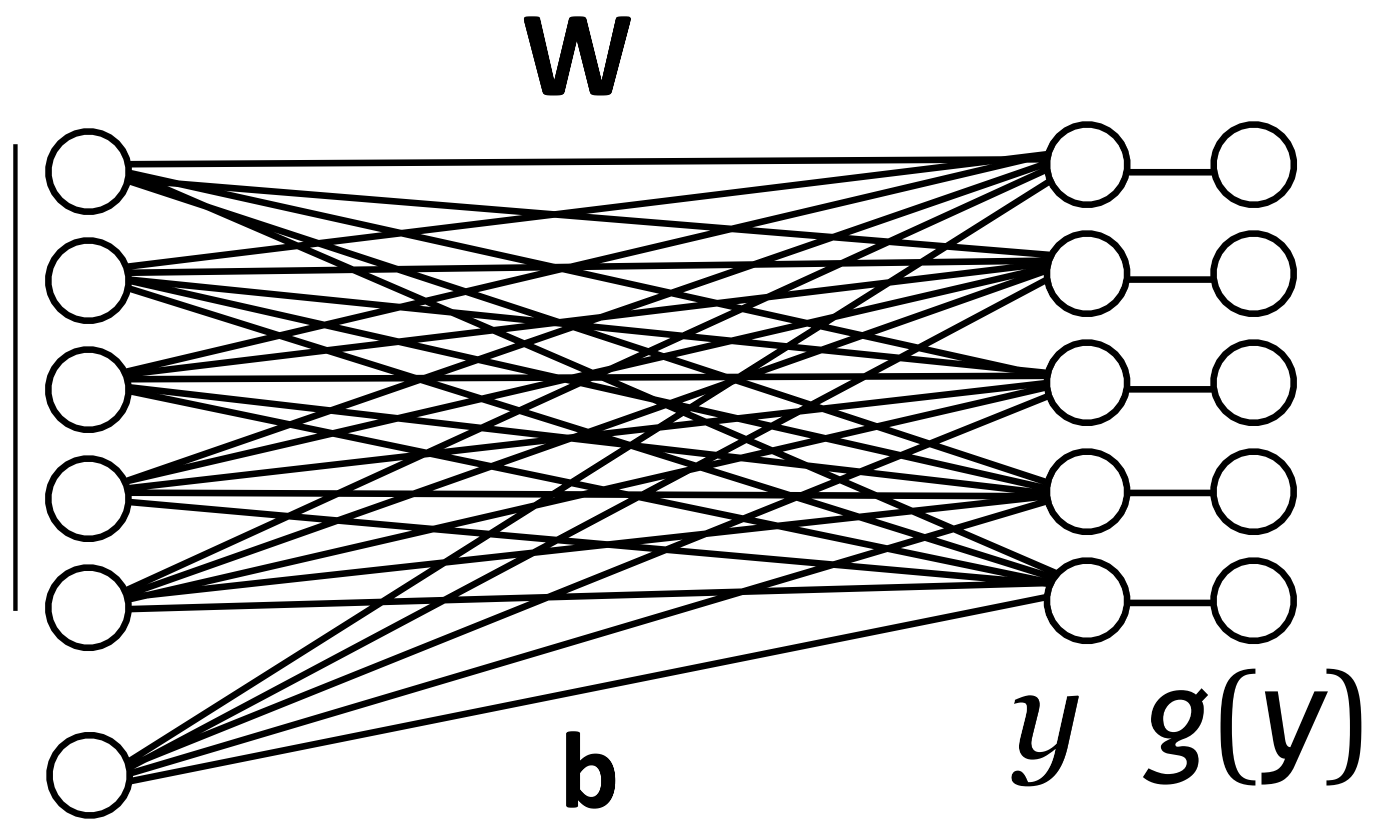


image **x** label y

Source: Isola, Torralba, Freeman

Idea #1: Fully-connected network

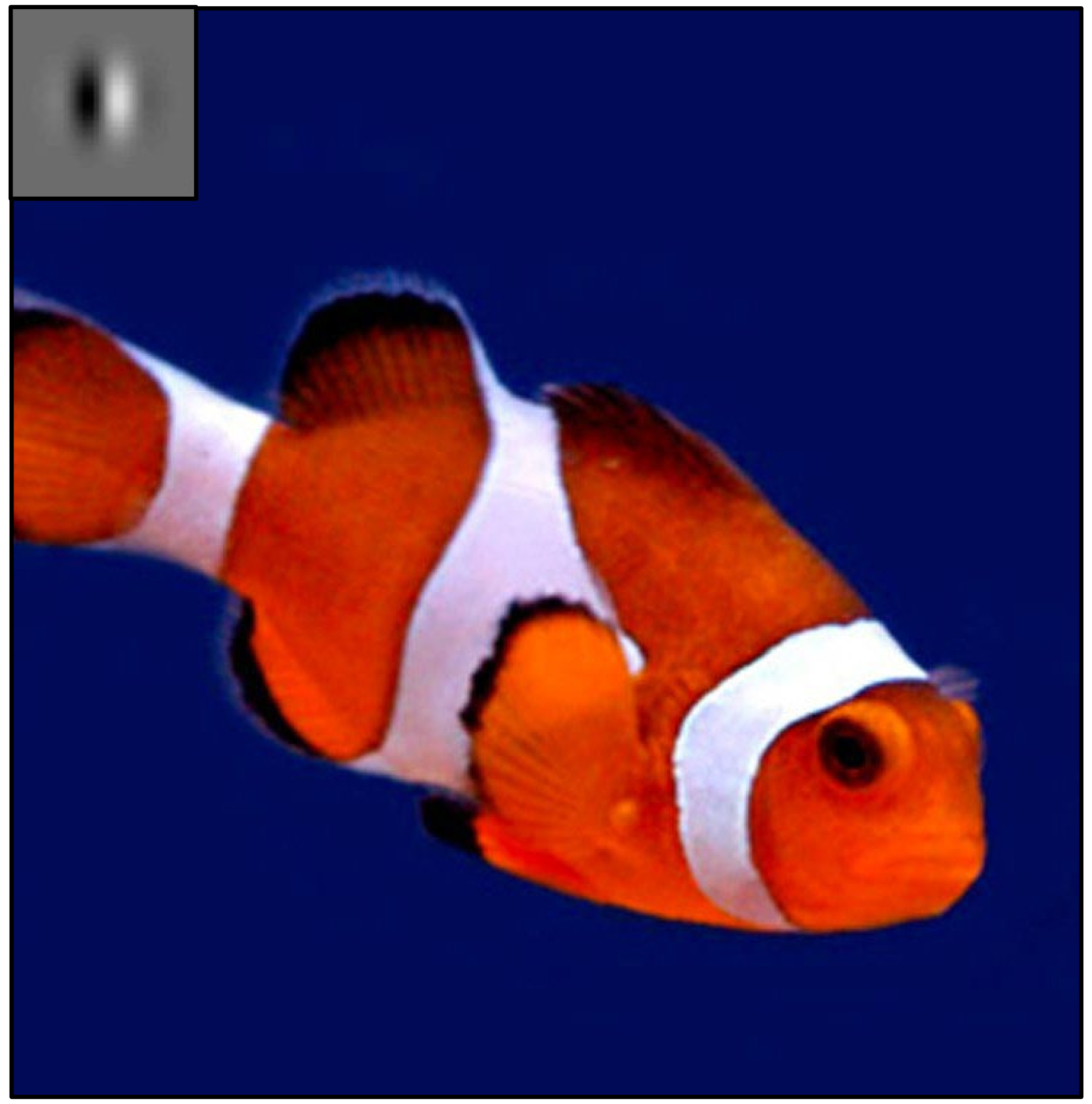
Fully-connected (linear) layer

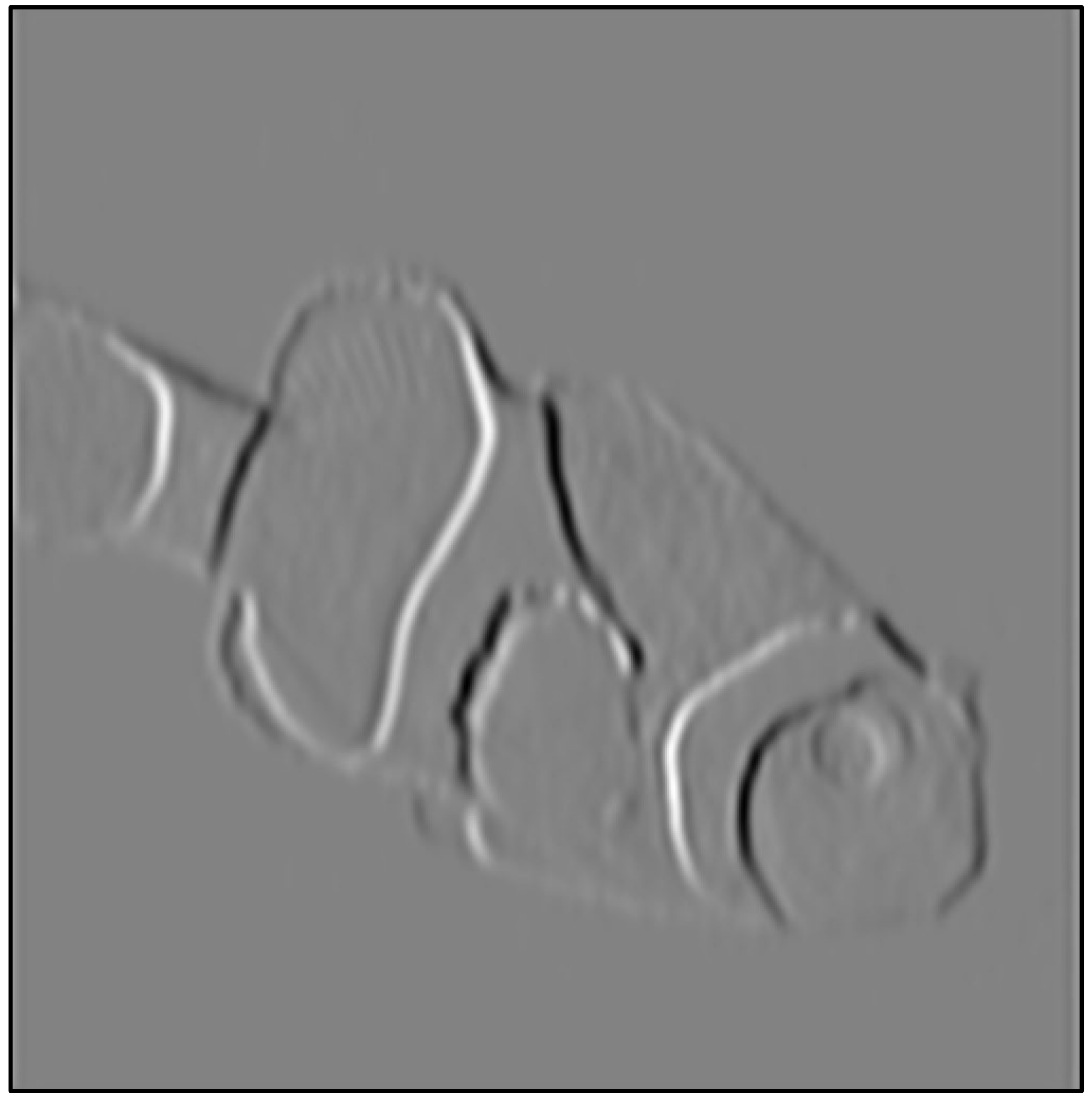


= 

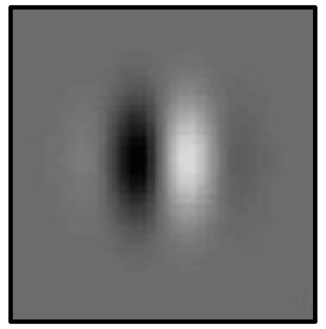
- is really big! E.g. 256 x 256

Can we use convolution in a neural network?



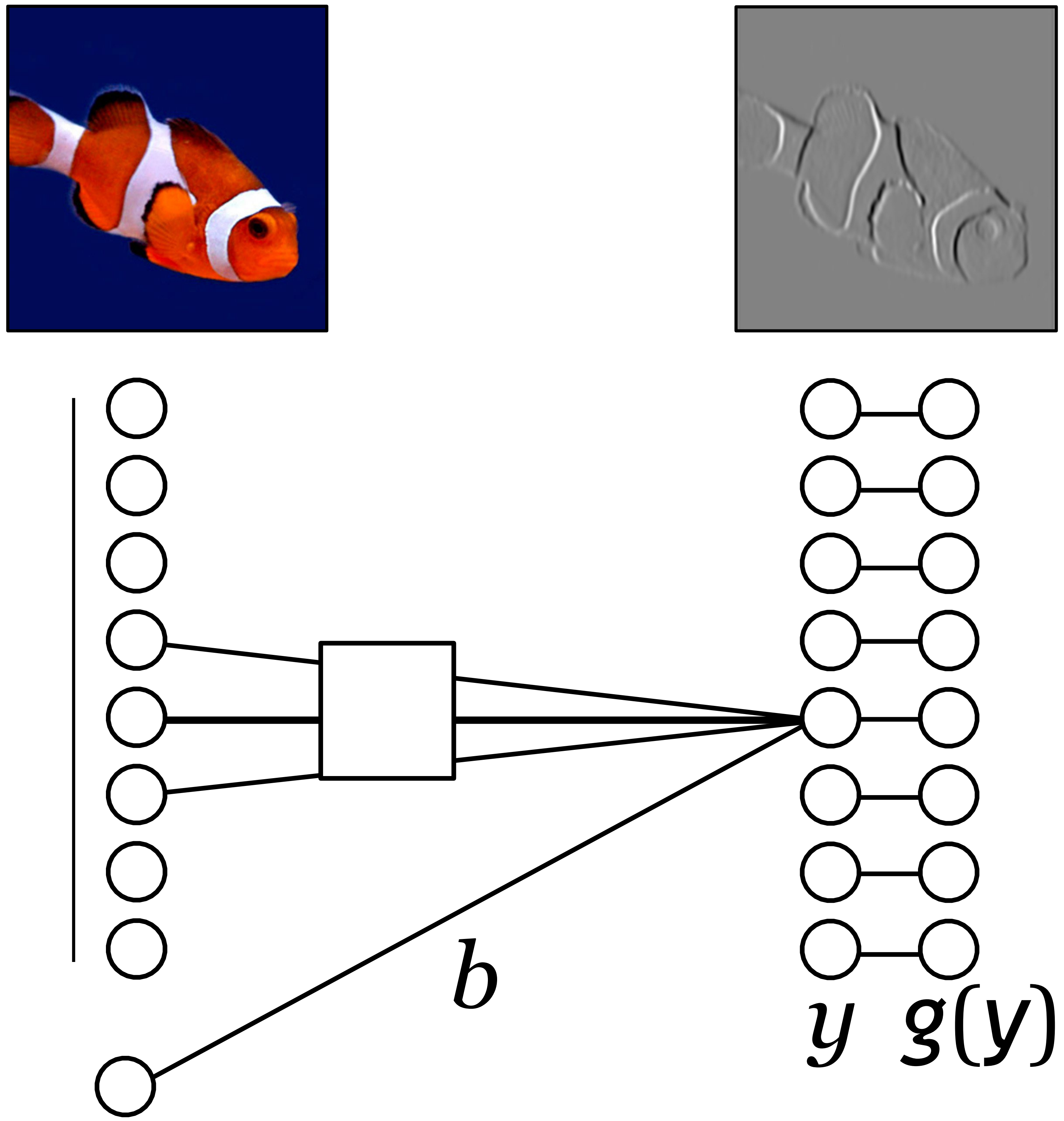


filter



Source: Isola, Torralba, Freeman

Recall: Sparsely connected network

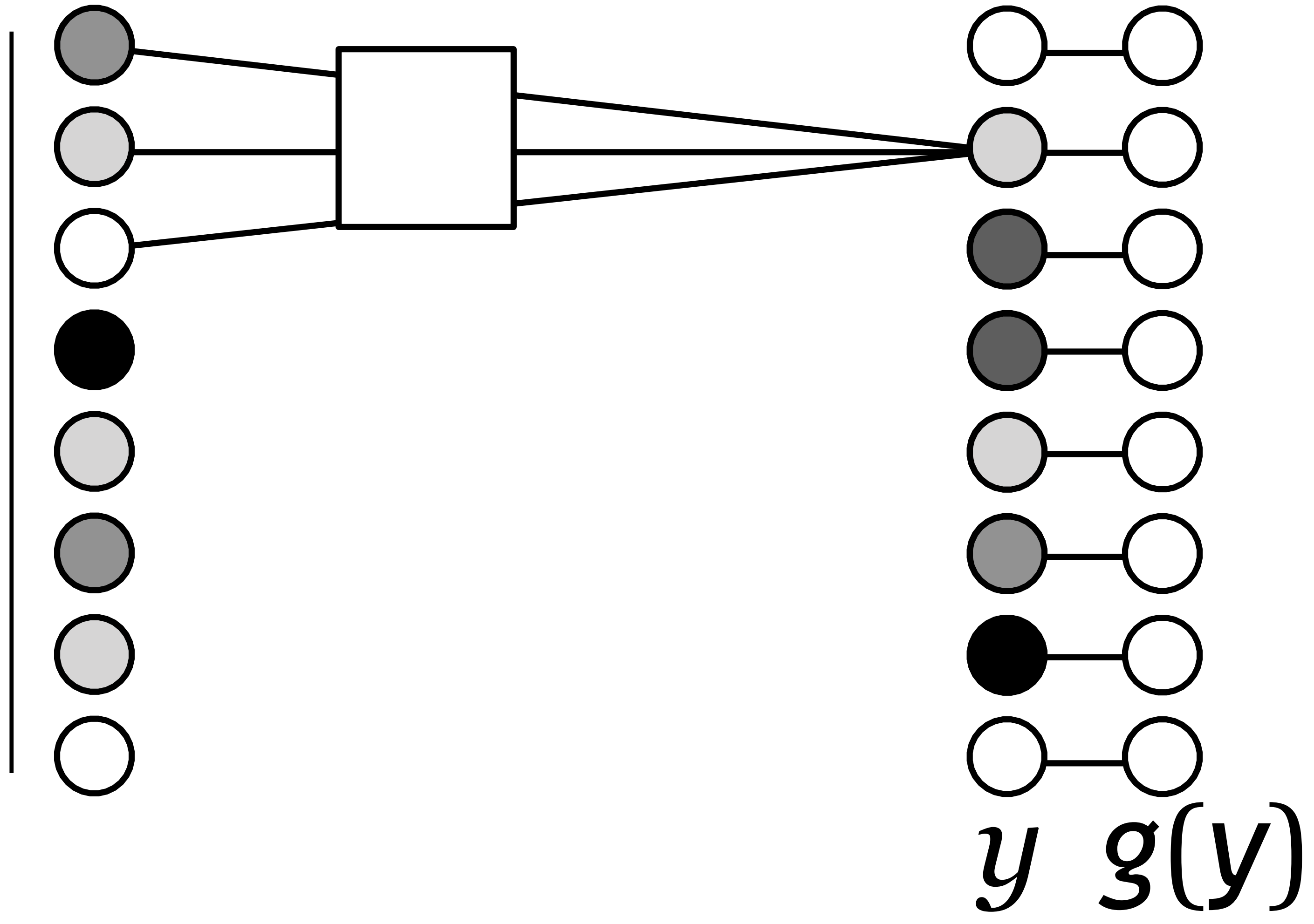


Each unit is connected to a subset of the units in the previous layer.

Source: Isola, Torralba, Freeman

Convolutional neural network

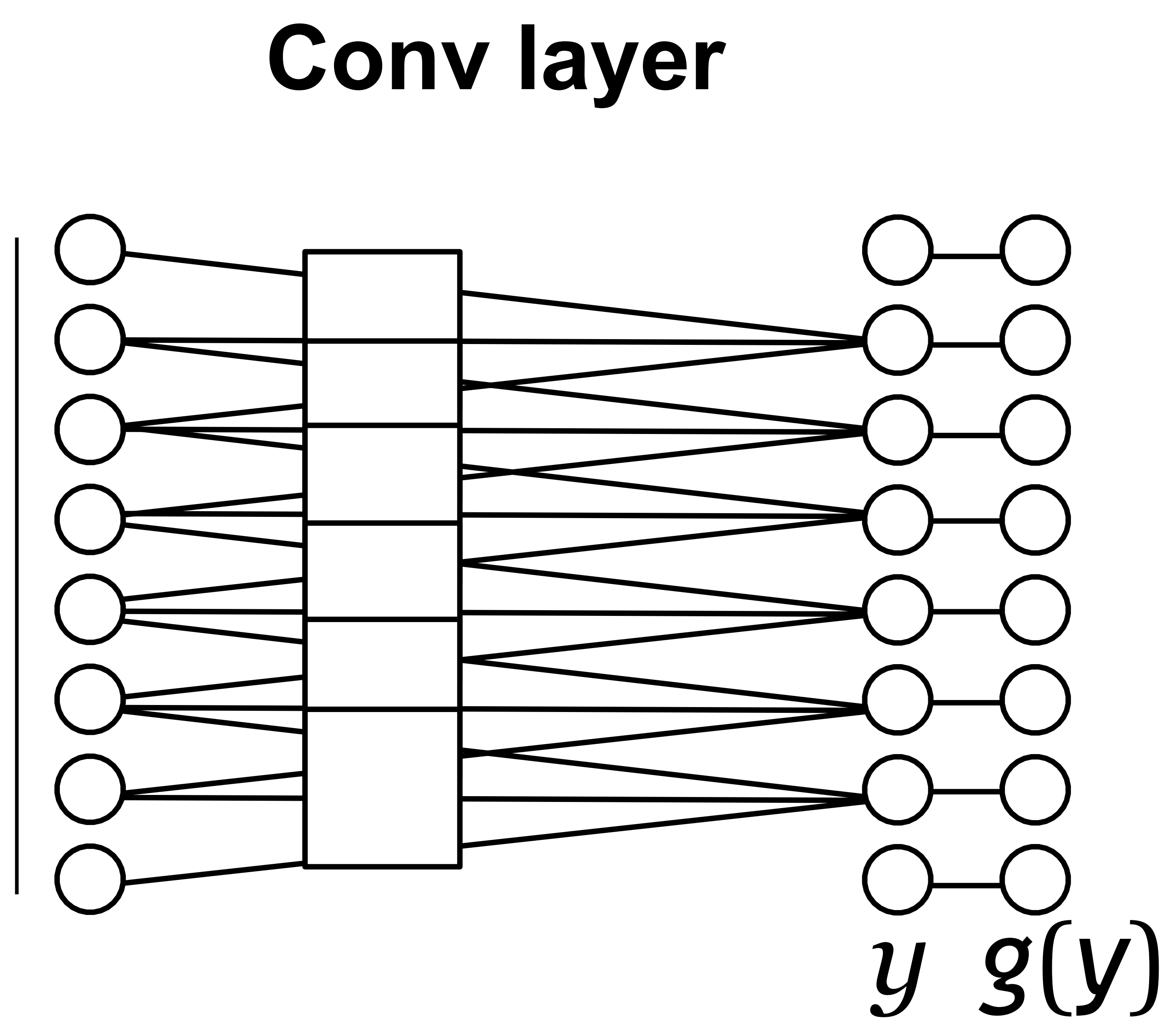
Conv layer



Each output unit is computed from an image patch.

Source: Isola, Torralba, Freeman

Weight sharing

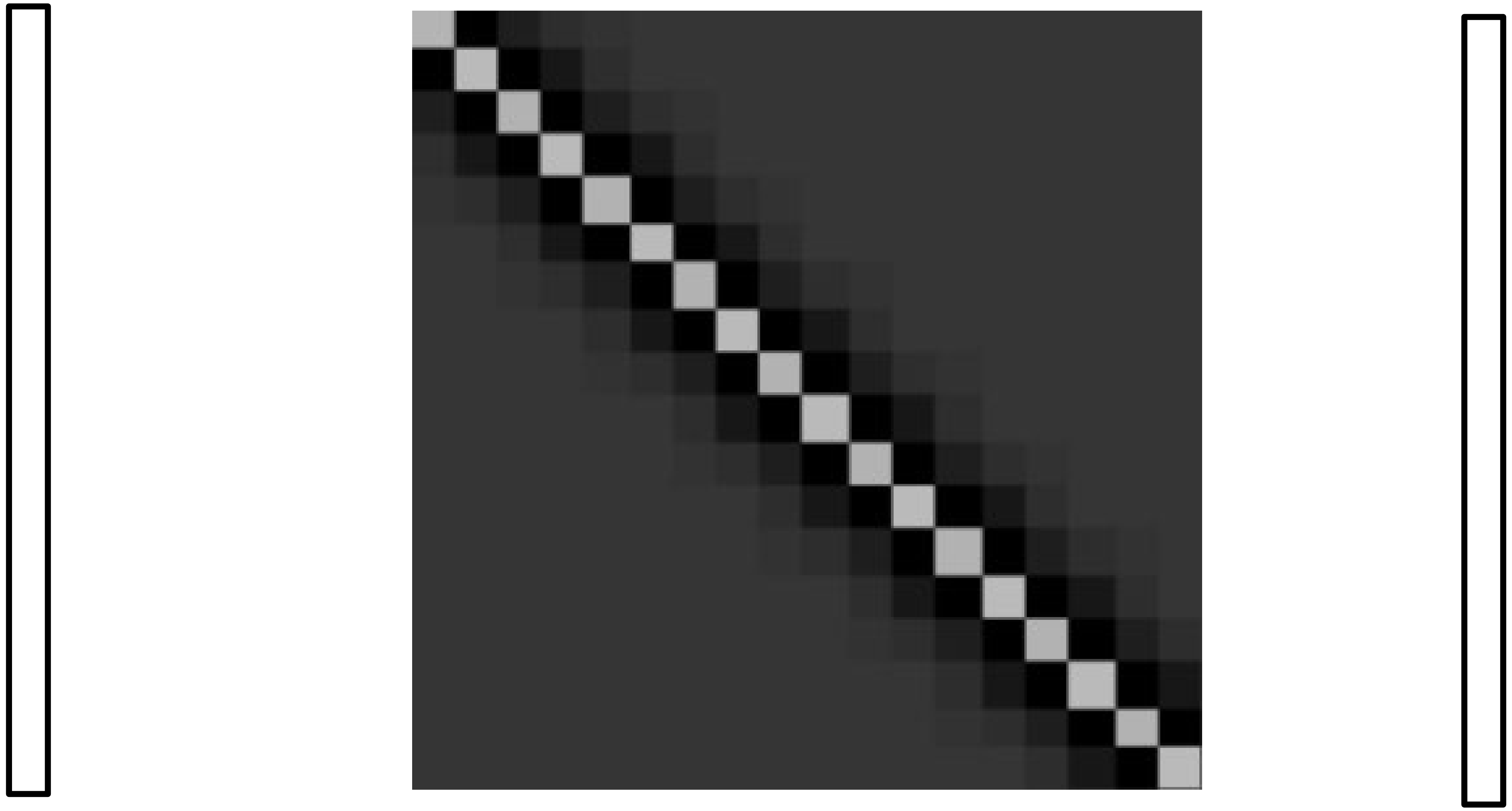


We "share" weights for each patch.

If a feature is useful in one position,
it should be useful in others, too.

Source: Isola, Torralba, Freeman

Convolution is a linear function



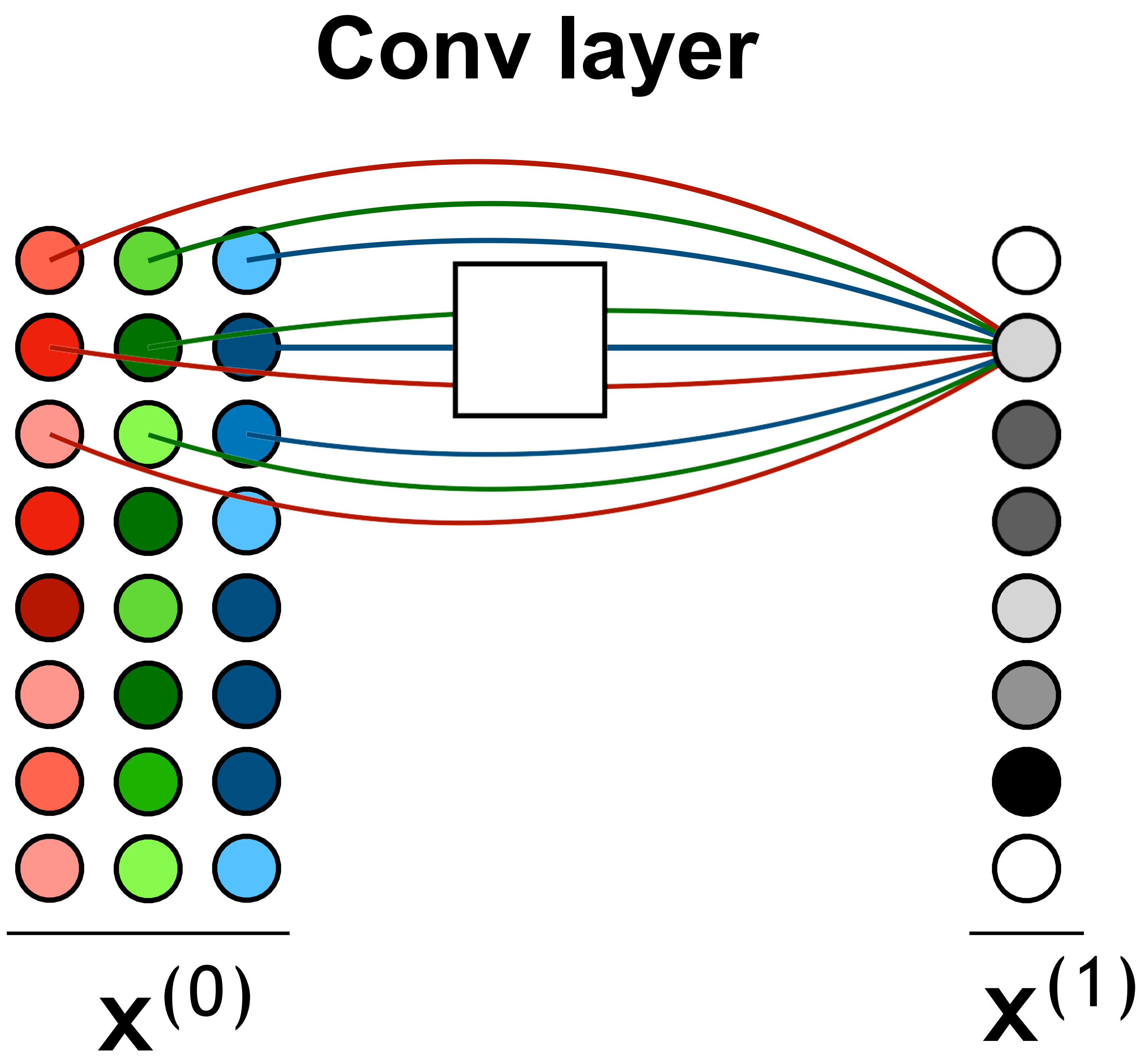
$\mathbf{x}^{(l+1)}$ $\mathbf{x}^{(l)}$

e.g., image

- Constrained linear layer
- Fewer parameters: easier to learn, less overfitting
- Usually use zero padding

Source: Isola, Torralba, Freeman

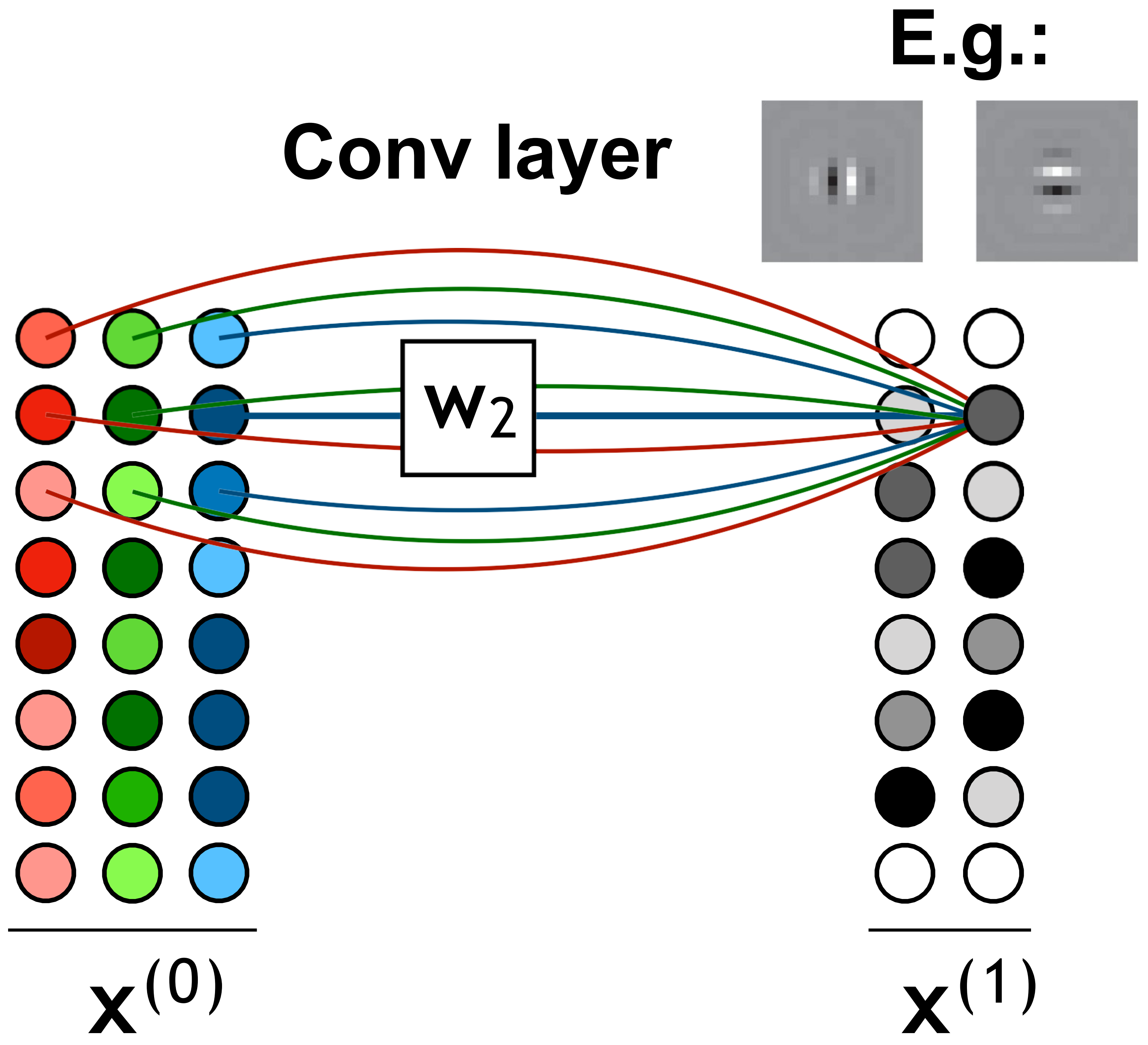
Multiple channels



$$\mathbb{R}^{N \times C} \to \mathbb{R}^{N \times 1}$$

Source: Isola, Torralba, Freeman

Multiple channels

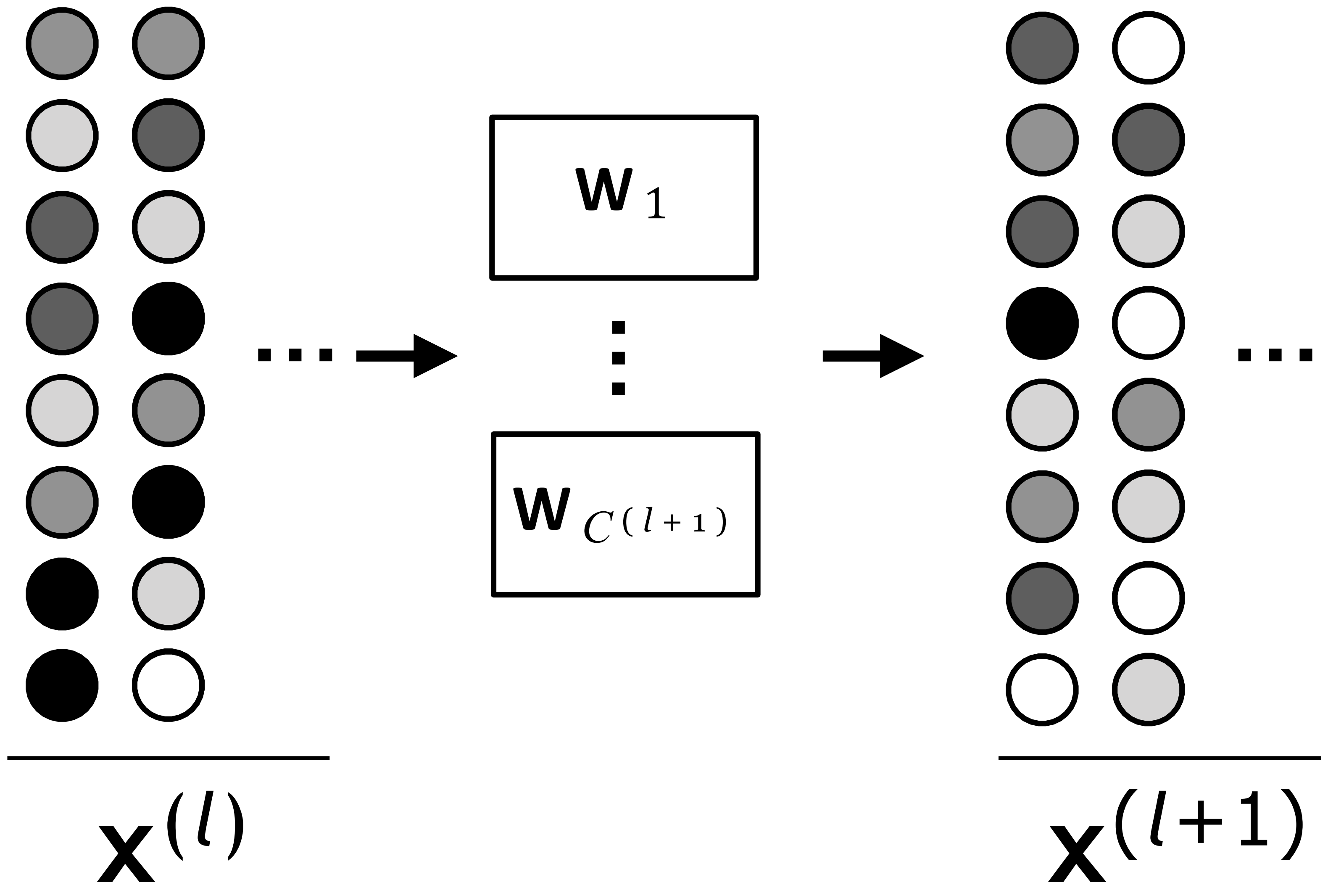


$$\mathbb{R}^{N \times C^{(0)}} \to \mathbb{R}^{N \times C^{(1)}}$$

Source: Isola, Torralba, Freeman

Multiple channels

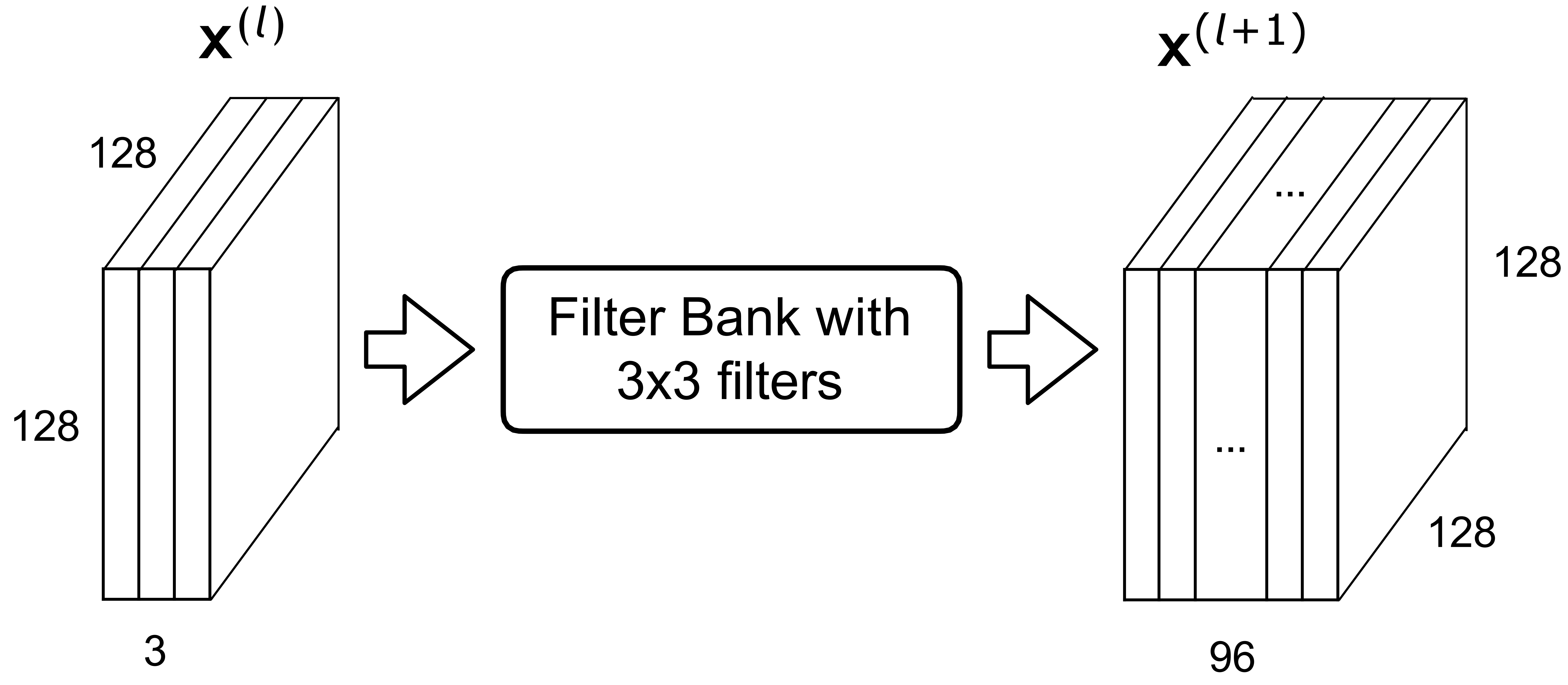
Conv layer



$$\mathbb{R}^{N \times C^{(l)}} \to \mathbb{R}^{N \times C^{(l+1)}}$$

Source: Isola, Torralba, Freeman

Multiple channels: Example

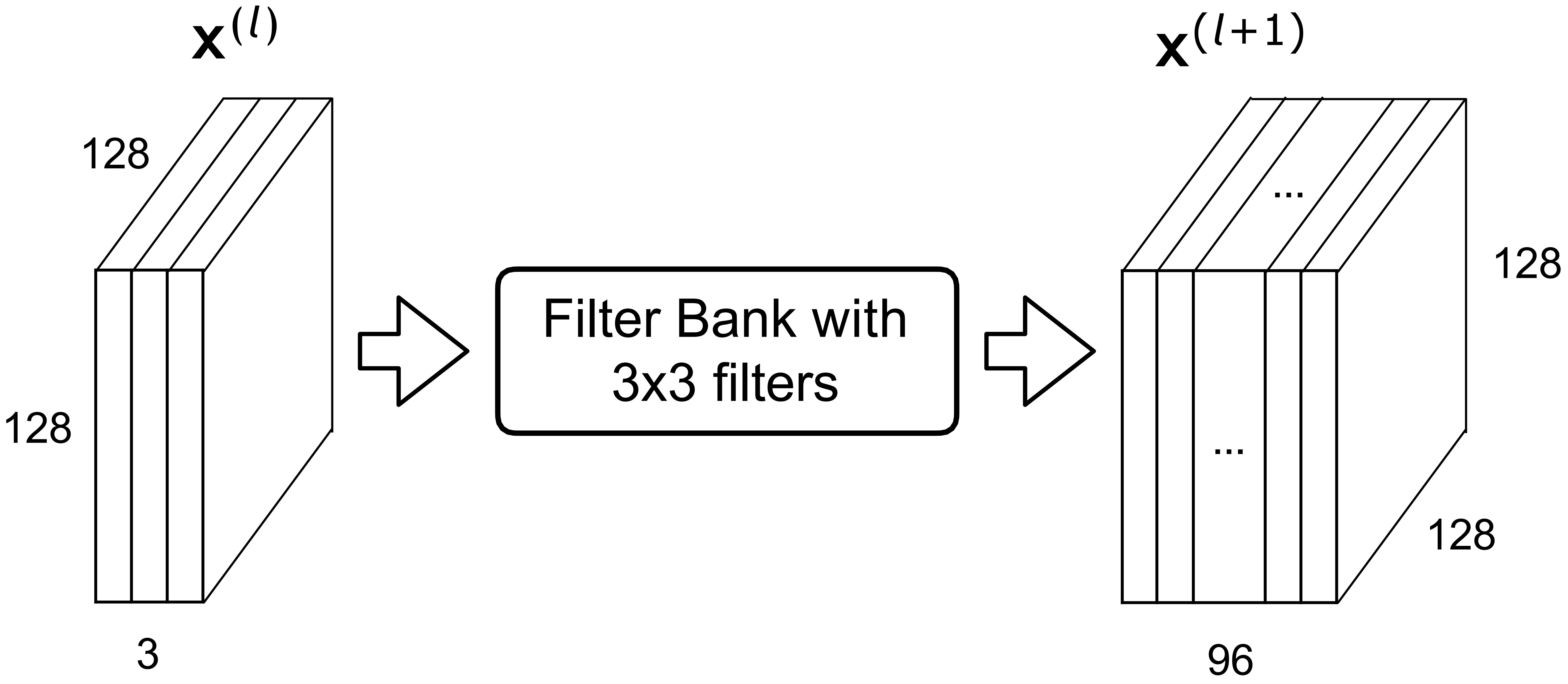


How many parameters does *each filter* have?

(a) 9 (b) 27 (c) 96 (d) 2592

Source: Isola, Torralba, Freeman

Multiple channels: Example

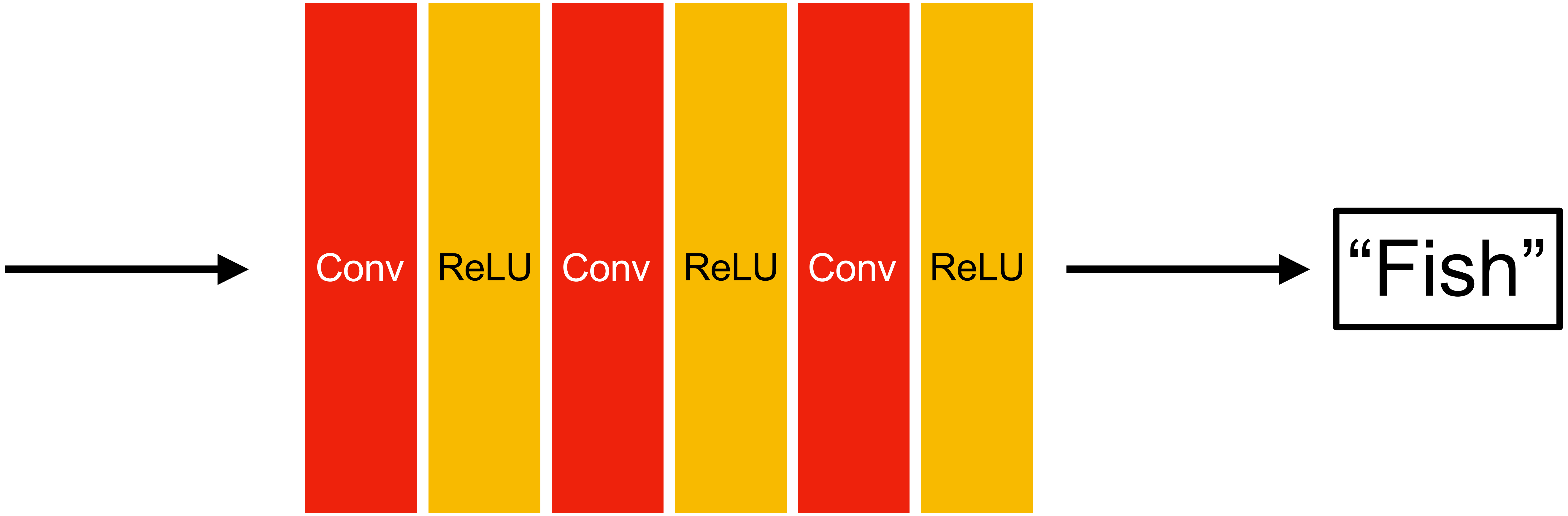


How many parameters *total* does this layer have?

(a) 9 (b) 27 (c) 96 (d) 2592

Image classification

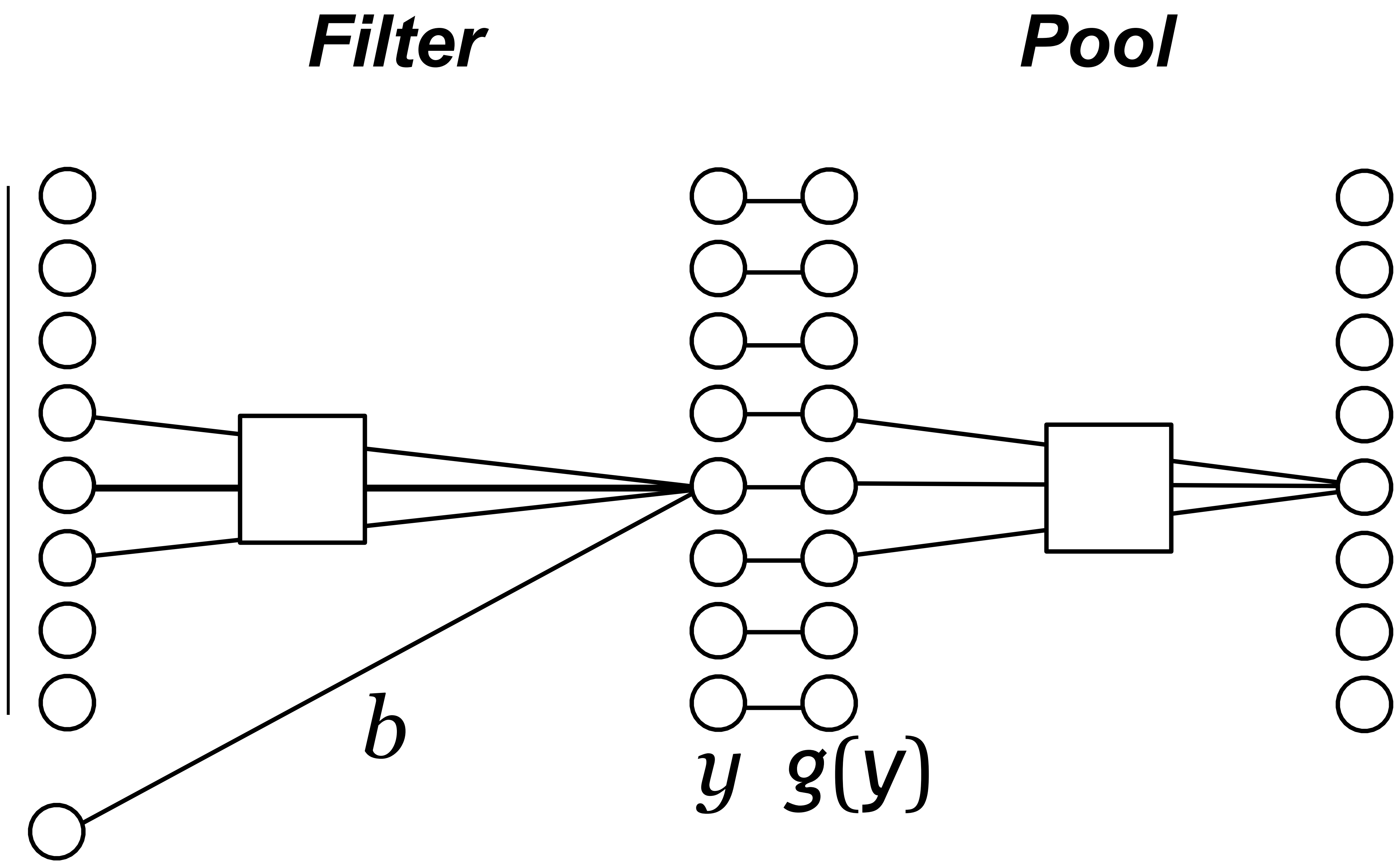
image **x**



label y

Source: Isola, Torralba, Freeman

Pooling



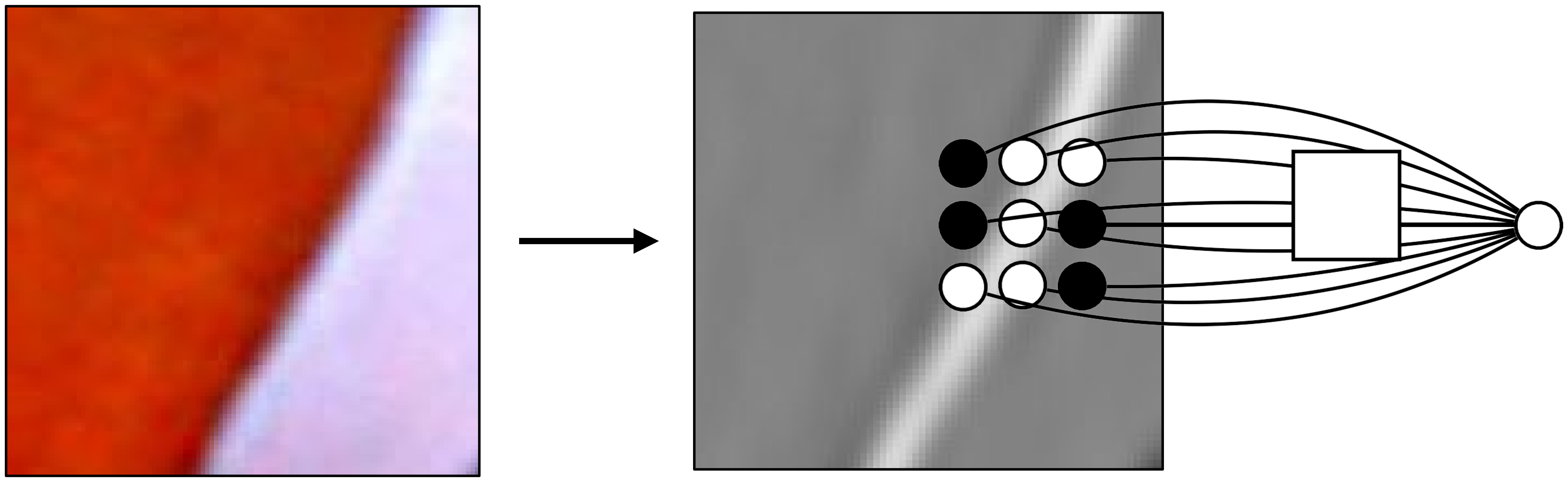
Max pooling

$$\mathbf{z}_k = \max_{j \in \mathcal{N}(j)} g(y_j)$$

Source: Isola, Torralba, Freeman

Pooling — Why?

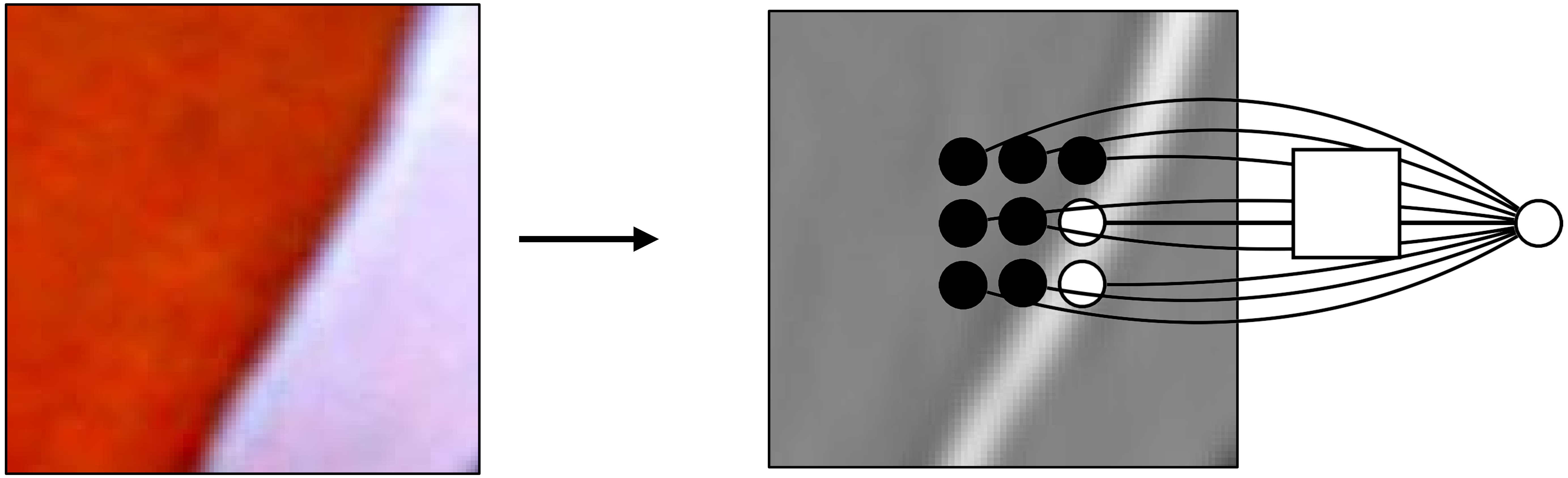
Pooling across spatial locations achieves stability w.r.t. small translations:



Source: Isola, Torralba, Freeman

Pooling — Why?

Pooling across spatial locations achieves stability w.r.t. small translations:

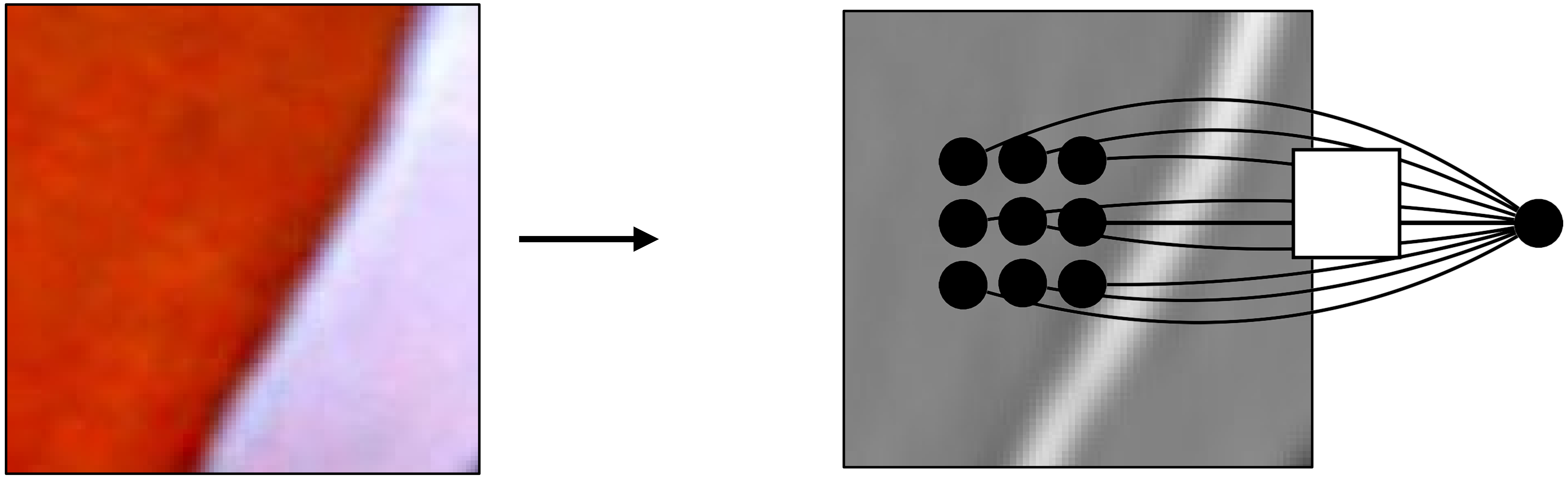


large response regardless of exact position of edge

Source: Isola, Torralba, Freeman

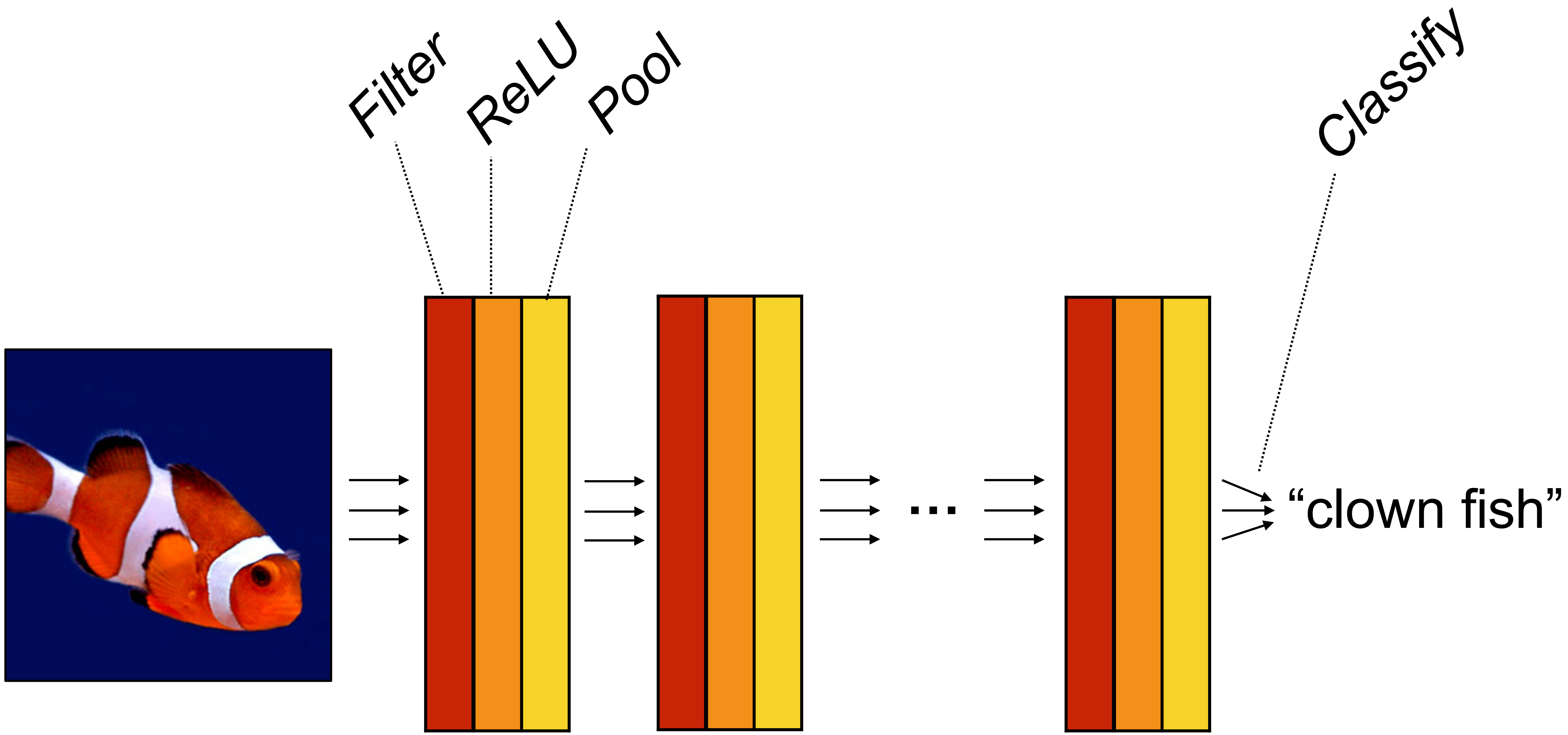
Pooling — Why?

Pooling across spatial locations achieves stability w.r.t. small translations:



Source: Isola, Torralba, Freeman

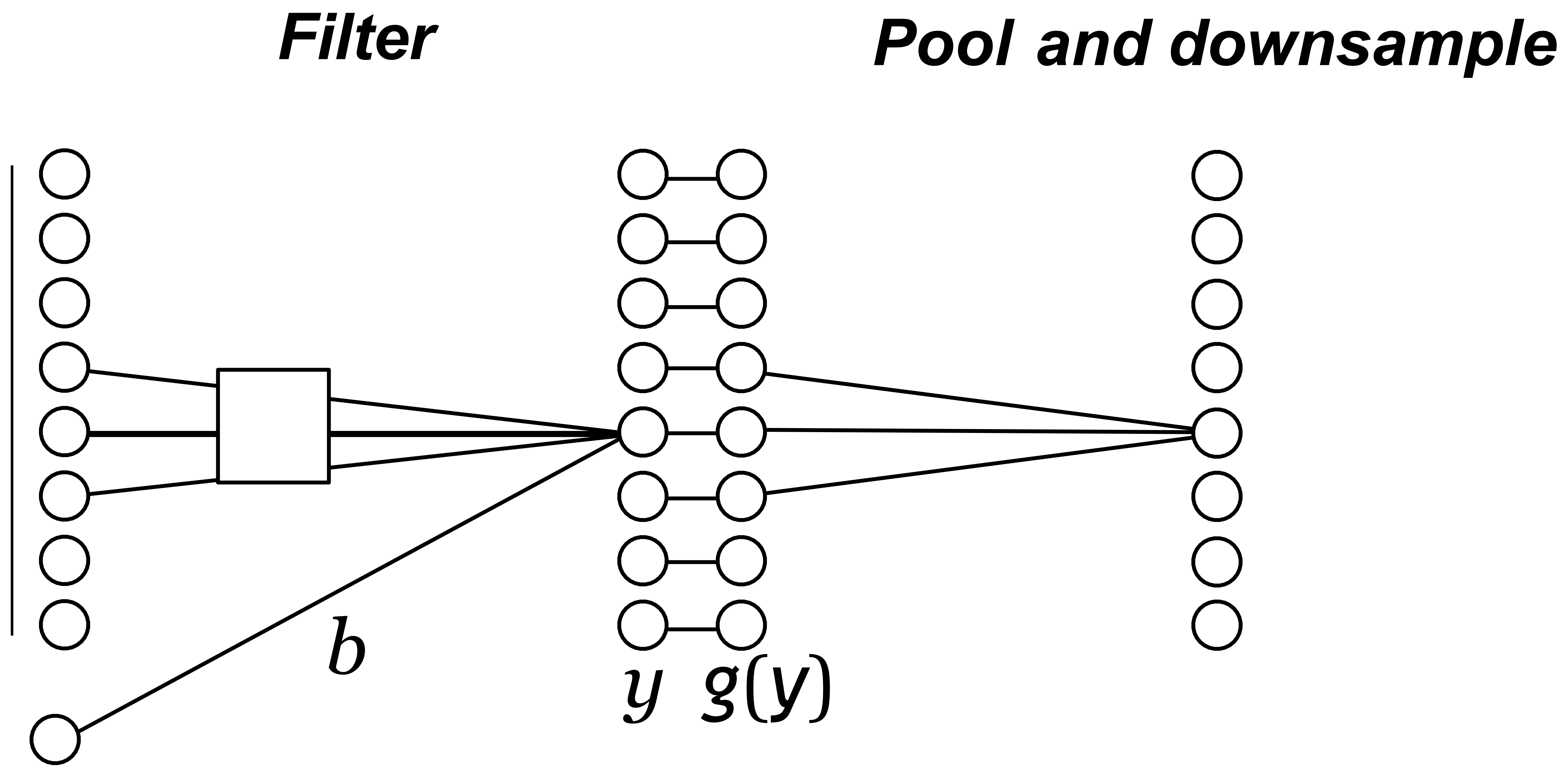
Computation in a neural net



$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$

Source: Isola, Torralba, Freeman

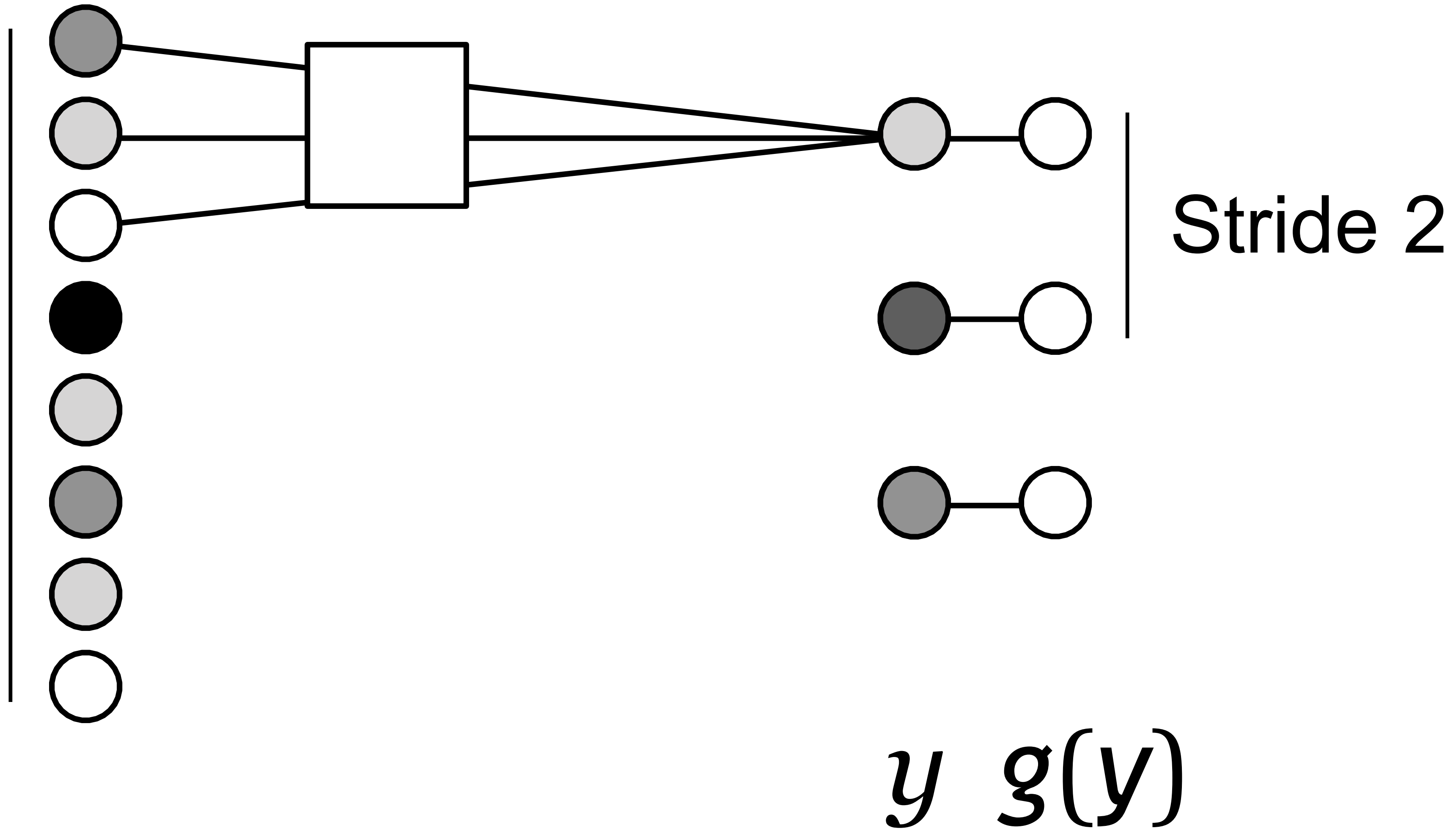
Downsampling



Source: Isola, Torralba, Freeman

Strided operations

Conv layer

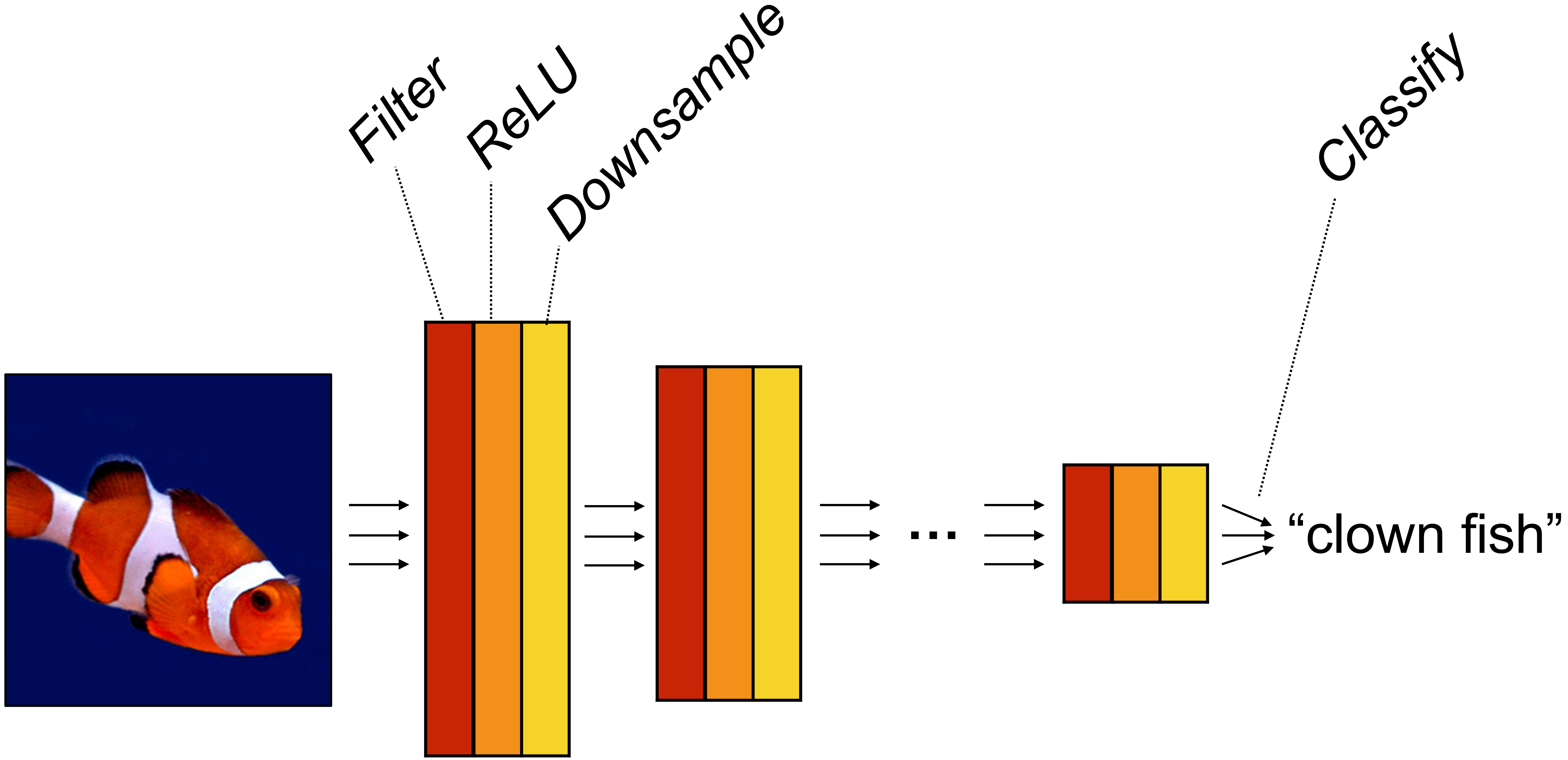


Strided operations combine a given operation (convolution or pooling) and downsampling into a single operation.

Strided convolution is an alternative to pooling layers: just do a strided convolution!

Source: Isola, Torralba, Freeman

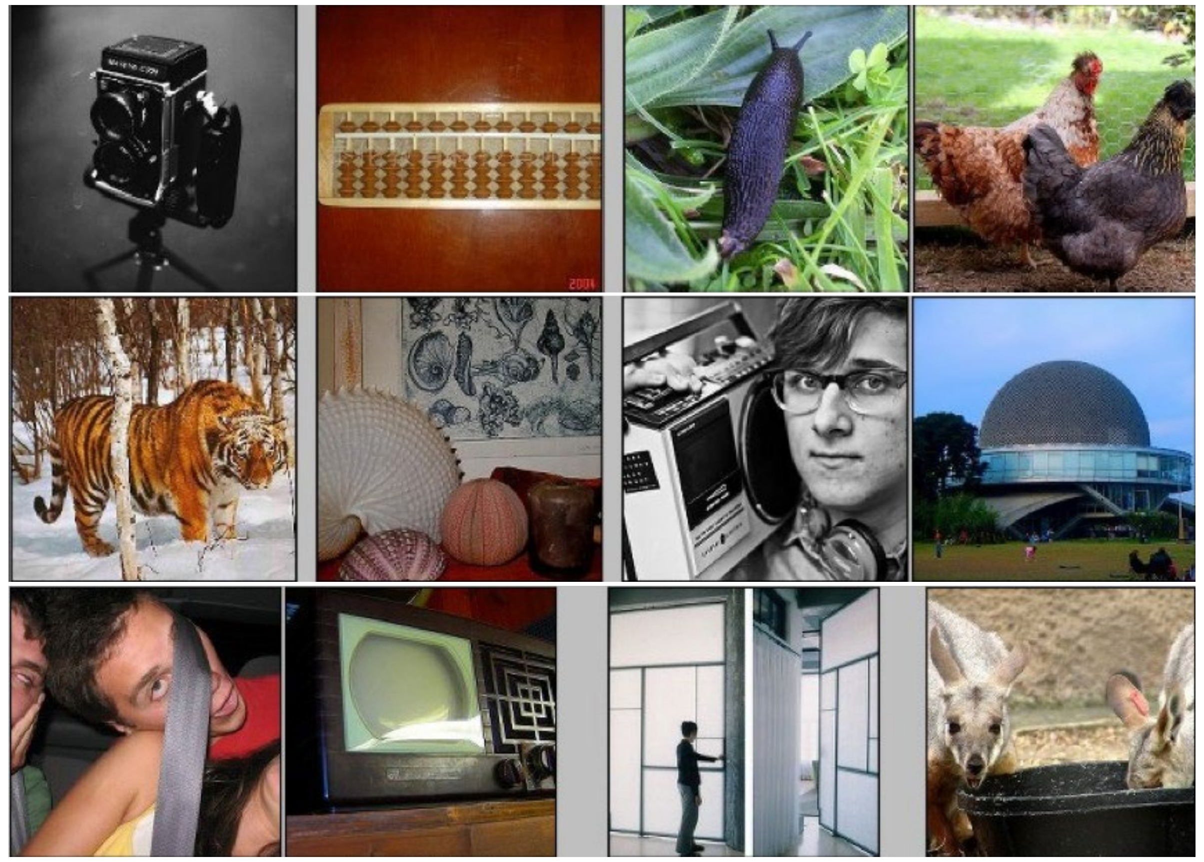
Computation in a neural net



$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$

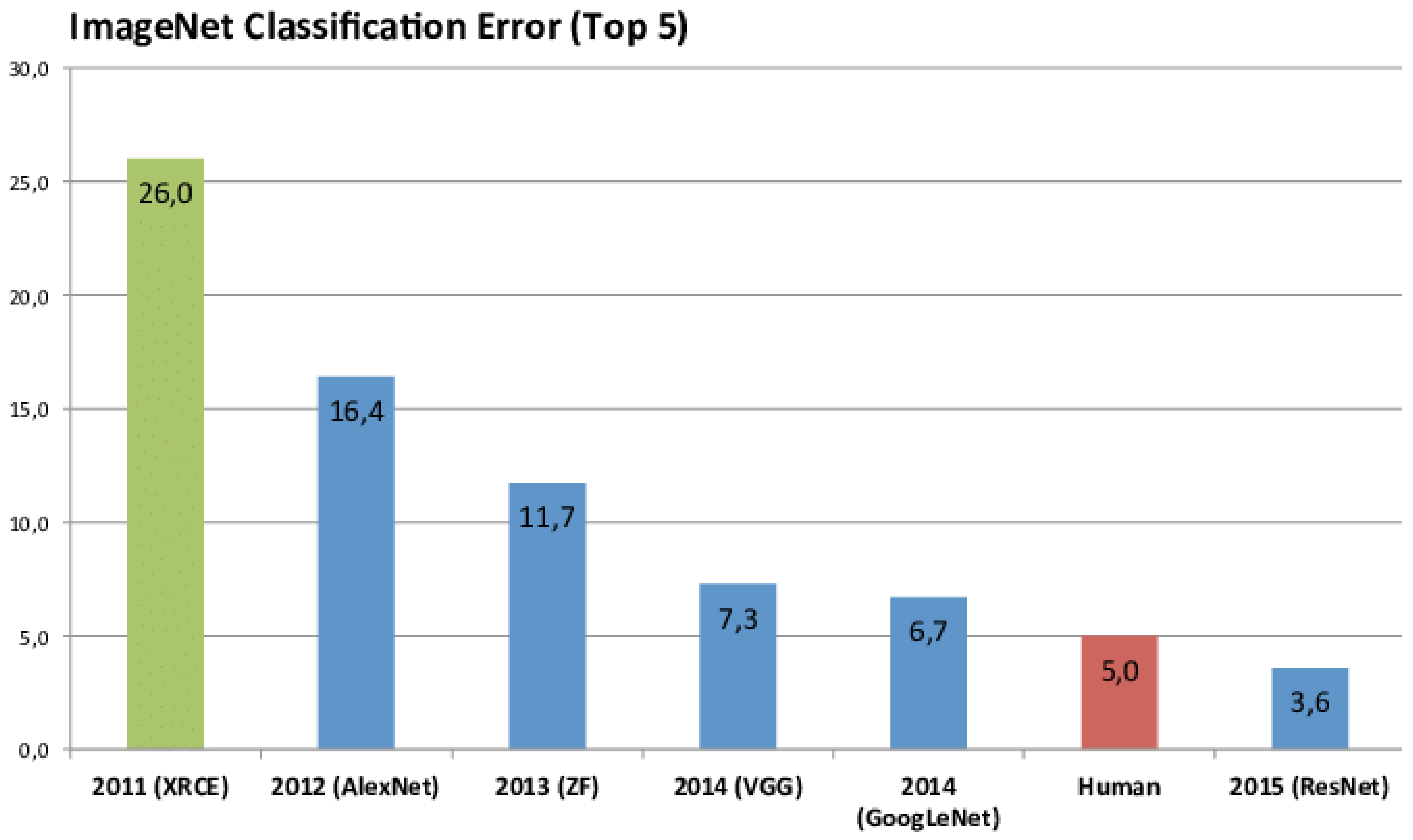
Source: Isola, Torralba, Freeman

ImageNet Challenge



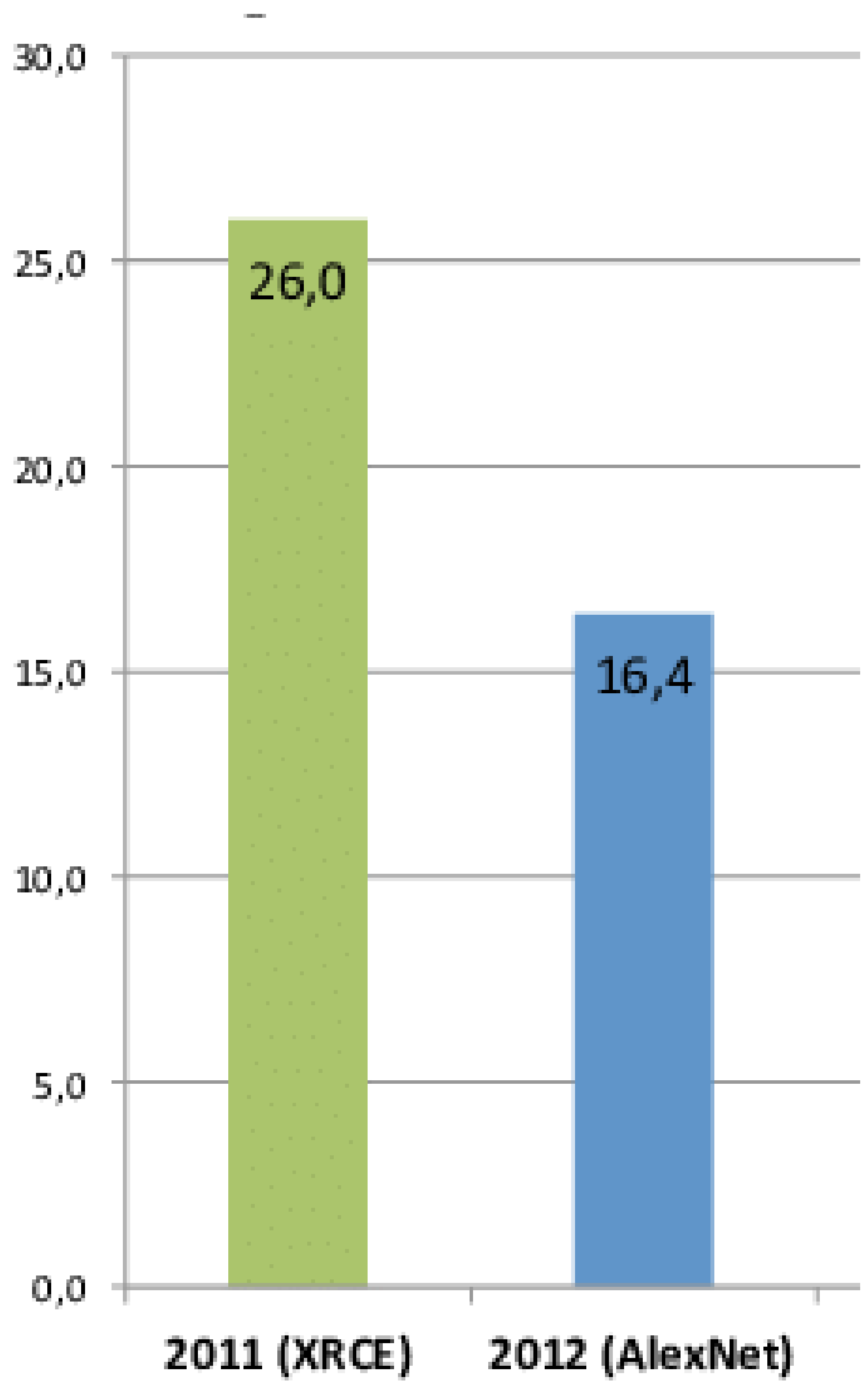
- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon MTurk
- ImageNet Large-Scale Visual Recognition Challenge (ILSVRC):
  1.2 million training images, 1000 classes

[Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein, Berge, Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge”, 2015]

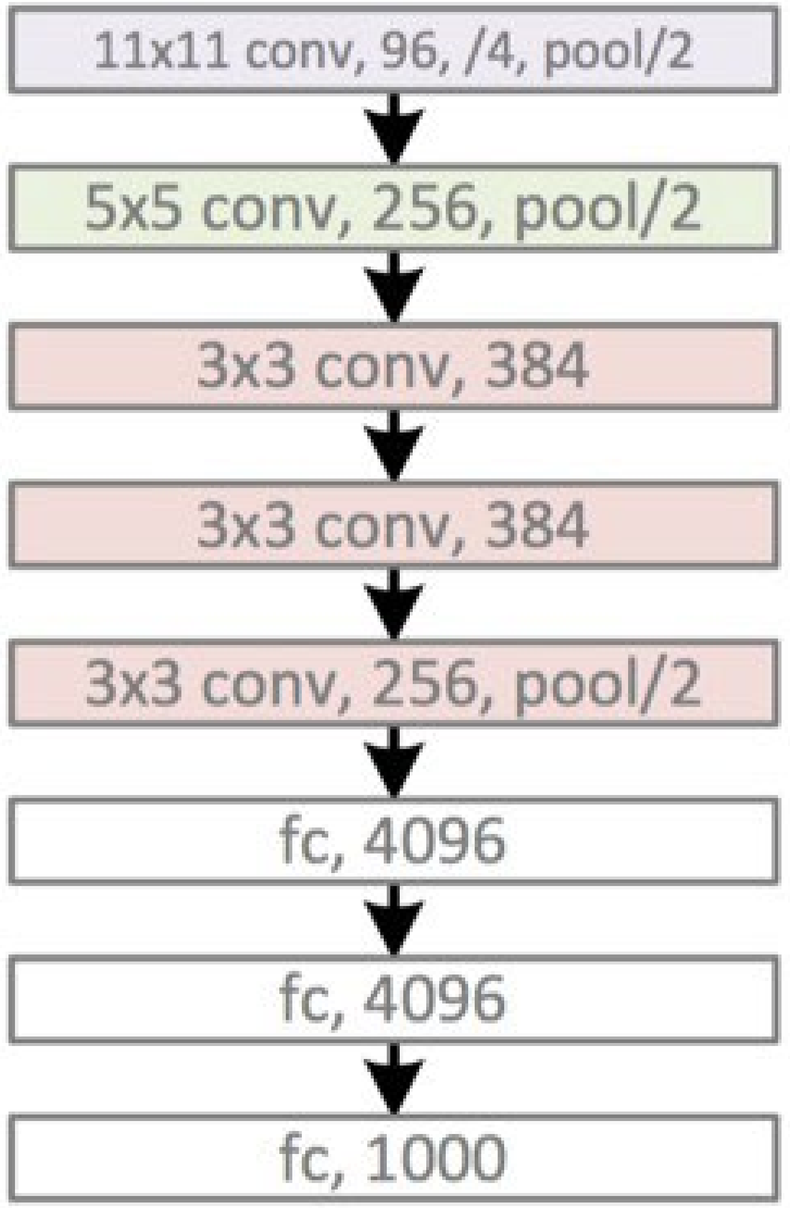

ImageNet Classification Error (Top 5)
30,0
25,0
20,0
15,0
10,0
5,0
0,0
26,0
16,4
11,7
7,3
6,7
5,0
3,6
2011 (XRCE)
2012 (AlexNet)
2013 (ZF)
2014 (VGG)
2014 (GoogLeNet)
Human
2015 (ResNet)

Network designs

ImageNet classification (top 5)



2012: AlexNet
5 conv. layers

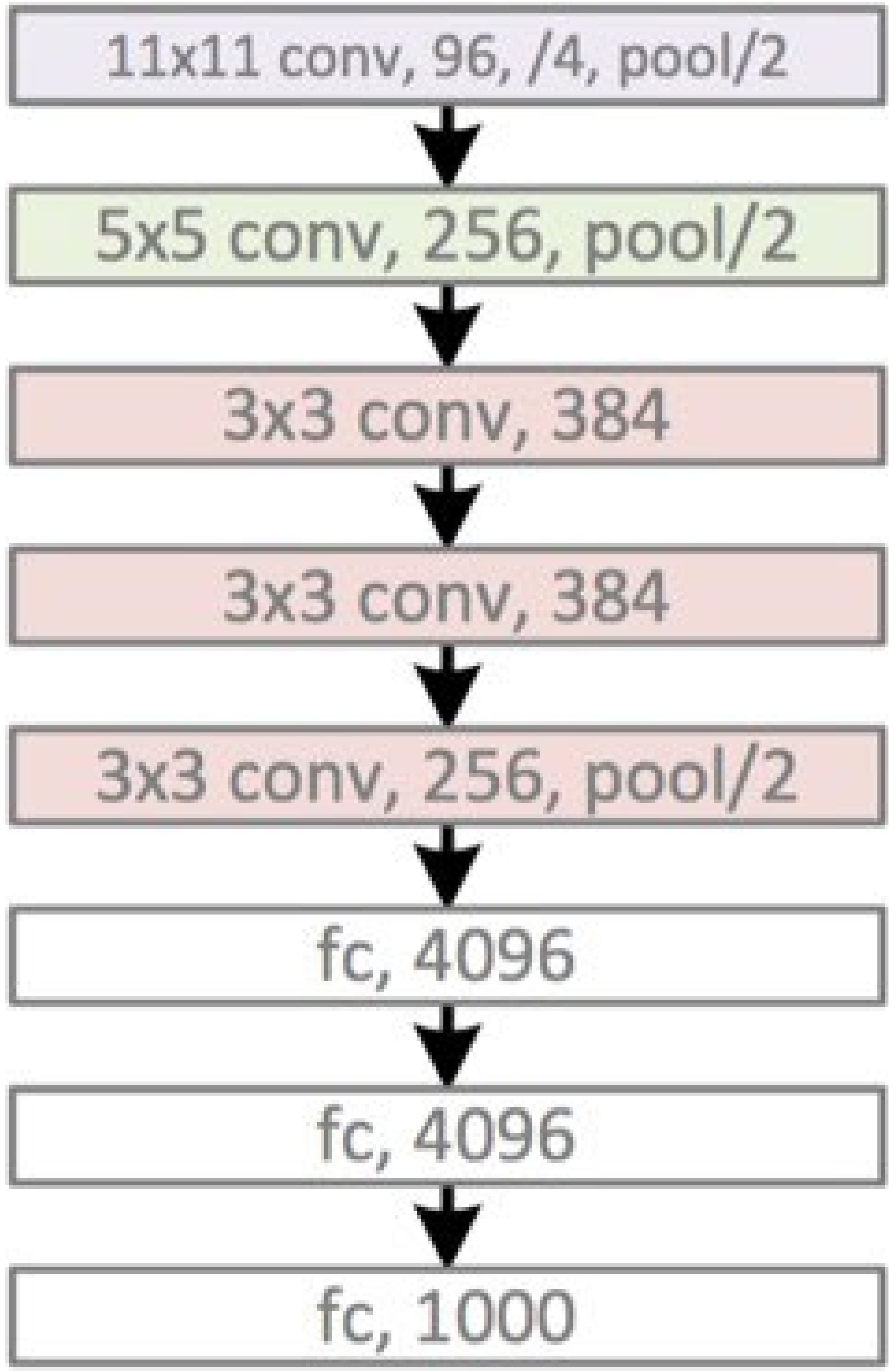


Error: 16.4%

[Krizhevsky et al: ImageNet Classification with Deep Convolutional Neural Networks, NeurIPS 2012]

Source: Isola, Torralba, Freeman

Alexnet — [Krizhevsky et al. NIPS 2012]



[227x227x3] INPUT

[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer

[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer

[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1

[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1

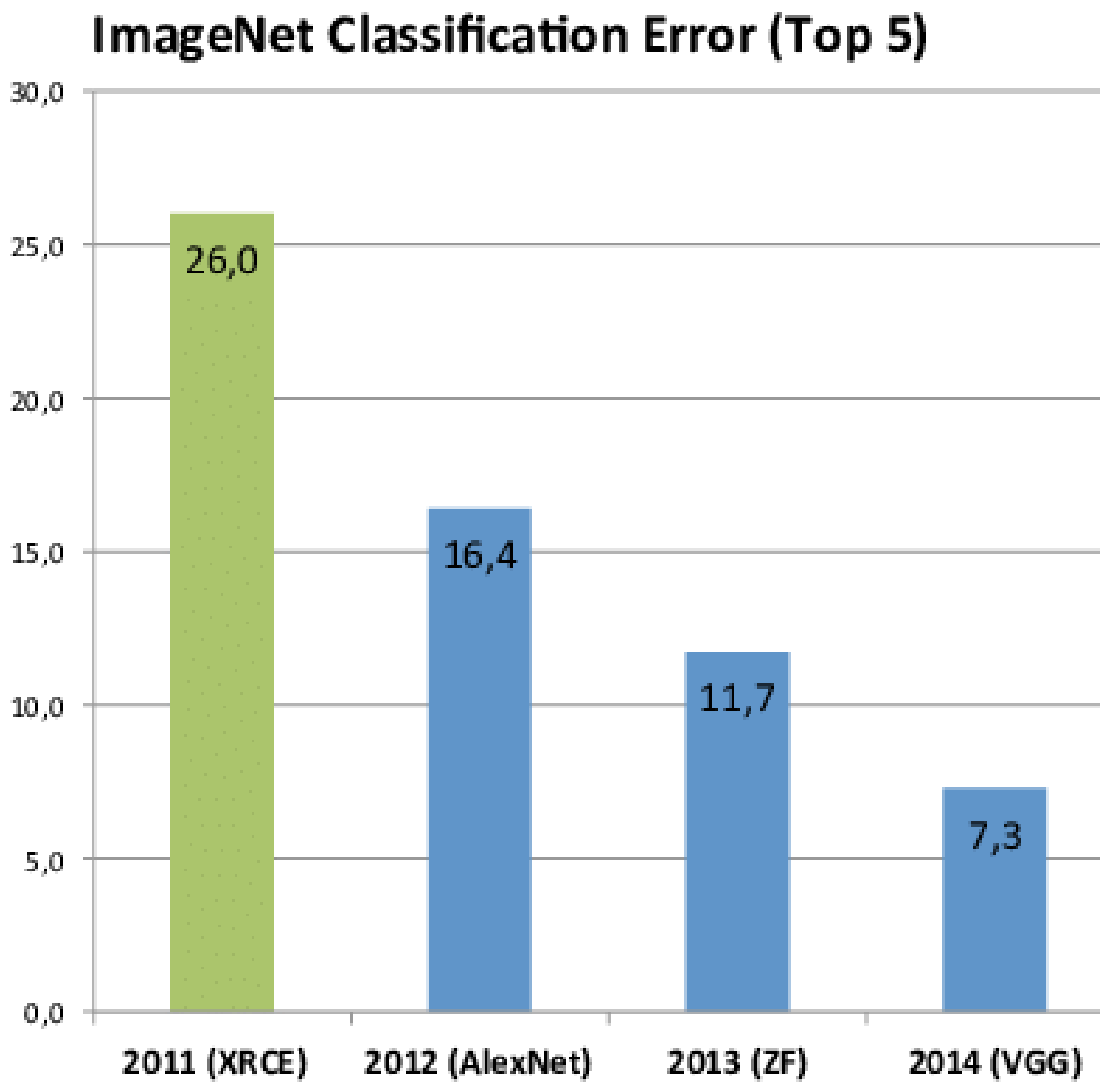
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2

[4096] FC6: 4096 neurons

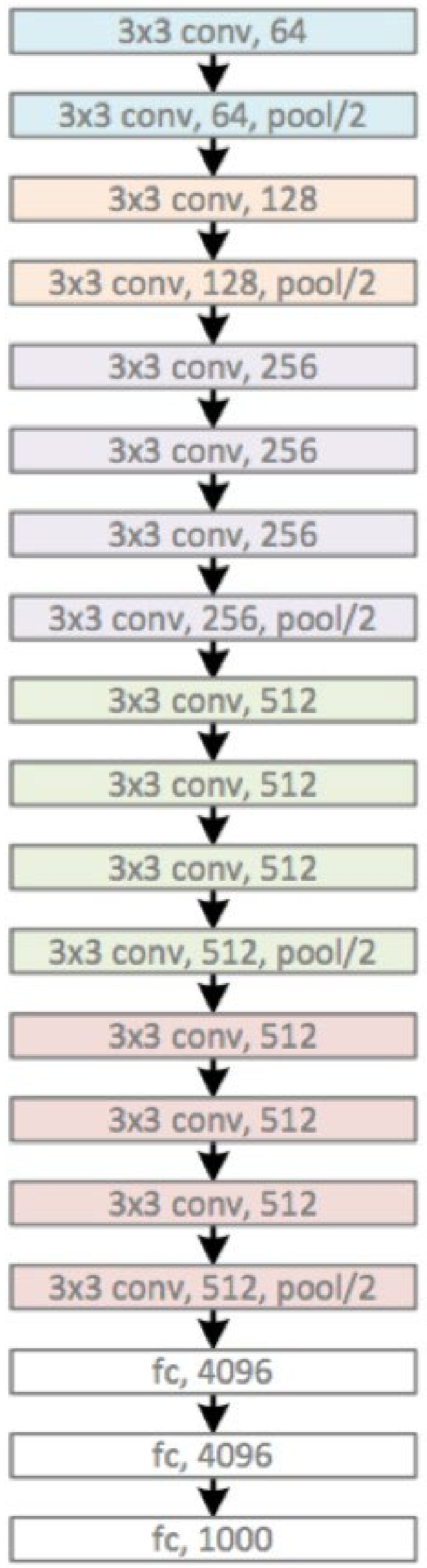
[4096] FC7: 4096 neurons

[1000] FC8: 1000 neurons (class scores)

Source: Isola, Torralba, Freeman



2014: VGG
16 conv. layers



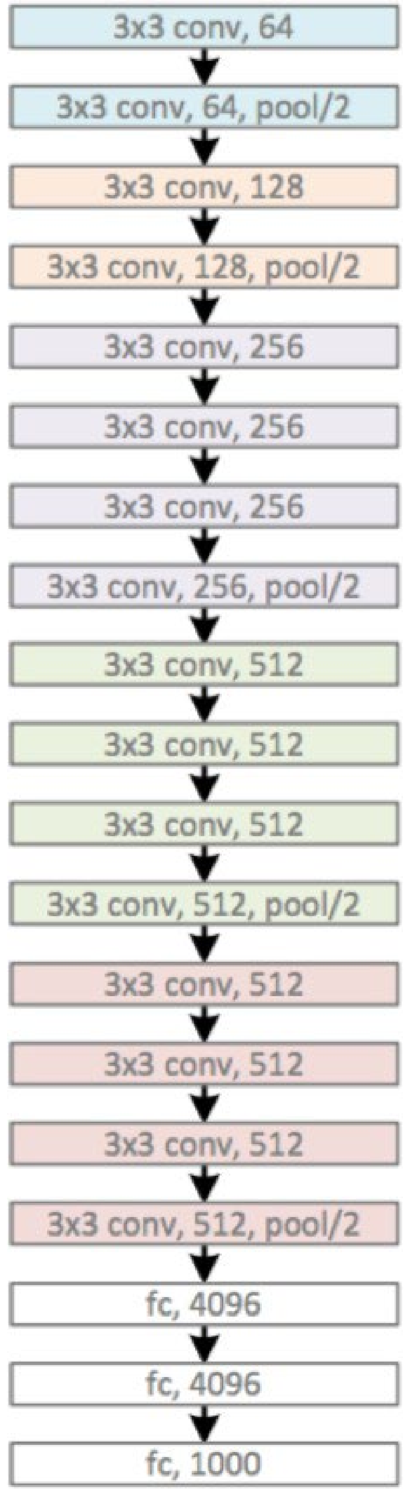
Error: 7.3%

[Simonyan & Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition, ICLR 2015]

Source: Isola, Torralba, Freeman

VGG-Net [Simonyan & Zisserman, 2015]

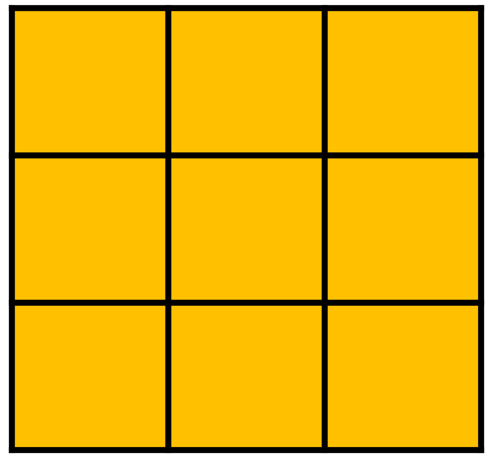
2014: VGG
16 conv. layers



Error: 7.3%

Main developments

- Small convolutional kernels: only 3x3

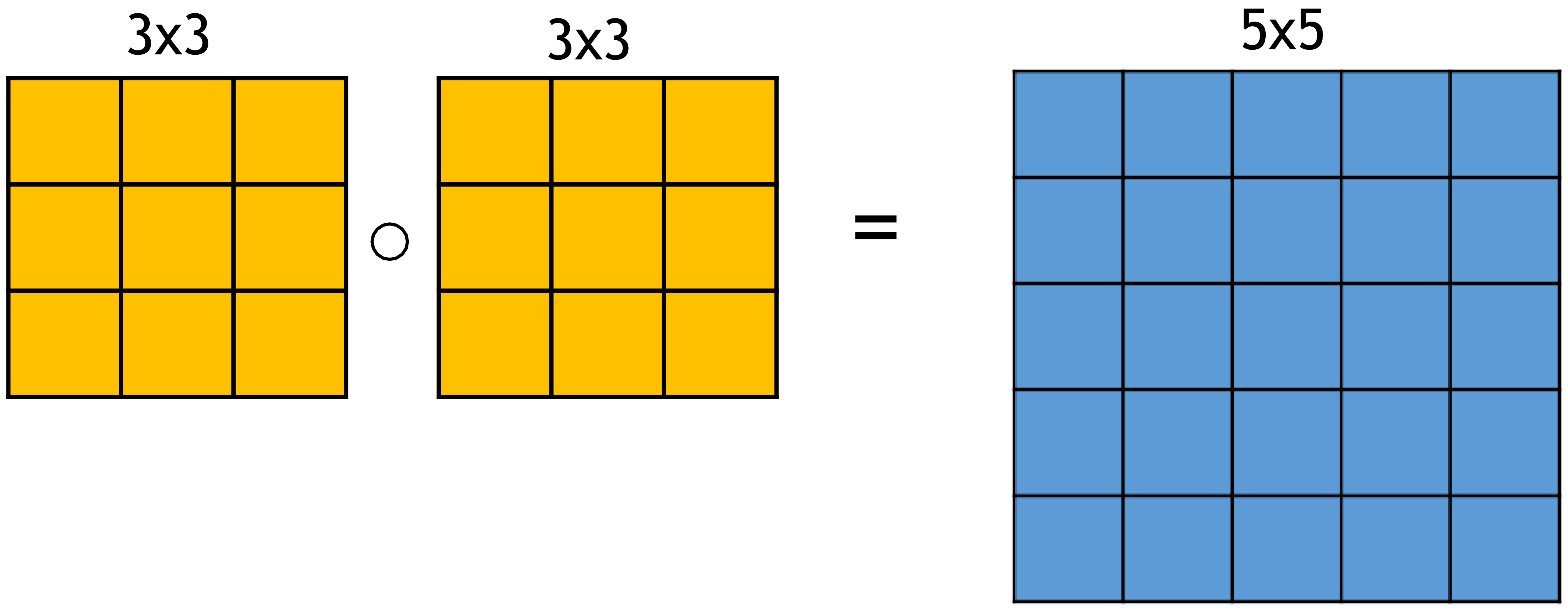


- Increased depth (5 -> 16/19 layers)

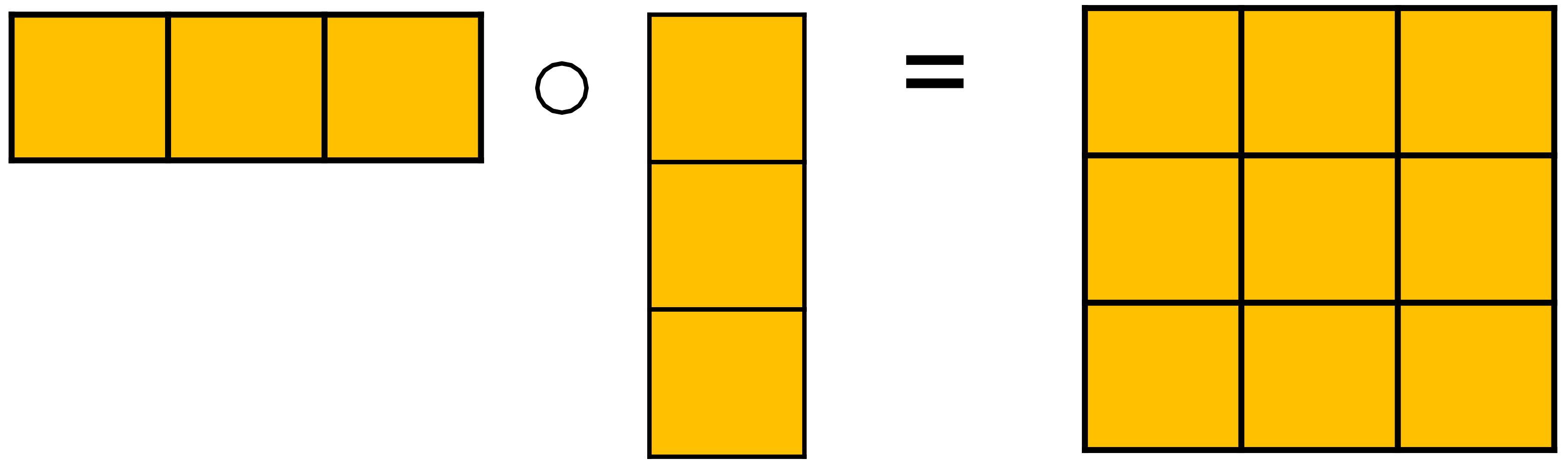
Source: Isola, Torralba, Freeman

Other tricks for designing convolutional nets

Chaining convolutions



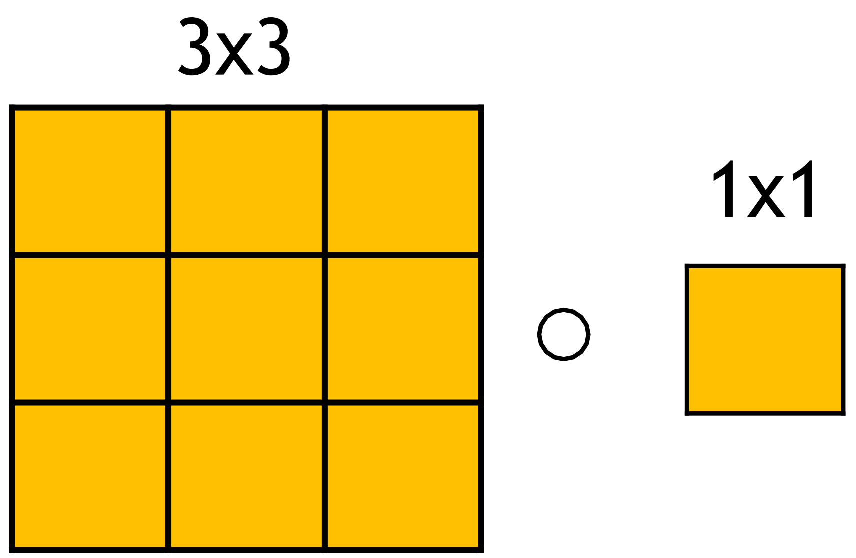
Nonlinearity in between.
25 coefficients, but only
18 degrees of freedom



9 coefficients, but only
6 degrees of freedom.
Less common.

Source: Isola, Torralba, Freeman

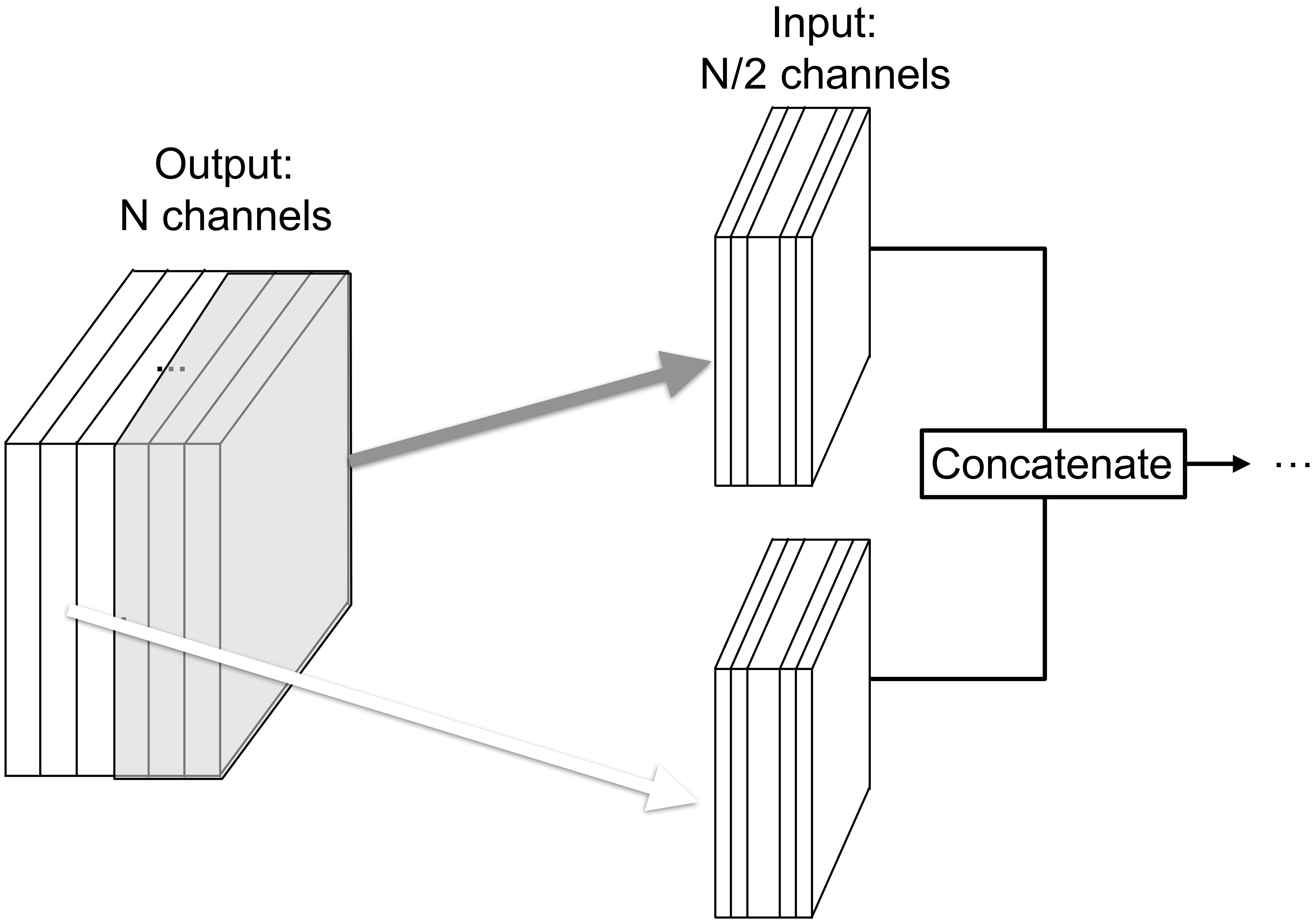
1x1 convolutions



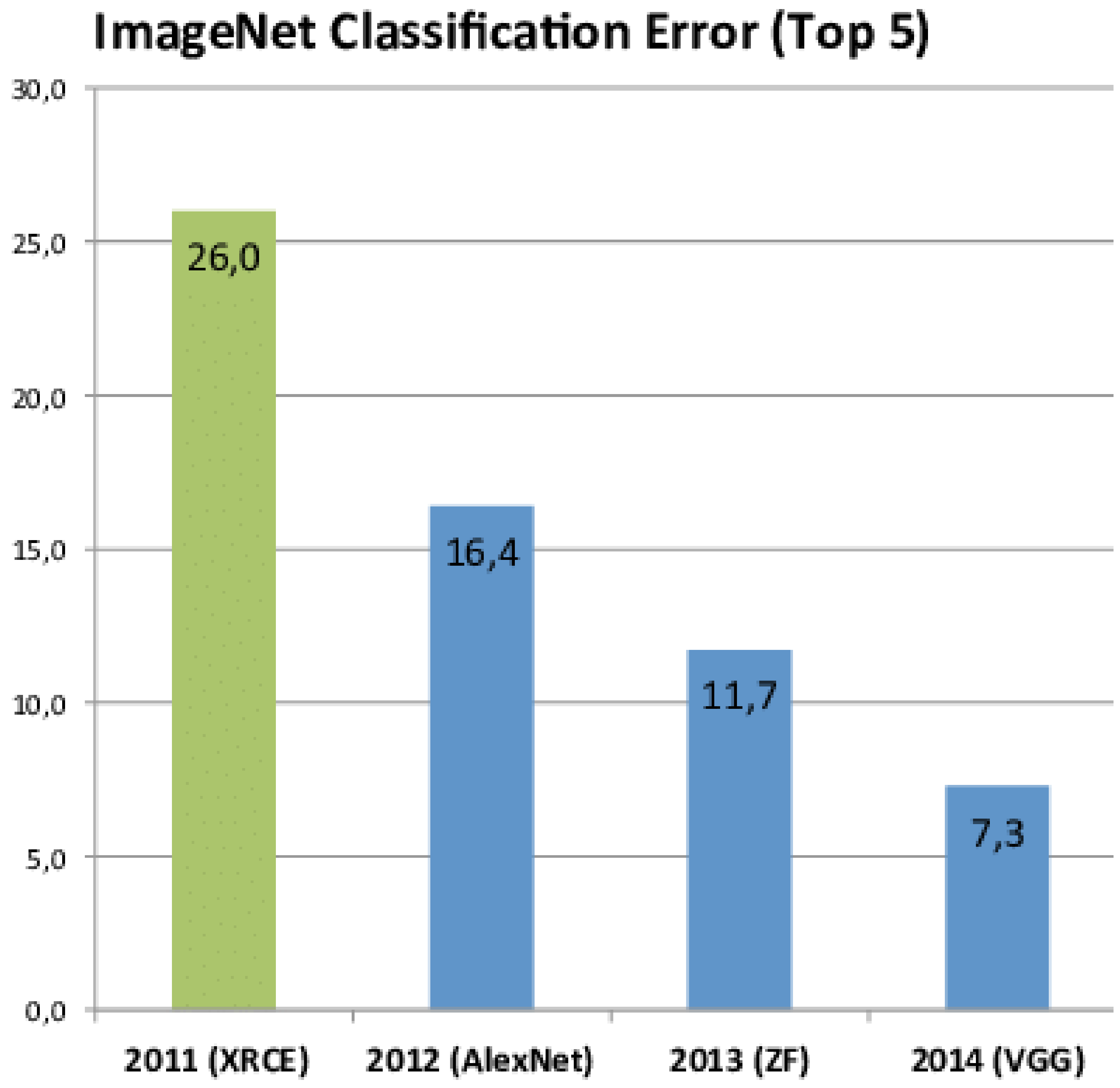
Why do this?

(nonlinearity in between)

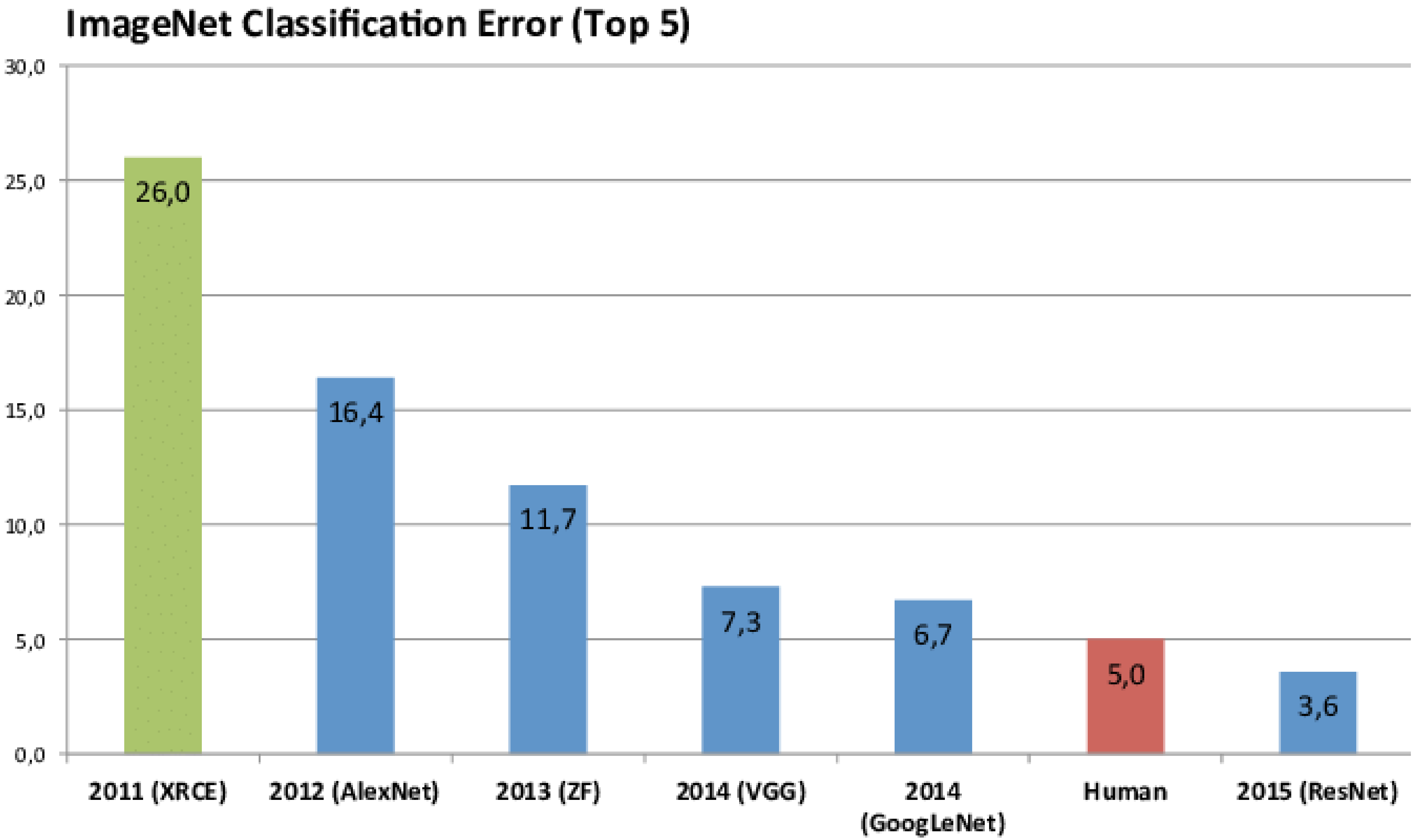
Grouped Convolutions



Split channels into N groups, and process separately with N convolution layers.



Source: Isola, Torralba, Freeman



2016: ResNet
>100 conv. layers

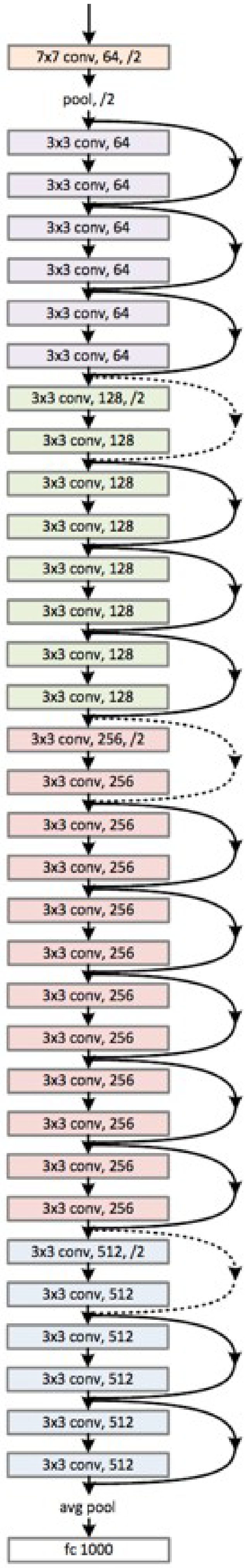
Error: 3.6%

[He et al: Deep Residual Learning for Image Recognition, CVPR 2016]

Source: Isola, Torralba, Freeman

2016: ResNet
>100 conv. layers

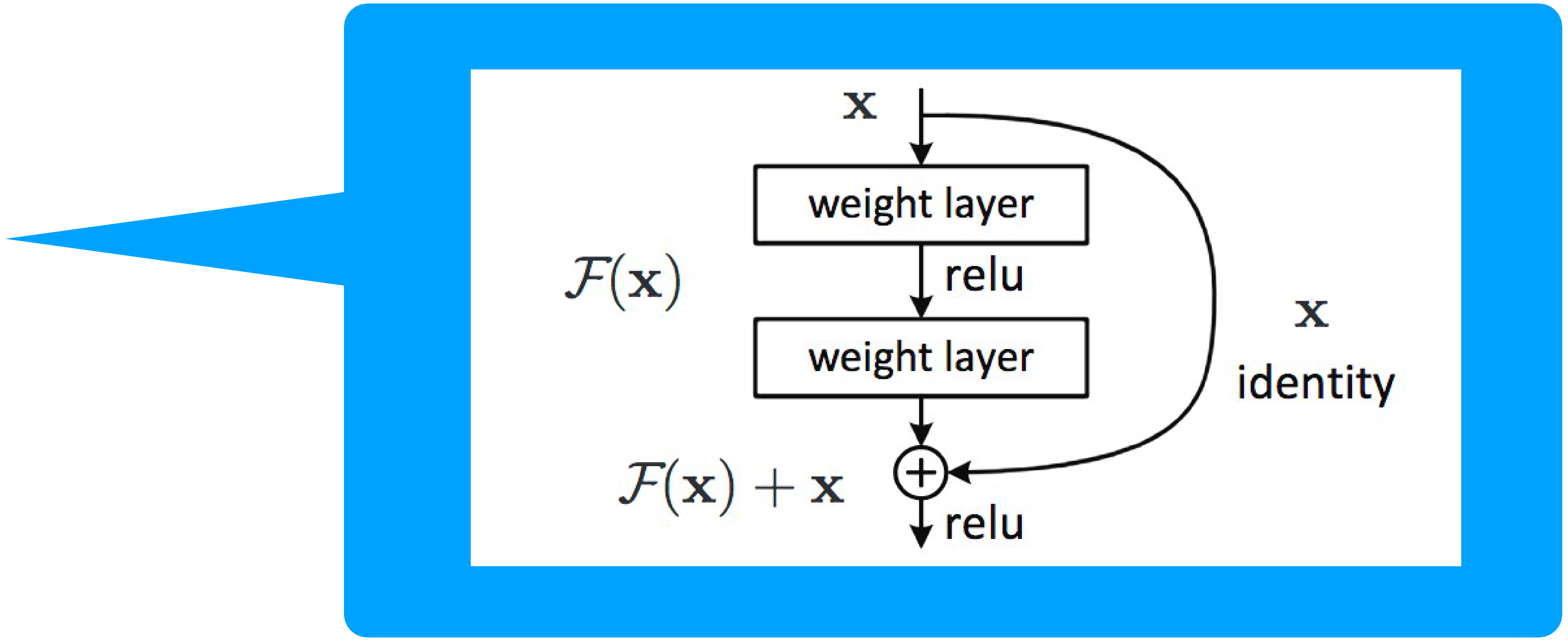
Error: 3.6%



ResNet [He et al, 2016]

Main developments

- Increased depth possible through residual blocks



Source: Isola, Torralba, Freeman

Residual Blocks

Problem: Hard to train very deep nets (50+ layers). This is an optimization issue, not overfitting: shallow models often get higher *training* accuracy than deep ones!

Idea: Make it easy to represent for the network to implement the identity.

Normal convolution + relu:

$$x_{i+1} = \mathrm{relu}(x_i \circ f)$$

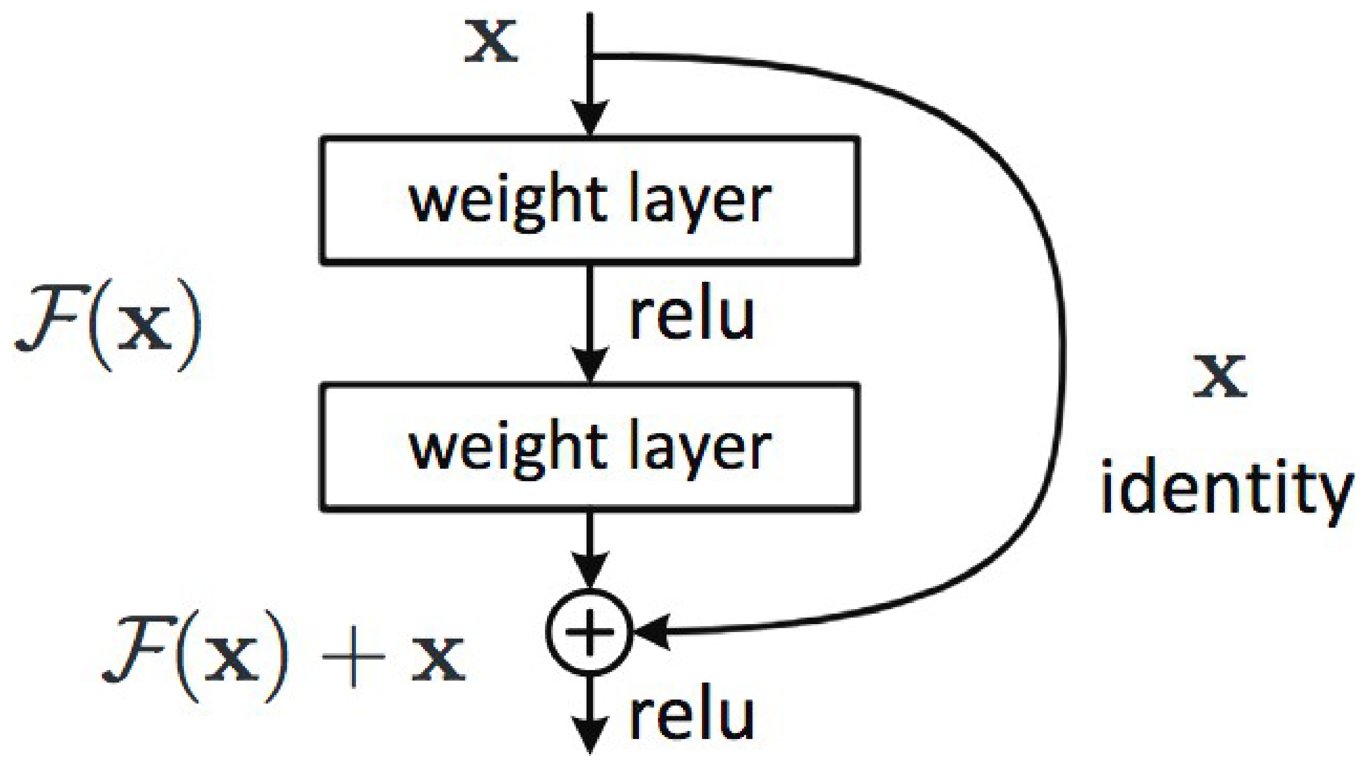
Residual connection

$$x_{i+1} = \mathrm{relu}((x_i \circ f) + x_i)$$

In general, do multiple convolutions (with nonlinearities) before summing:

$$x_{i+1} = \mathrm{relu}(F(x_i) + x_i)$$

Residual Blocks



Why do they work?

- Gradients can propagate faster (via the identity mapping)
- Within each block, only small residuals have to be learned

Source: Isola, Torralba, Freeman

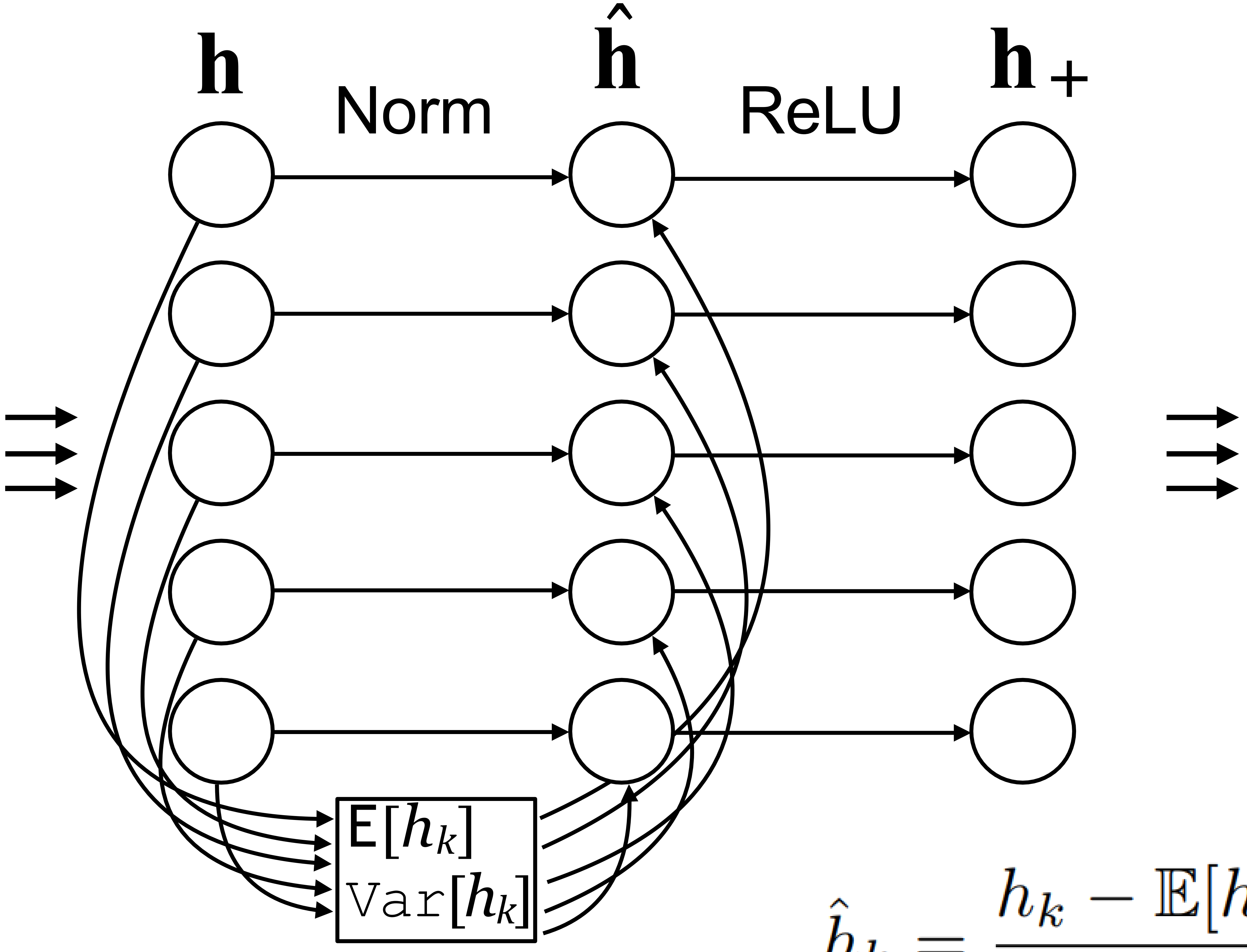
Normalization layers

Standardize activations by subtracting mean and dividing by standard deviation (averaged over all spatial locations).

This provides a constant “interface” for later layers of the networks. Ensures that the previous layer will have zero mean,

Obtains invariance to mean and variance.

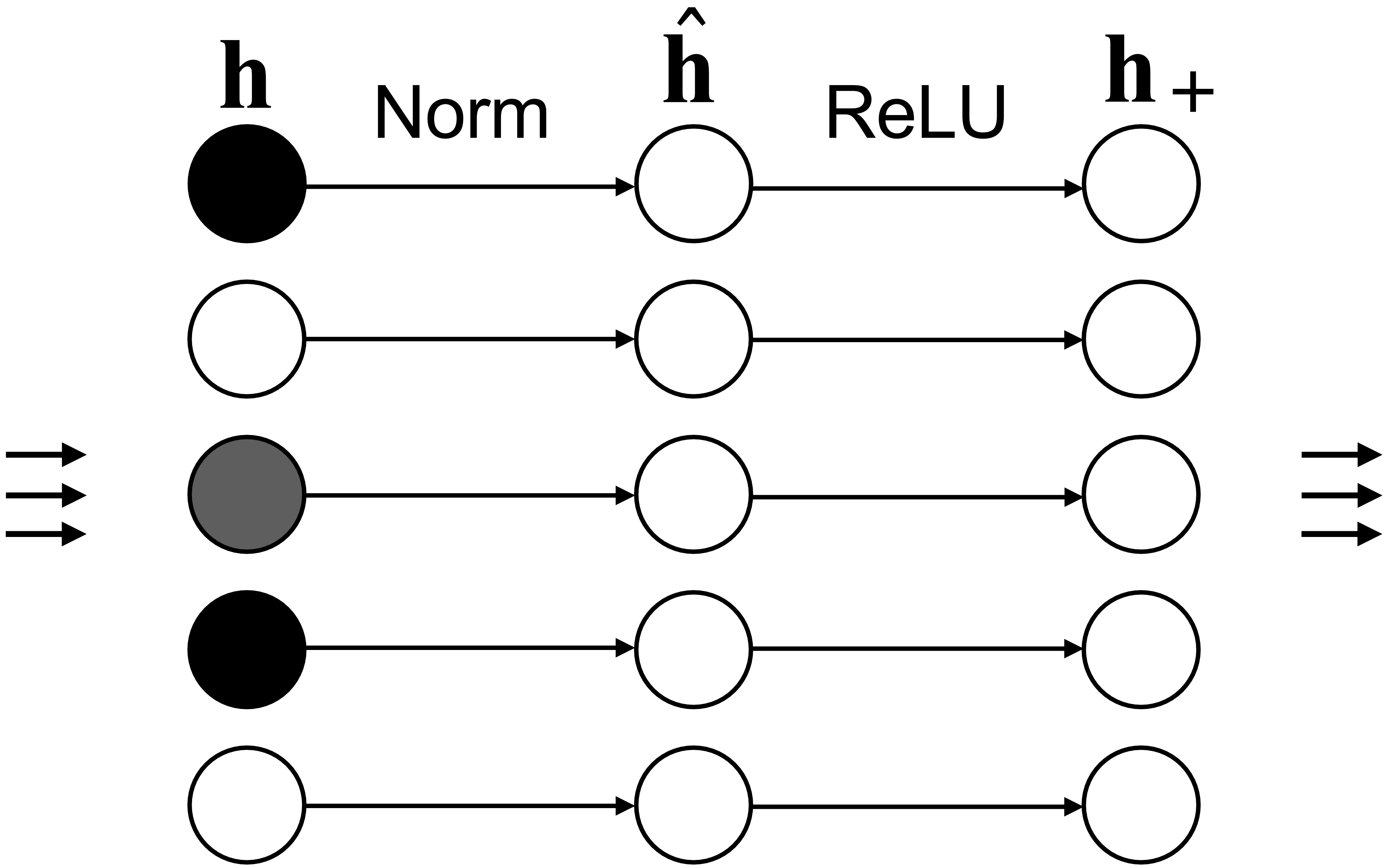
Normalization layers



$$\hat{h}_k = \frac{h_k - \mathbb{E}[h_k]}{\sqrt{\mathtt{Var}[h_k]}}$$

Source: Isola, Torralba, Freeman

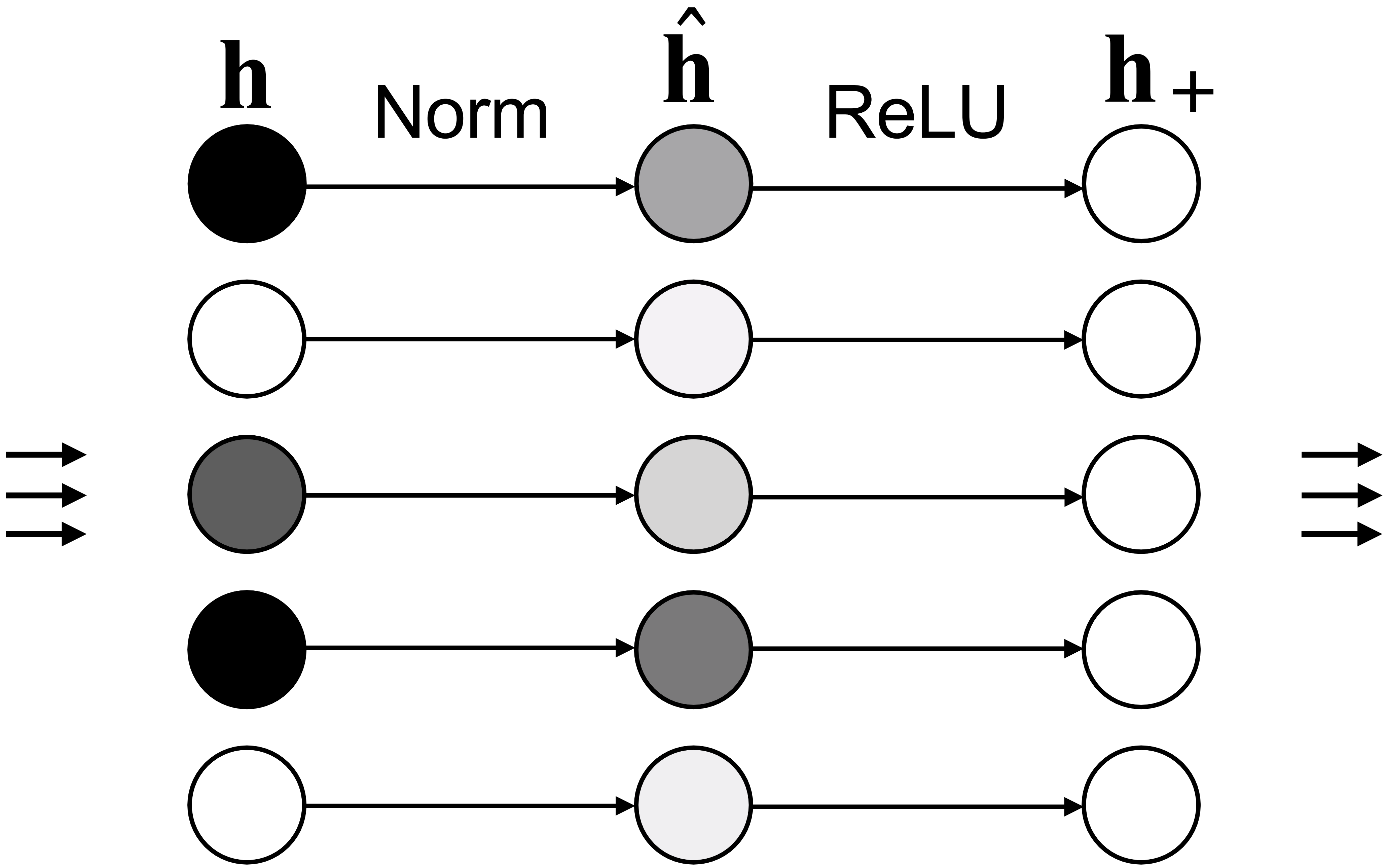
Normalization layers



$$\hat{h}_k = \frac{h_k - \mathbb{E}[h_k]}{\sqrt{\mathtt{Var}[h_k]}}$$

Source: Isola, Torralba, Freeman

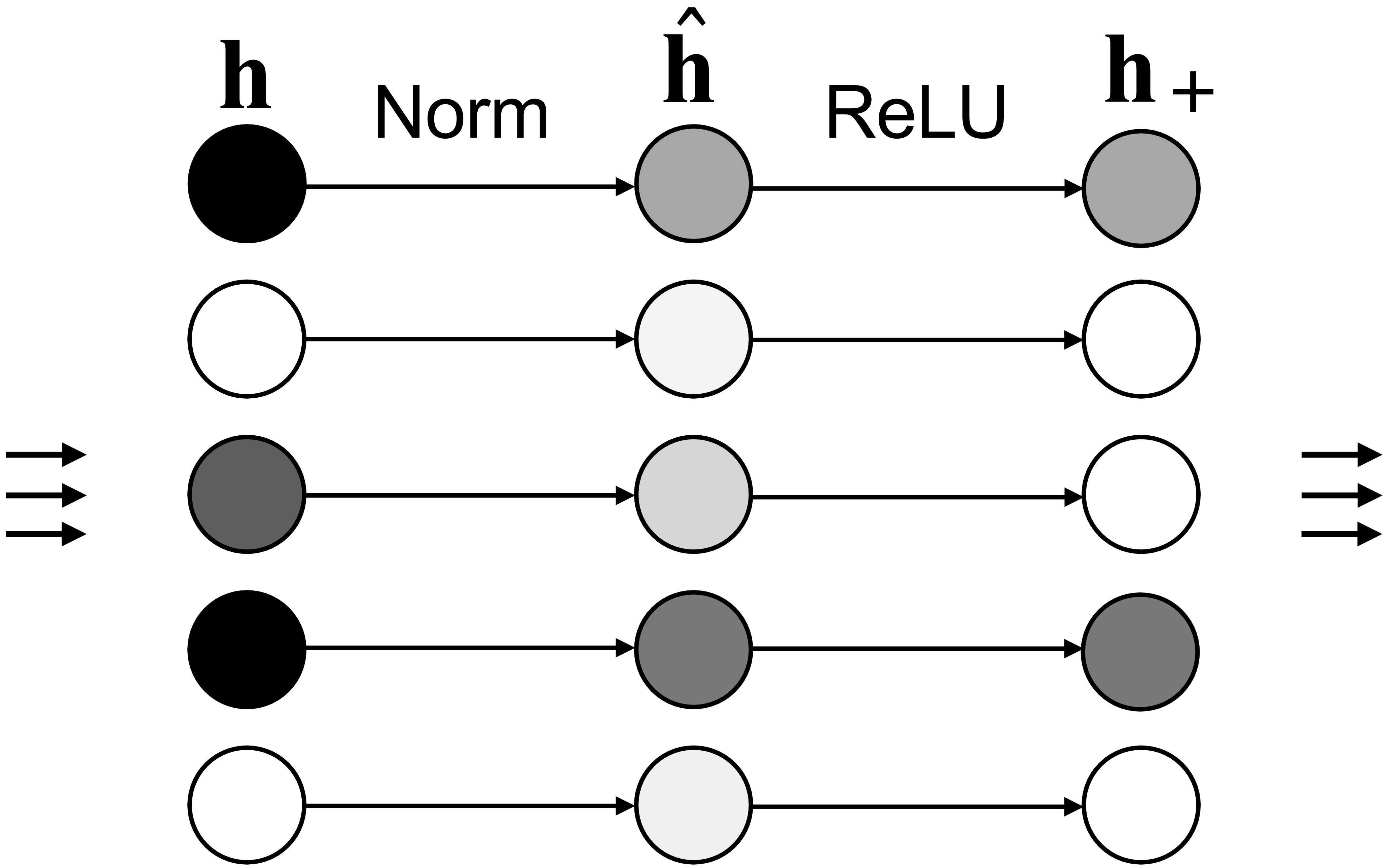
Normalization layers



$$\hat{h}_k = \frac{h_k - \mathbb{E}[h_k]}{\sqrt{\mathtt{Var}[h_k]}}$$

Source: Isola, Torralba, Freeman

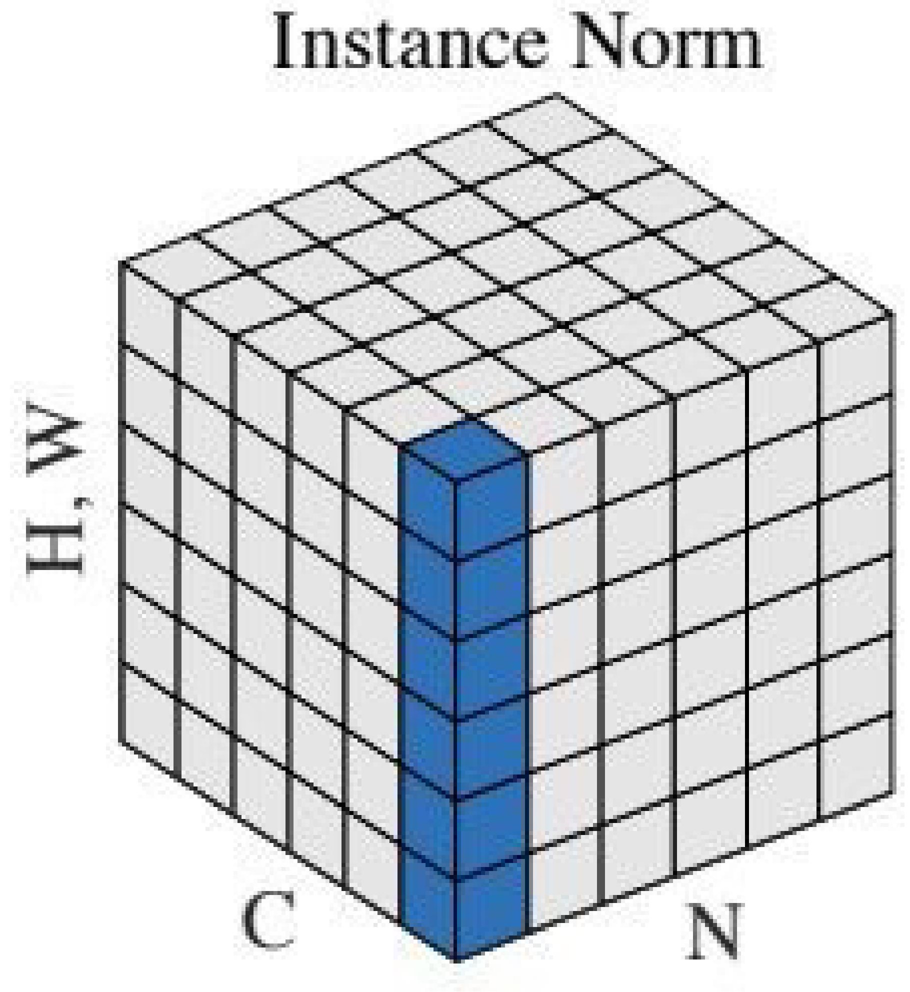
Normalization layers



$$\hat{h}_k = \frac{h_k - \mathbb{E}[h_k]}{\sqrt{\mathtt{Var}[h_k]}}$$

Source: Isola, Torralba, Freeman

Instance normalization



Filter value in a CNN layer

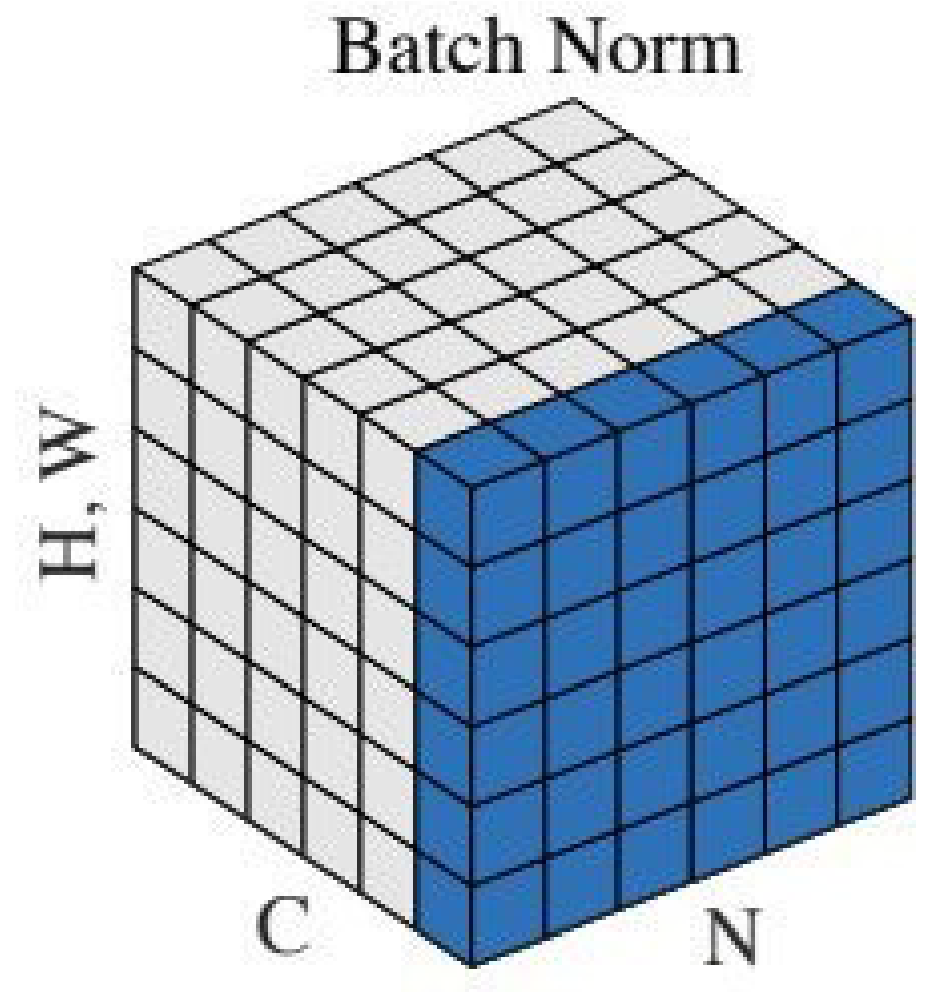
Average and standard deviation each filter response, estimated over **instance**.

$$\hat{h}_k = \frac{h_k - \mathbb{E}[h_k]}{\sqrt{\texttt{Var}[h_k]}}$$

Normalize a single hidden unit's activations to be mean 0, standard deviation 1.

[Figure from Wu & He, arXiv 2018] [Ulyanov et al., 2015]

Batch normalization



Filter value in a CNN layer

Average and standard deviation each filter response, estimated over **whole batch**.

$$\hat{h}_k = \frac{h_k - \mathbb{E}[h_k]}{\sqrt{\mathtt{Var}[h_k]}}$$

Normalize a single hidden unit's activations to be mean 0, standard deviation 1.

At test time, remember the mean and standard deviation seen during training.

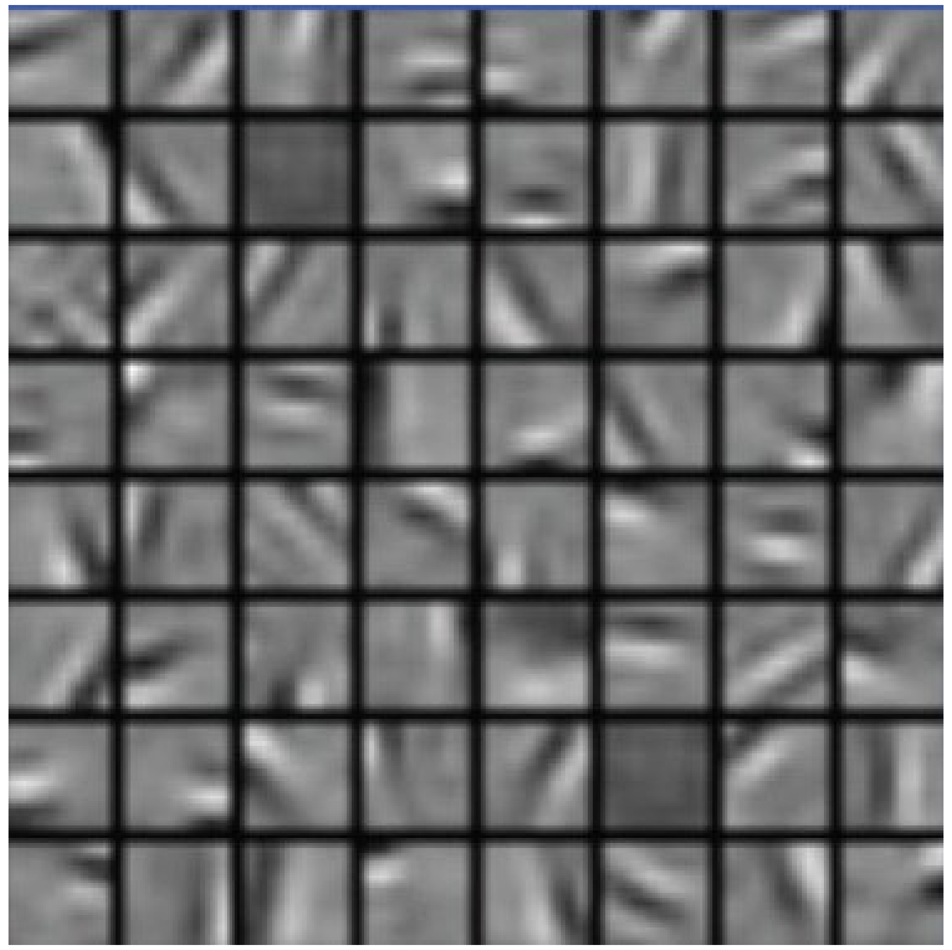
Can allow you to train with larger learning rate and significantly speed up training!

[Figure from Wu & He, arXiv 2018] [Ioffe & Szegedy, 2015]

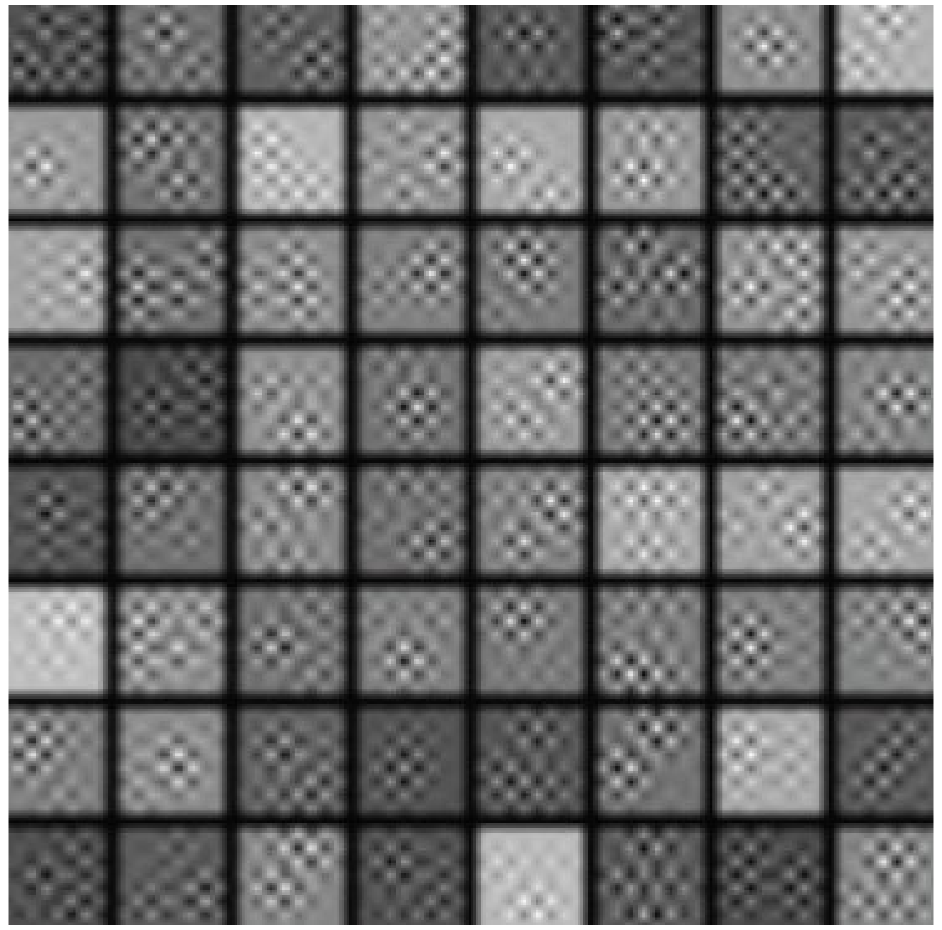
What filters are learned?

What filters are learned?

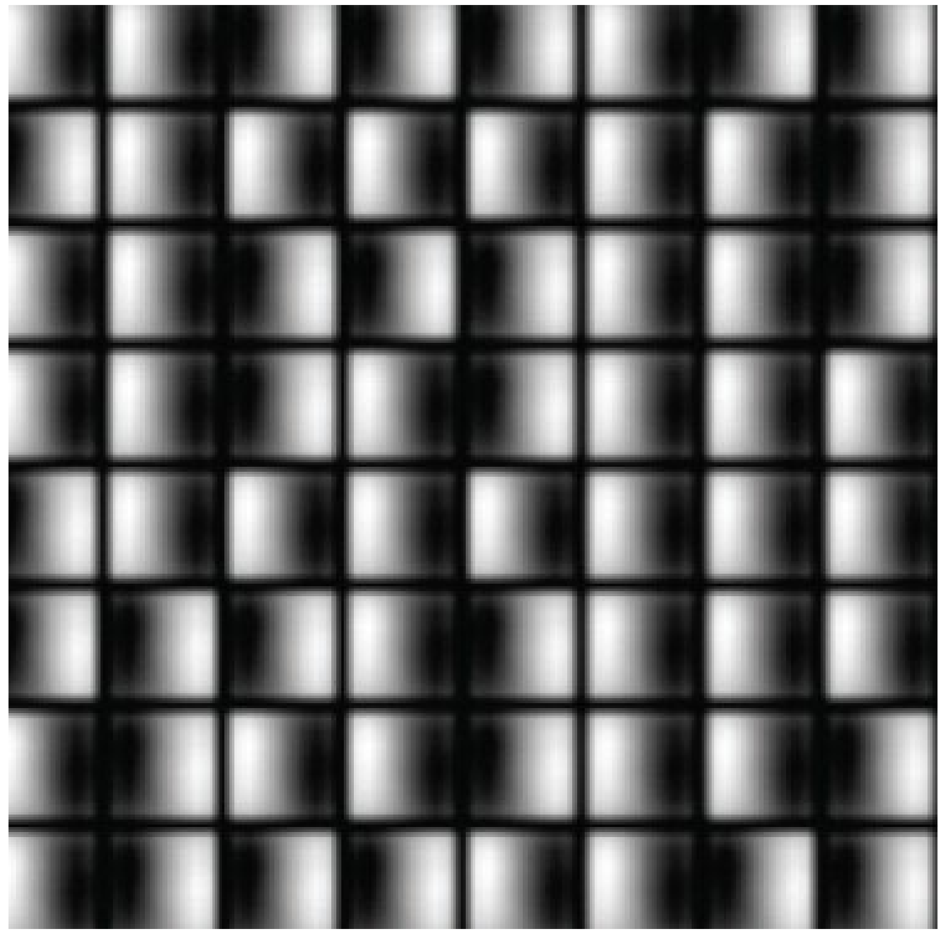
A



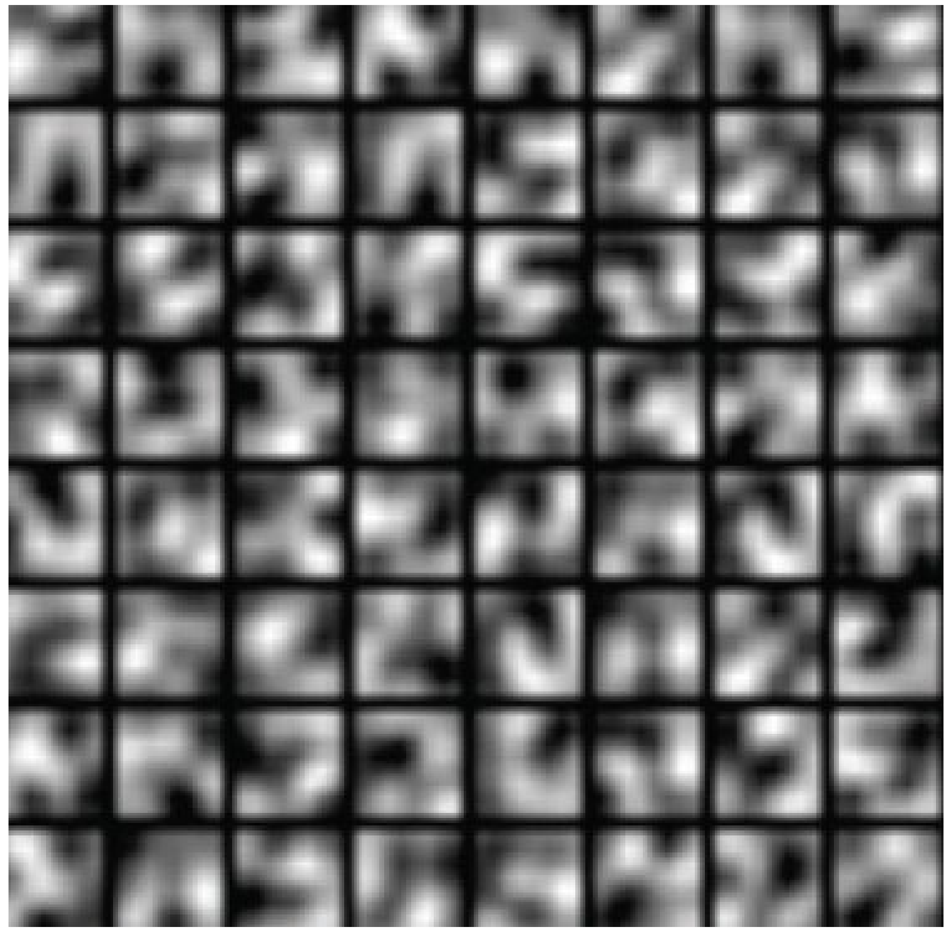
B



C

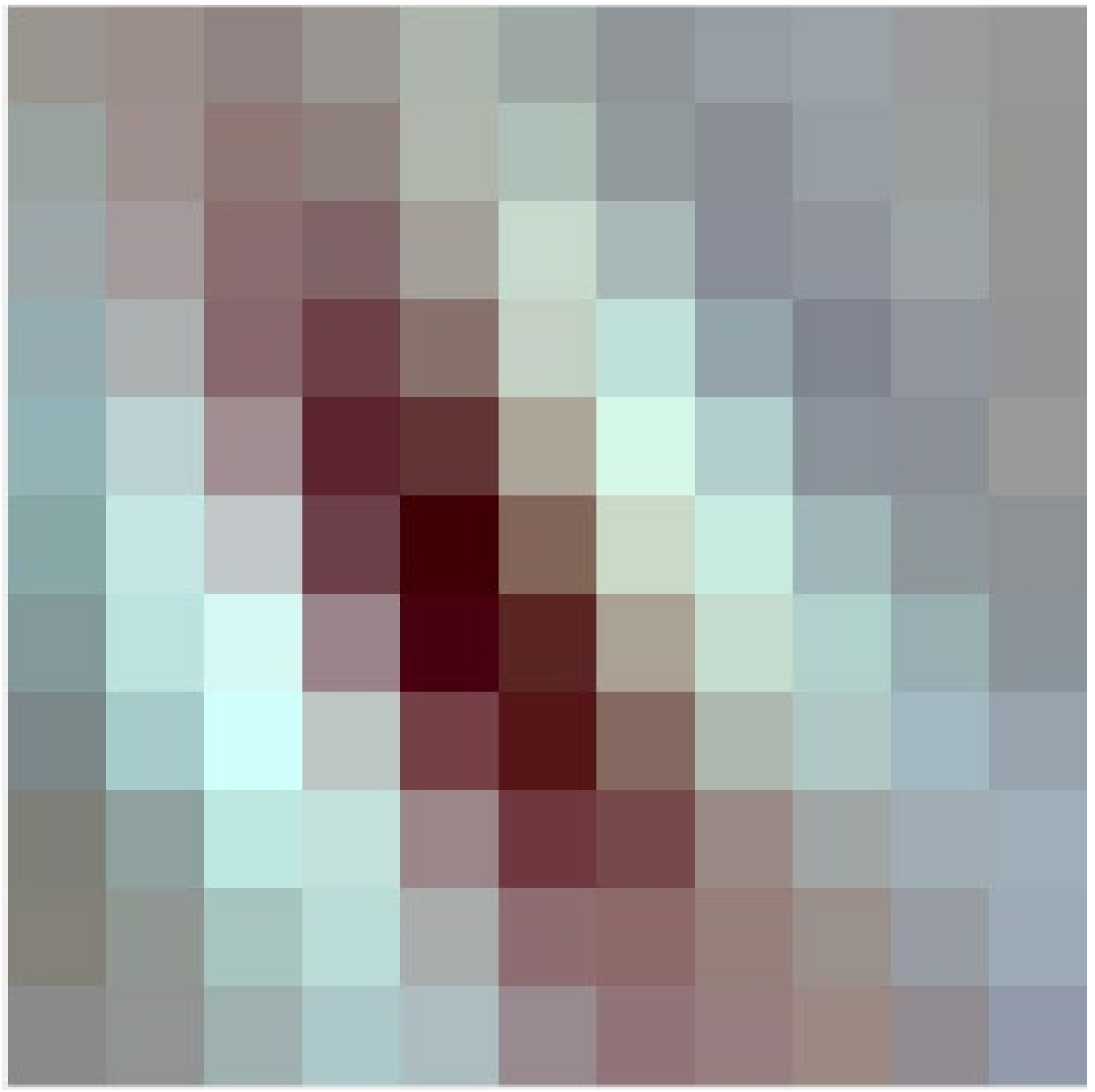


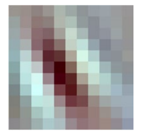
D

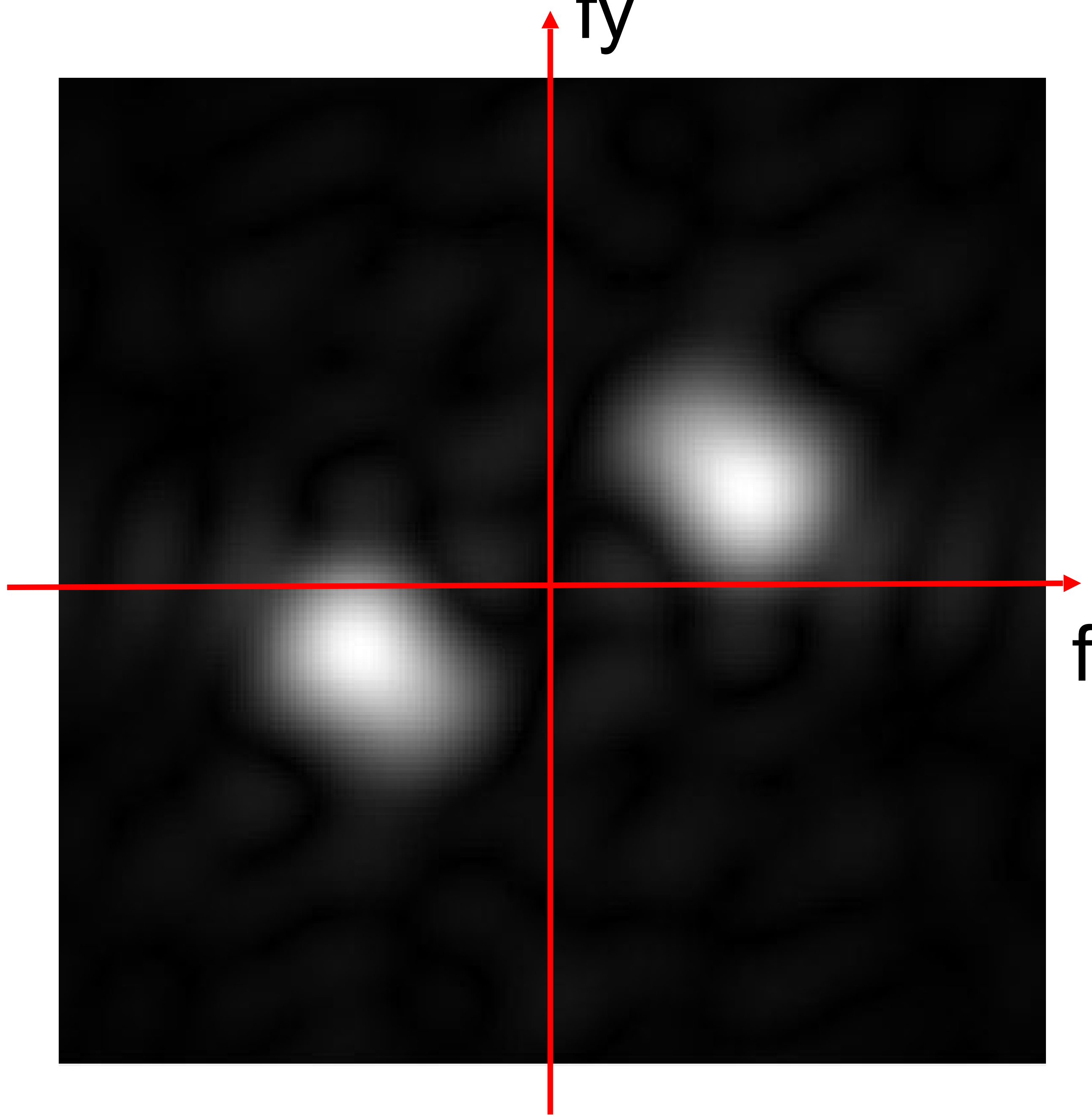


Source: Marc'Aurelio Ranzato & Isola, Torralba, Freeman

Filters in first layer



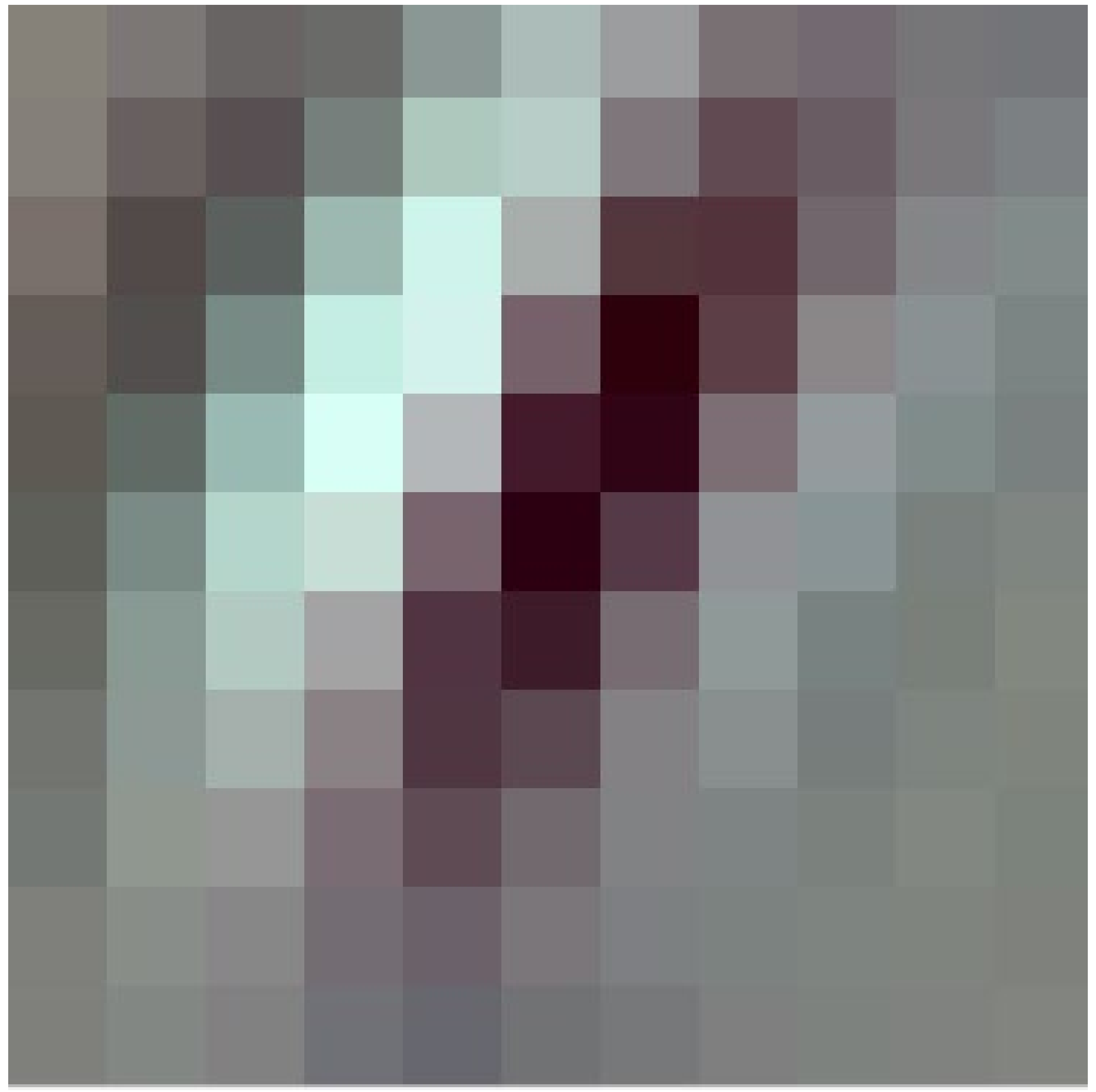


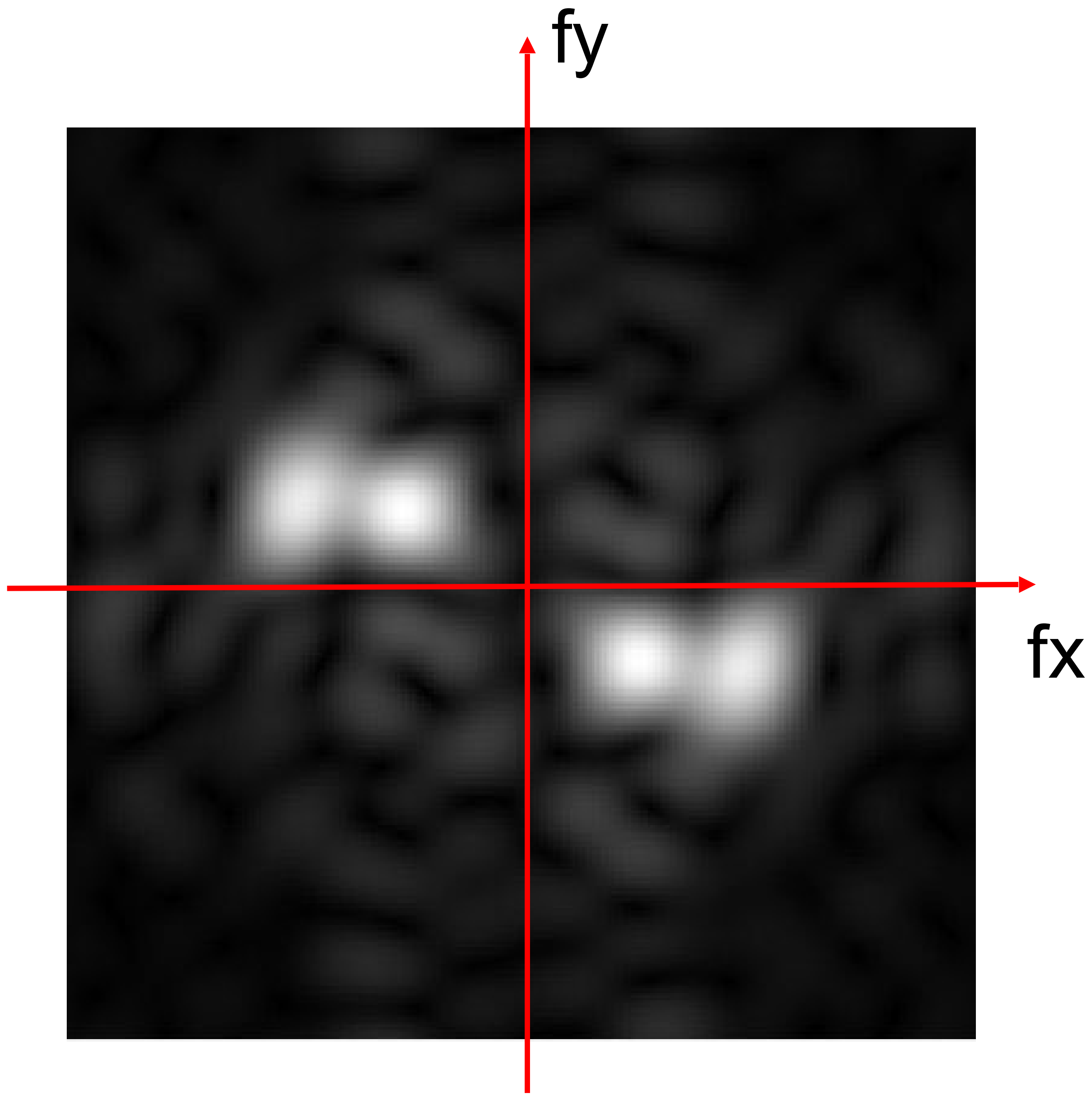


11x11 convolution kernel
(3 color channels)

Source: Isola, Torralba, Freeman

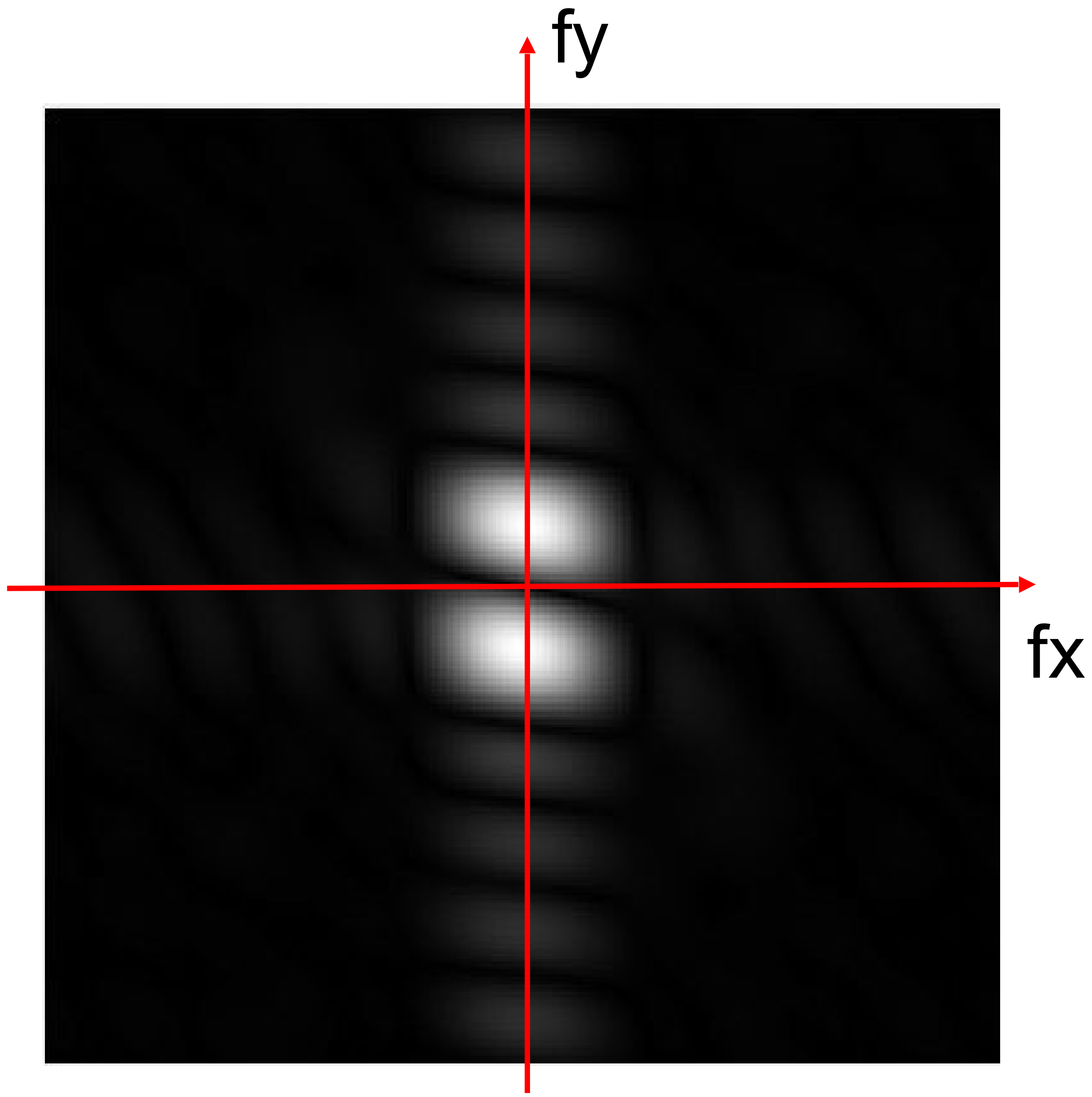
Filters in first layer





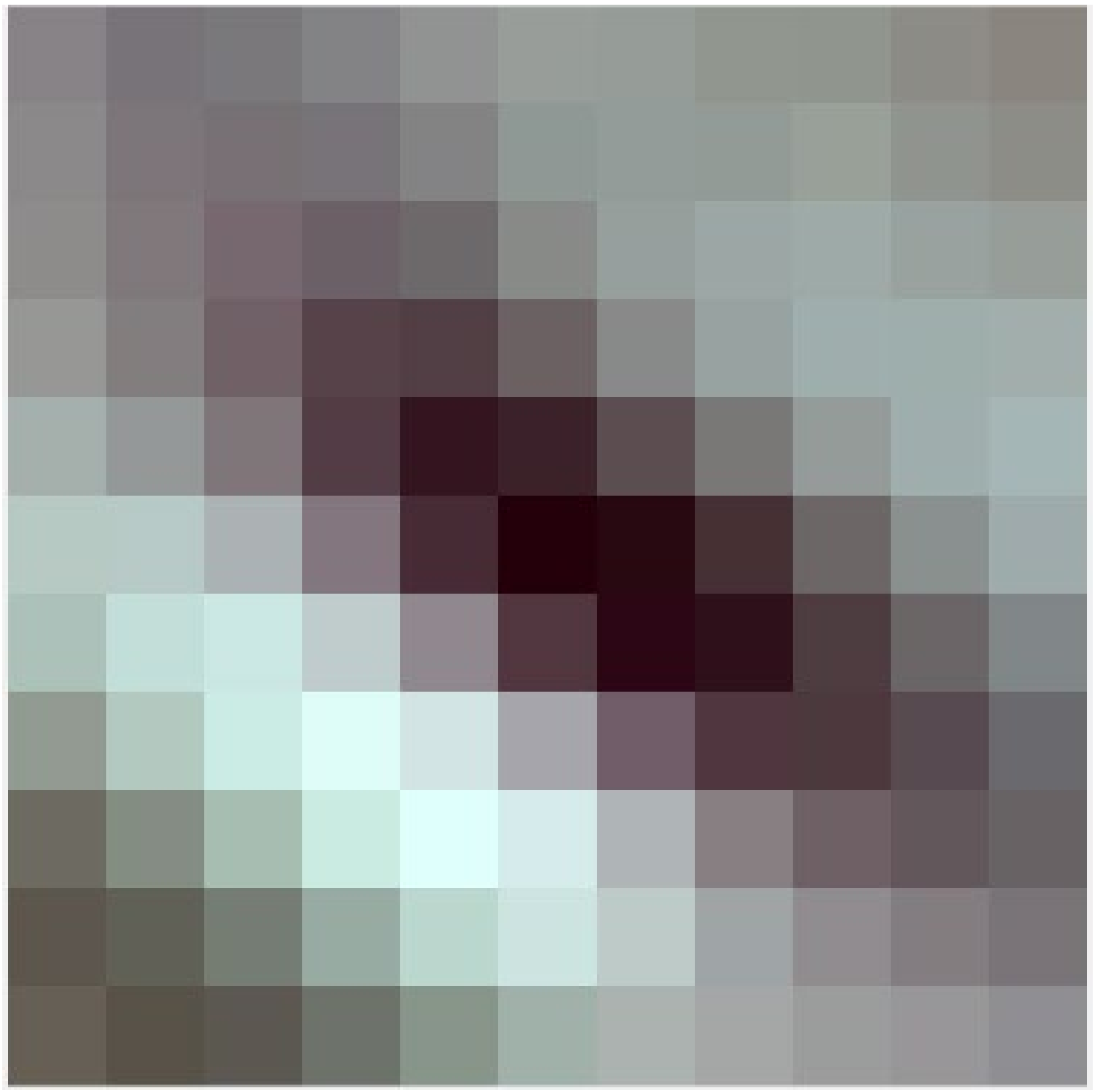
Source: Isola, Torralba, Freeman

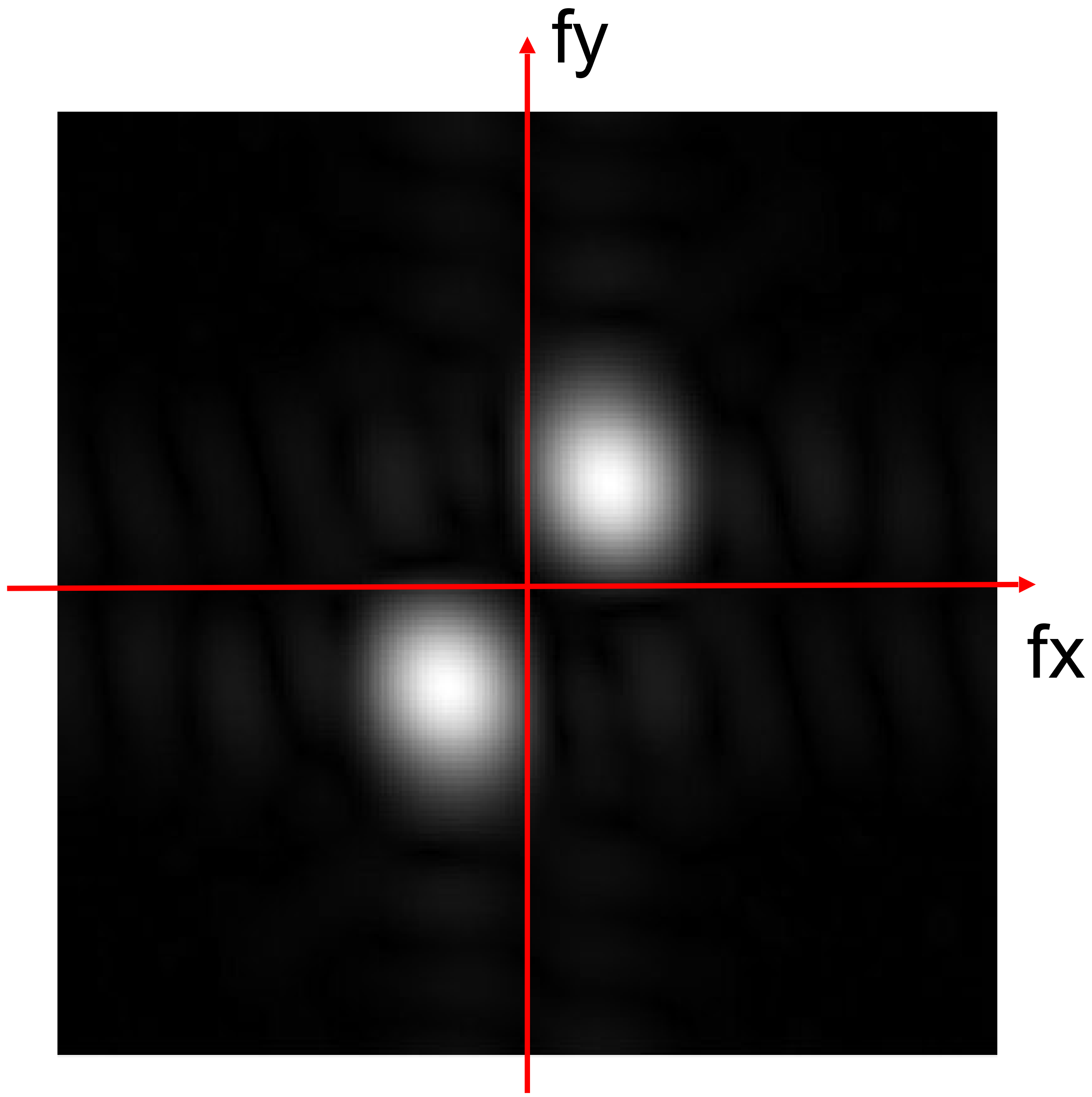
Filters in first layer



Source: Isola, Torralba, Freeman

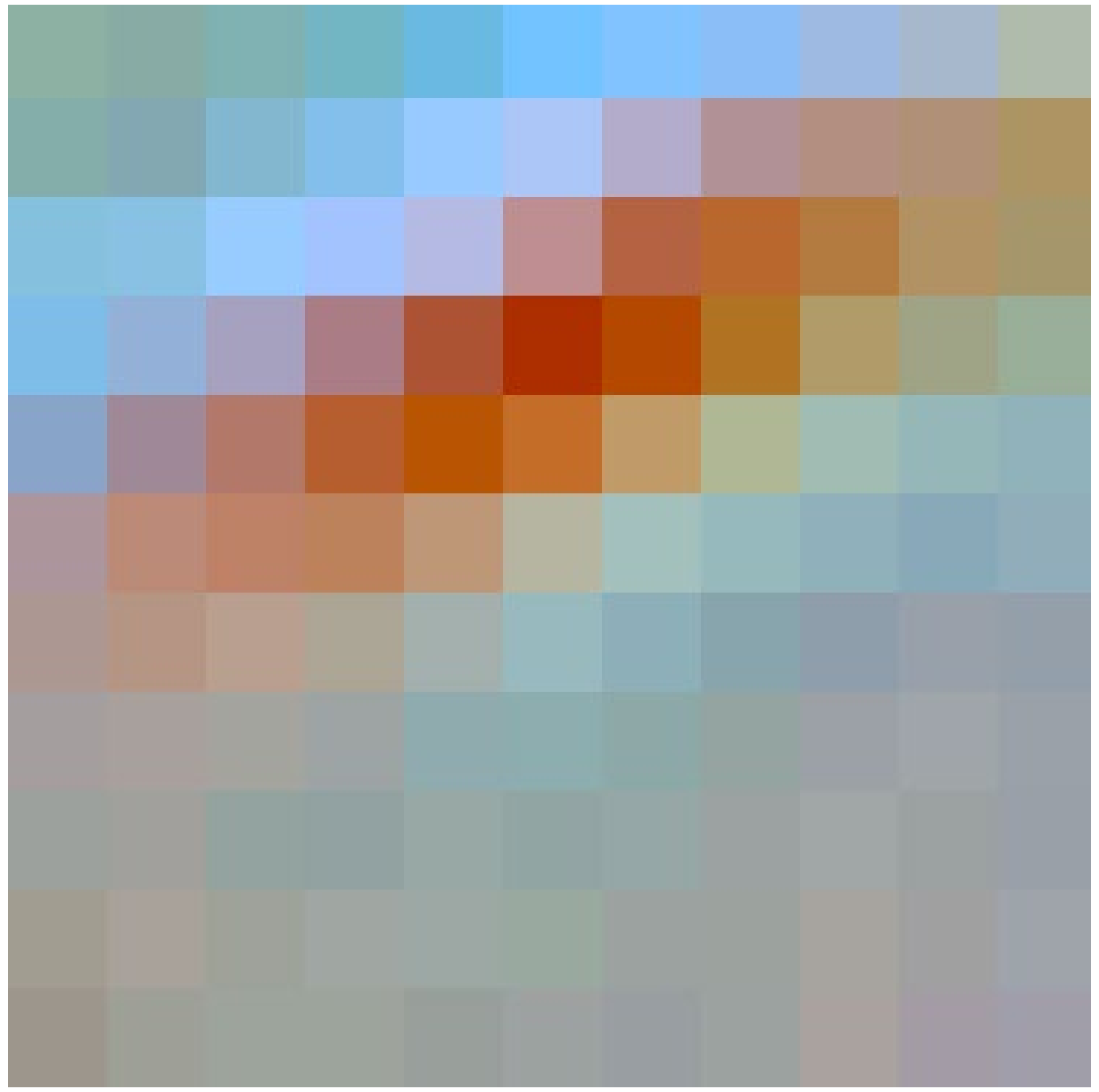
Filters in first layer

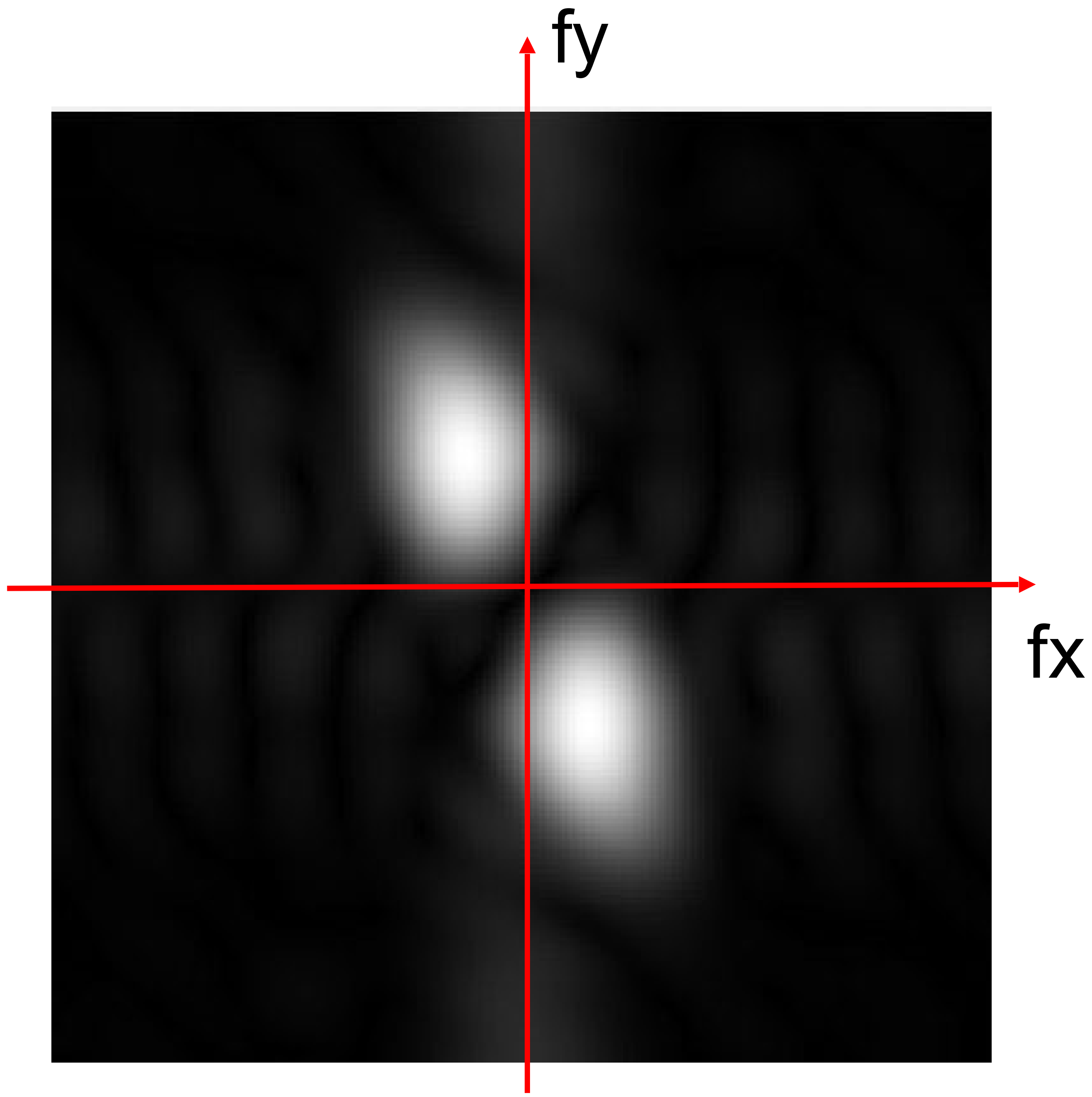




Source: Isola, Torralba, Freeman

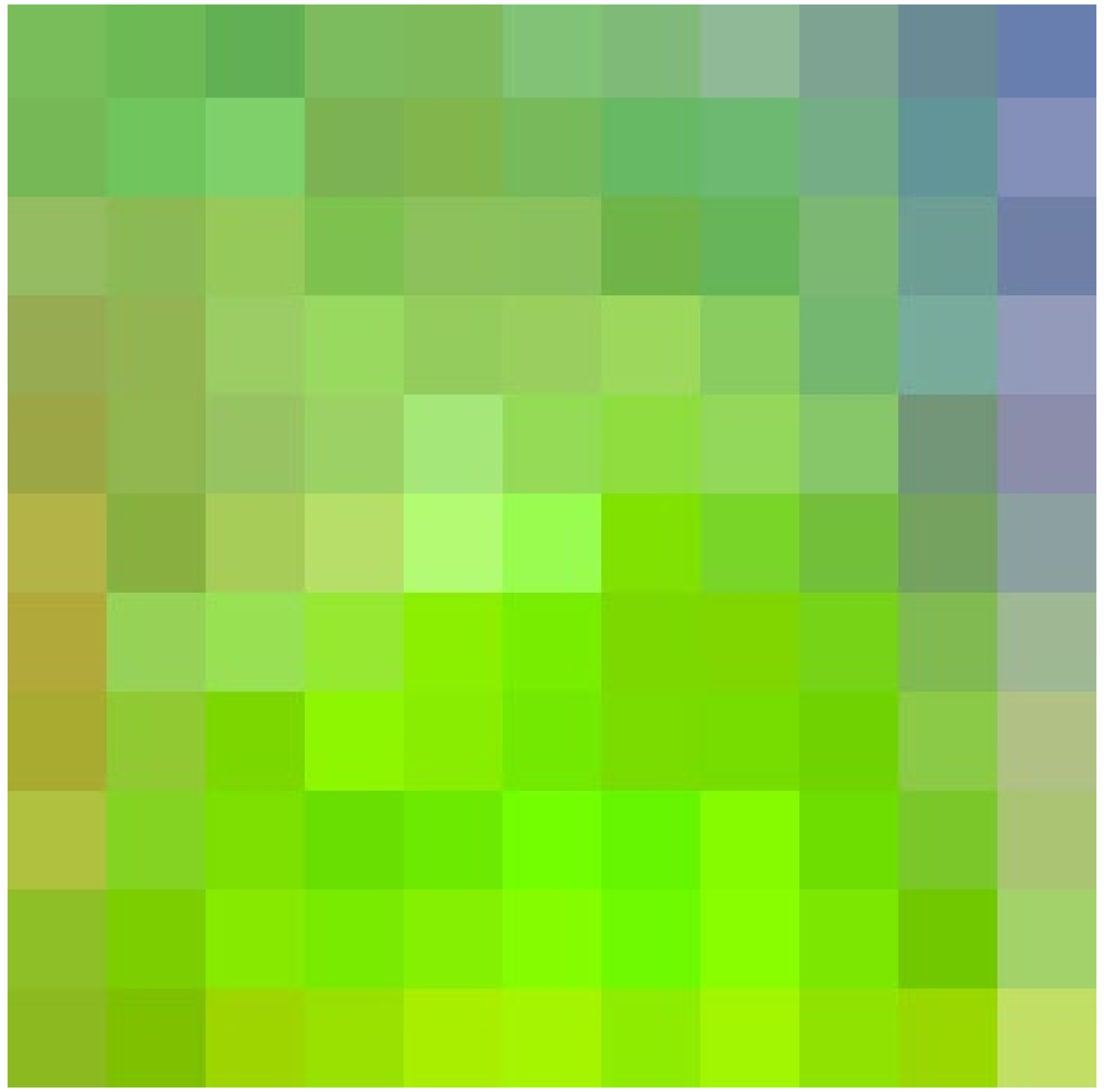
Filters in first layer

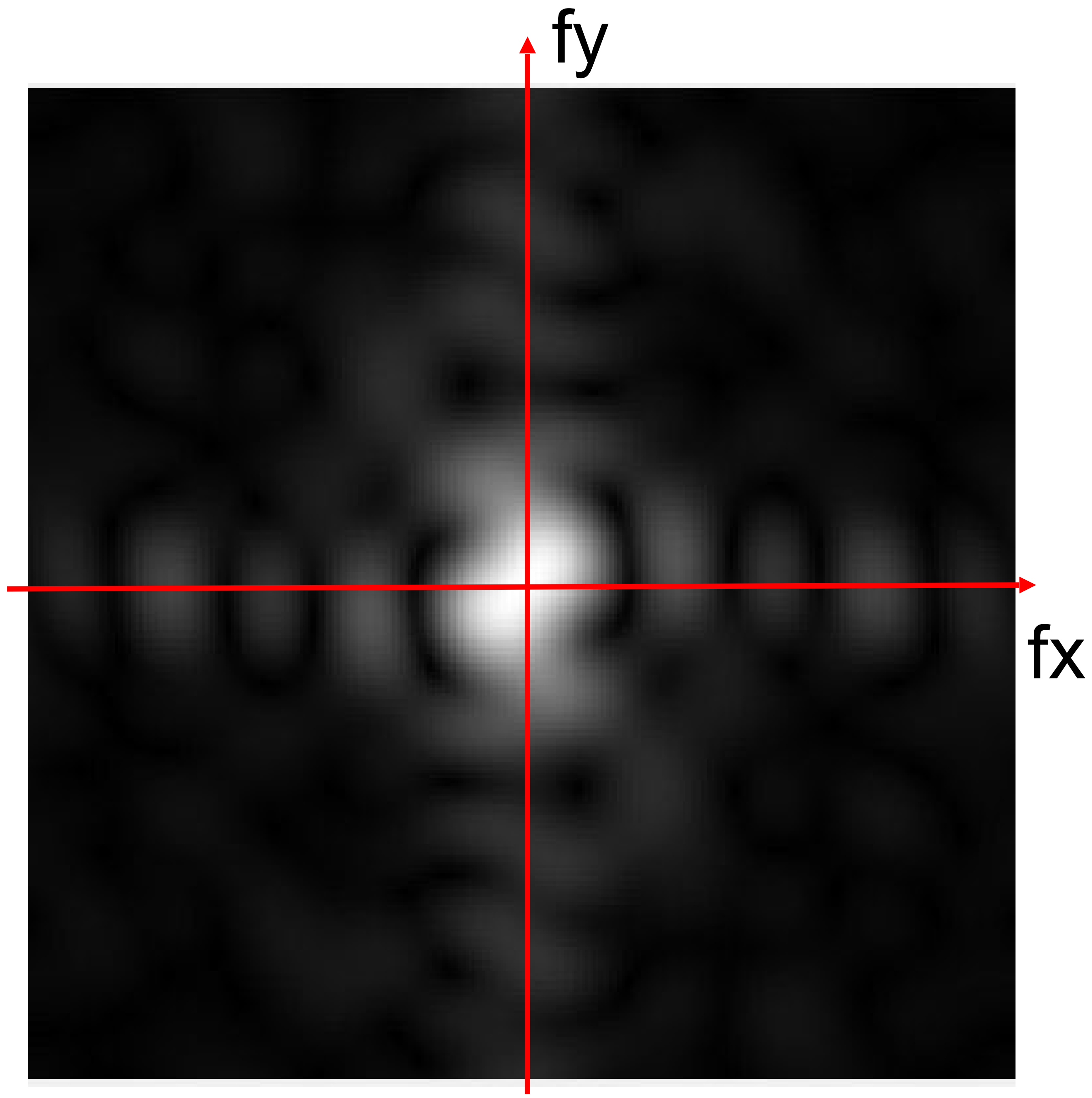




Source: Isola, Torralba, Freeman

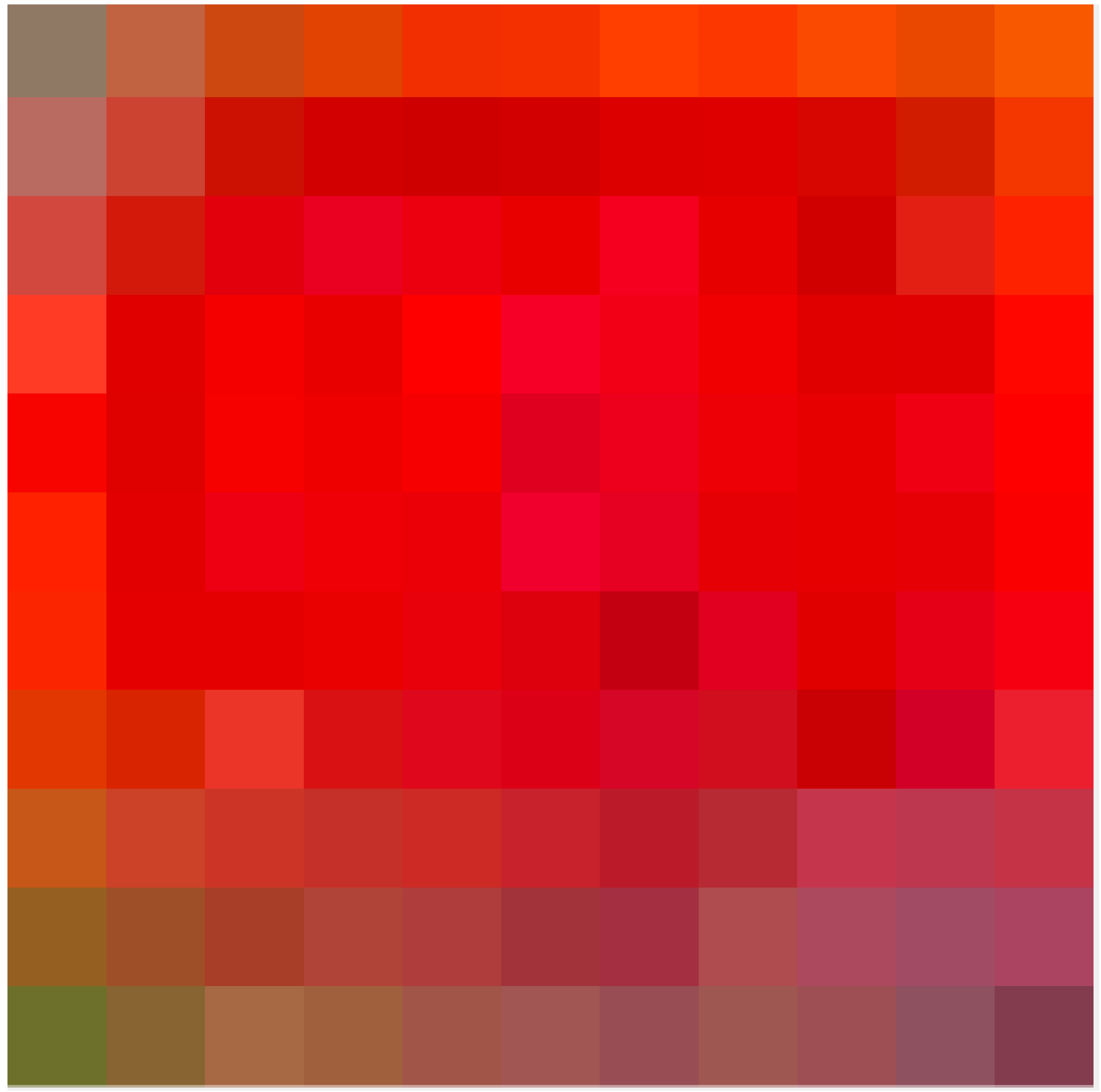
Filters in first layer

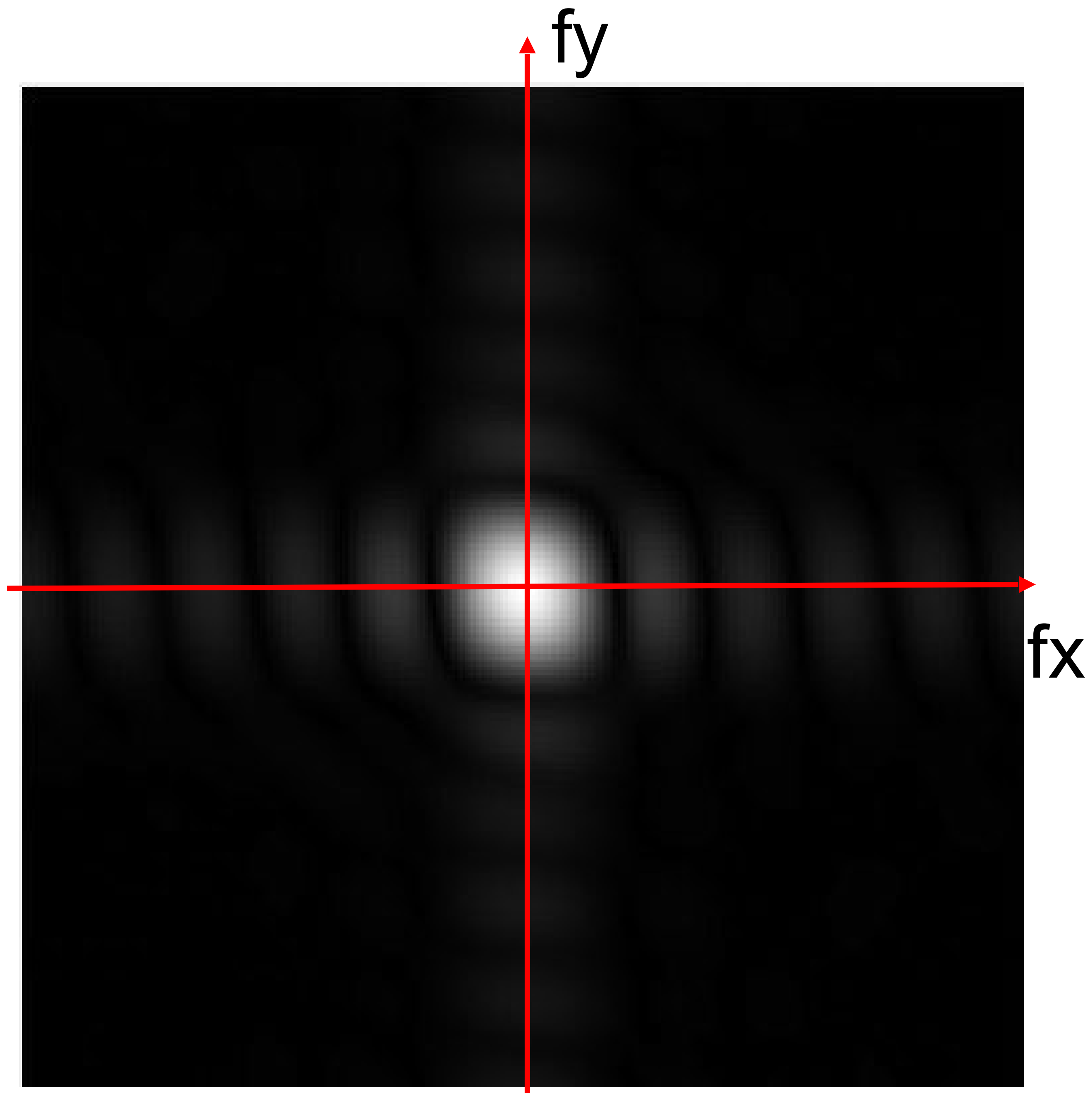




Source: Isola, Torralba, Freeman

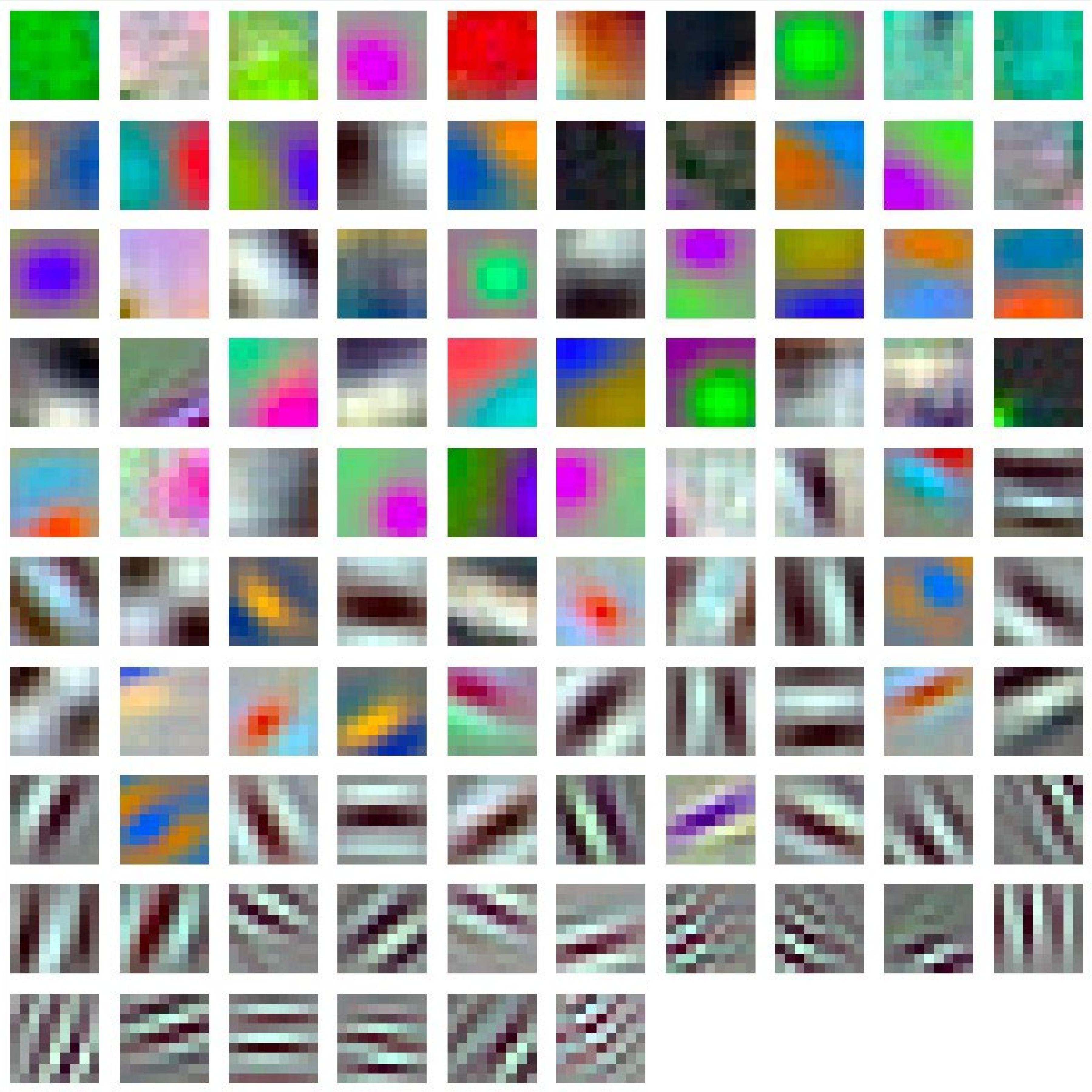
Filters in first layer

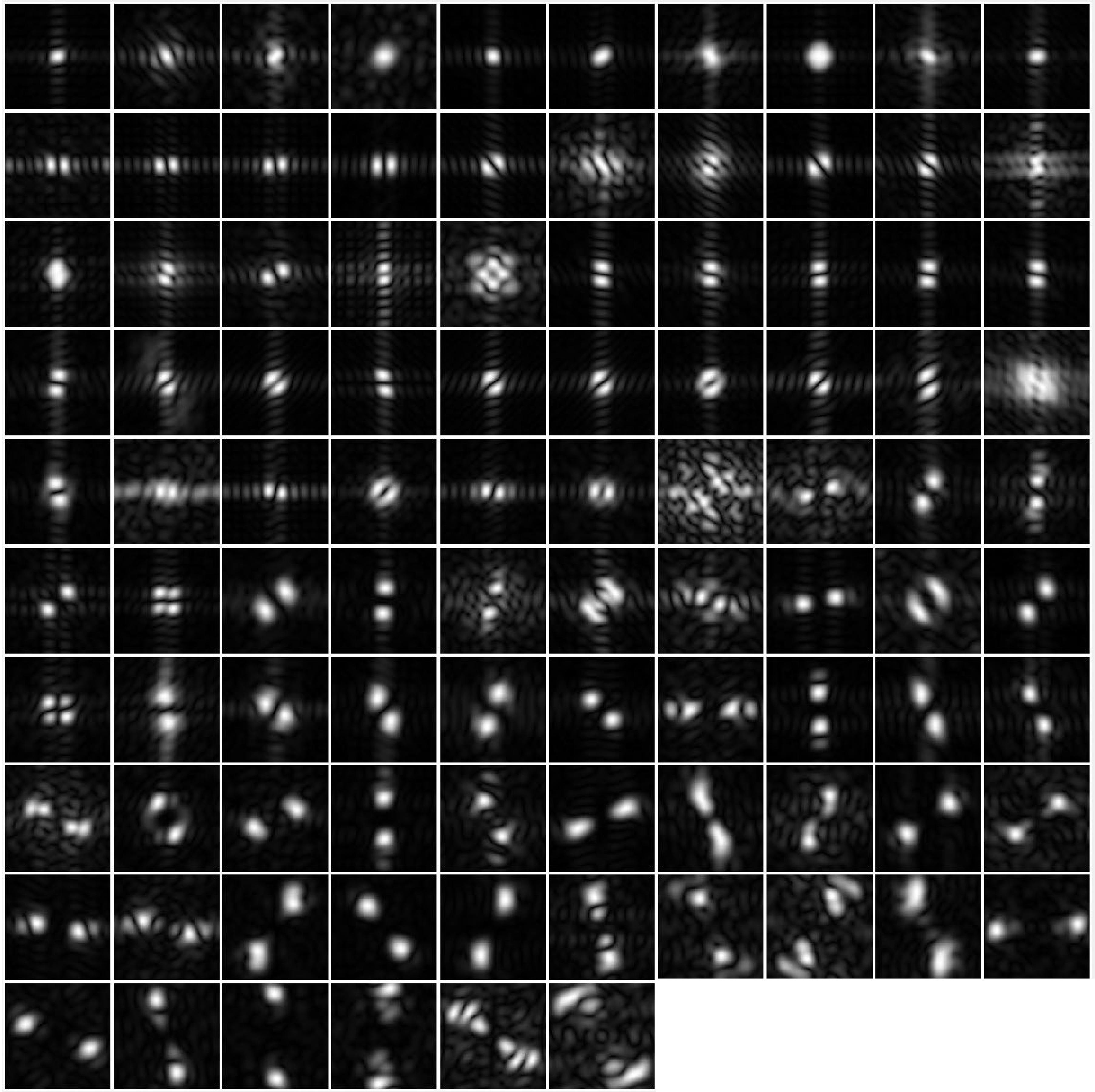




Source: Isola, Torralba, Freeman

Filters in first layer





96 Units in conv1

Source: Isola, Torralba, Freeman

A few practical issues

Dealing with rectangular images

Rectangular
images

Resize, then take square
crop from center

Training with data augmentation

Original image

Flipping

- Less susceptible to overfitting.
- Improves performance by simulating examples.

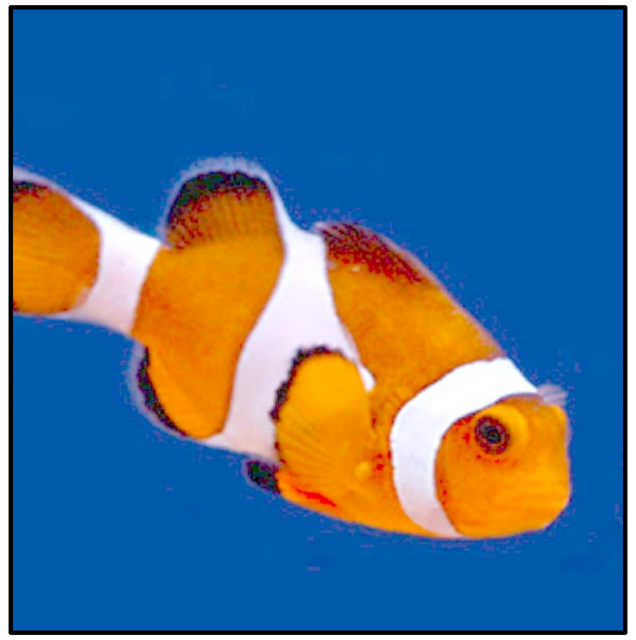
Training with data augmentation

Original image

Flipping

Cropping

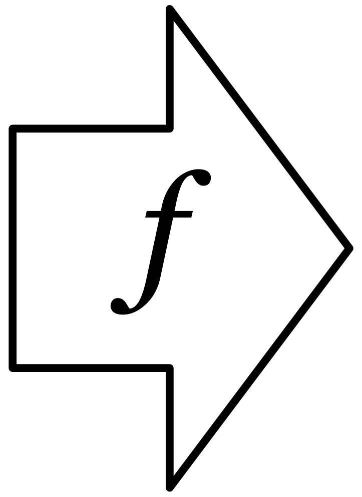
Scaling



Color jittering

Beyond image labeling

Object recognition: what objects are in the image?

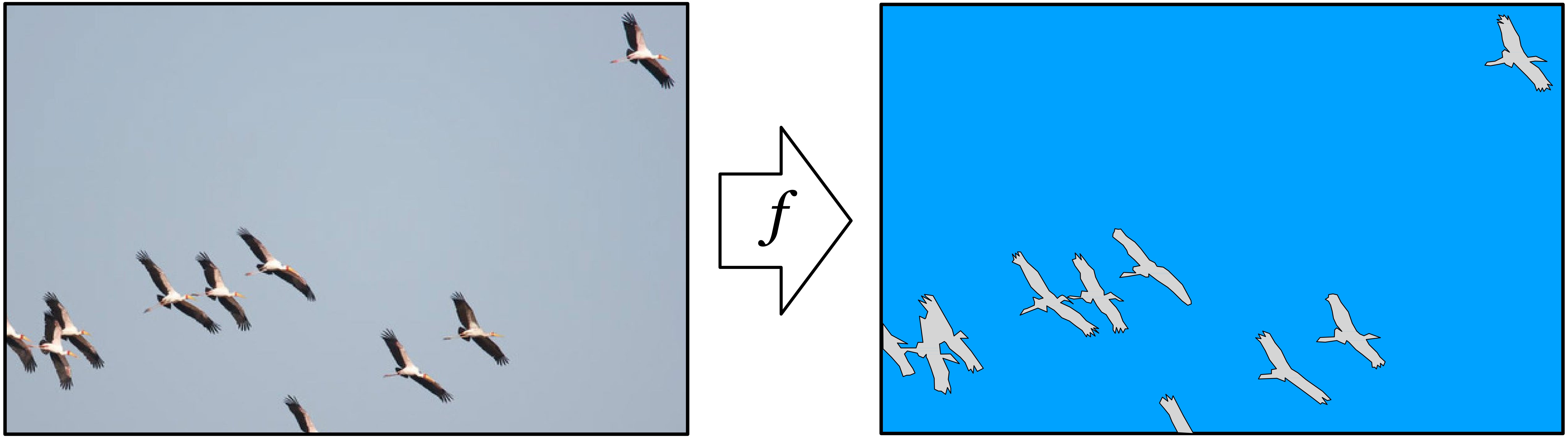


f

"Birds"

Source: Torralba, Freeman, Isola

Semantic segmentation

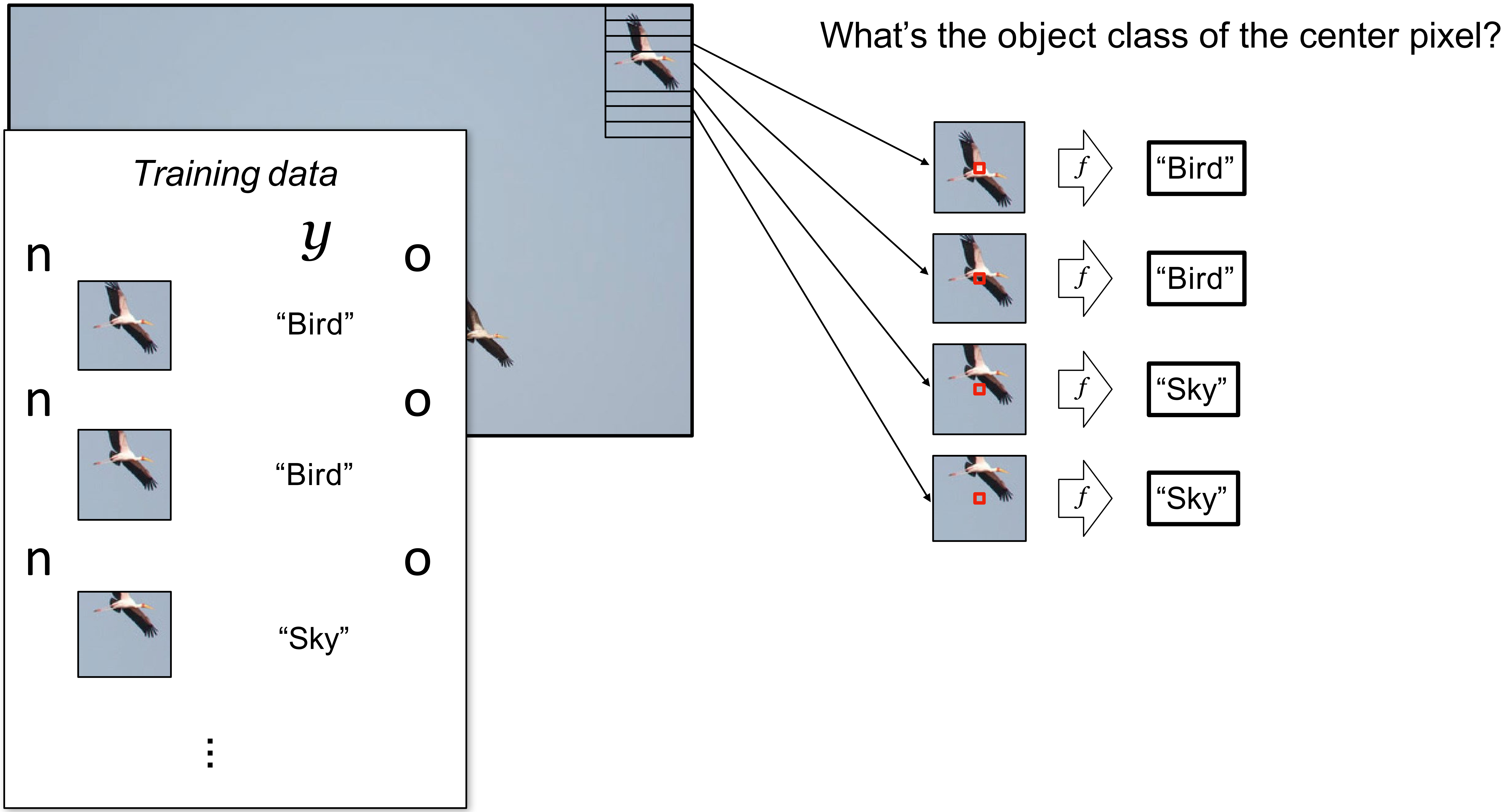


(Colors represent categories)

General technique: predict something at every pixel!

Source: Torralba, Freeman, Isola

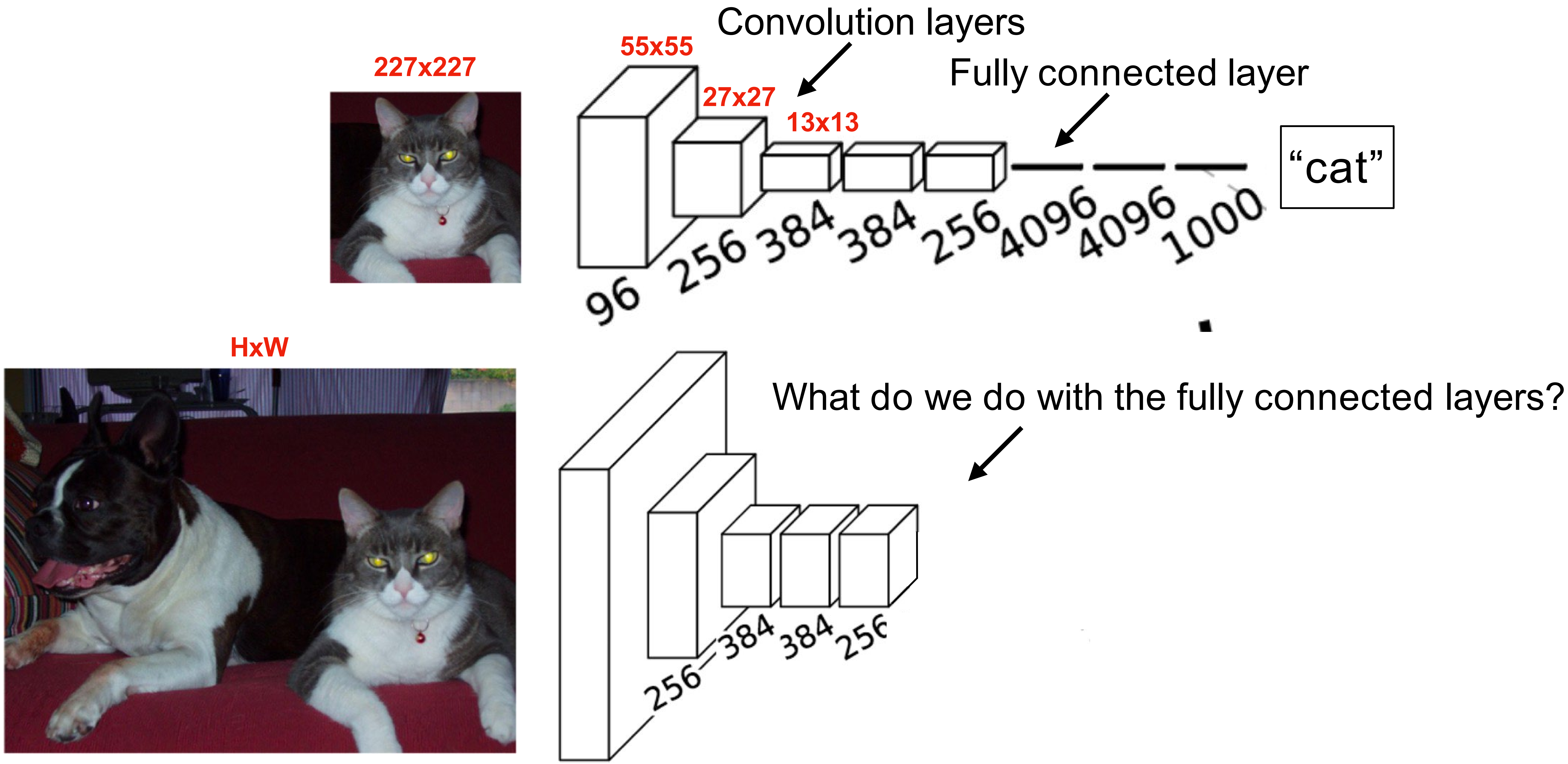
Idea #1: Independently classify windows



Source: Torralba, Freeman, Isola

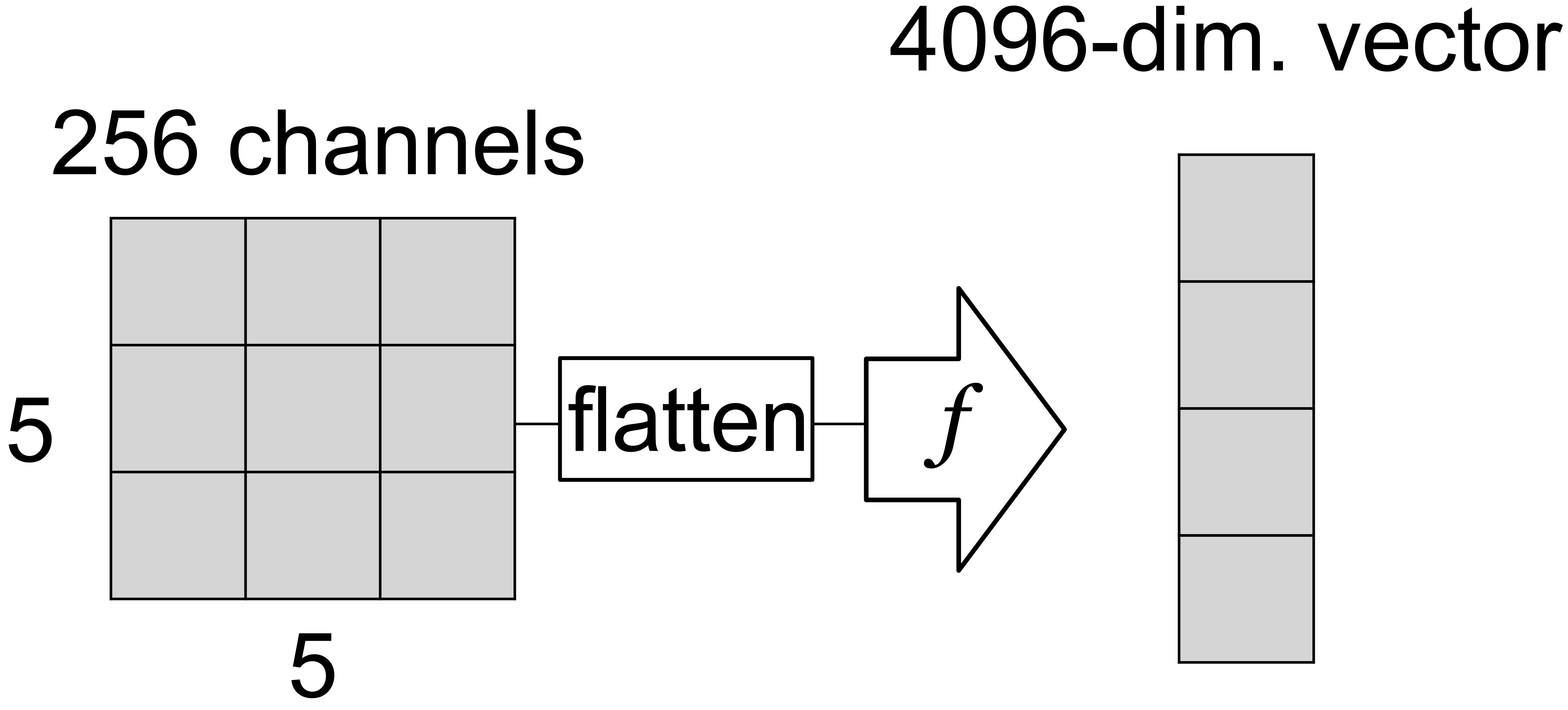
Idea #2: Fully convolutional networks

Reuse features across windows



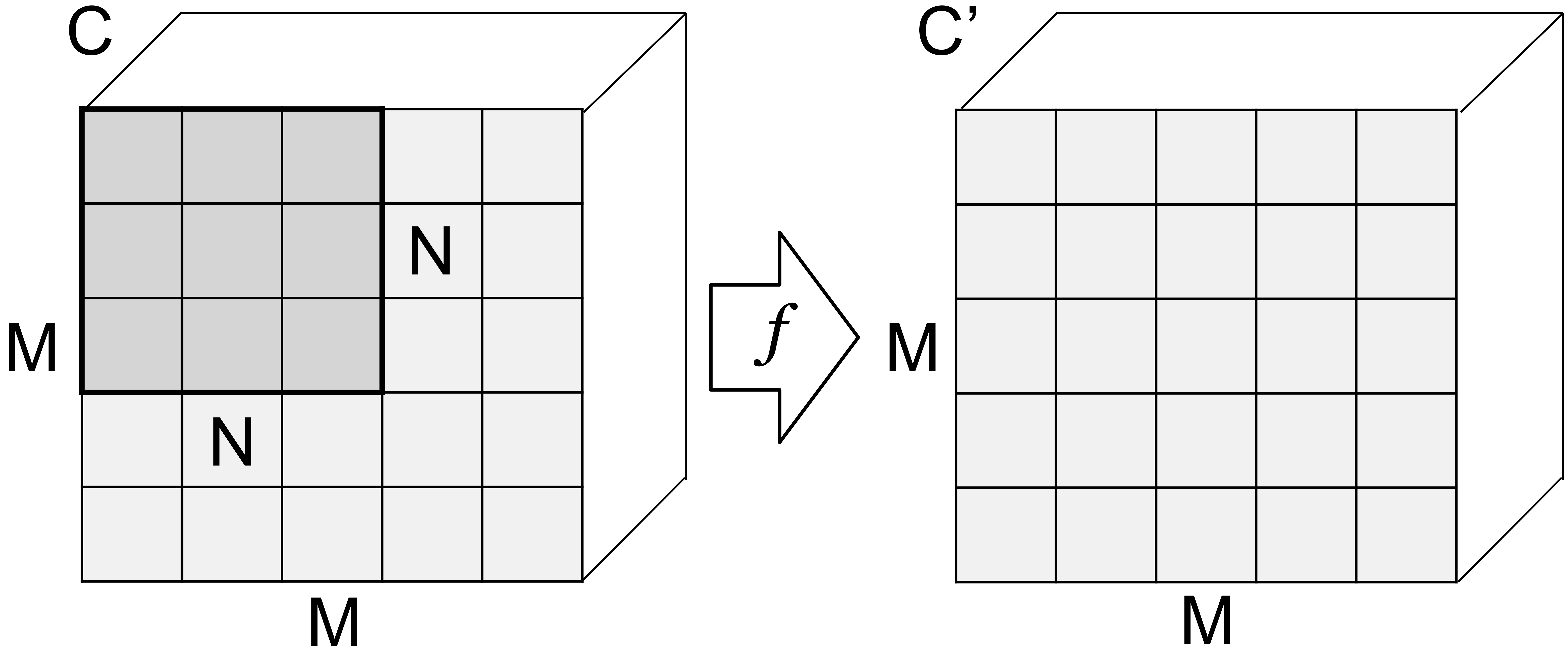
[Shelhamer, Long, Darrell, "Fully convolutional networks for semantic segmentation", CVPR 2015]

Converting fully connected layer to convolution



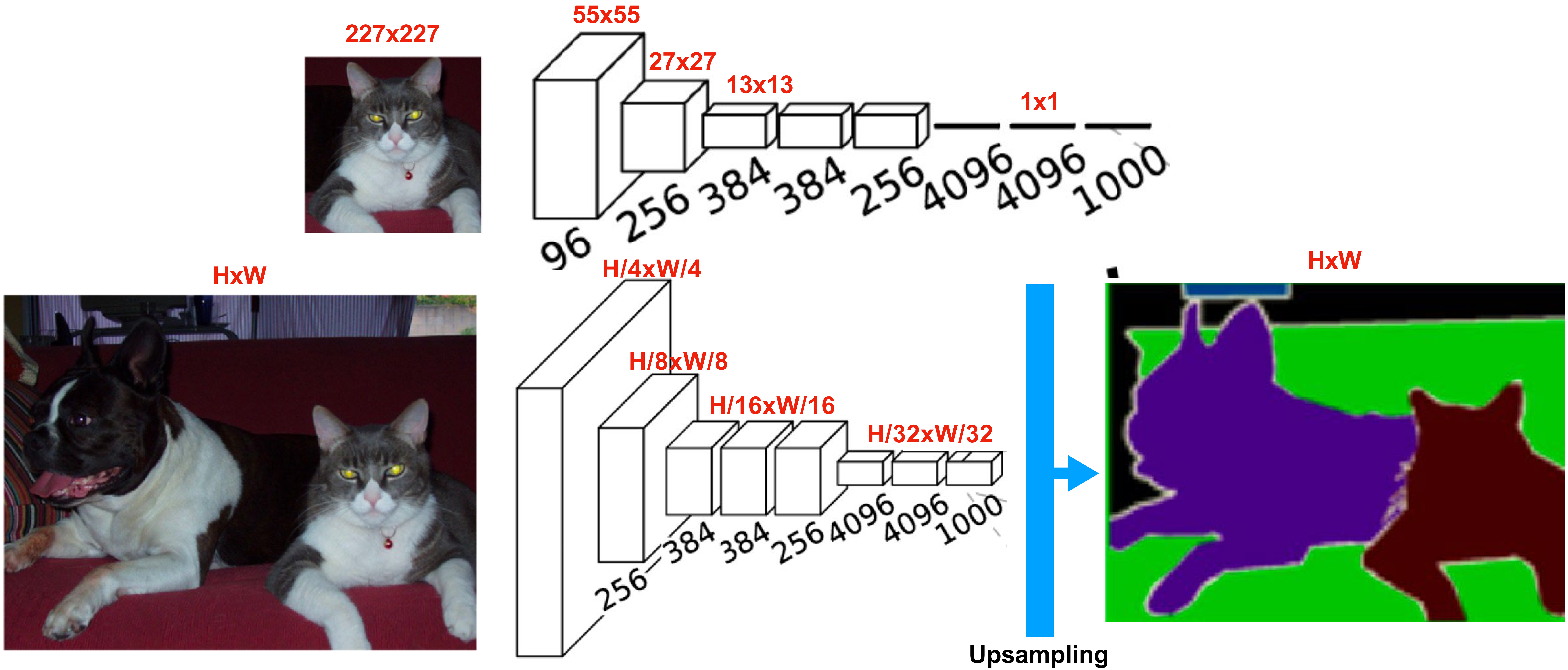
Fully connected layer: $\mathbb{R}^{N \times N \times C \to C'}$

Converting fully connected layer to convolution



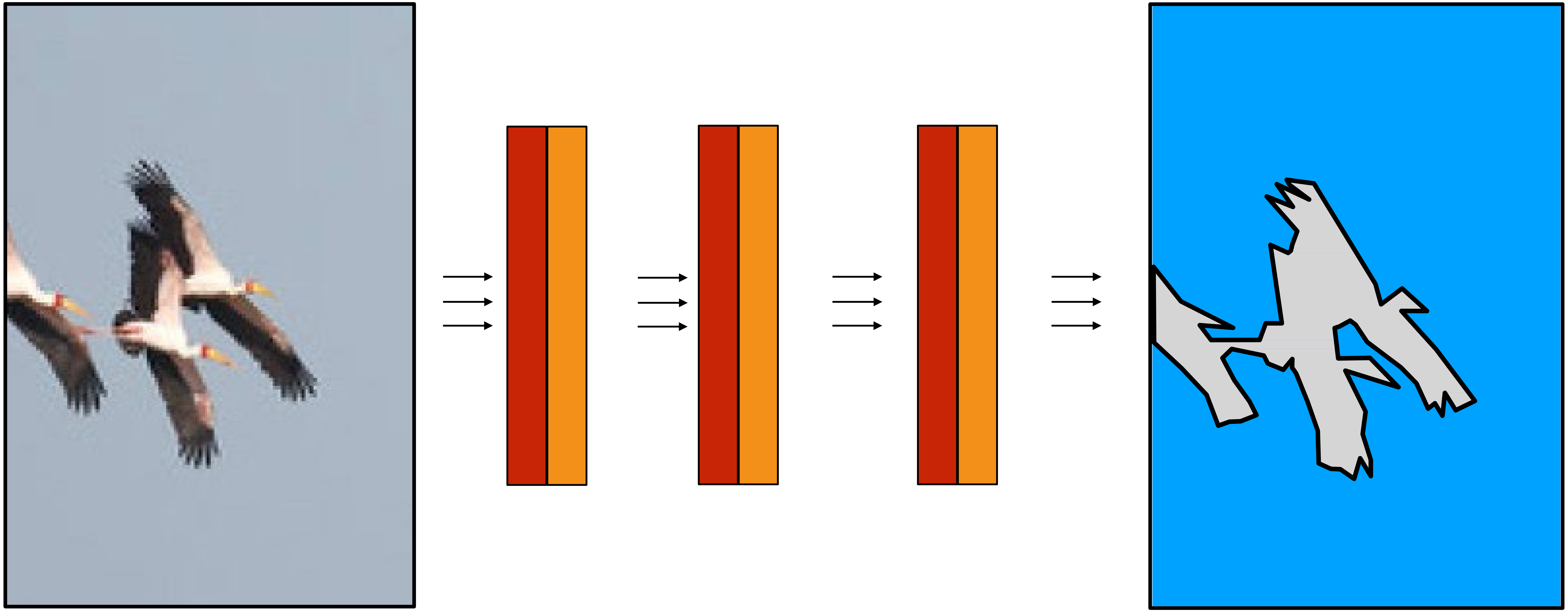
Convert into an $N \times N$ convolution!

Fully Convolutional Networks



Source: Isola, Torralba, Freeman

What if we remove subsampling layers?



Problems: small receptive fields (and expensive)

Idea #3: Dilated convolutions

Dilated convolutions

3x3

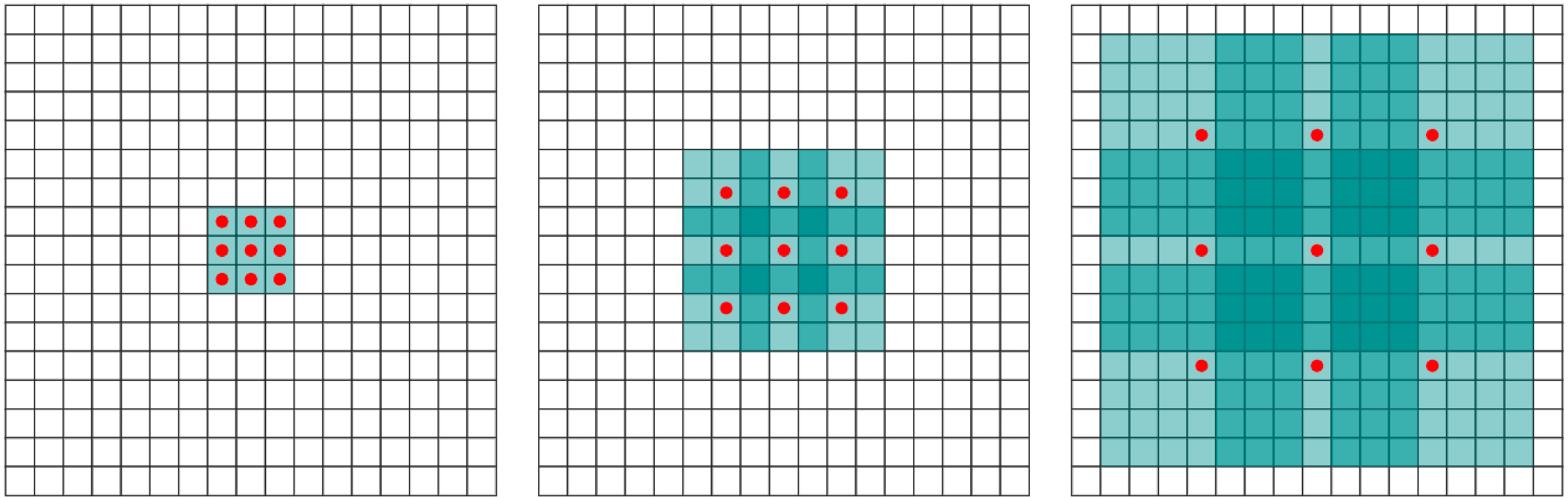
a	**b**	**c**
d	**e**	**f**
g	**h**	**i**

5x5

a	0	**b**	0	**c**
0	0	0	0	0
d	0	**e**	0	**f**
0	0	0	0	0
g	0	**h**	0	**i**

- Alternative to pooling that preserves input size
- 9 degrees of freedom
- 5x5 receptive field

Architectures with dilated convolutions

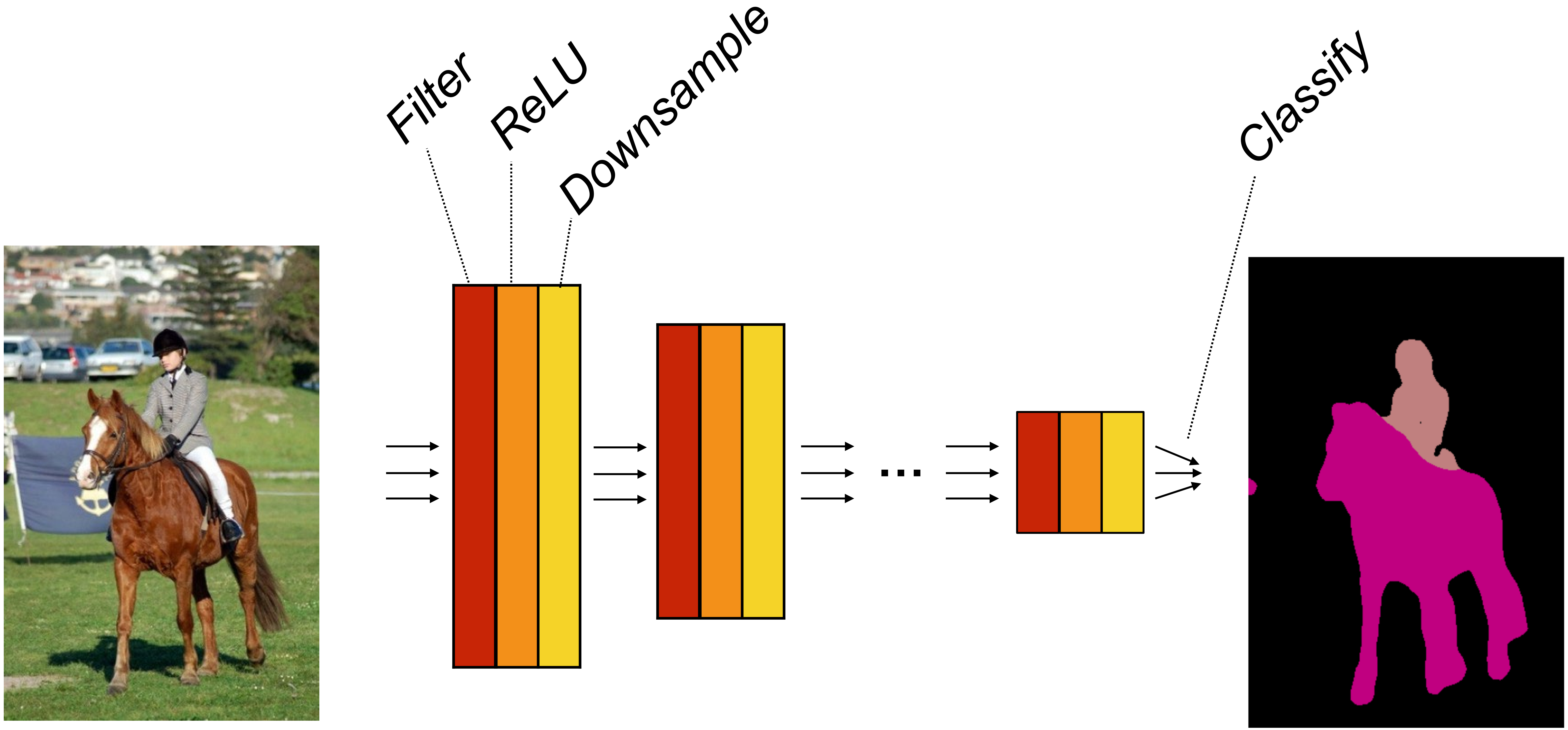


- Architecture design: dilation by 2^L instead of striding times
- Obtains comparable receptive field to CNN with strides.

[Yu and Koltun 2016, https://arxiv.org/pdf/1511.07122.pdf]

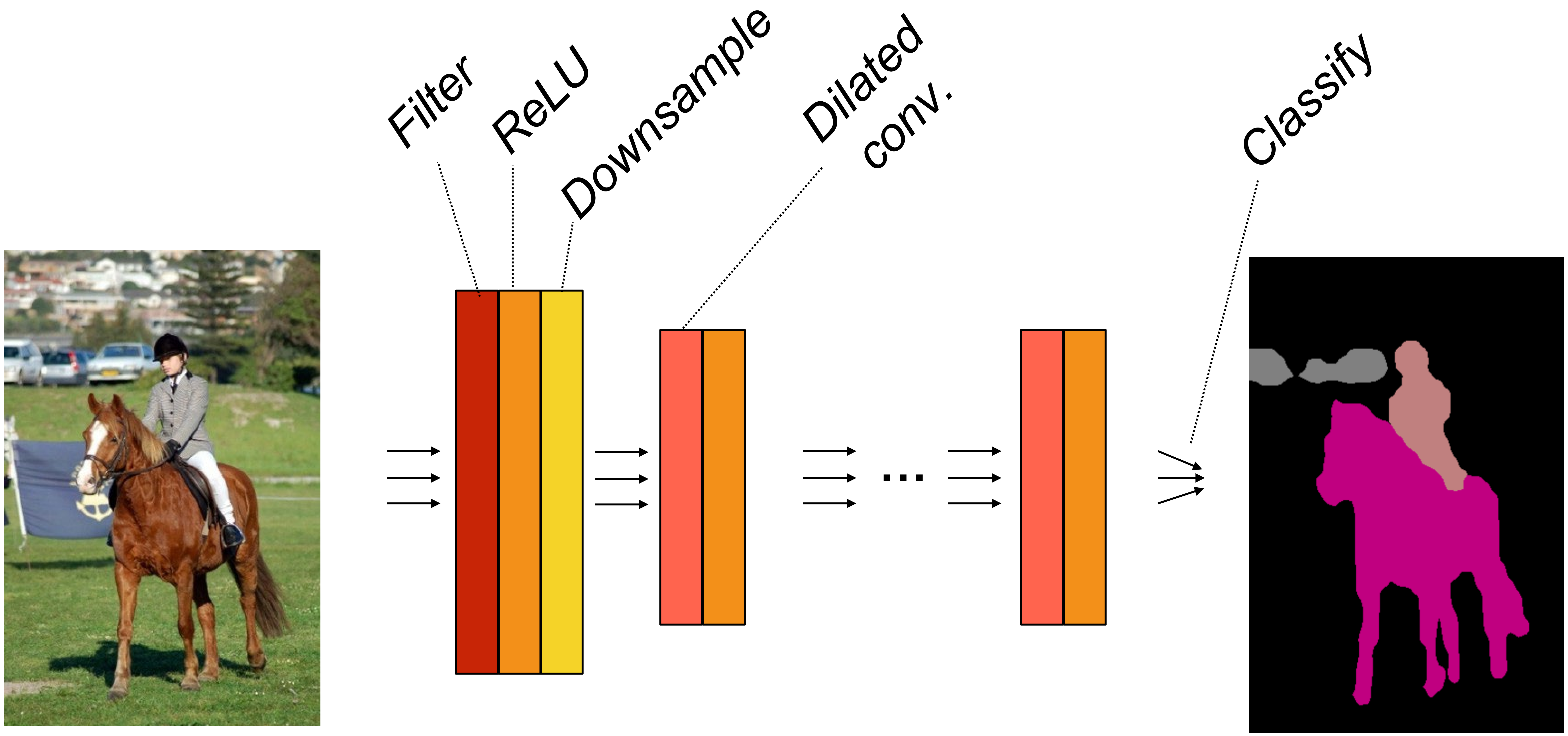
Adapted from: Isola, Torralba, Freeman

CNN without dilated convolutions



Apply CNN convolutionally

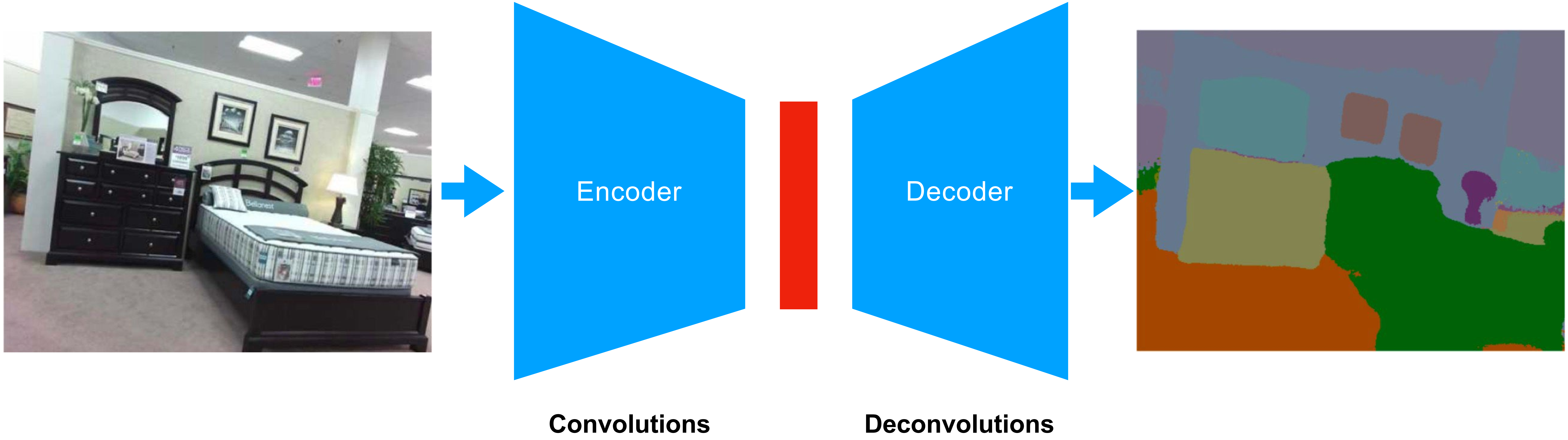
CNN with dilated convolutions



Even with dilated convolutions, still not full resolution

Idea #4: Encoder-decoder models

Encoder-decoder architectures



Source: Torralba, Freeman, Isola

Upsampling

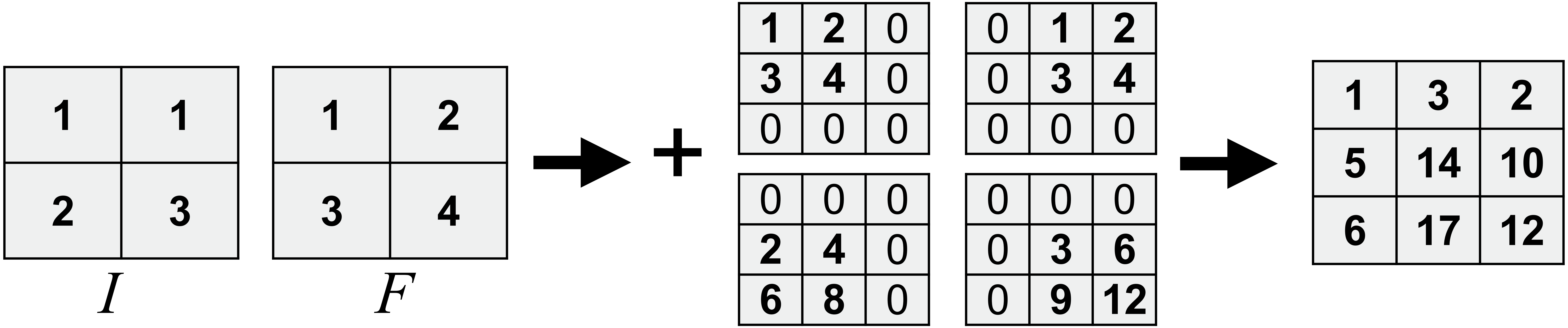
1	2
3	4

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Convolution

- Often using nearest-neighbor upsampling
- Can also use interpolation.
- Produces fewer “checkerboard” artifacts

Transposed convolution



- Weight the filter by the image coefficient and sum.
- Also sometimes called “upconvolution” or “deconvolution”.

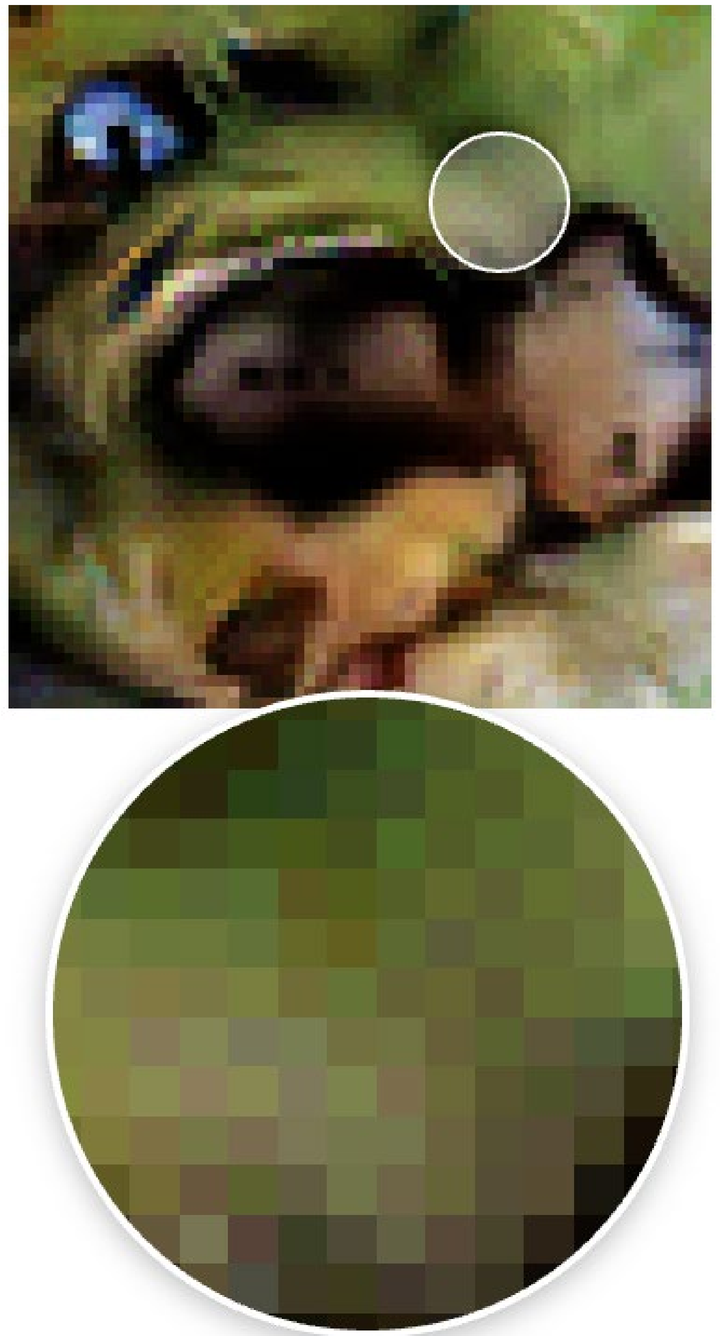
Transposed convolution

1	1
2	3

I

1	2
3	4

F

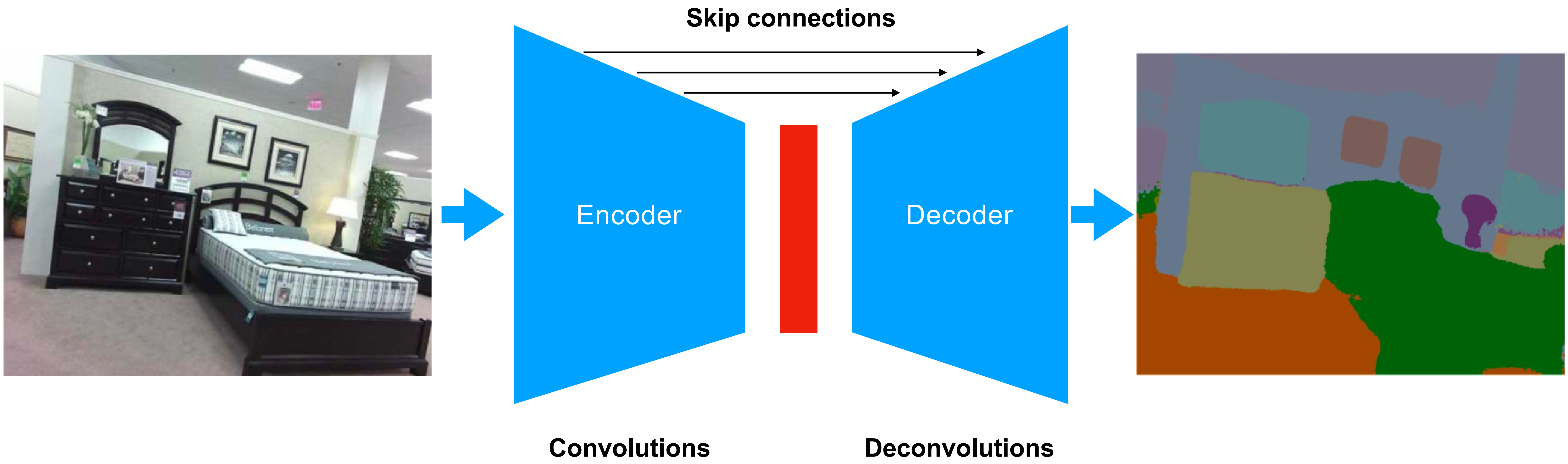


Donahue, et al., 2016 [3]

Can lead to “checkerboard” artifacts.

[Odena et al. Distill article]

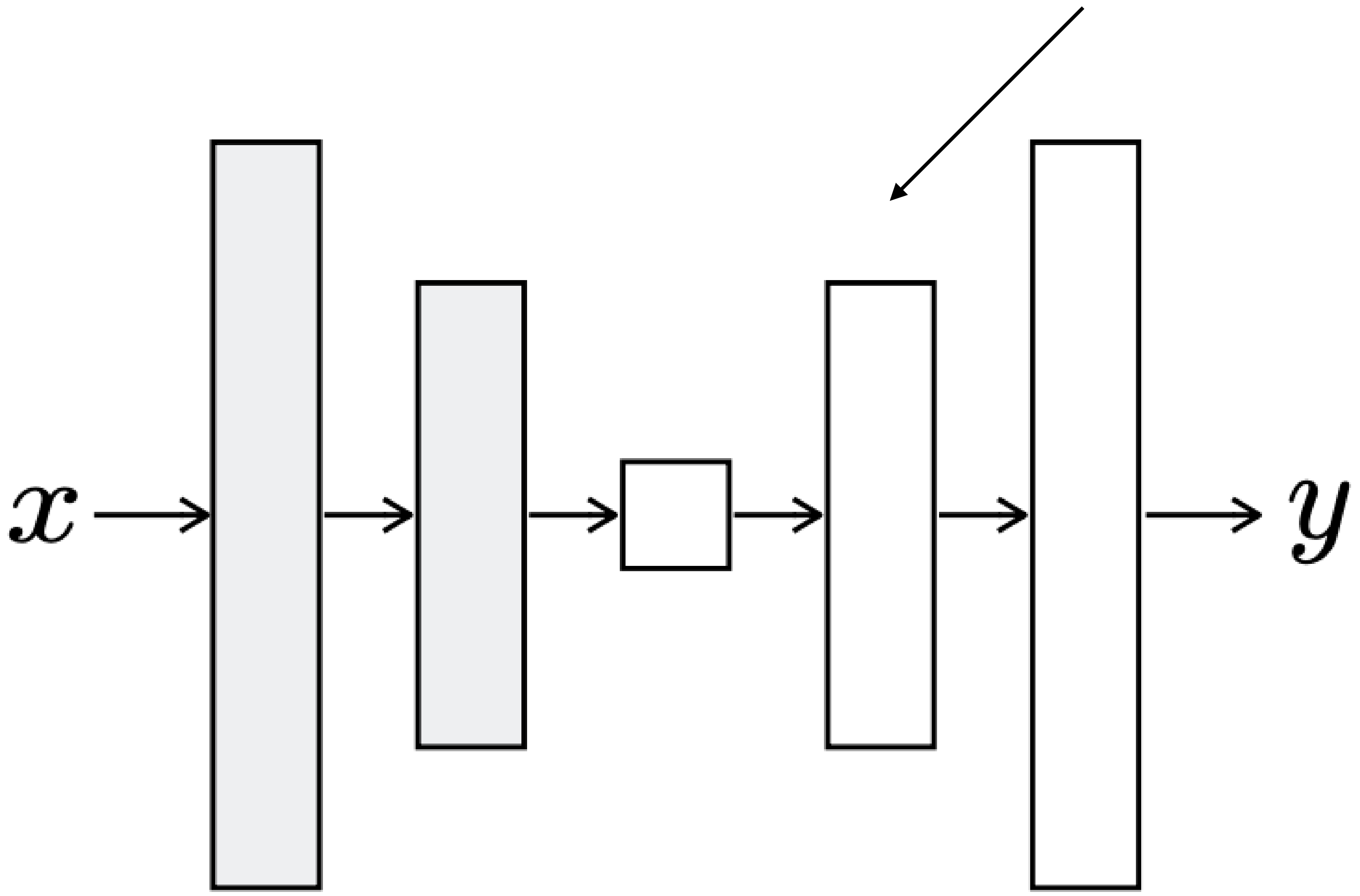
Encoder-decoder architectures



Source: Torralba, Freeman, Isola

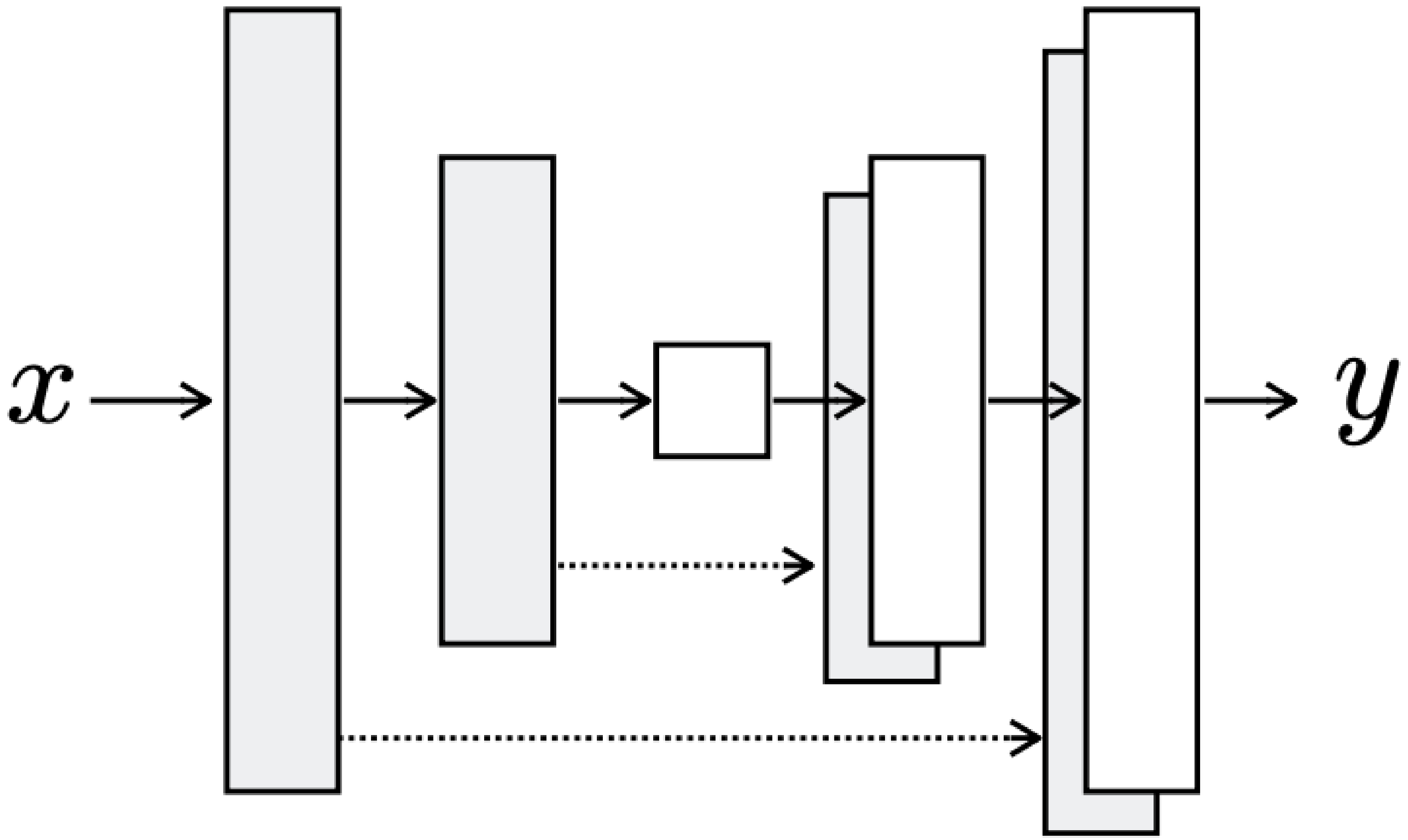
Encoder-decoder architectures

Transposed convolution



"Vanilla" encoder-decoder architecture

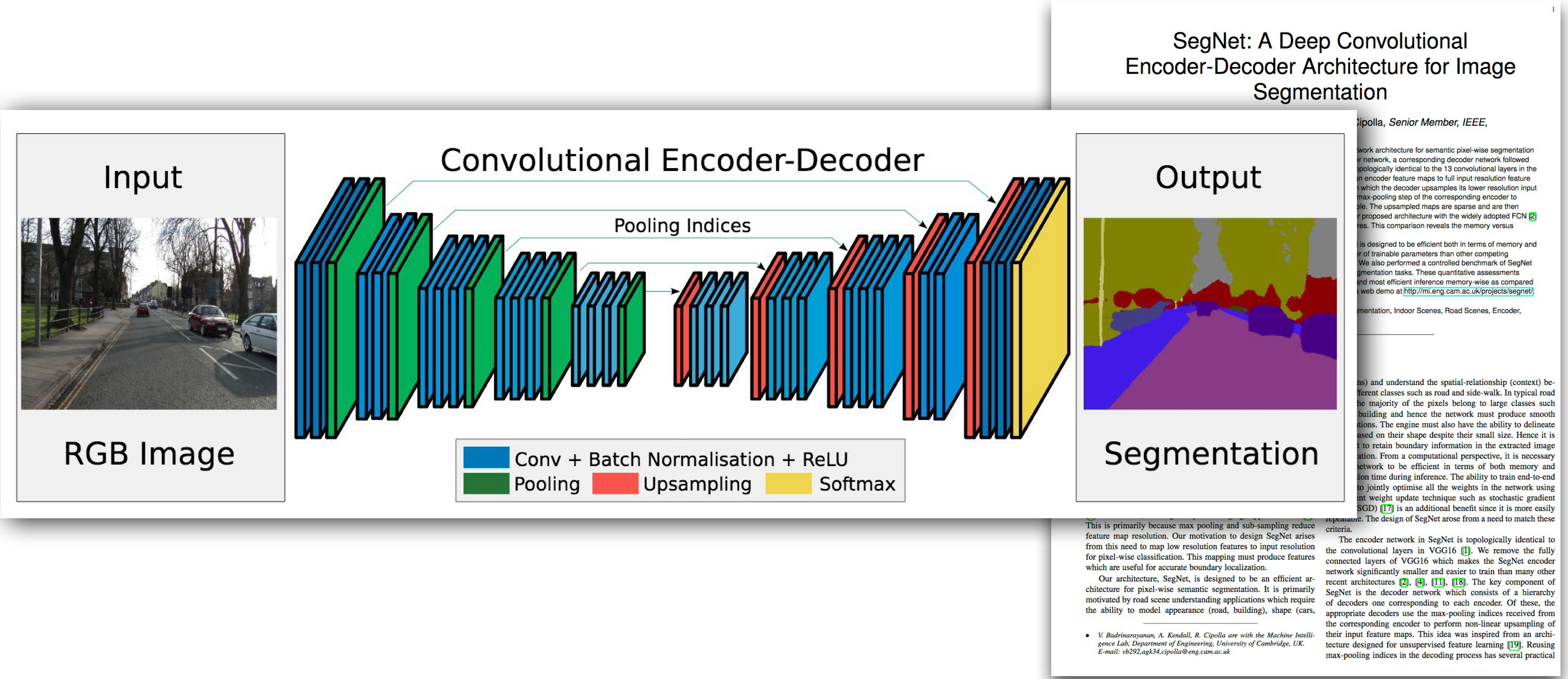
Early layers and late layers have same shape. Concatenate channel-wise!



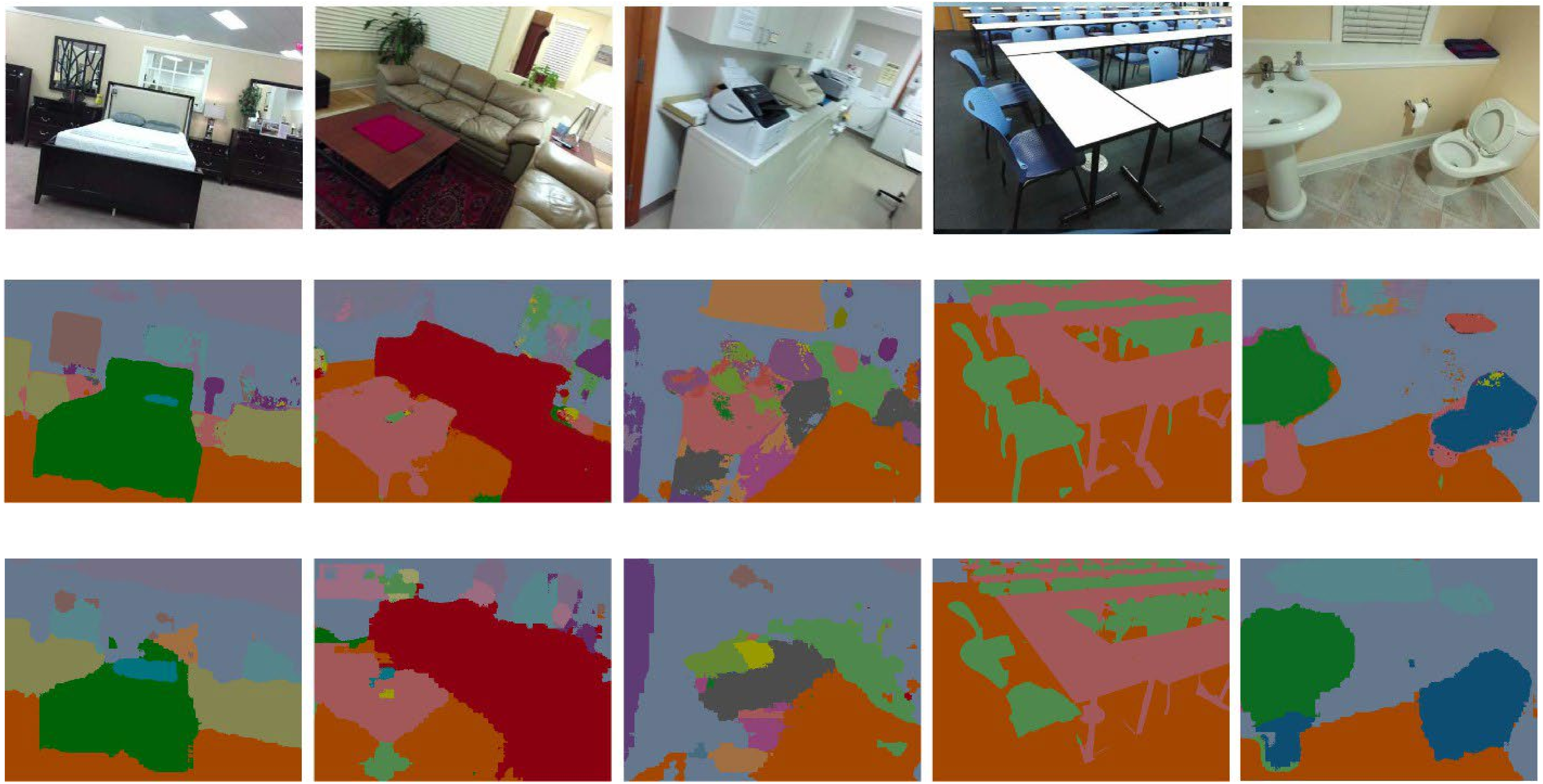
U-Net

Figures from [Isola et al., "Image-to-Image Translation with Conditional Adversarial Networks", 2017]

Encoder-decoder architectures



Source: Torralba, Freeman, Isola



Input

Segnet

FCN

Source: Torralba, Freeman, Isola