Artificial Intelligence II

Part 2: Lecture 7

Yalda Mohsenzadeh

Slides are from Antonio Torralba and Philip Isola (MIT)

Convolutional Neural Networks

Today

- How to use networks for images
- Why “C”-NNs
- Standard building blocks of CNNs
- Some important networks & their tricks
- Some debugging tools

Image classification

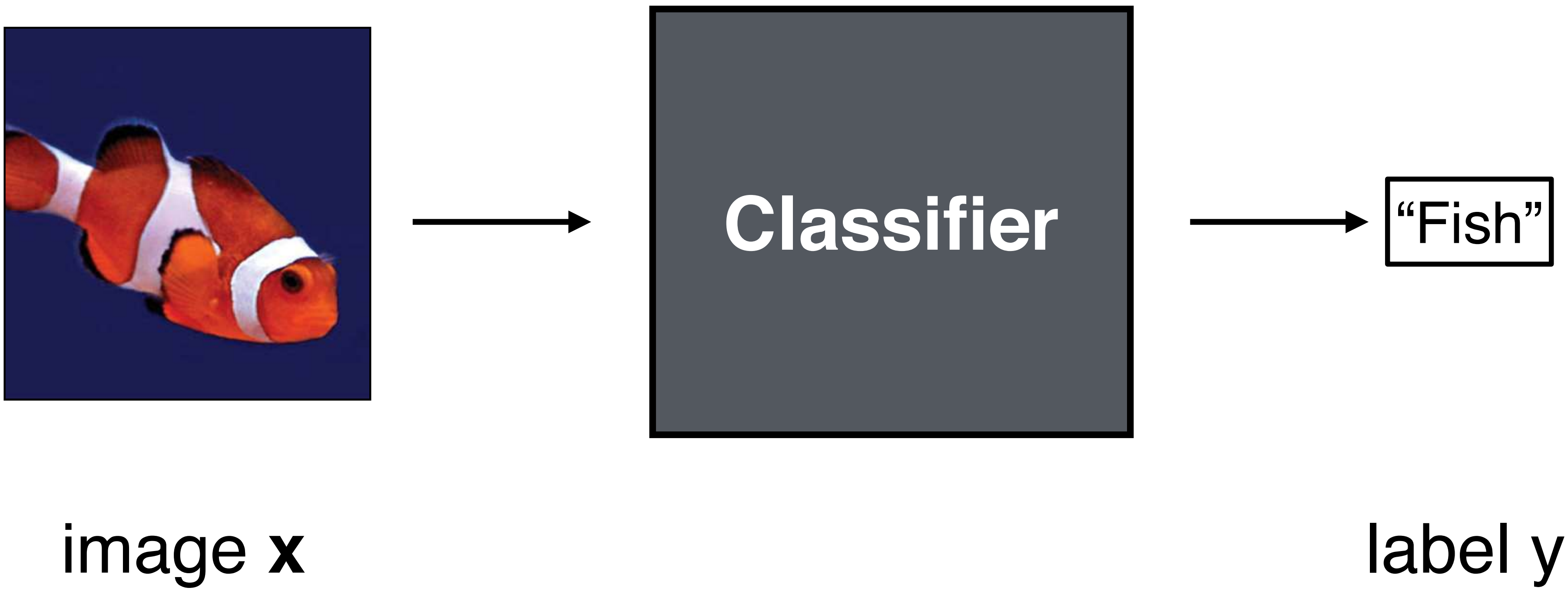
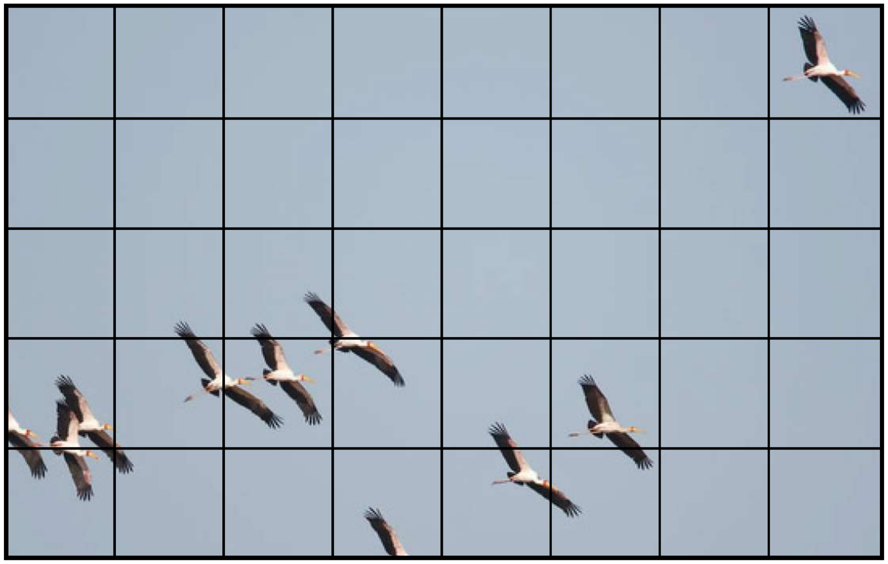
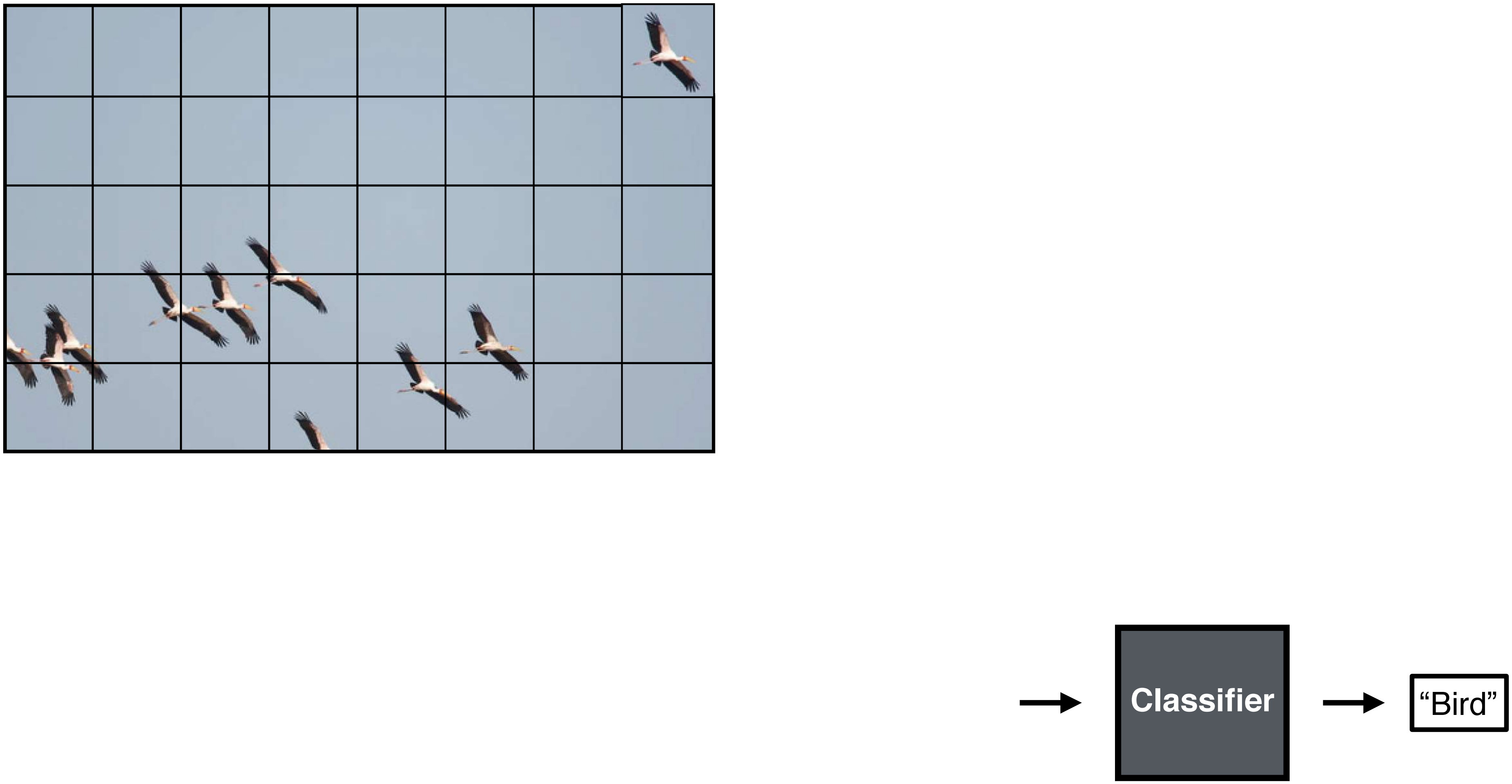
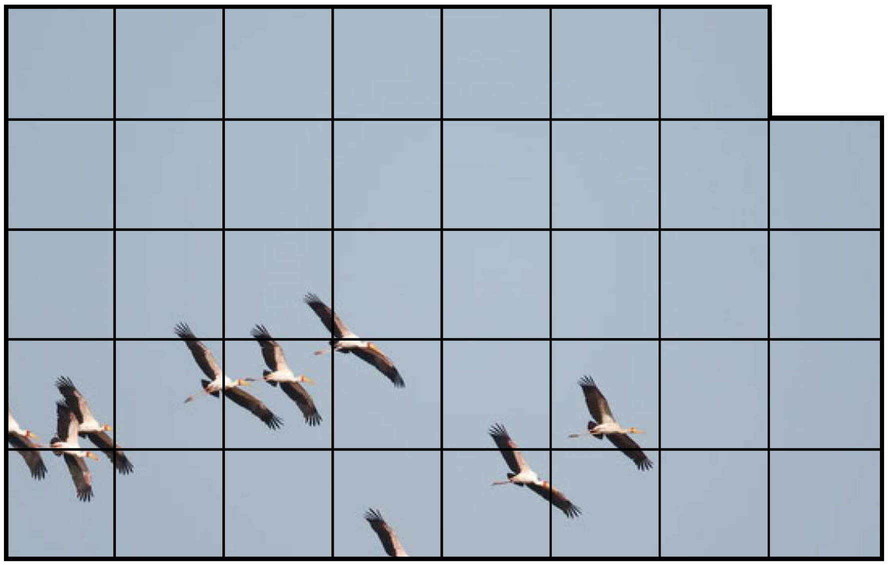
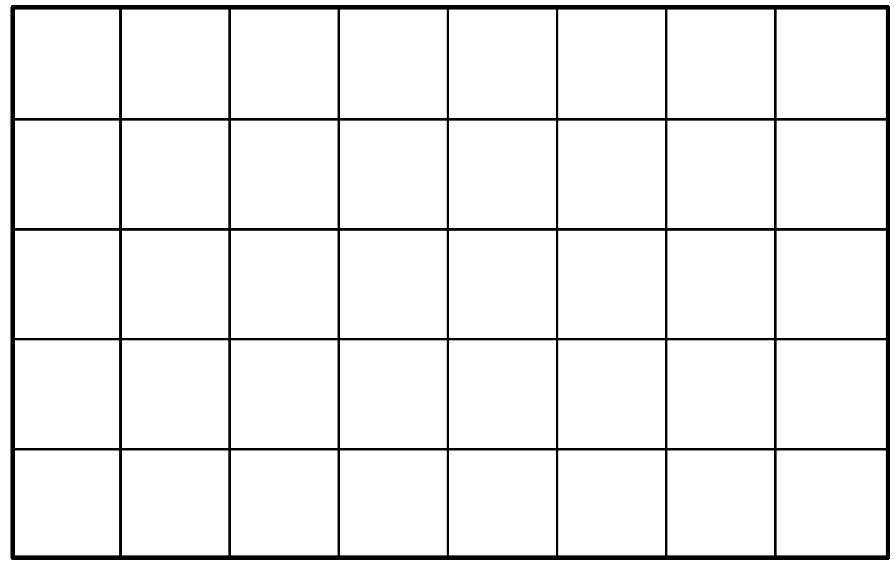



Photo credit: Fredo Durand

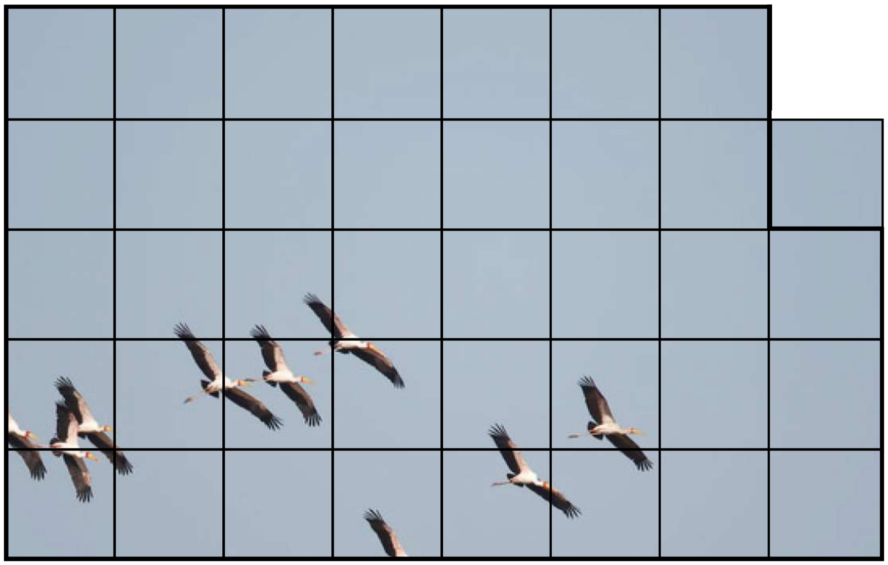


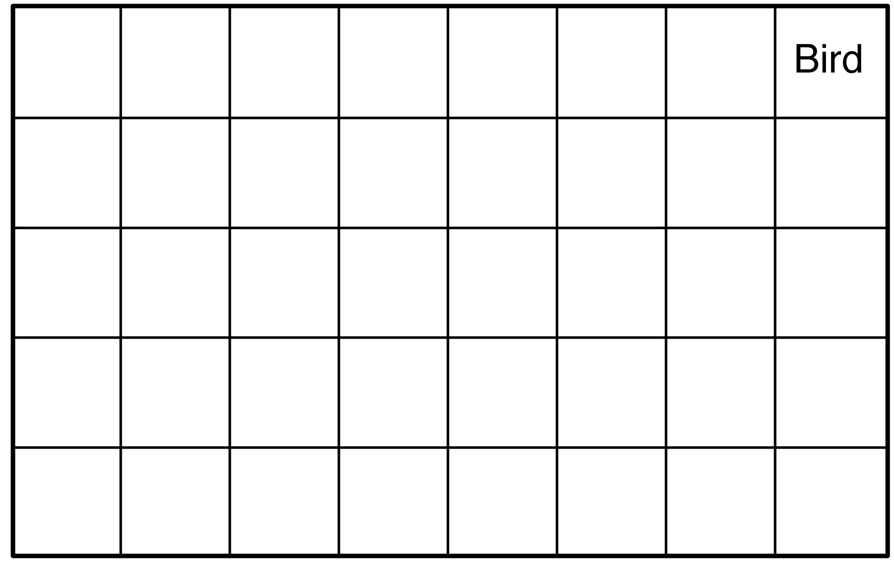

Classifier
“Bird”

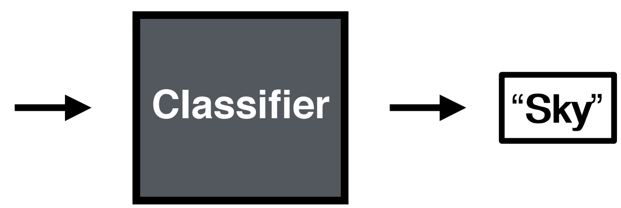


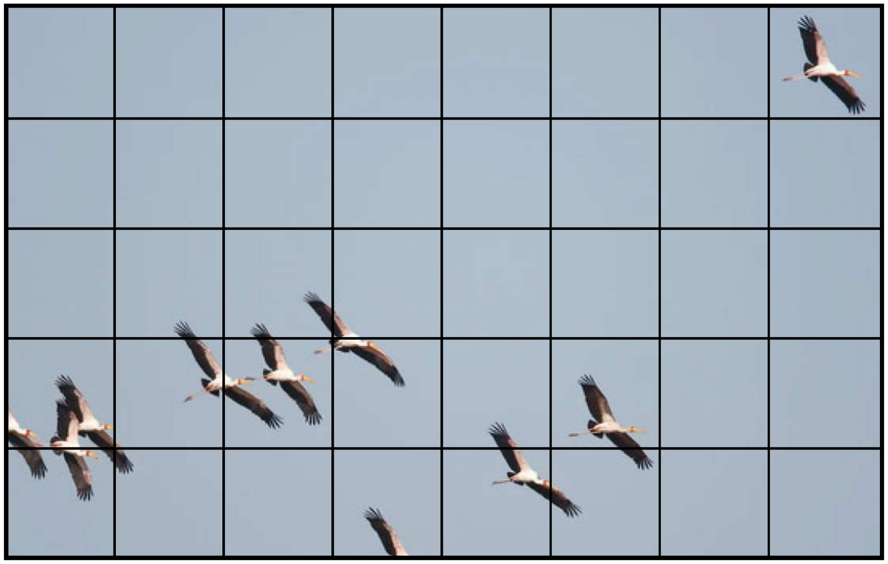



Classifier
"Bird"

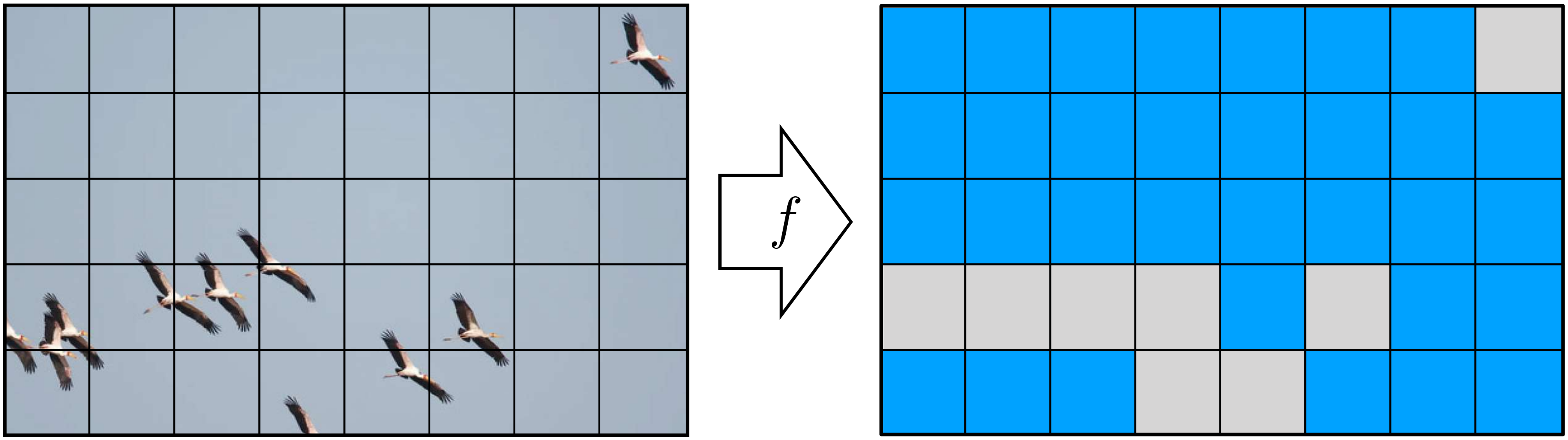



Bird


Classifier
"Sky"



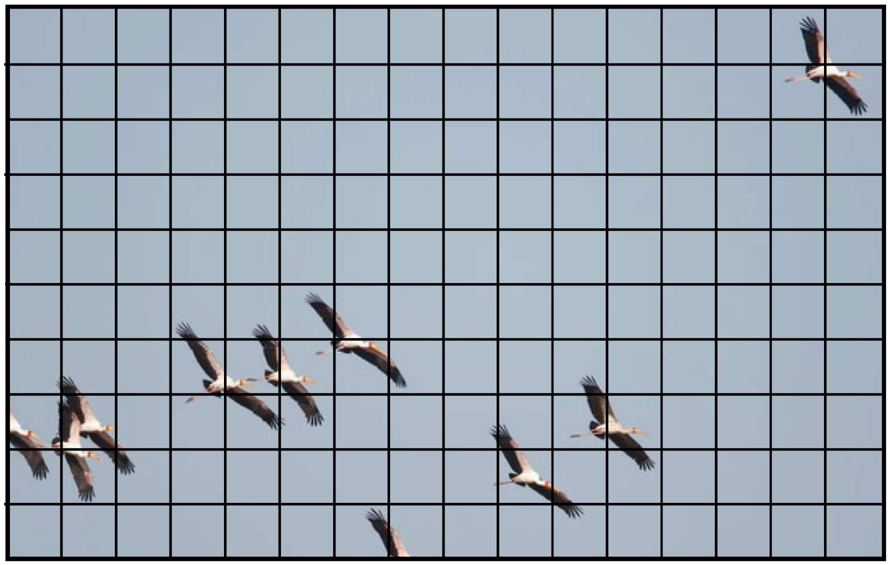
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Bird
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Sky
Sky	Sky	Sky	Sky	Sky	Sky	Sky	Sky
Bird	Bird	Bird	Sky	Bird	Sky	Sky	Sky
Sky	Sky	Sky	Bird	Sky	Sky	Sky	Sky



Problem:

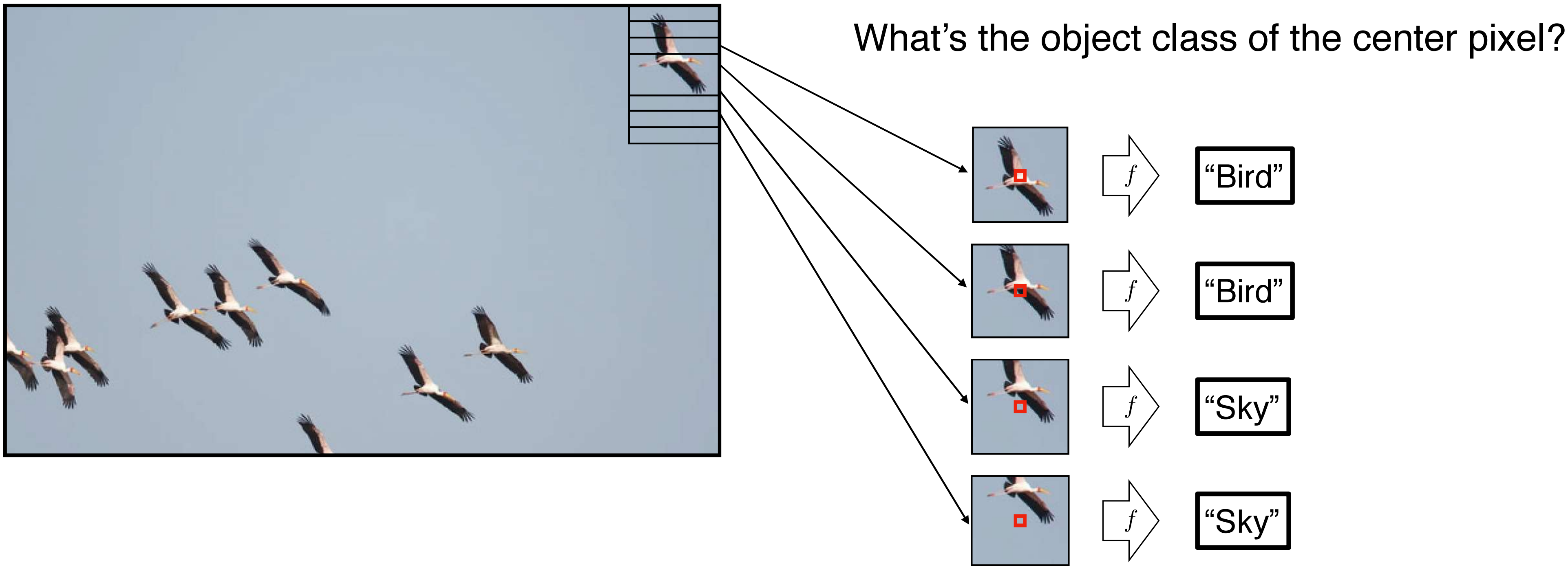
What happens to objects that are bigger?

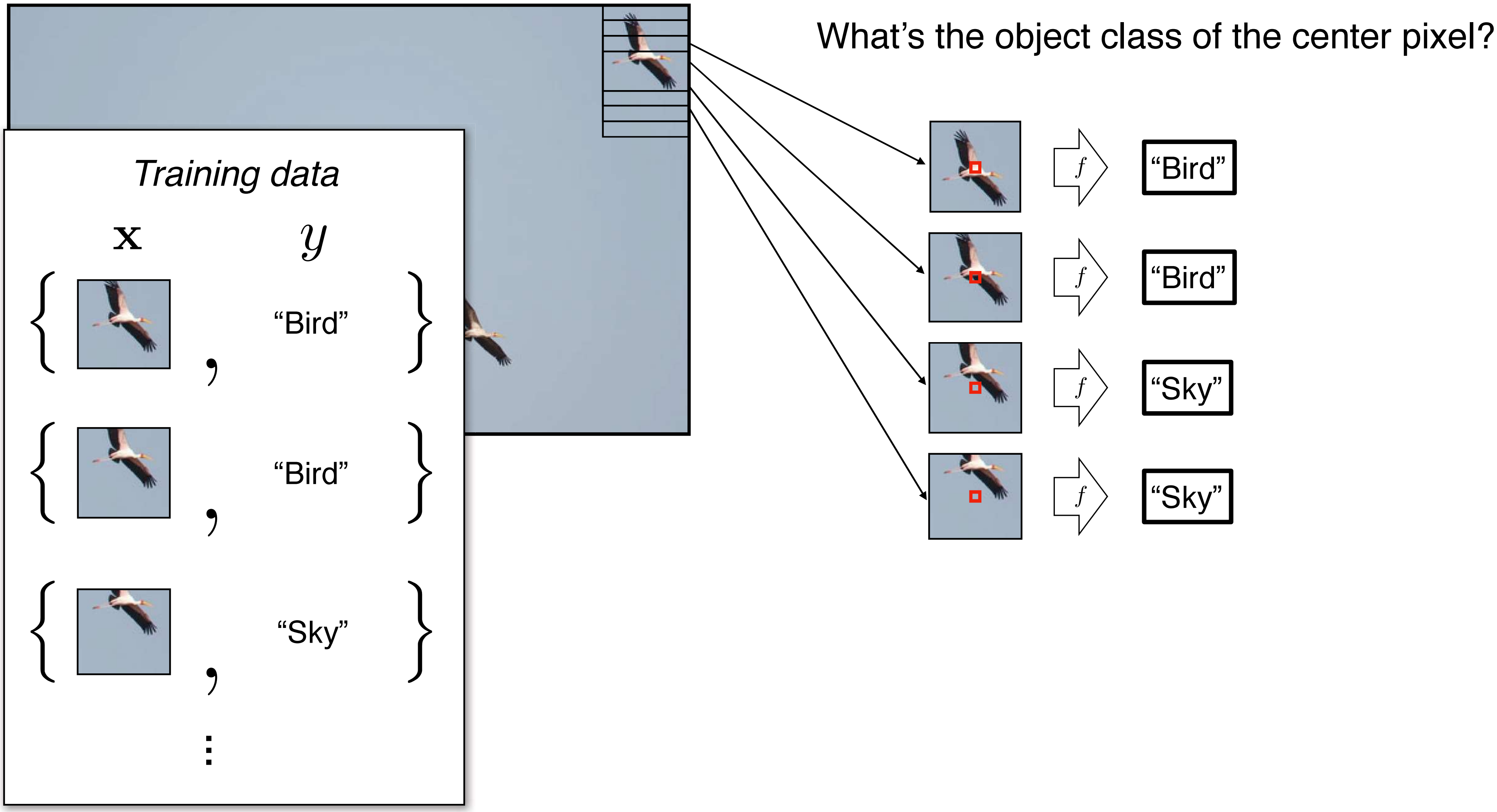
What if an object crosses multiple cells?

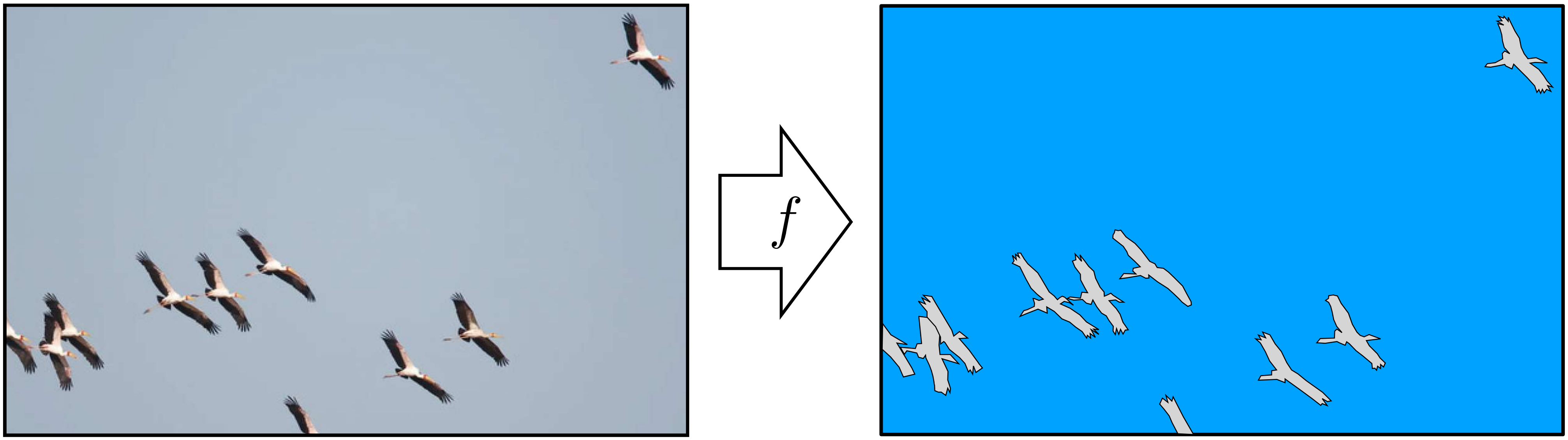


“Cell”-based approach is limited.

What can we do instead?


What's the object class of the center pixel?
f
"Bird"
f
"Bird"
f
"Sky"
f
"Sky"

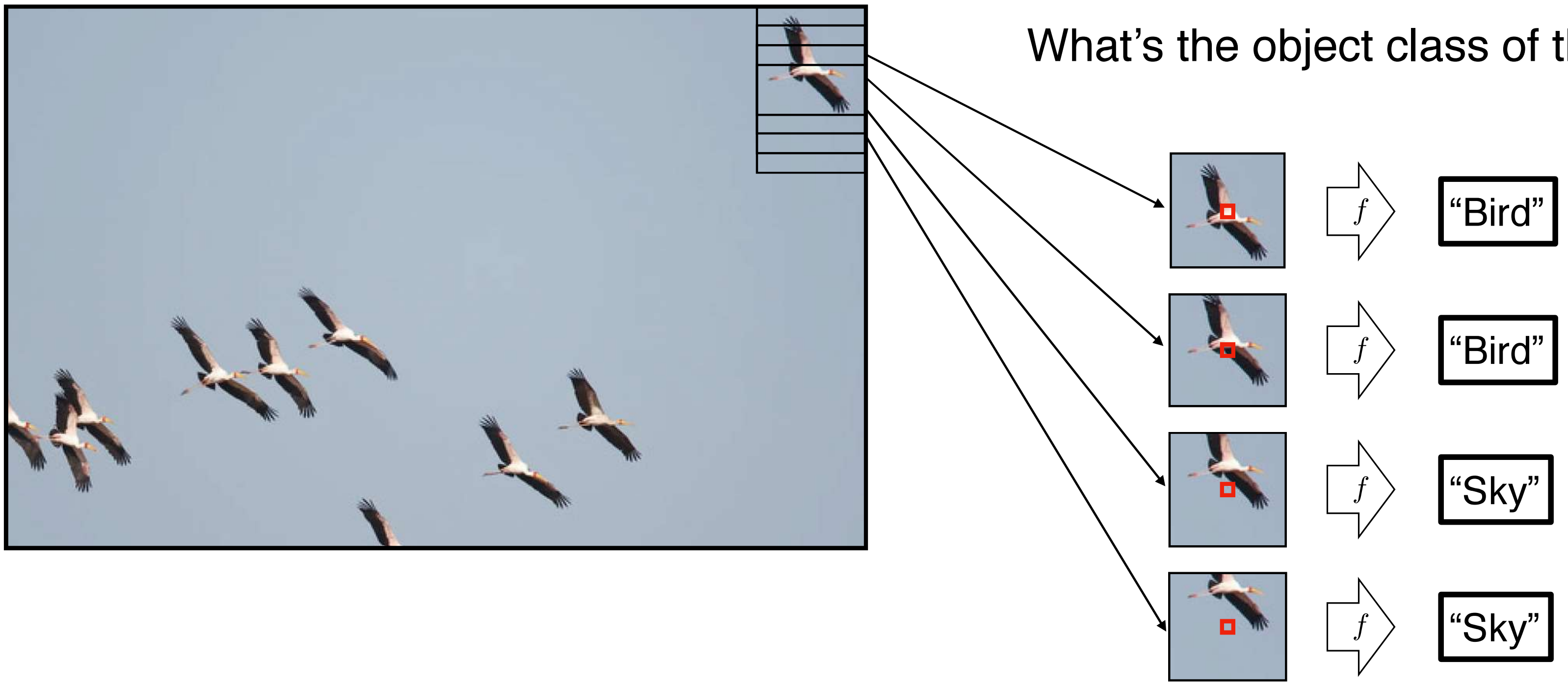

What's the object class of the center pixel?
Training data
$\mathbf{x}$
y
"Bird"
"Bird"
"Sky"
f
"Bird"
"Bird"
"Sky"
"Sky"



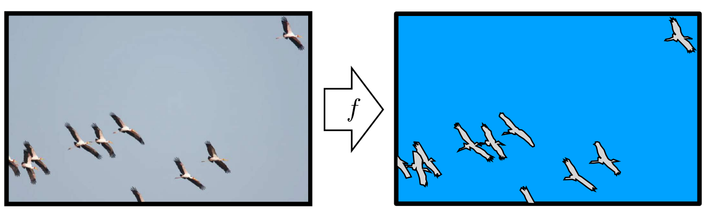
(Colors represent one-hot codes)

This problem is called **semantic segmentation**

What’s the object class of the center pixel?



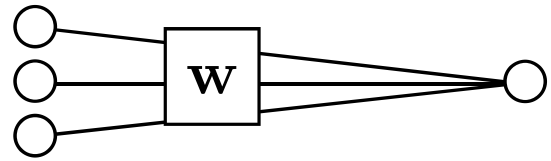
Translation invariance: process each patch in the same way.

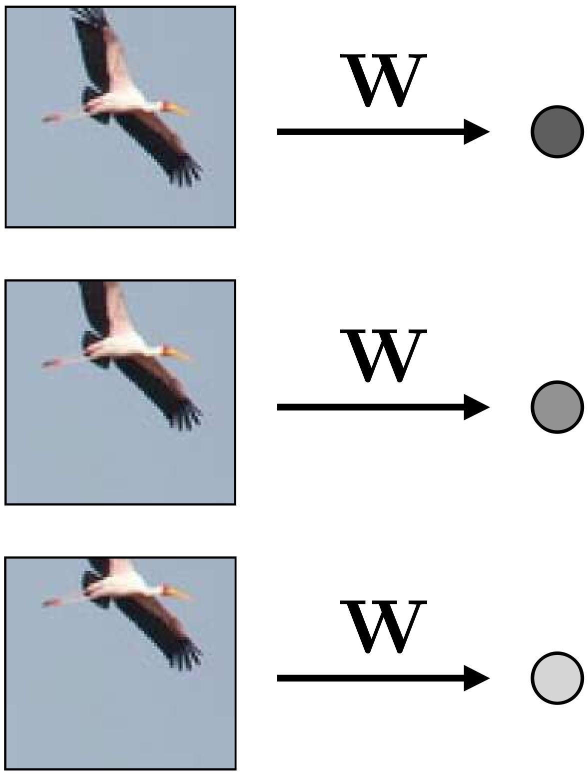


An *equivariant* mapping:

$$f(\texttt{translate}(x)) = \texttt{translate}(f(x))$$

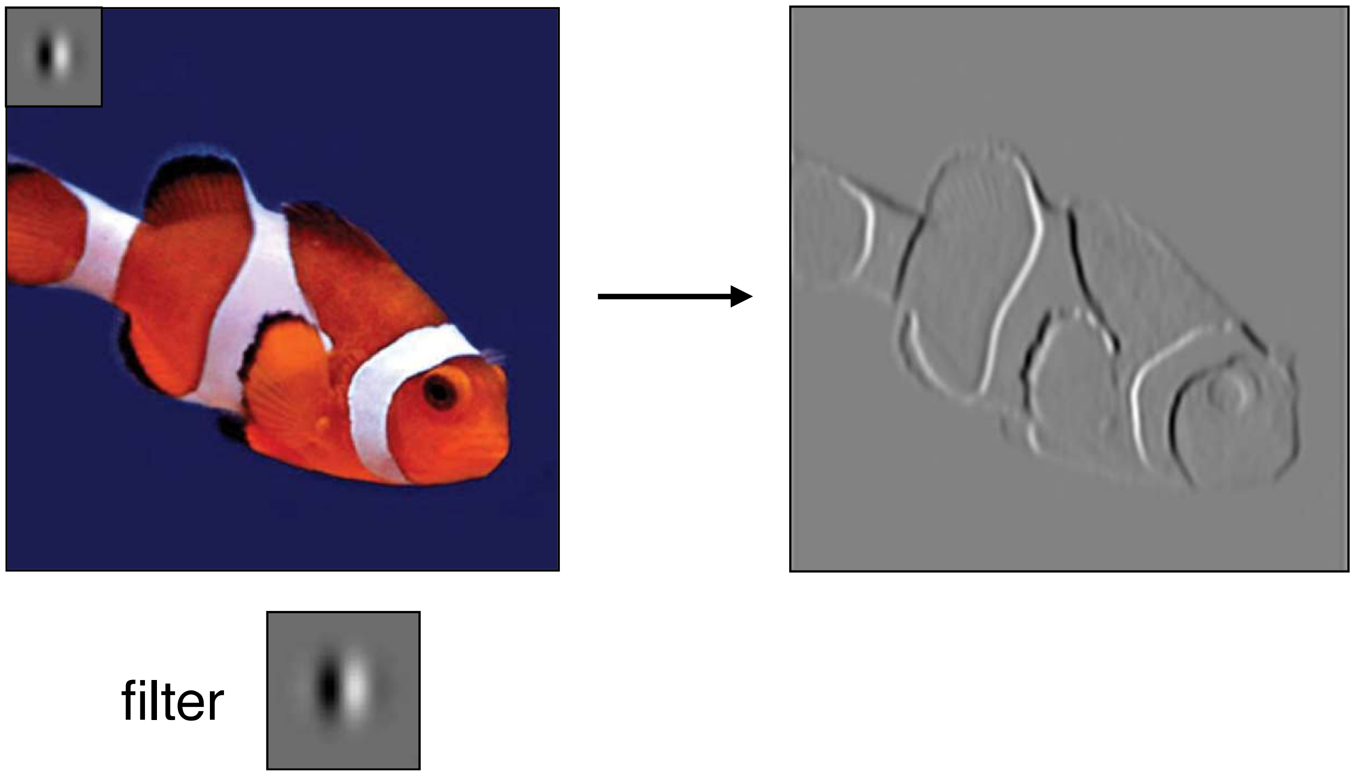
W computes a weighted sum of all pixels in the patch





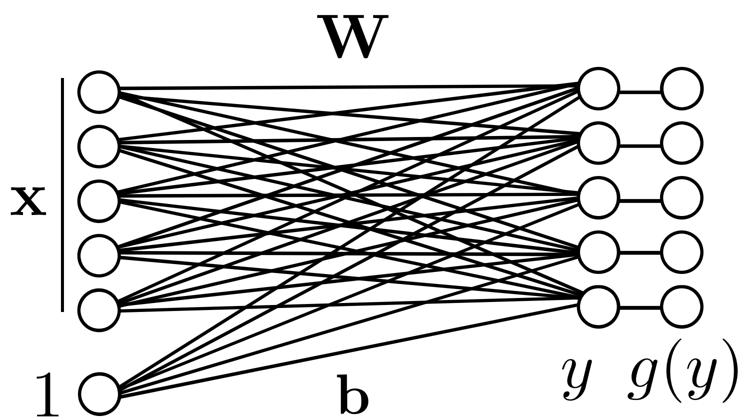
W is a convolutional kernel applied to the full image!

Convolution

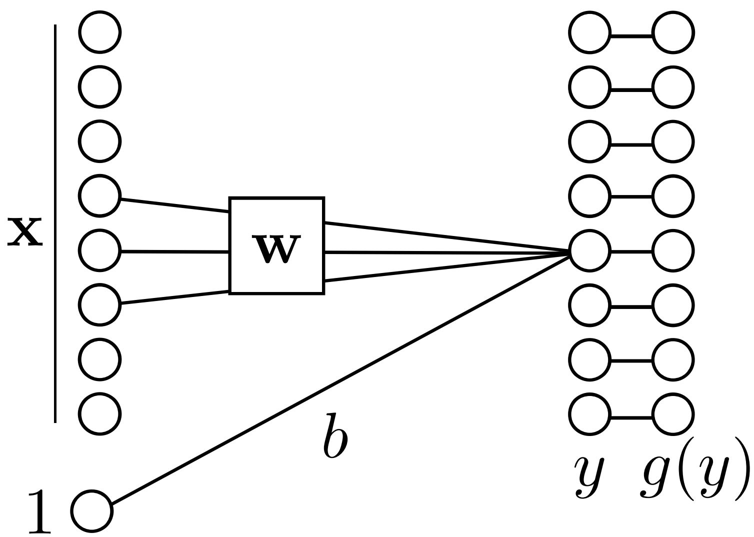


Fully-connected network

Fully-connected (fc) layer



Locally connected network

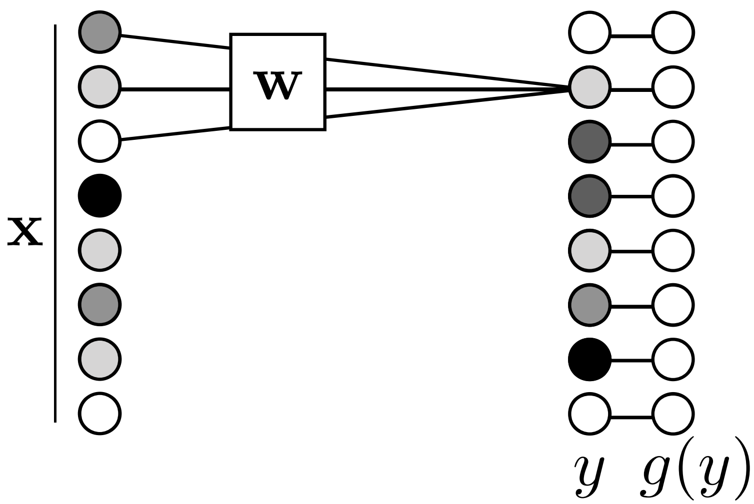


Often, we assume output is a **local** function of input.

If we use the same weights (**weight sharing**) to compute each local function, we get a convolutional neural network.

Convolutional neural network

Conv layer

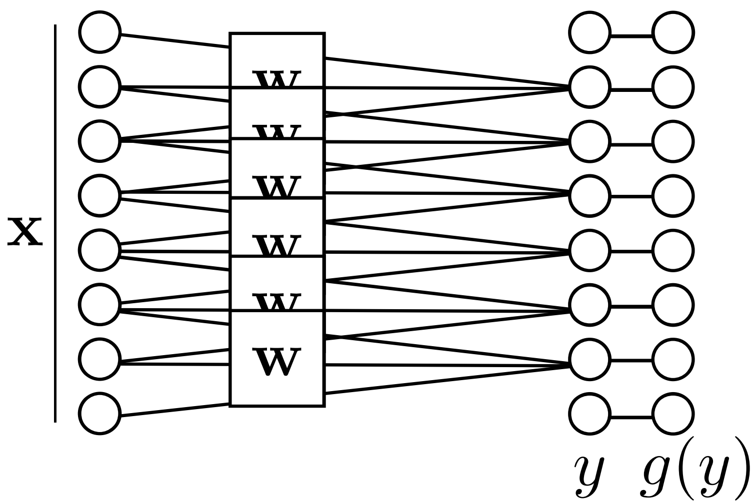


Often, we assume output is a **local** function of input.

If we use the same weights (**weight sharing**) to compute each local function, we get a convolutional neural network.

Weight sharing

Conv layer



Often, we assume output is a **local** function of input.

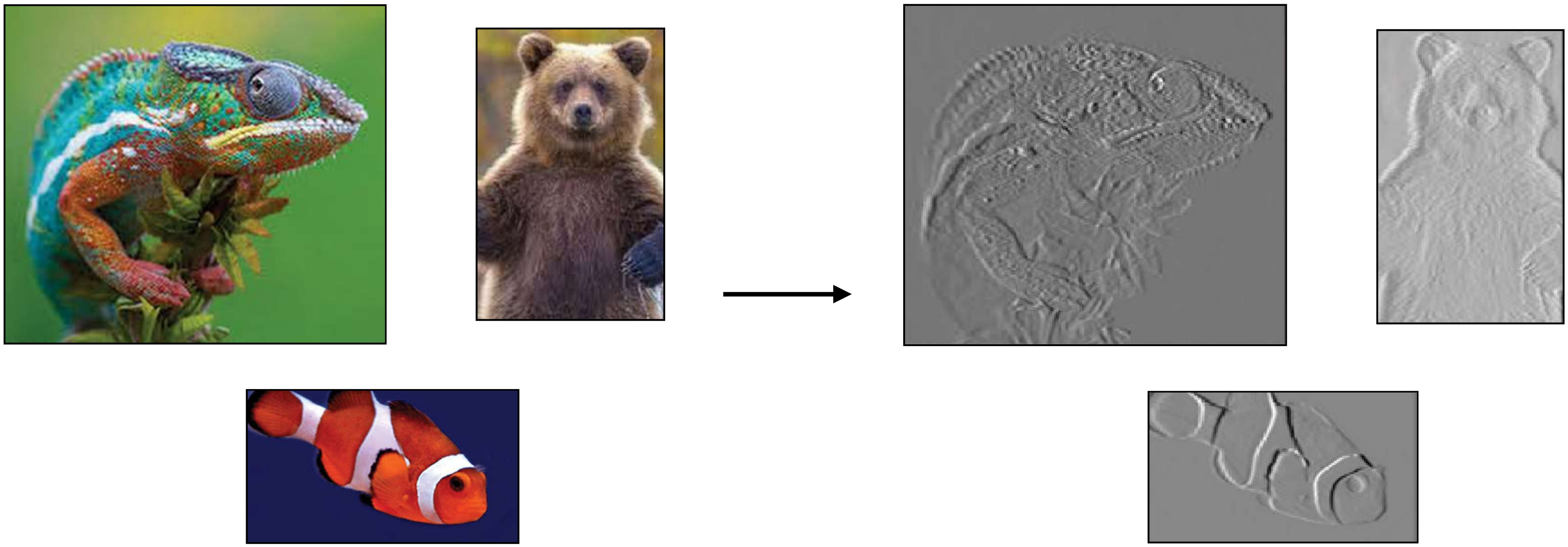
If we use the same weights (**weight sharing**) to compute each local function, we get a convolutional neural network.

Toeplitz matrix

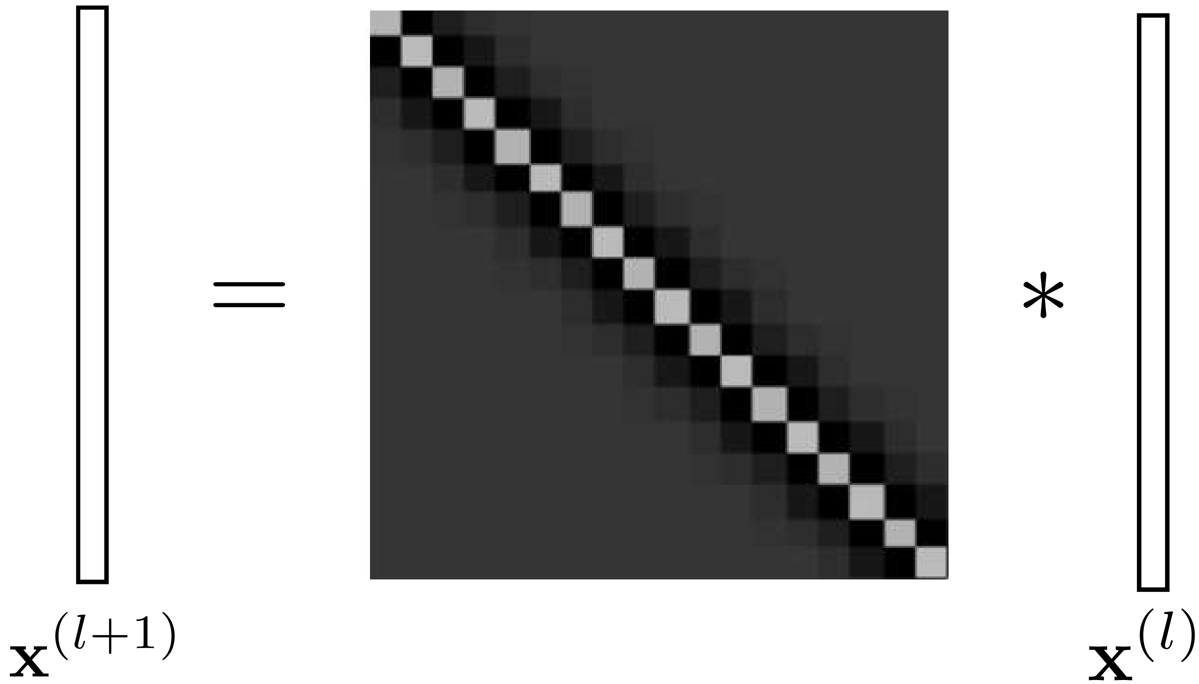
$$\begin{pmatrix} a & b & c & d & e \\ f & a & b & c & d \\ g & f & a & b & c \\ h & g & f & a & b \\ i & h & g & f & a \end{pmatrix}$$

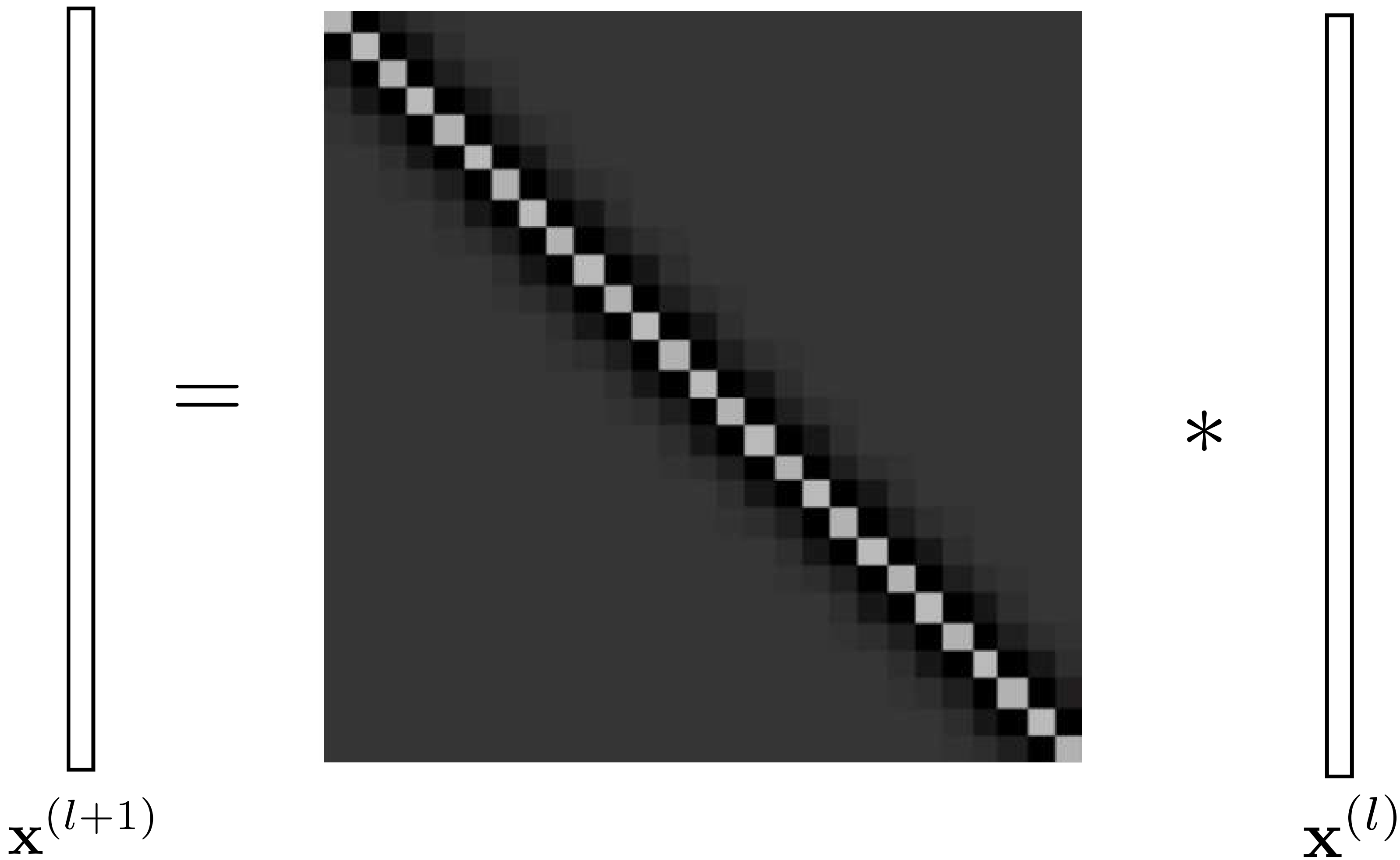
e.g., pixel image

- Constrained linear layer
- Fewer parameters —> easier to learn, less overfitting



Conv layers can be applied to arbitrarily-sized inputs


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

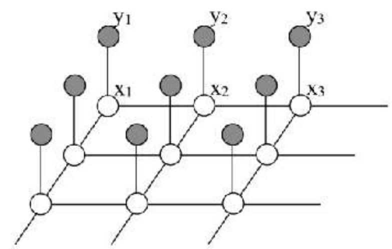


=
*
$\mathbf{x}^{(l+1)}$
$\mathbf{x}^{(l)}$

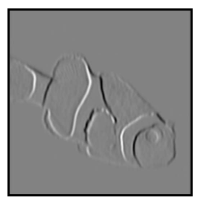
Five views on convolutional layers

1. Equivariant with translation (stationarity) $f(\texttt{translate}(x)) = \texttt{translate}(f(x))$

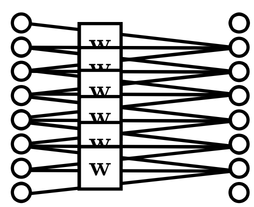
2. Patch processing (Markov assumption)



3. Image filter



4. Parameter sharing

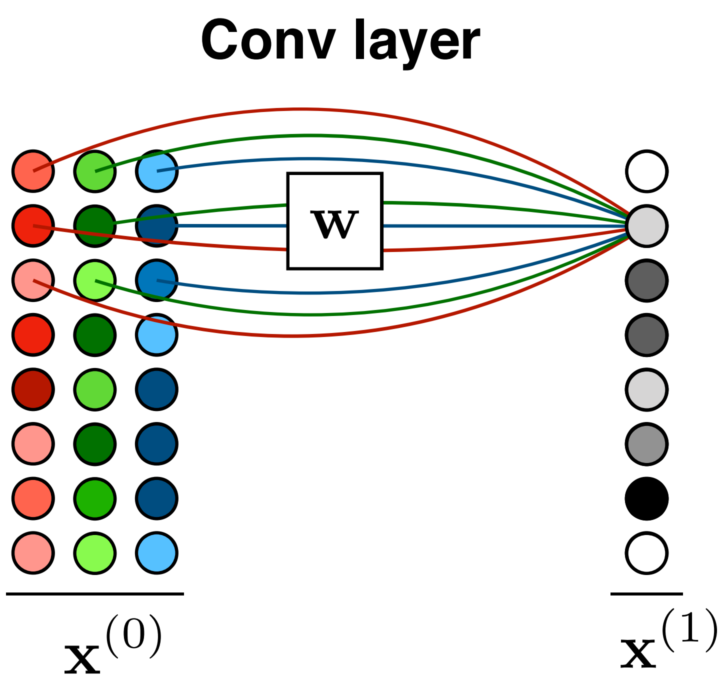


5. A way to process variable-sized tensors

What if we have color?

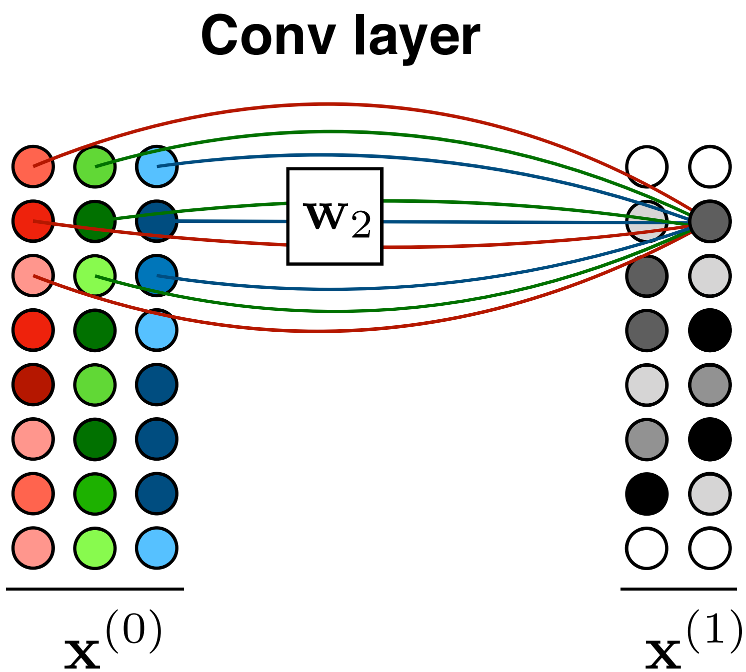
(aka multiple input channels?)

Multiple channels



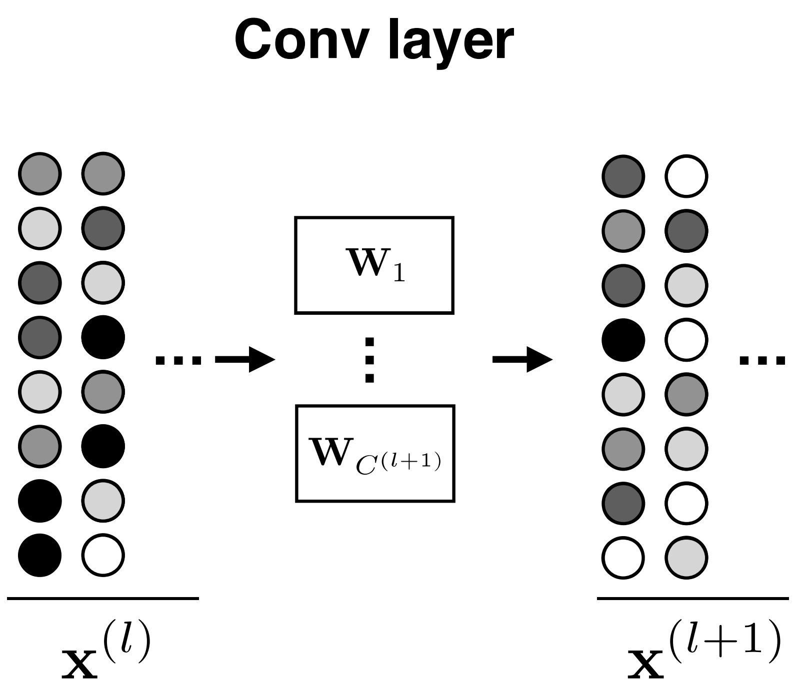
$$\mathbb{R}^{N\times C} \to \mathbb{R}^{N\times 1}$$

Multiple channels



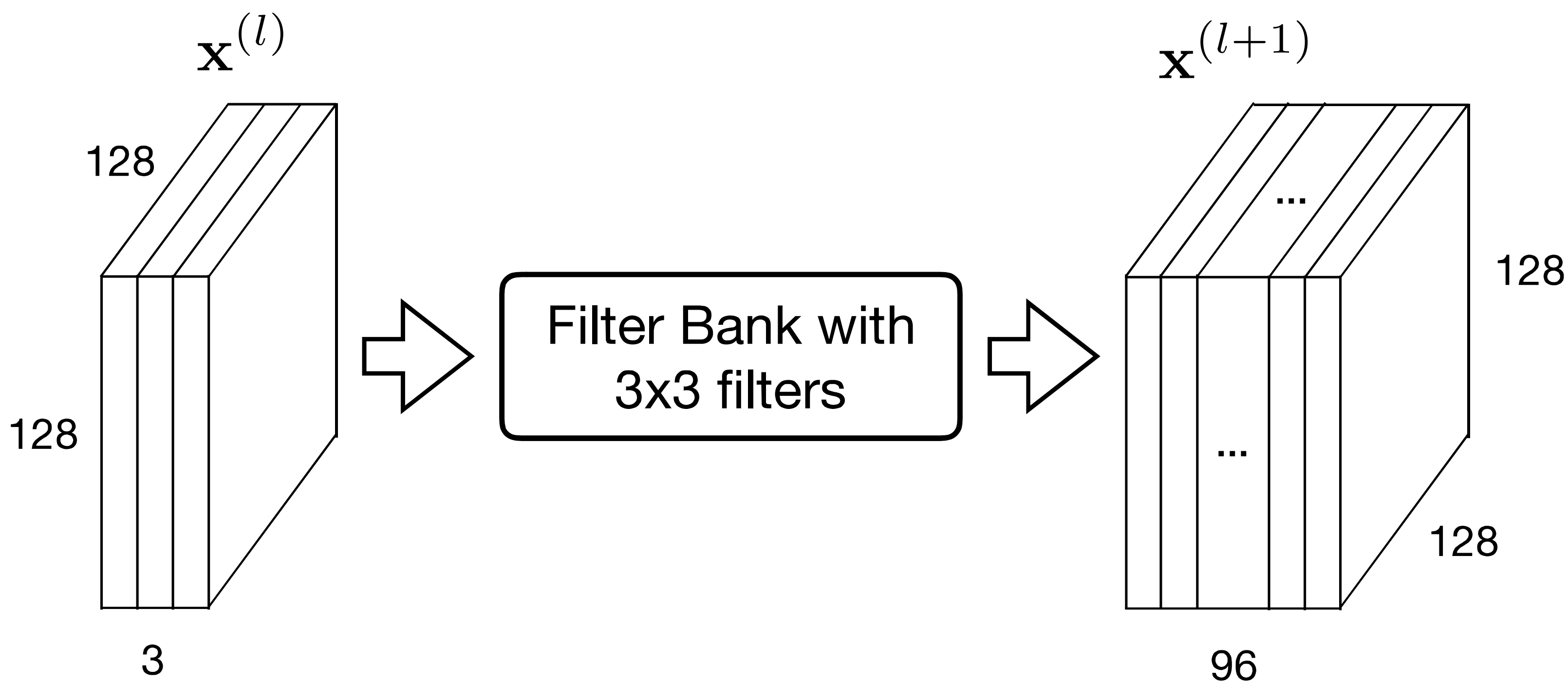
$$\mathbb{R}^{N \times C^{(0)}} \rightarrow \mathbb{R}^{N \times C^{(1)}}$$

Multiple channels



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

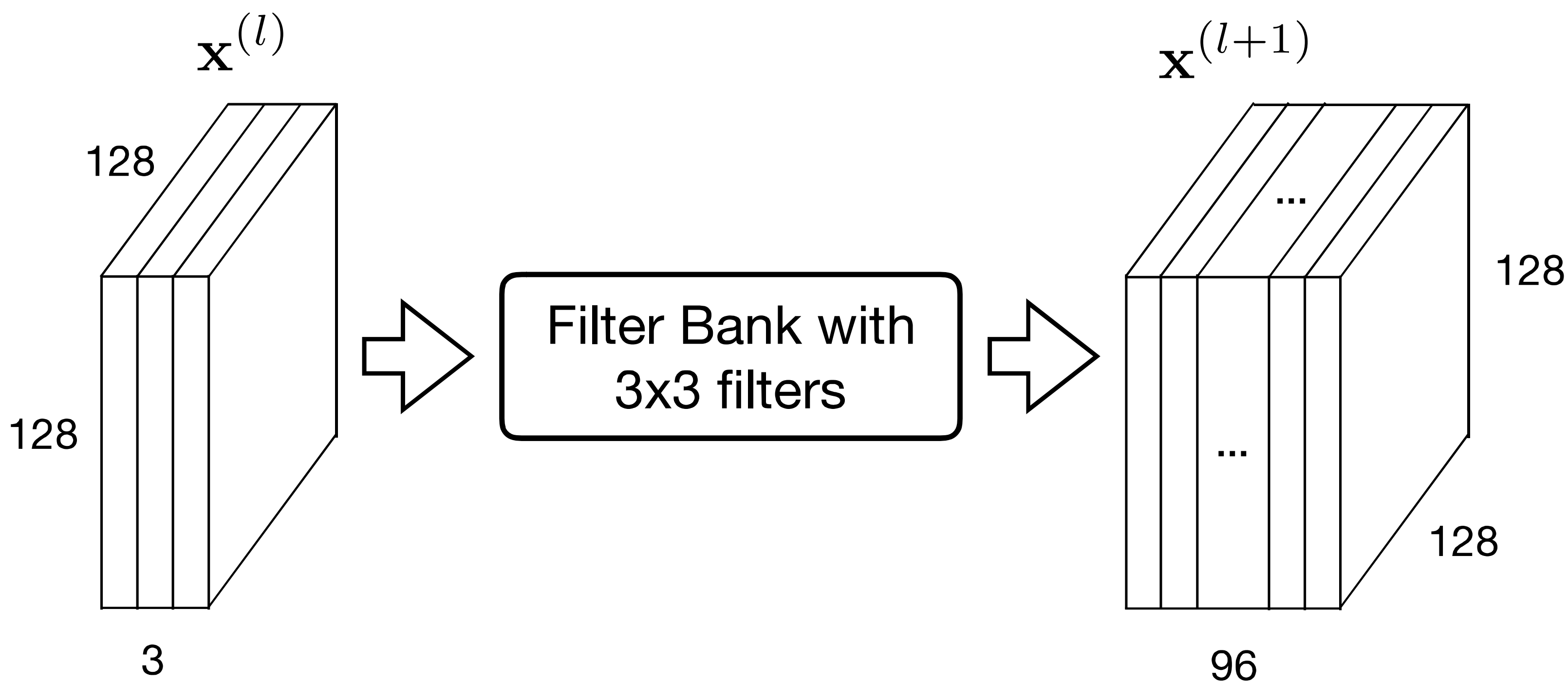
Multiple channels: Example



How many parameters does *each filter* have?

(a) 9 (b) 27 (c) 96 (d) 864

Multiple channels: Example



How many filters are in the bank?

(a) 3 (b) 27 (c) 96 (d) can't say

Filter sizes

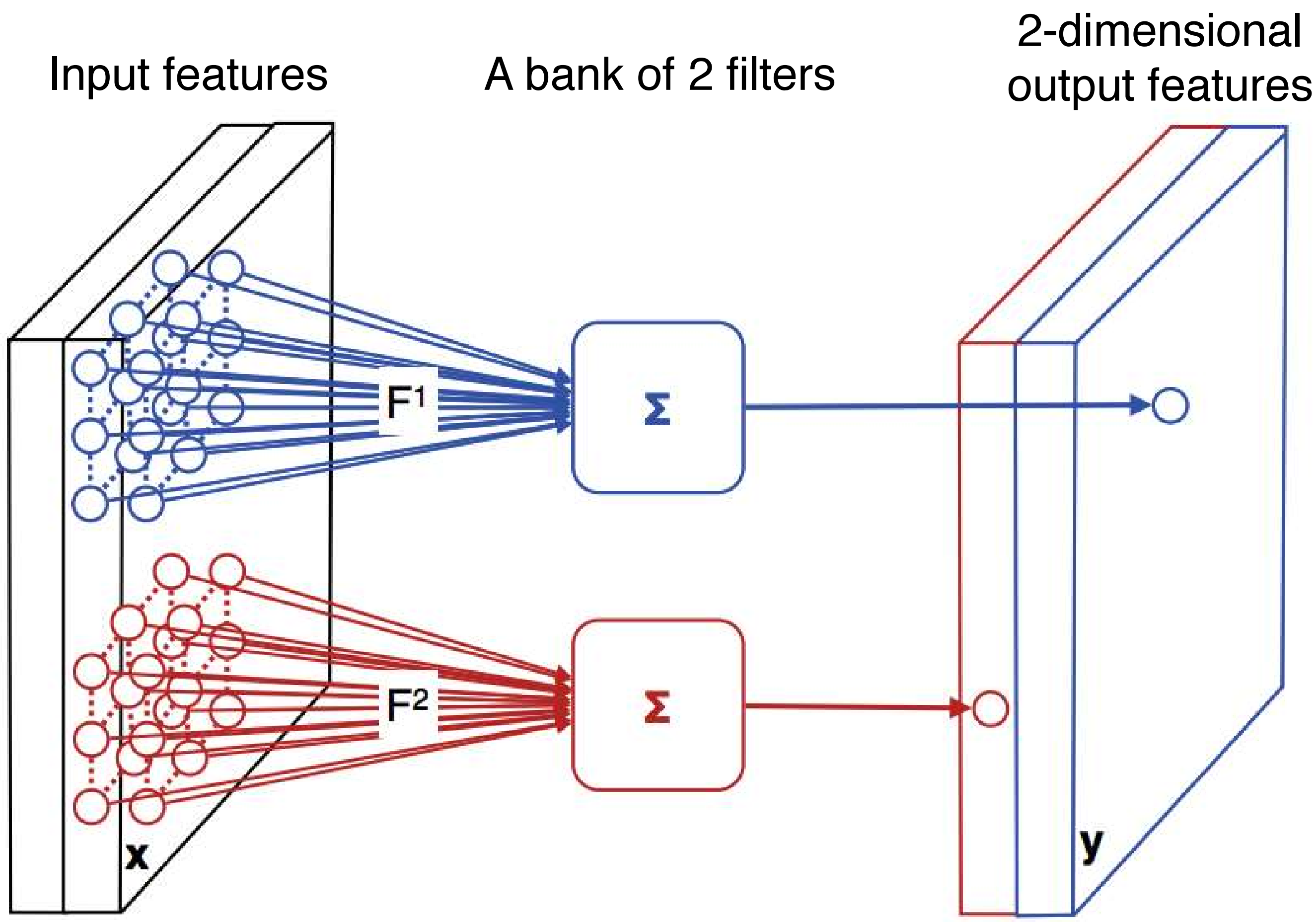
When mapping from

$$\mathbf{x}^{(l)} \in \mathbb{R}^{H \times W \times C^{(l)}} \rightarrow \mathbf{x}^{(l+1)} \in \mathbb{R}^{H \times W \times C^{(l+1)}}$$

using an filter of spatial extent $M \times N$

Number of parameters per filter: $M \times N \times C^{(l)}$

Number of filters: $C^{(l+1)}$



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

[Figure from Andrea Vedaldi]

Image classification

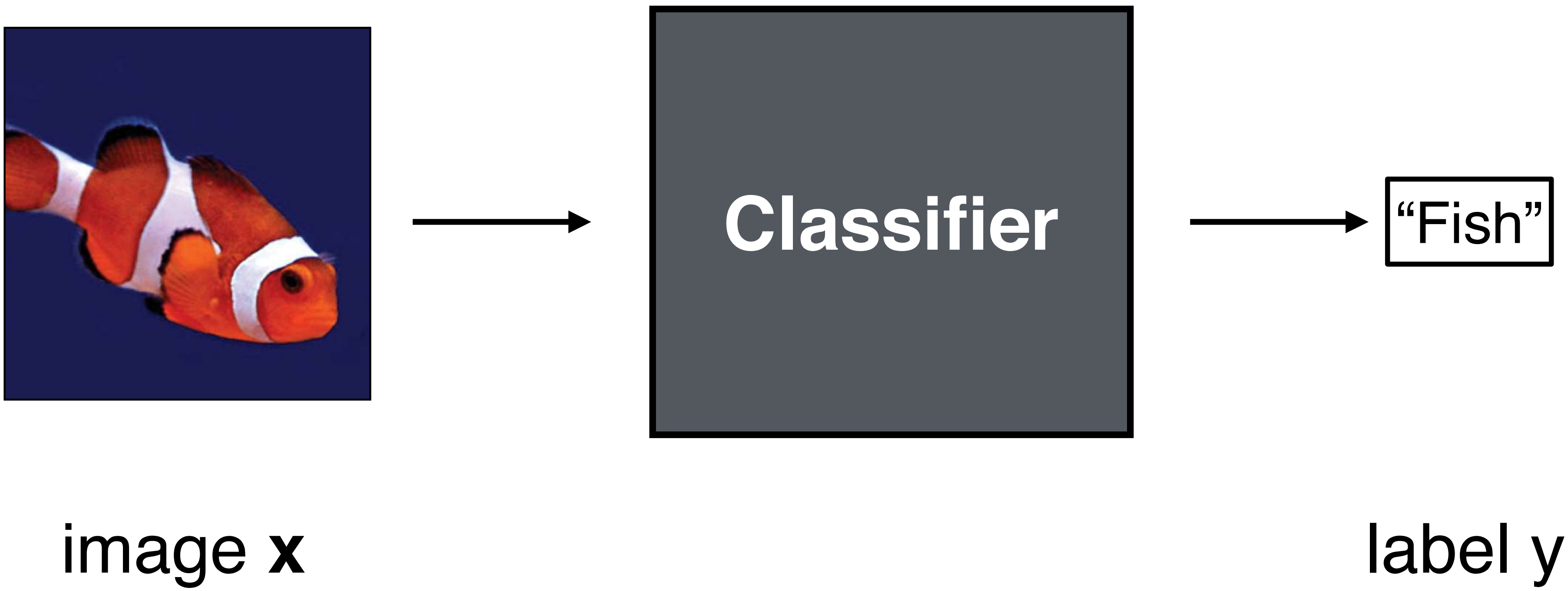


Image classification

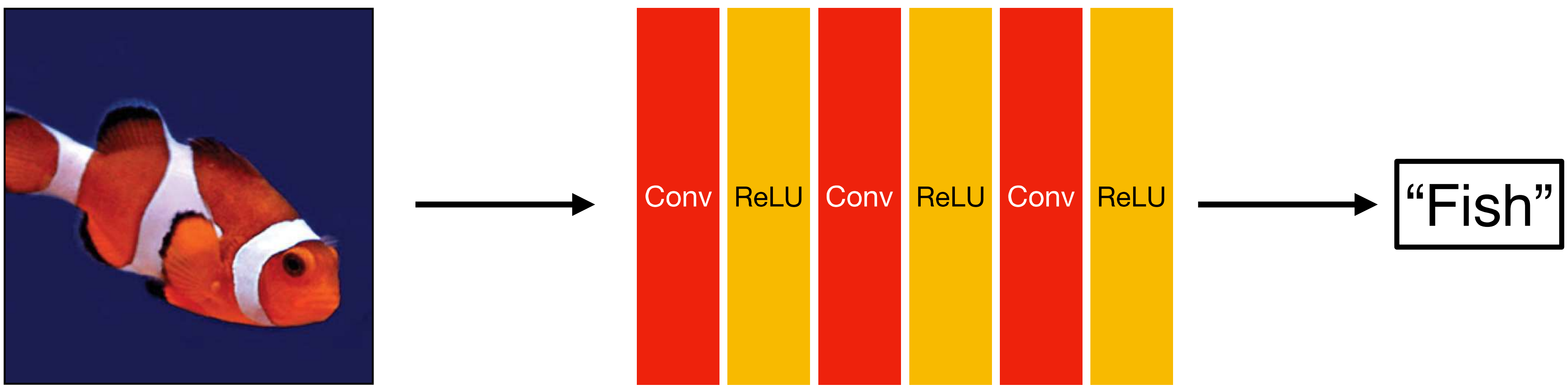
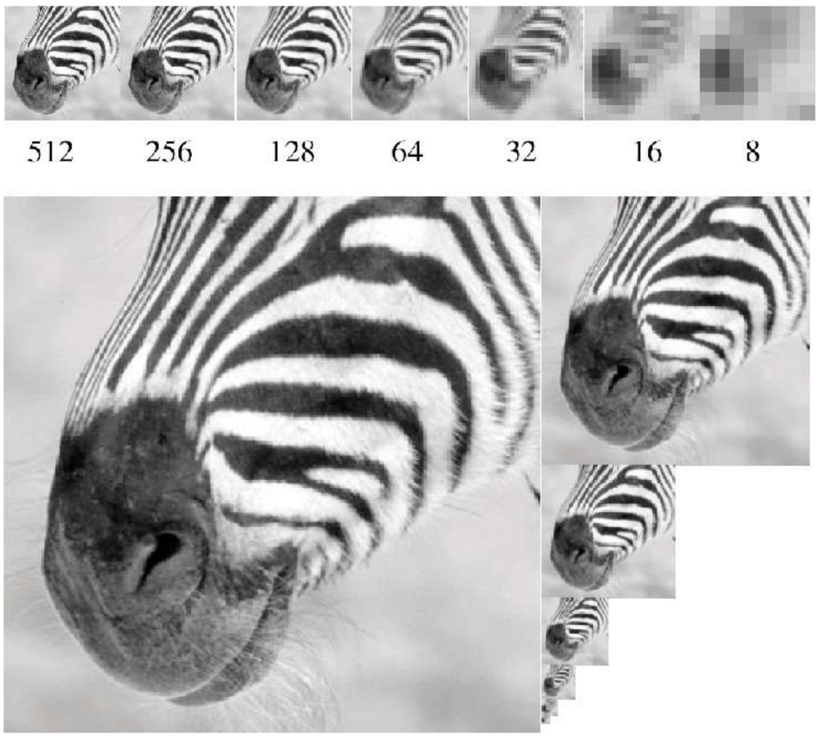


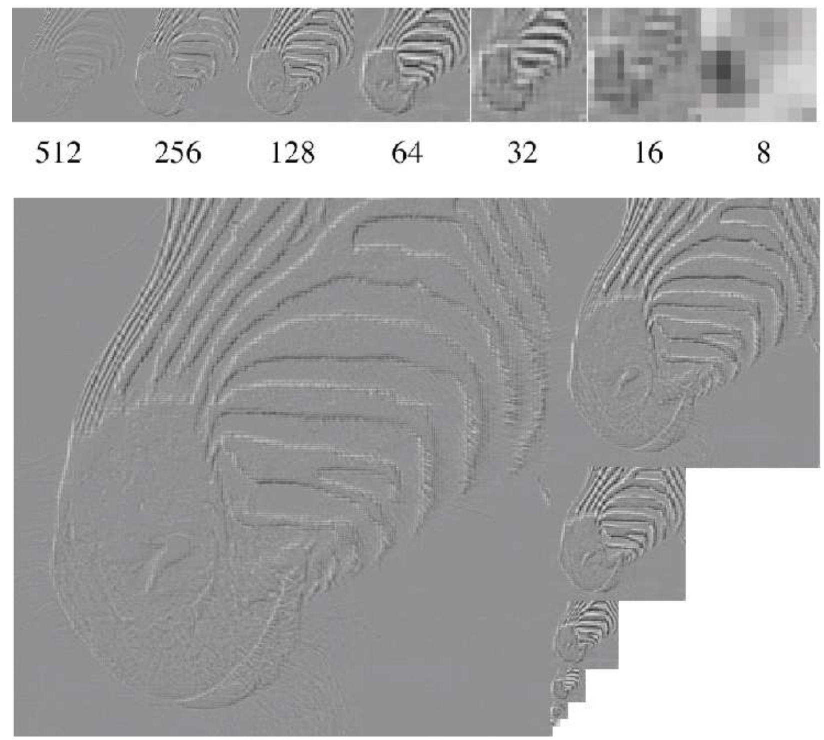
image **x**

label y

Multiscale representations are great!



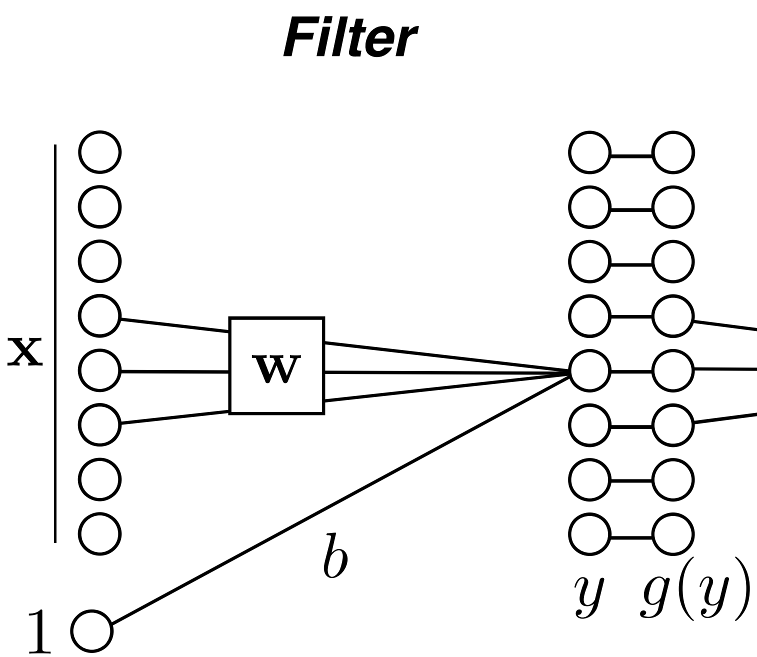
Gaussian Pyr



Laplacian Pyr

How can we use multi-scale modeling in Convnets?

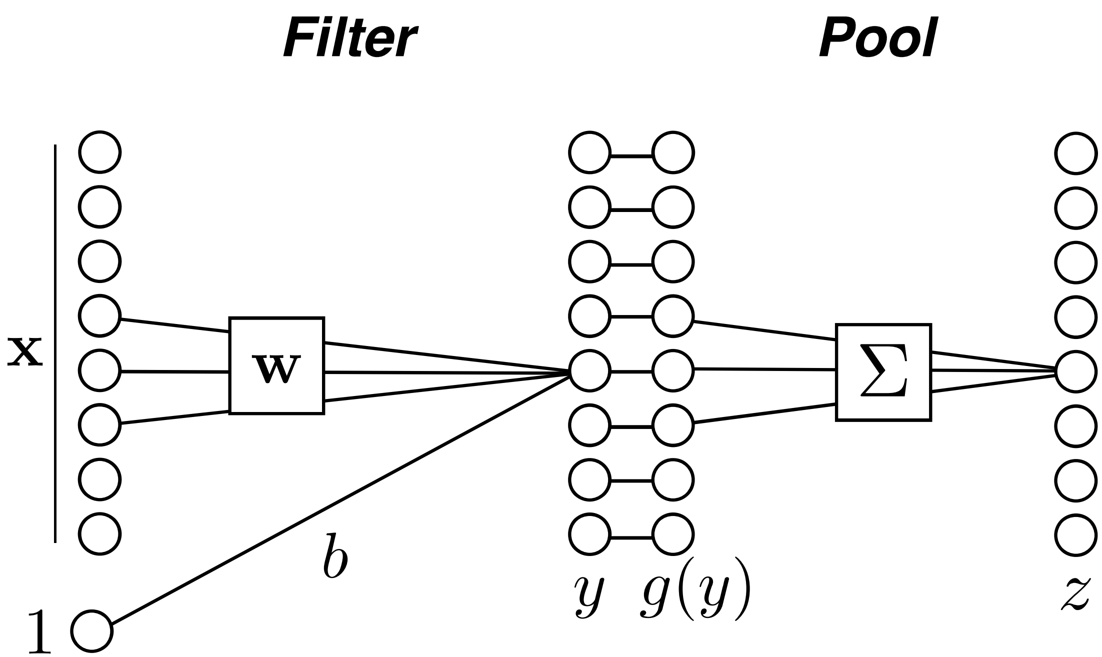
Pooling



Max pooling

$$z_k = \max_{j \in \mathcal{N}(j)} g(y_j)$$

Pooling



Max pooling

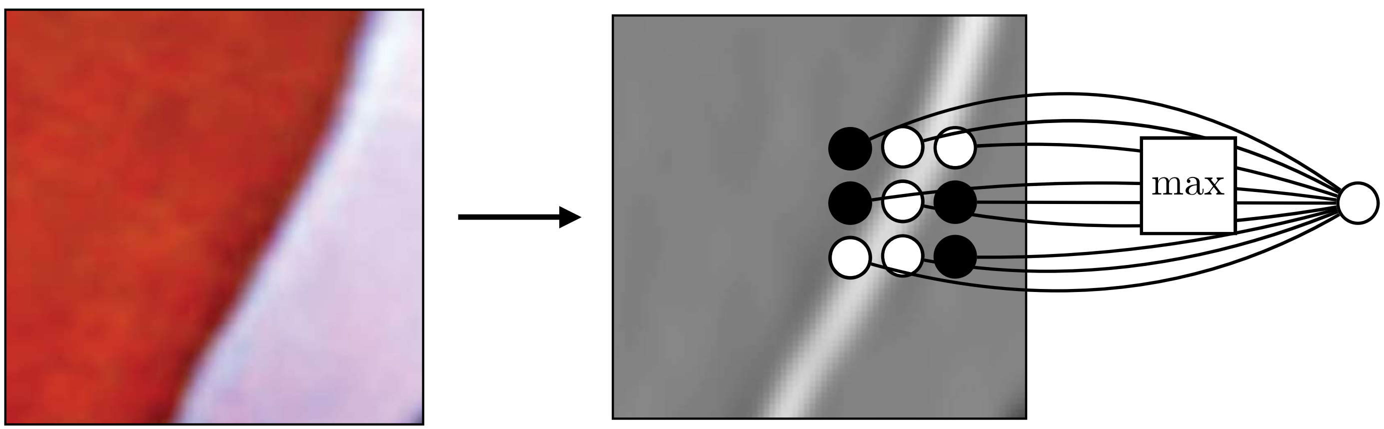
$$z_k = \max_{j \in \mathcal{N}(j)} g(y_j)$$

Mean pooling

$$z_k = \frac{1}{|\mathcal{N}|} \sum_{j \in \mathcal{N}(j)} g(y_j)$$

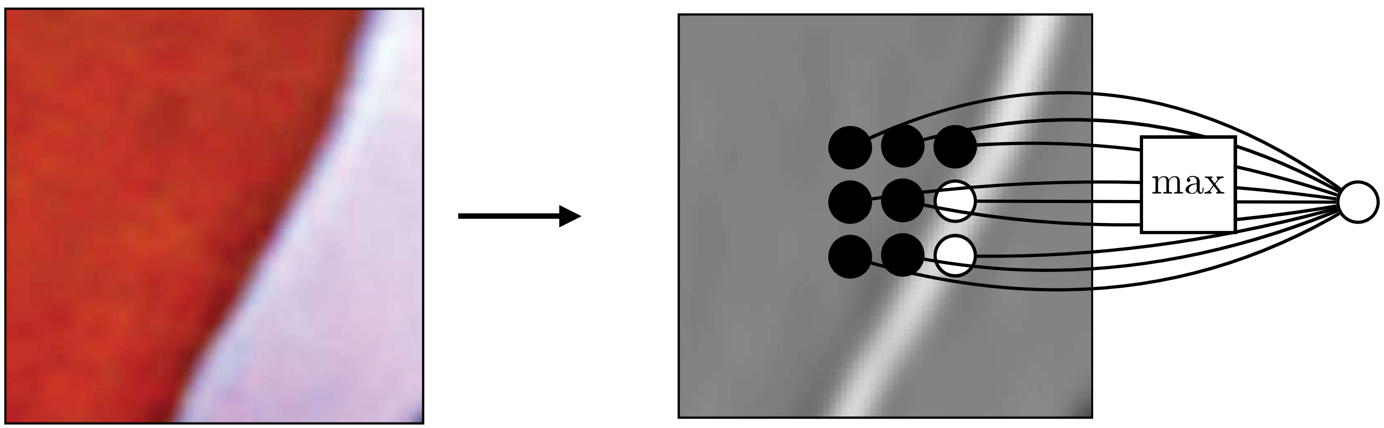
Pooling — Why?

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



Pooling — Why?

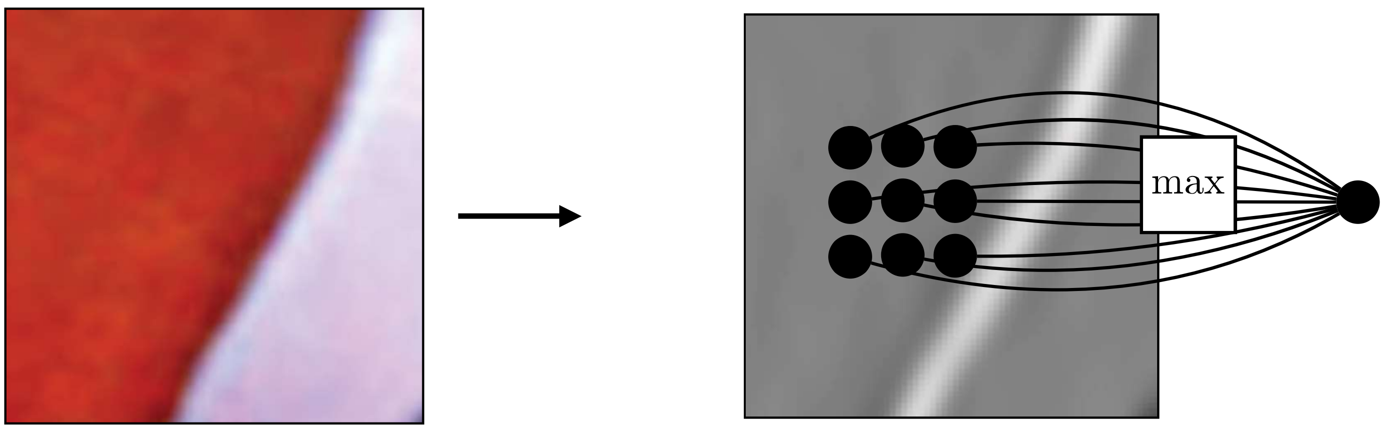
Pooling across spatial locations achieves stability w.r.t. small translations:



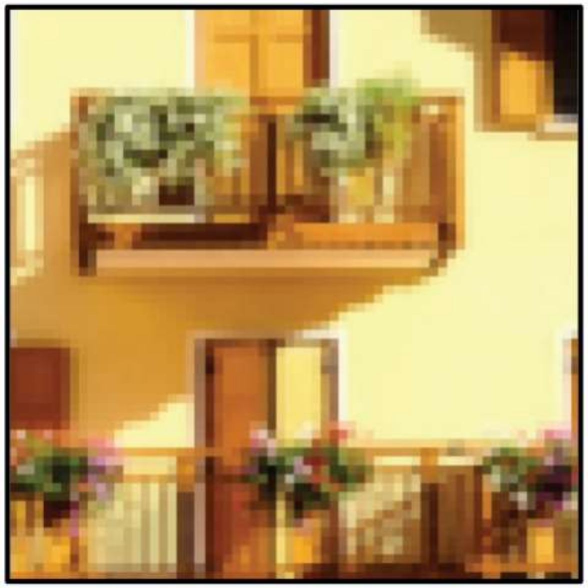
large response regardless of exact position of edge

Pooling — Why?

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



CNNs are stable w.r.t. diffeomorphisms

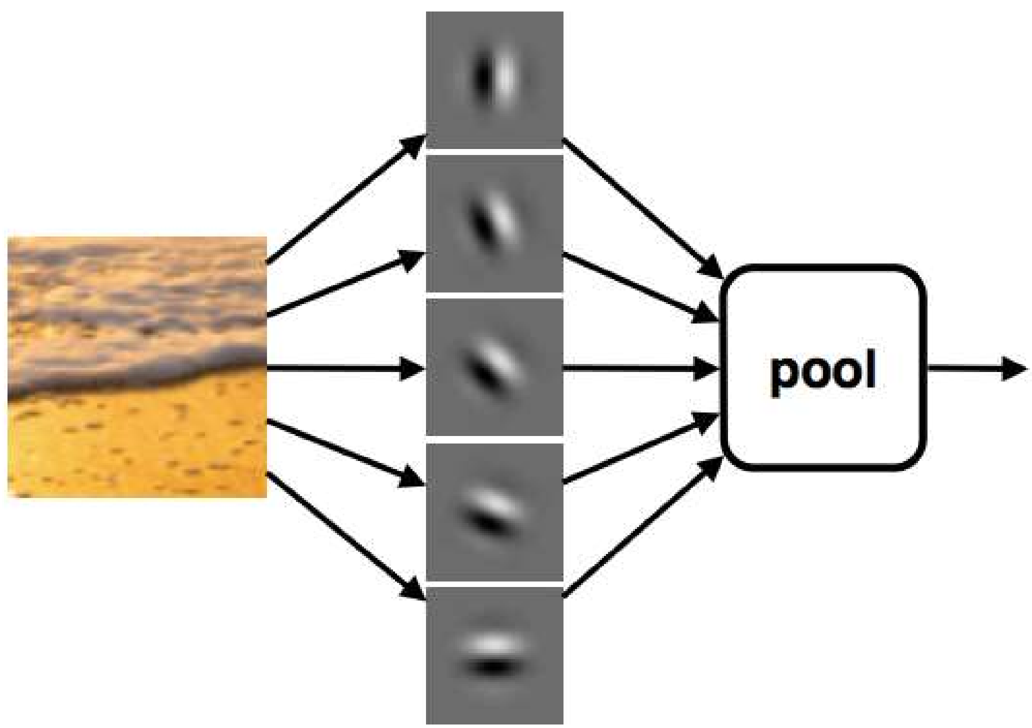


$\approx$

[“Unreasonable effectiveness of Deep Features as a Perceptual Metric”, Zhang et al. 2018]

Pooling — Why?

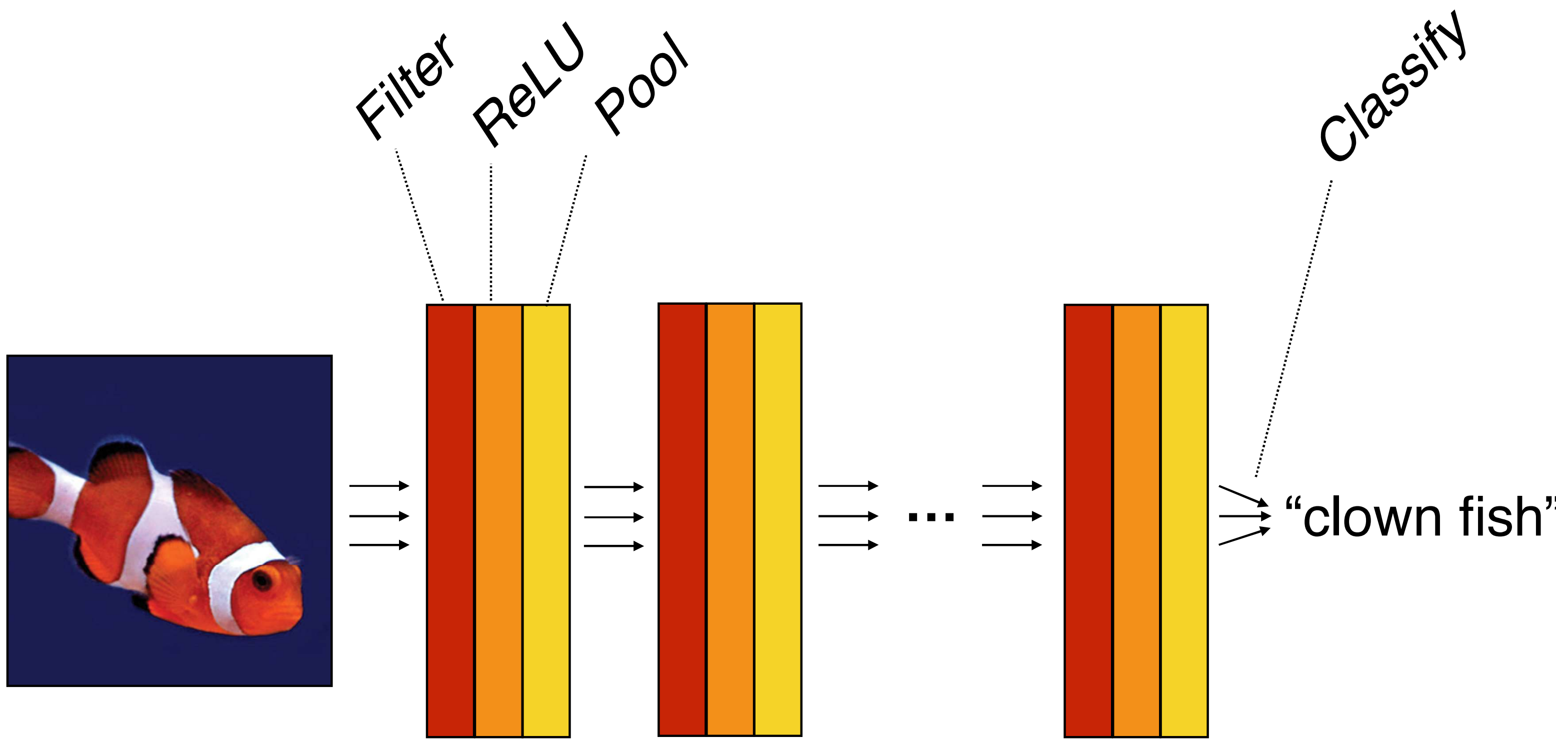
Pooling across feature channels (filter outputs) can achieve other kinds of invariances:



large response for any edge, regardless of its orientation

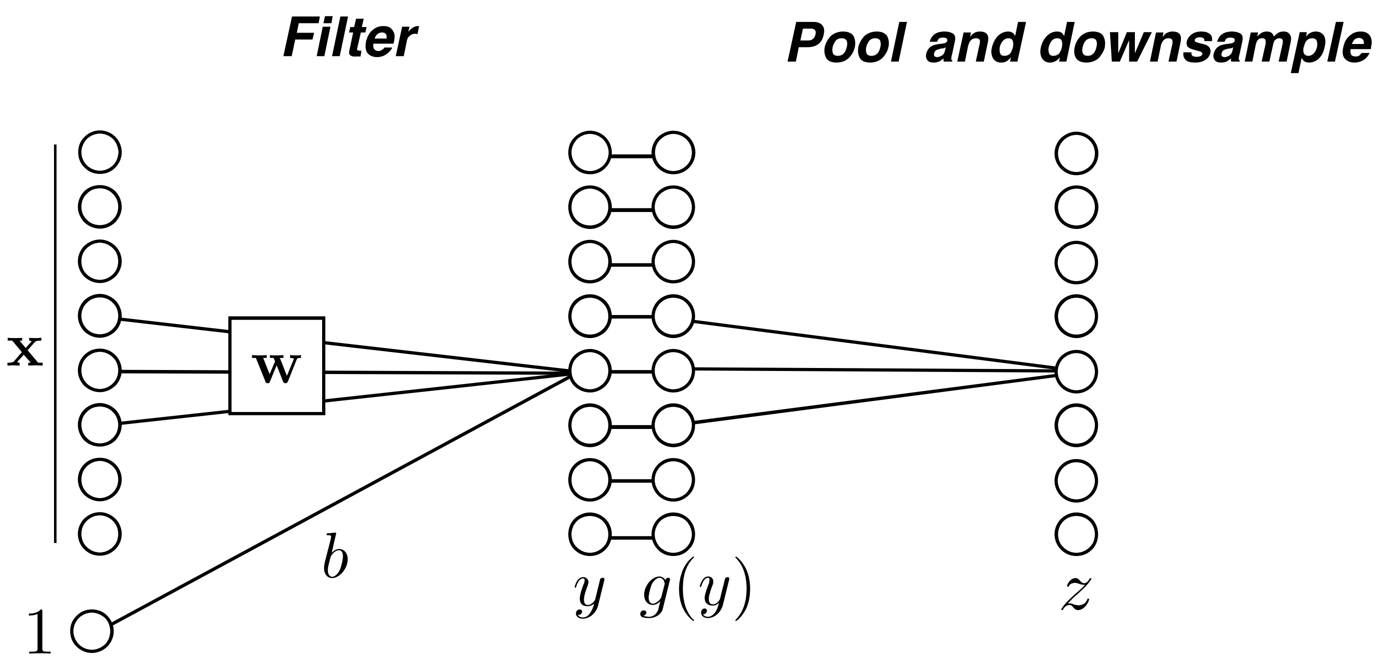
[Derived from slide by Andrea Vedaldi]

Computation in a neural net

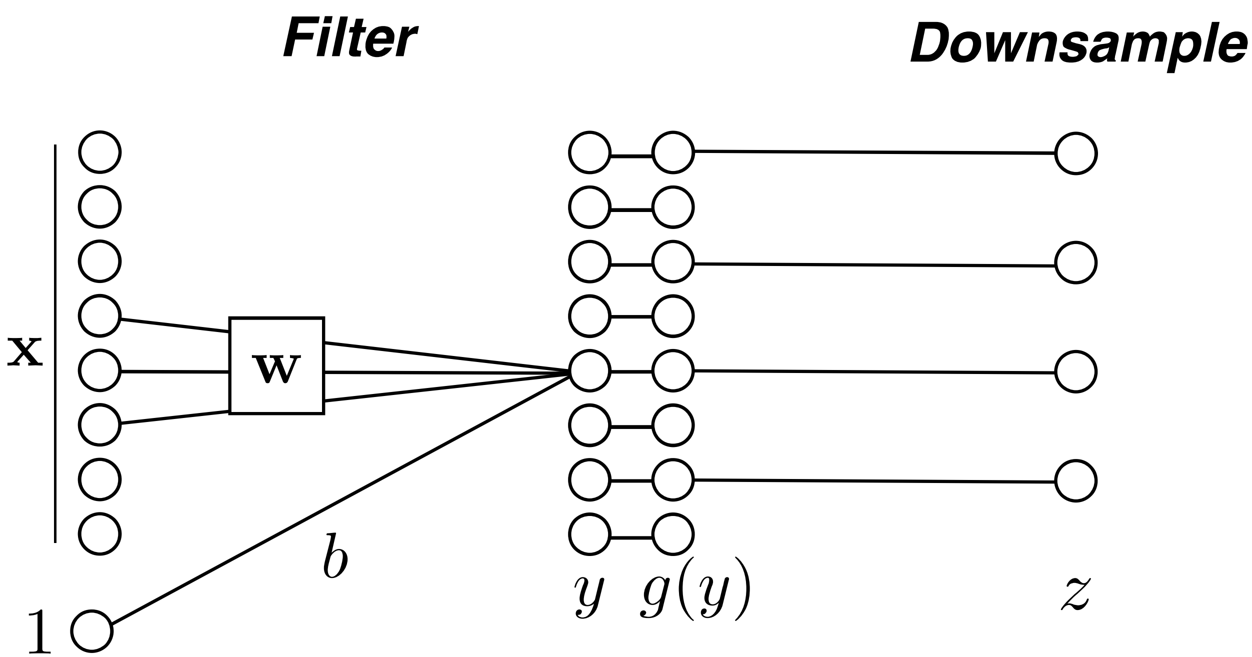


$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$

Downsampling

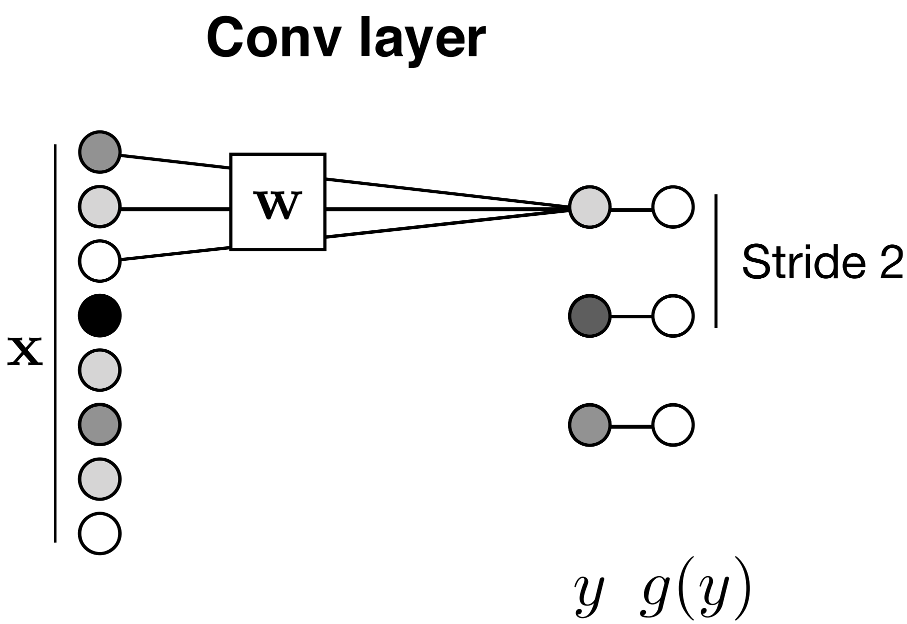


Downsampling



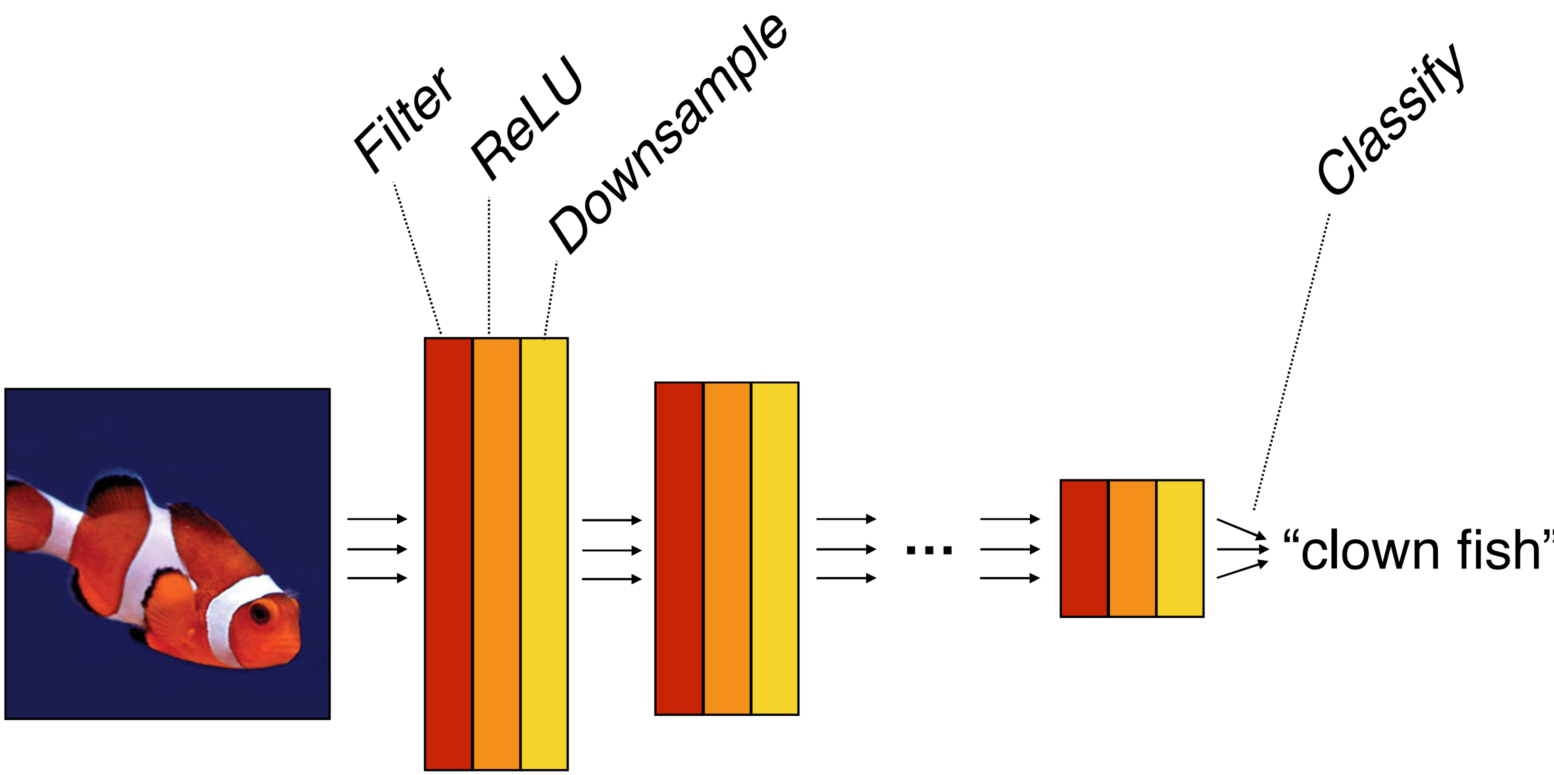
$$\mathbb{R}^{H^{(l)} \times W^{(l)} \times C^{(l)}} \rightarrow \mathbb{R}^{H^{(l+1)} \times W^{(l+1)} \times C^{(l+1)}}$$

Strided operations



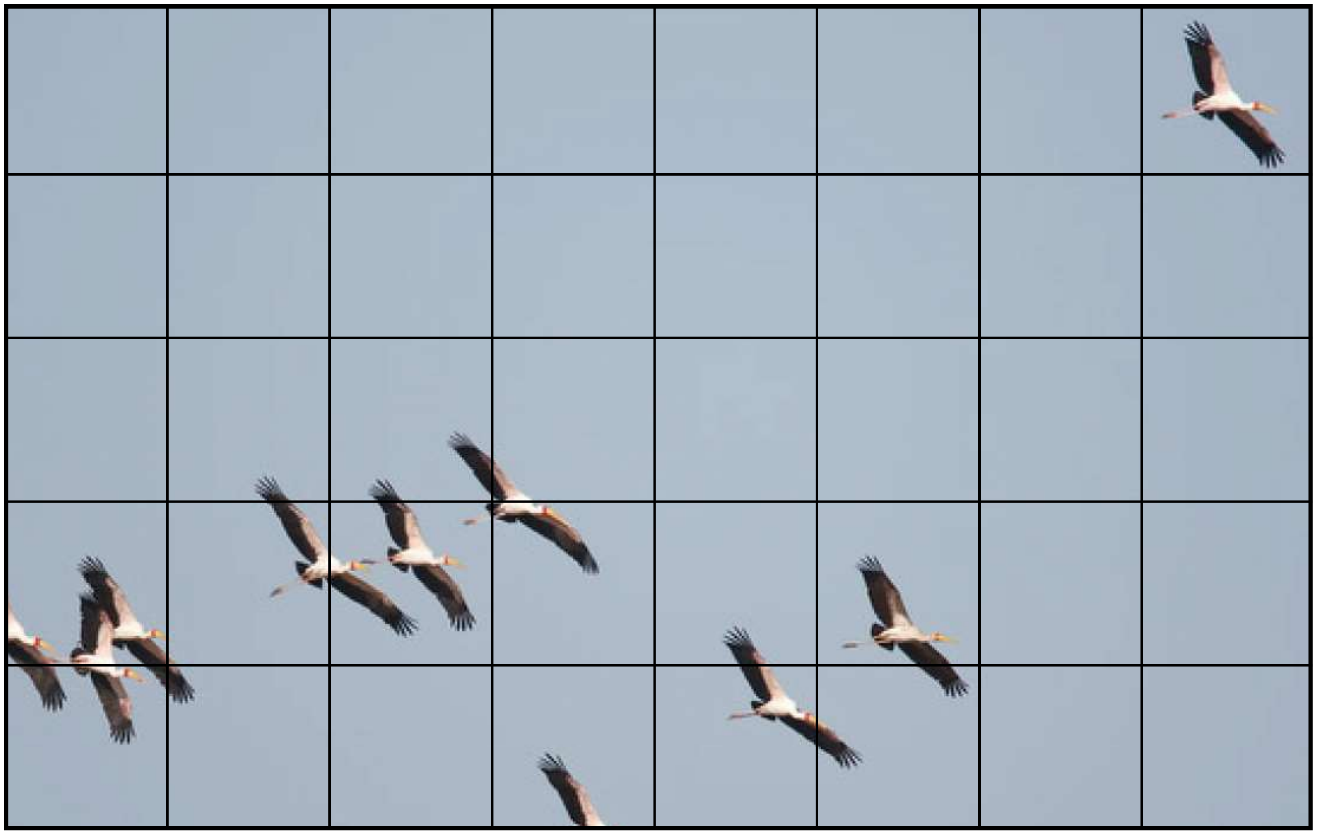
Strided operations combine a given operation (convolution or pooling) and downsampling into a single operation.

Computation in a neural net

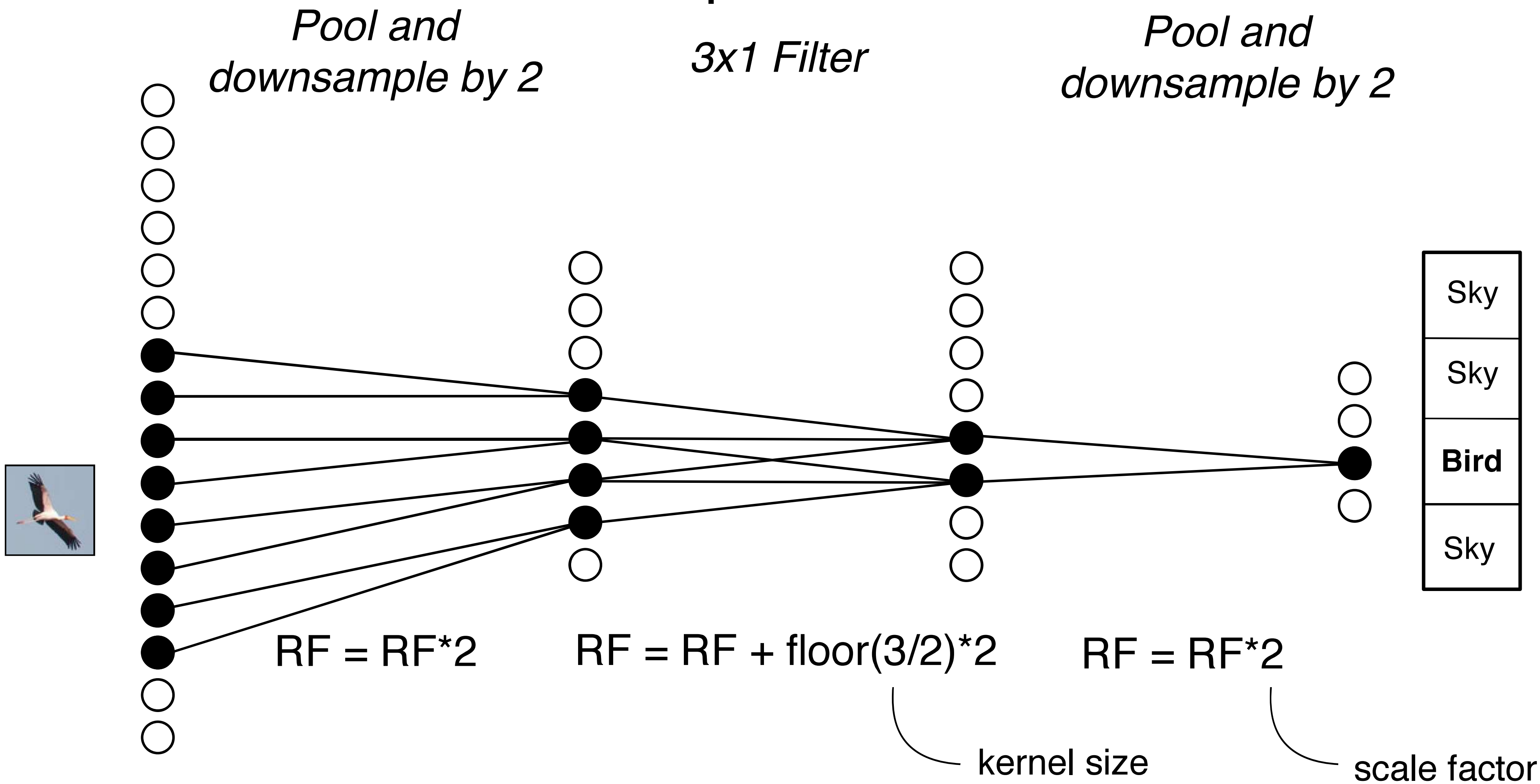


$$f(\mathbf{x}) = f_L(\ldots f_2(f_1(\mathbf{x})))$$

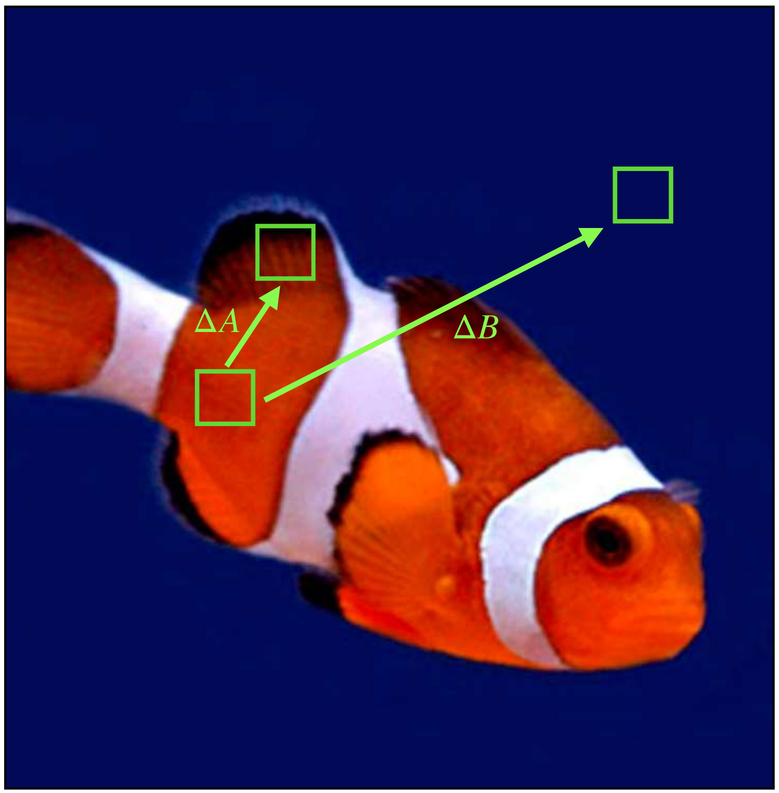
Receptive fields



Receptive fields



Local vs. global processing



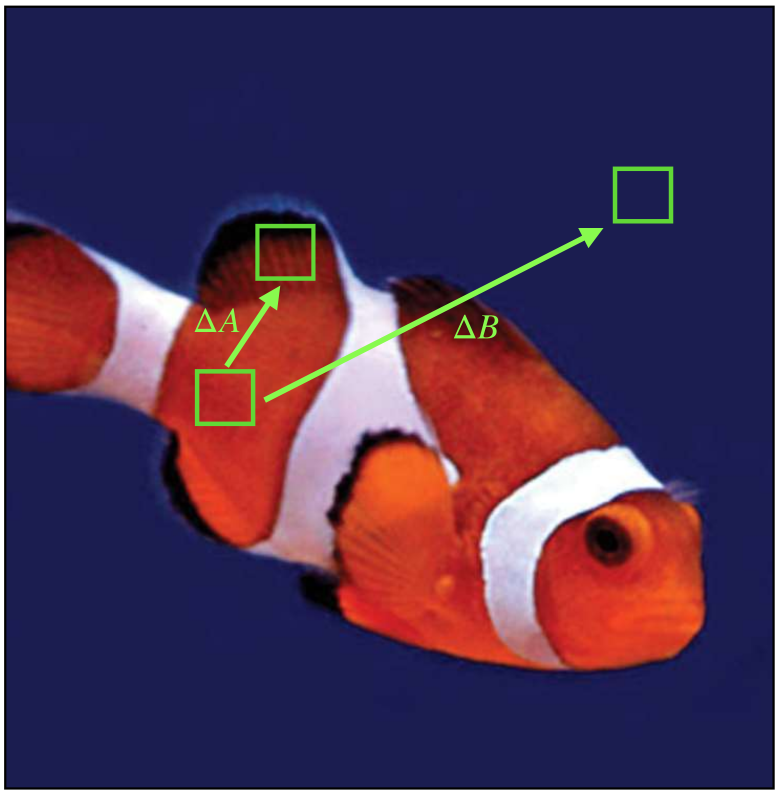
Across all images, which is higher:

(1) correlation between points with distance ΔA

(2) correlation between points with distance ΔB

(3) can't say

Local vs. global processing



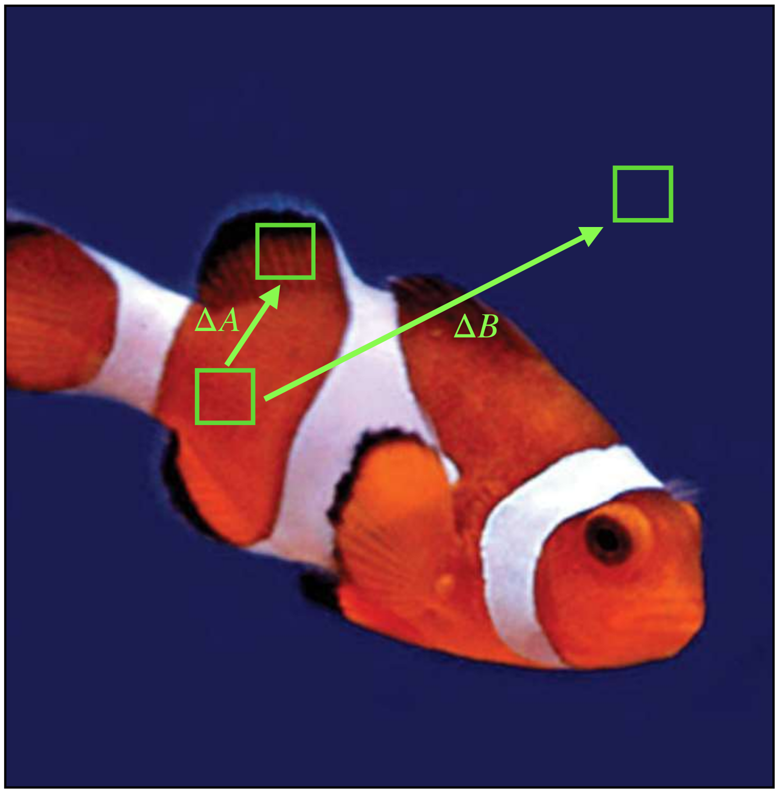
Across all images, which is higher:

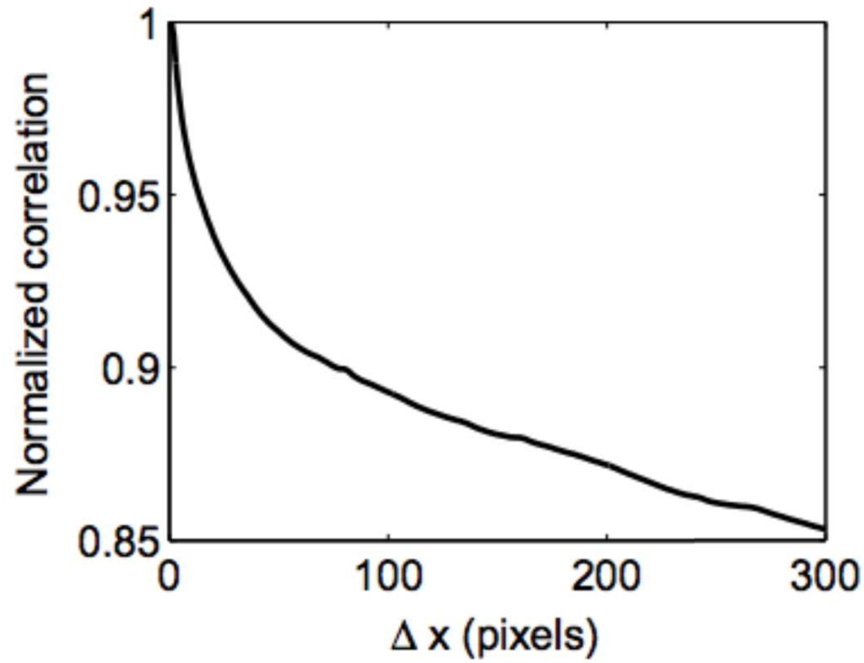
(1) correlation between points with distance ΔA

(2) correlation between points with distance ΔB

(3) can't say

Local vs. global processing





[Simoncelli: Statistical Modelling of Photographic Images, 2005]

CNNs — Why?

Statistical dependences between pixels decay as a power law of distance between the pixels.

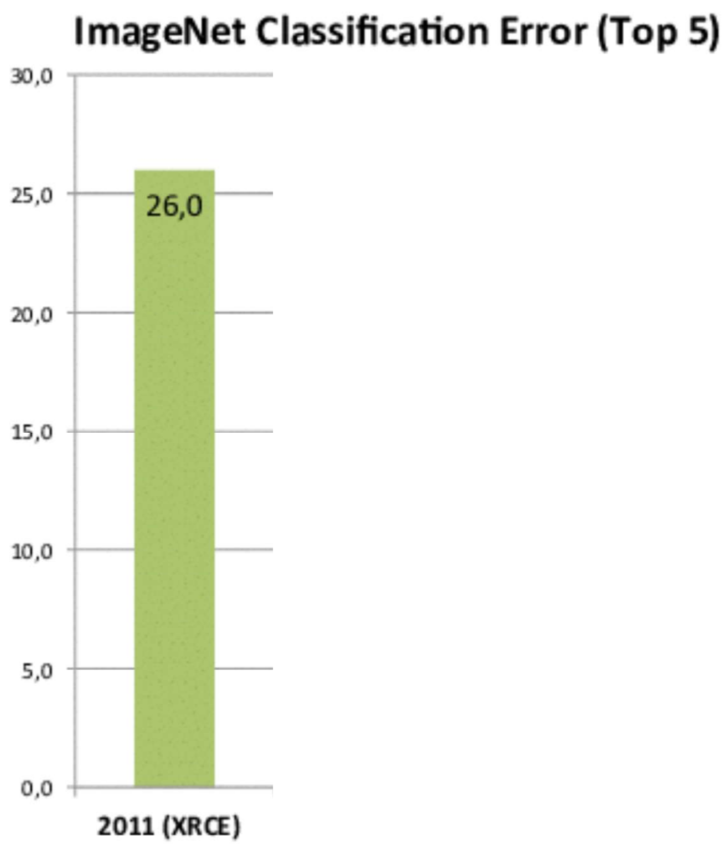
It is therefore often sufficient to model local dependences only. —> **Convolution**

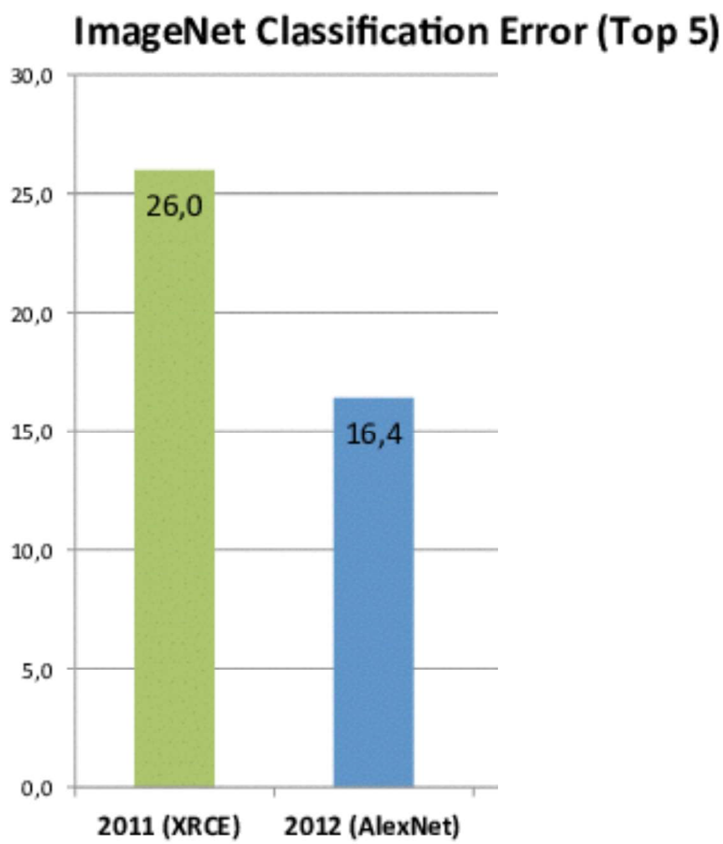
More generally, we should allocate parameters that model dependencies in proportion to the strength of those dependences. —> **Multiscale, hierarchical representations**

[For more discussion, see "Why does Deep and Cheap Learning Work So Well?", Lin et al. 2017]

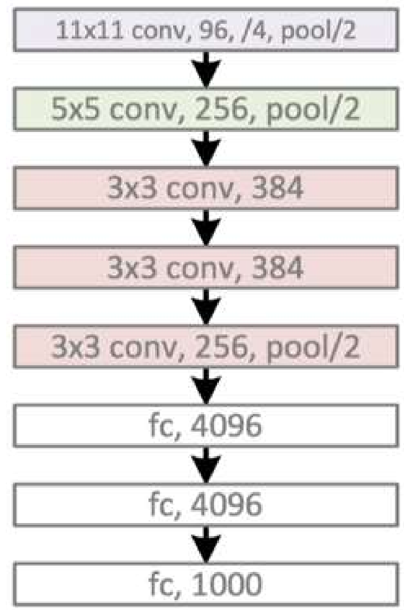
Some networks

… and what makes them work


ImageNet Classification Error (Top 5)
30,0
25,0
20,0
15,0
10,0
5,0
0,0
26,0
2011 (XRCE)



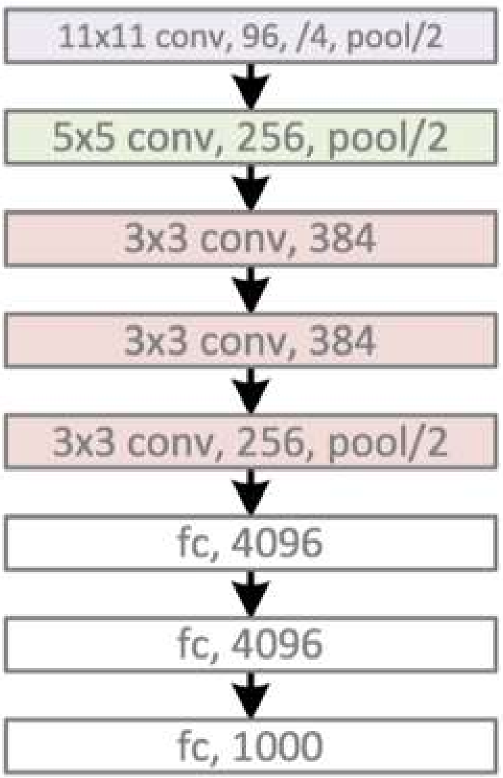
2012: AlexNet
5 conv. layers



Error: 16.4%

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

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

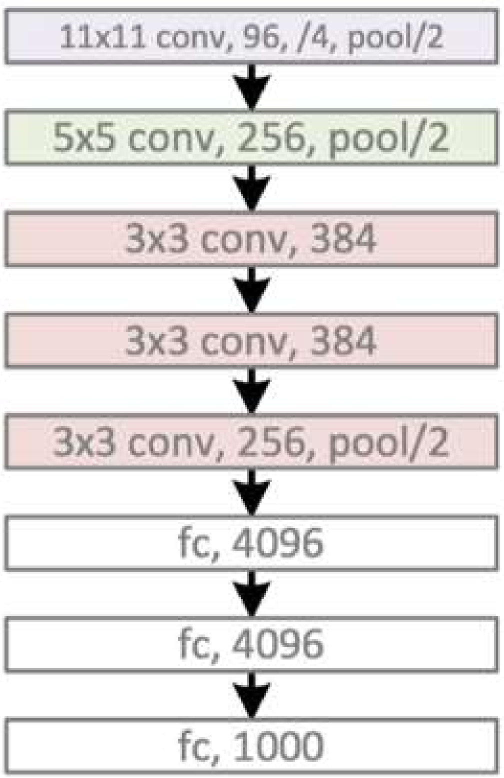
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1

[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

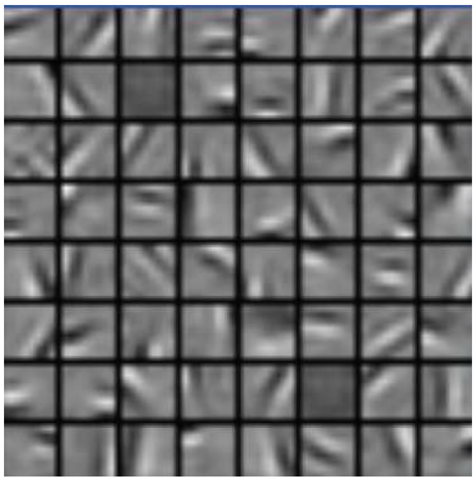
[1000] FC8: 1000 neurons (class scores)



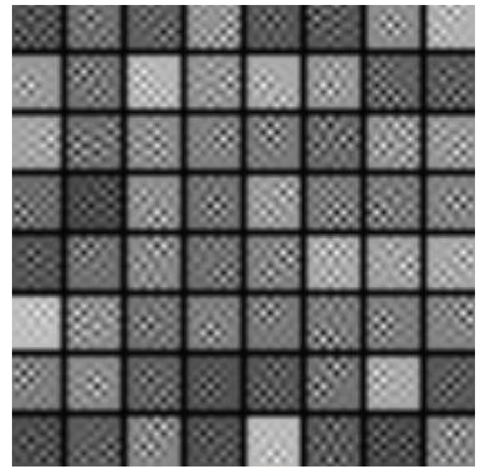
What filters are learned?

What filters are learned?

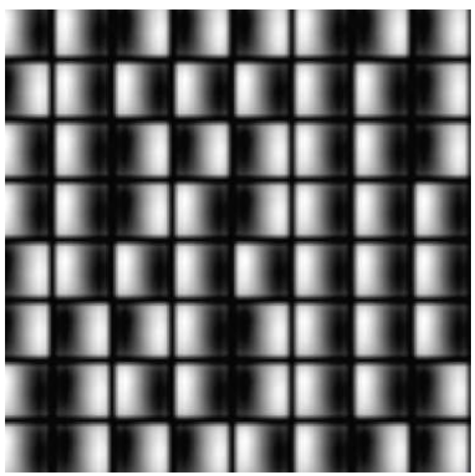
A



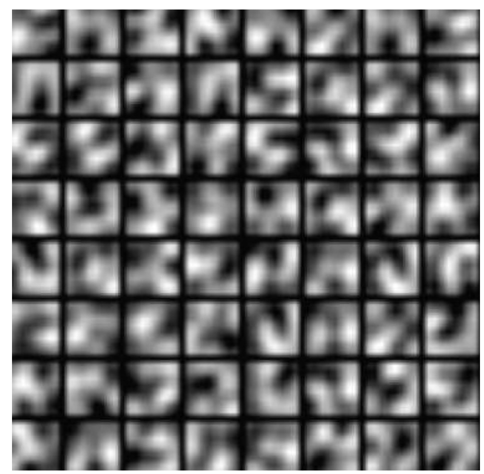
B



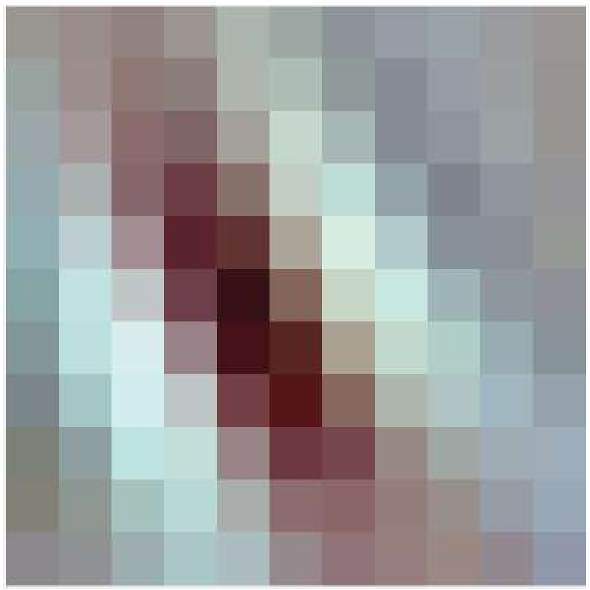
C

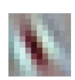


D

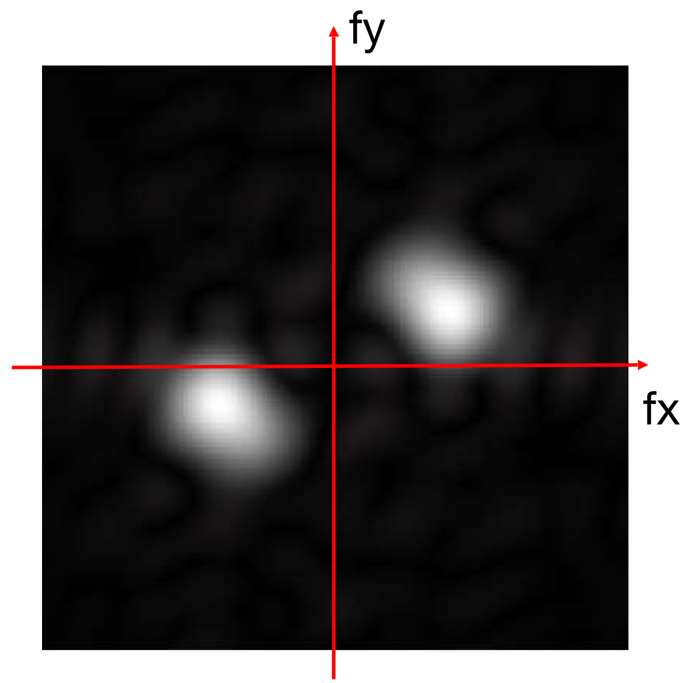


Get to know your units

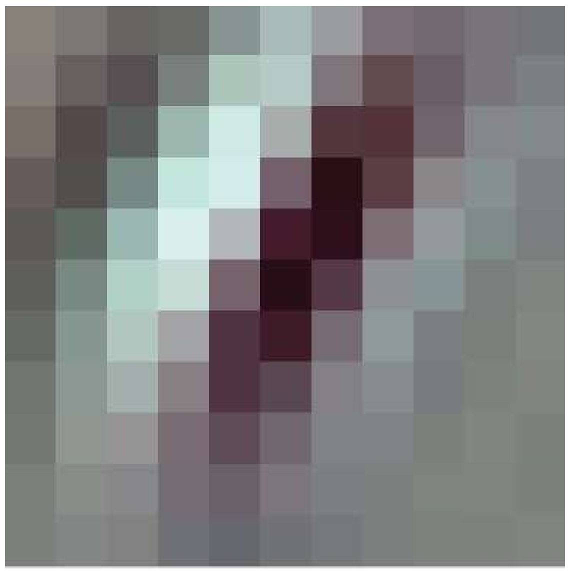


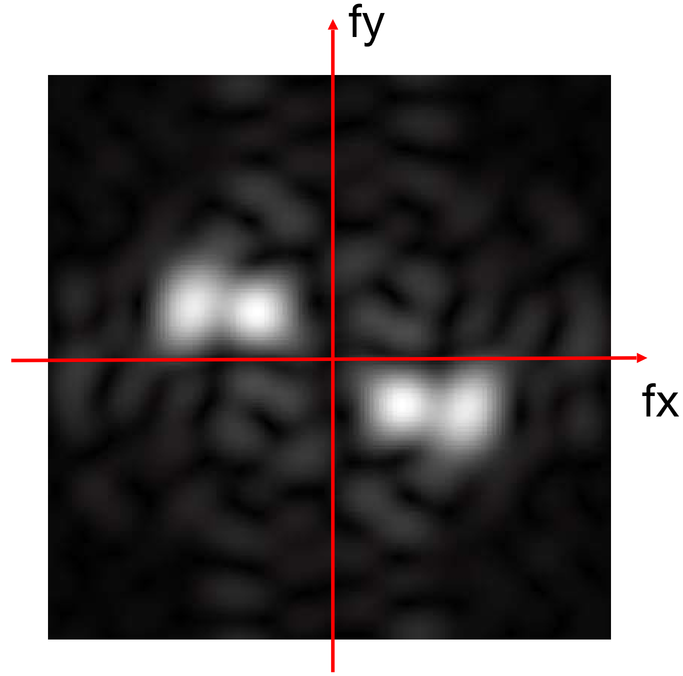


11x11 convolution kernel
(3 color channels)

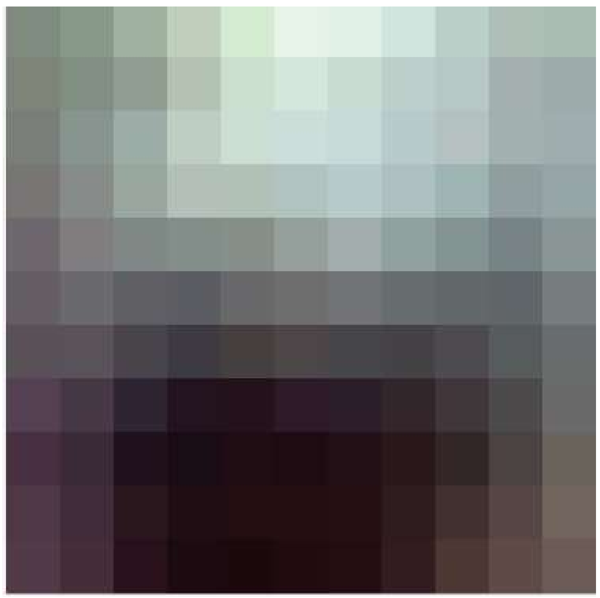


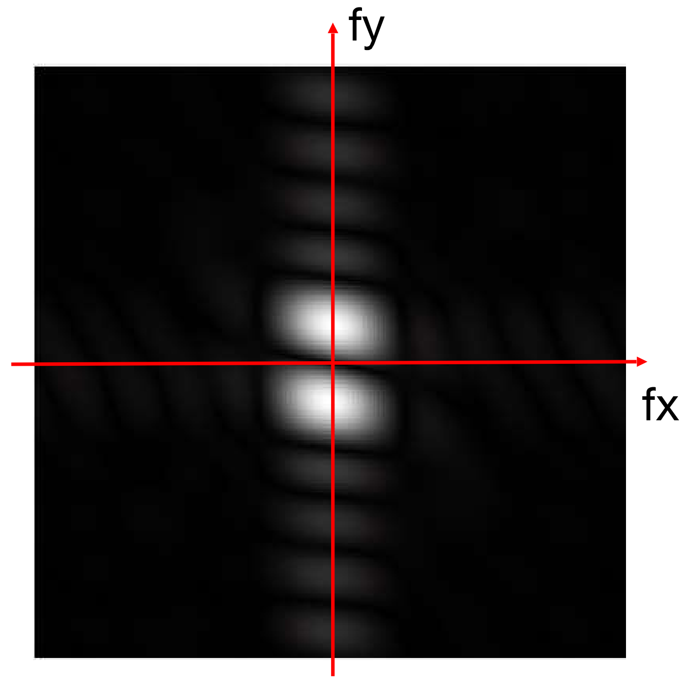
Get to know your units



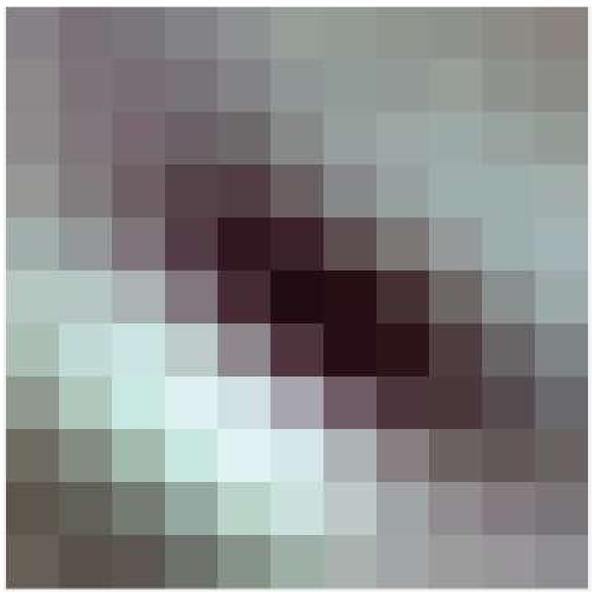


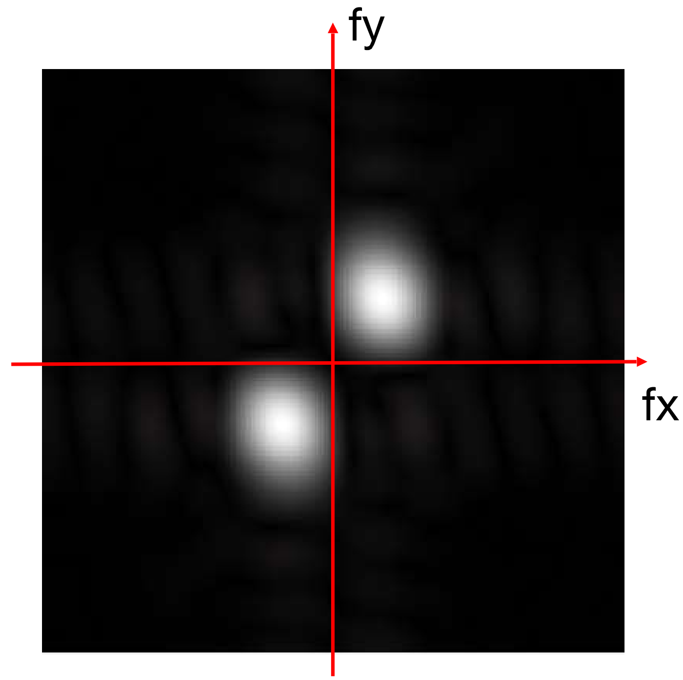
Get to know your units



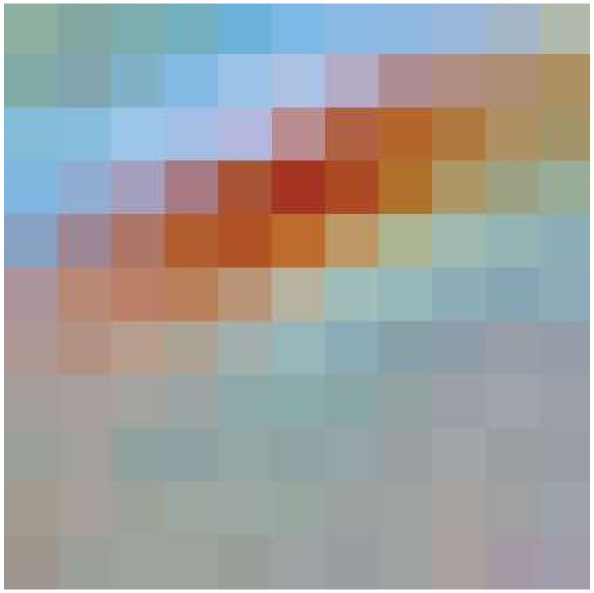


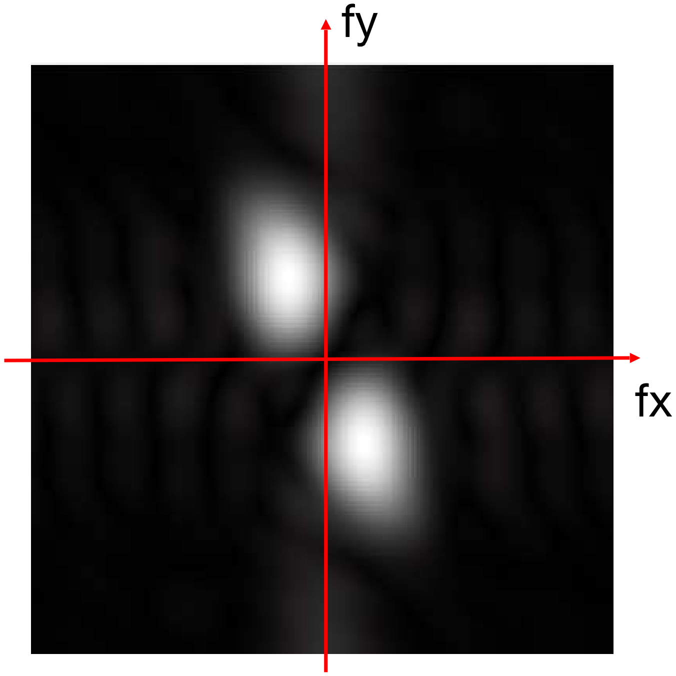
Get to know your units



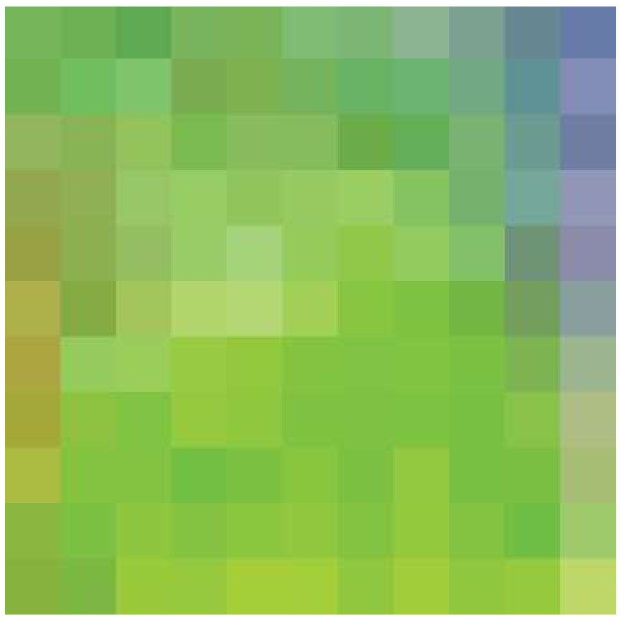


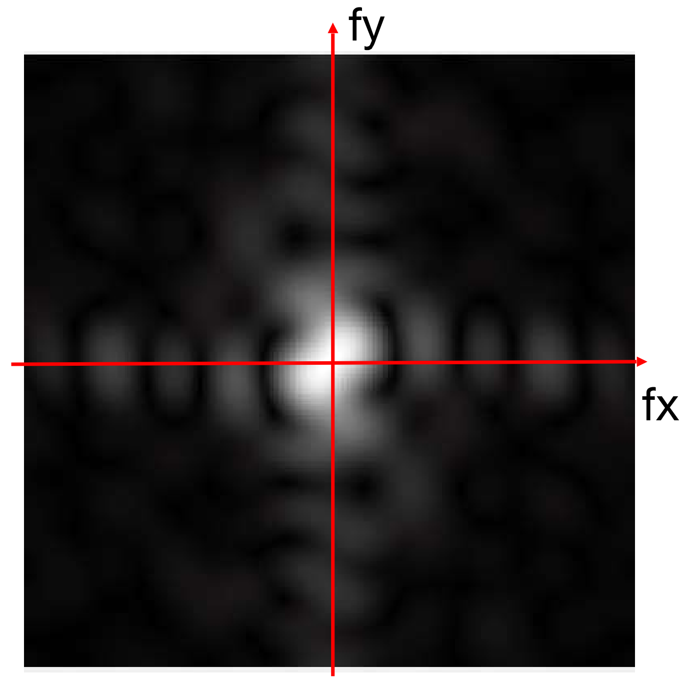
Get to know your units



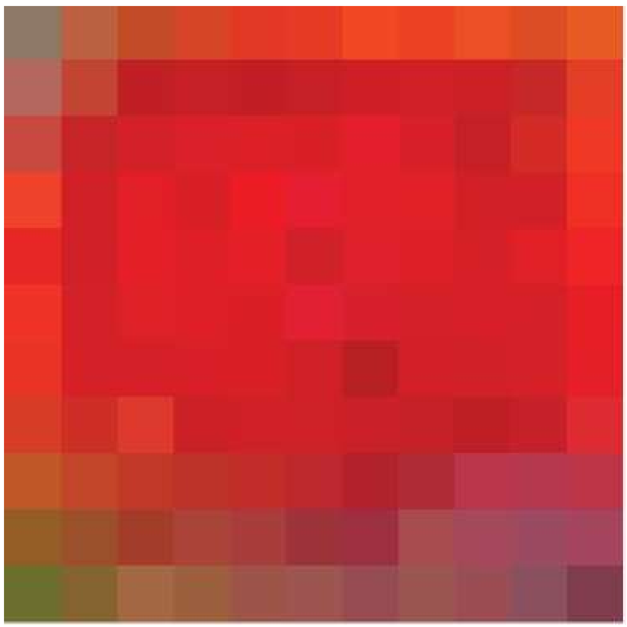


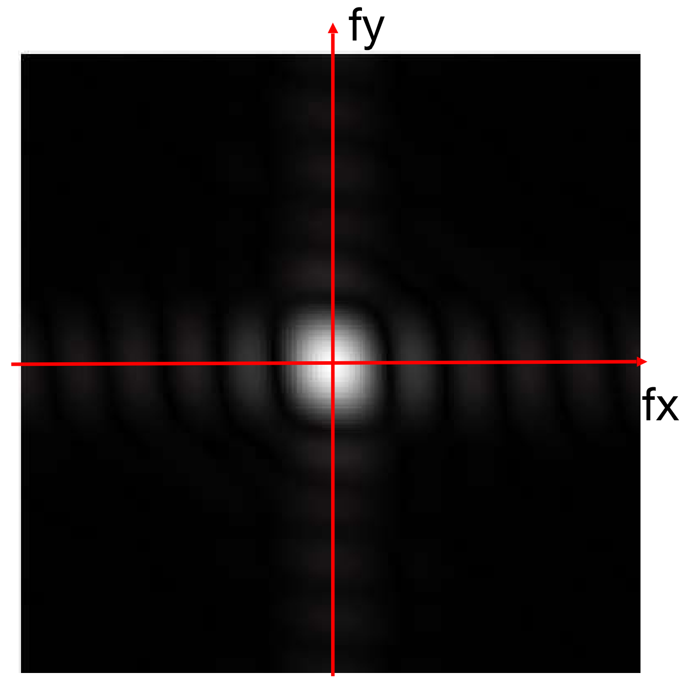
Get to know your units



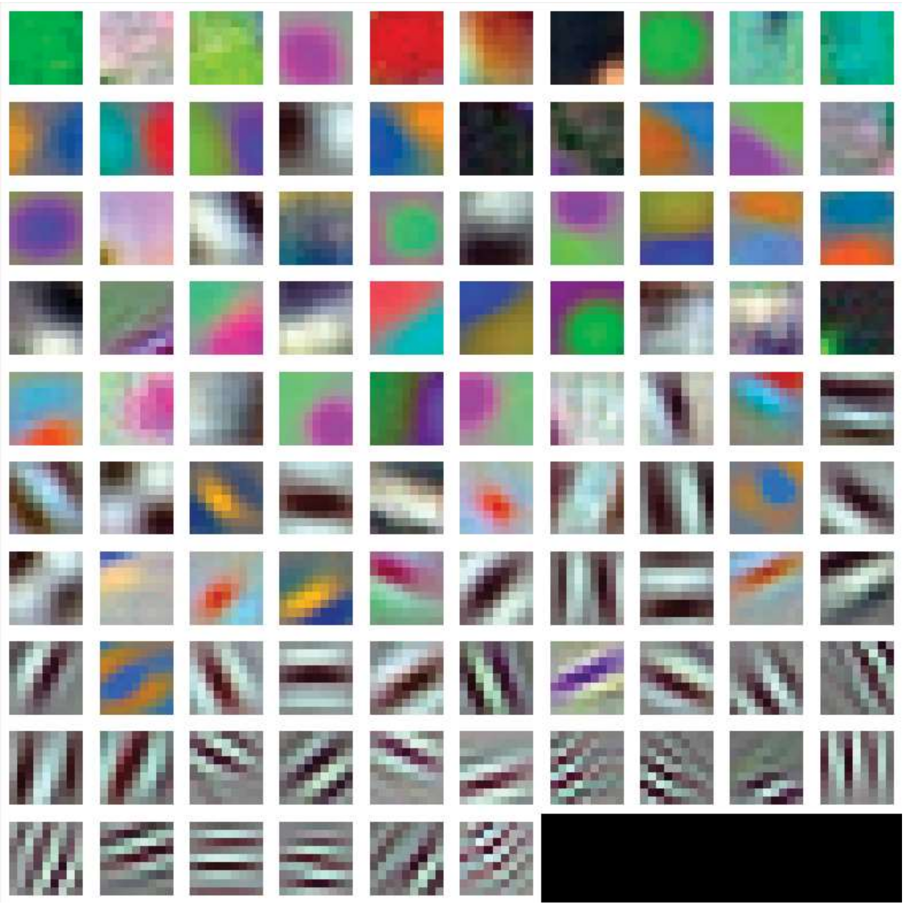


Get to know your units

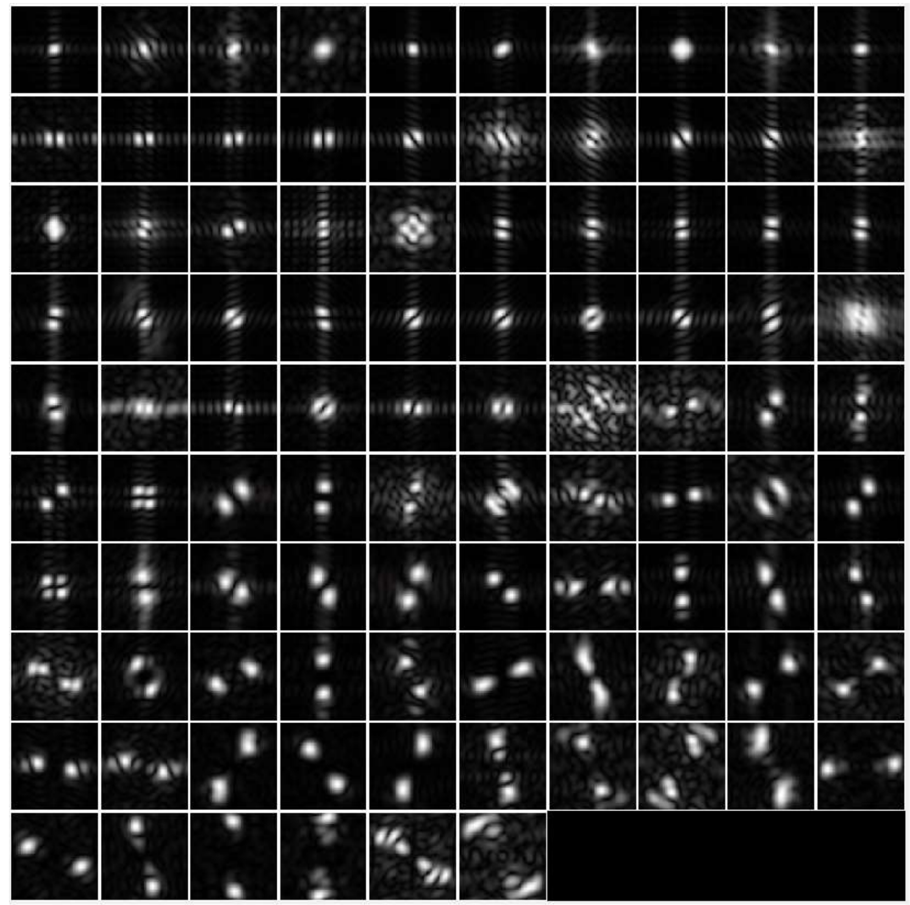




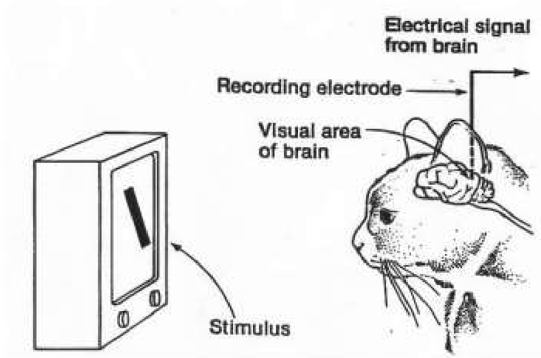
Get to know your units

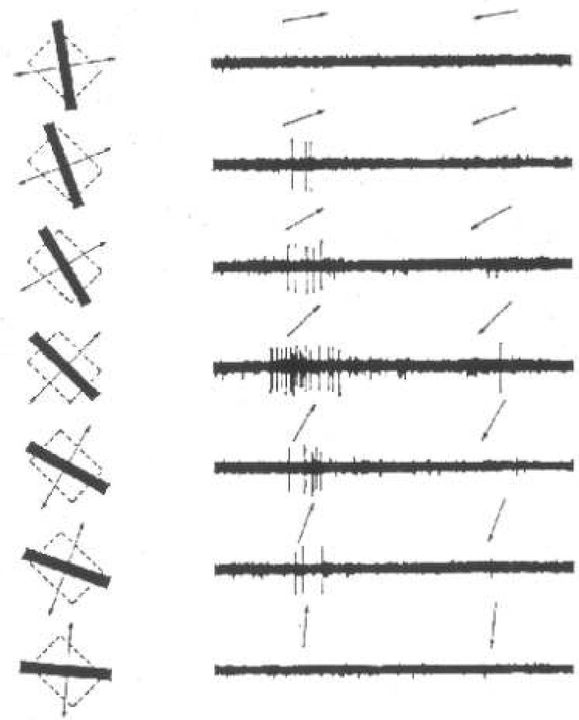


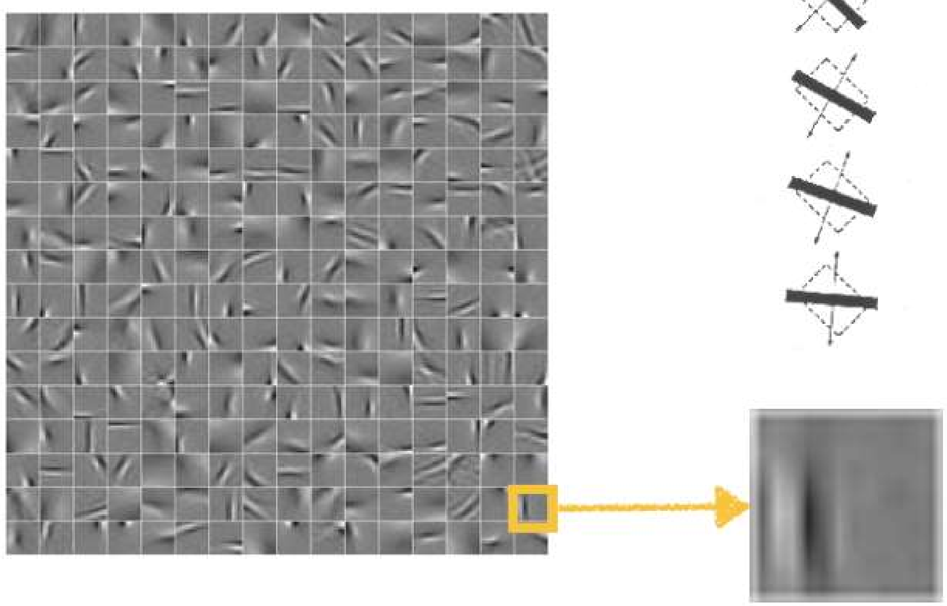
96 Units in conv1



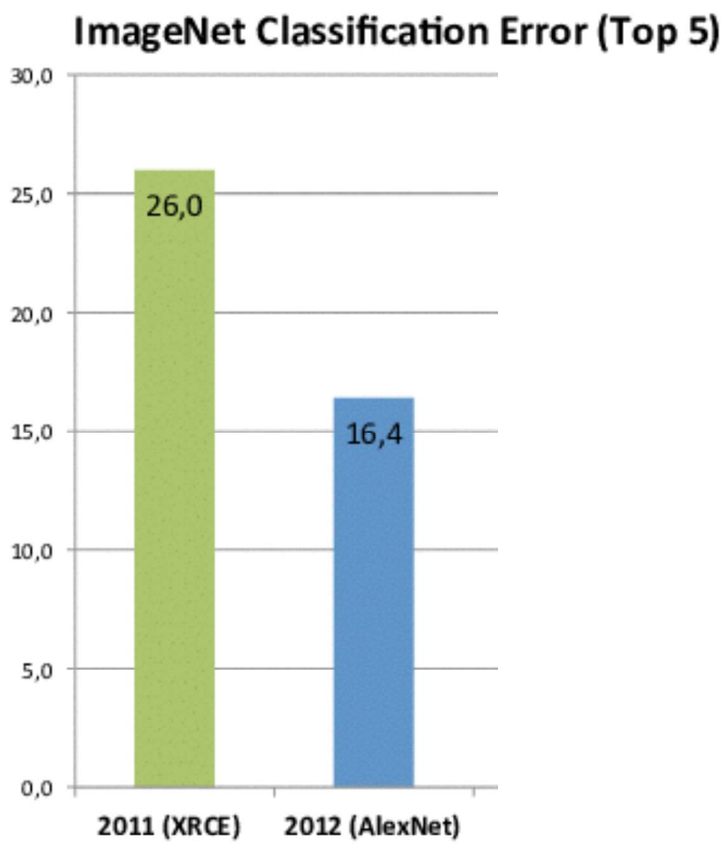
[Hubel and Wiesel 59]



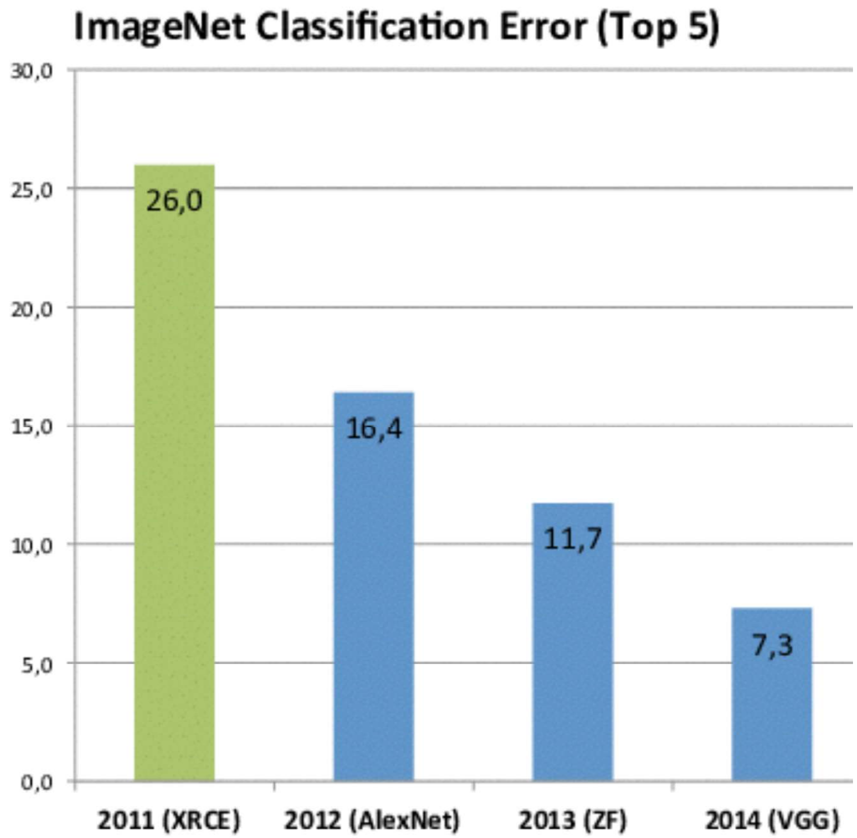


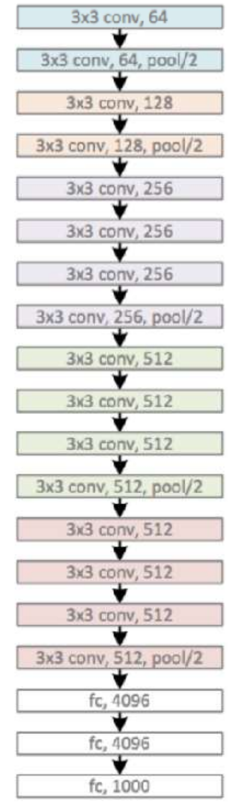


[Slide from Andrea Vedaldi]


ImageNet Classification Error (Top 5)
30,0
25,0
20,0
15,0
10,0
5,0
0,0
26,0
16,4
2011 (XRCE)
2012 (AlexNet)

2014: VGG
16 conv. layers



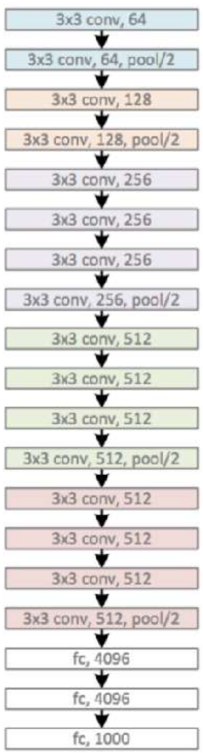


Error: 7.3%

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

VGG-Net [Simonyan & Zisserman, 2015]

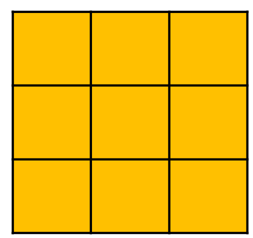
2014: VGG
16 conv. layers



Error: 7.3%

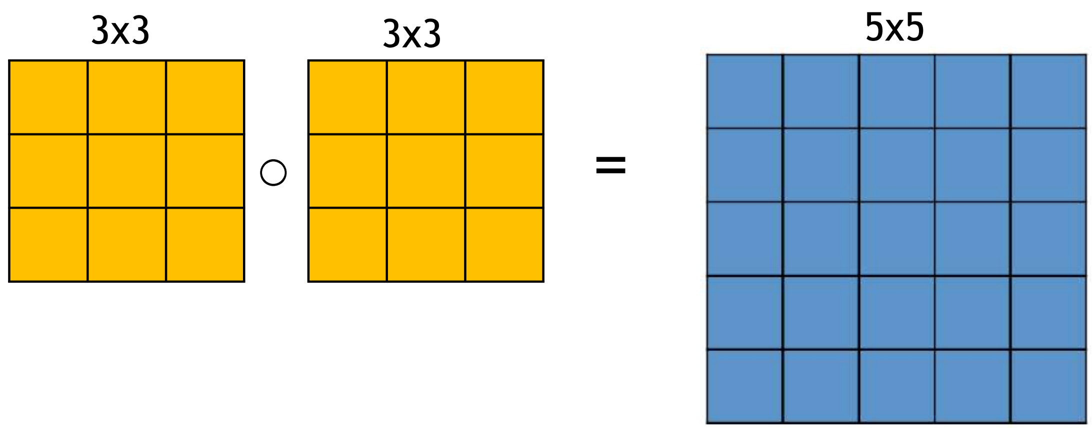
Main developments

- Small convolutional kernels: only 3x3

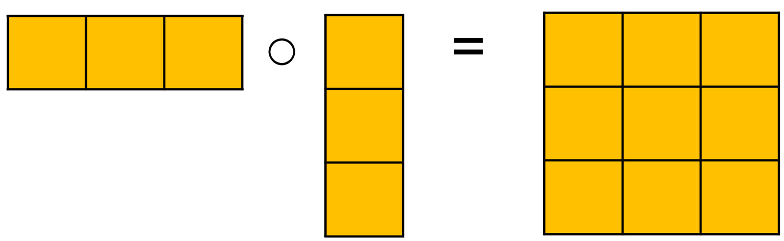


- Increased depth (5 -> 16/19 layers)

Chaining convolutions

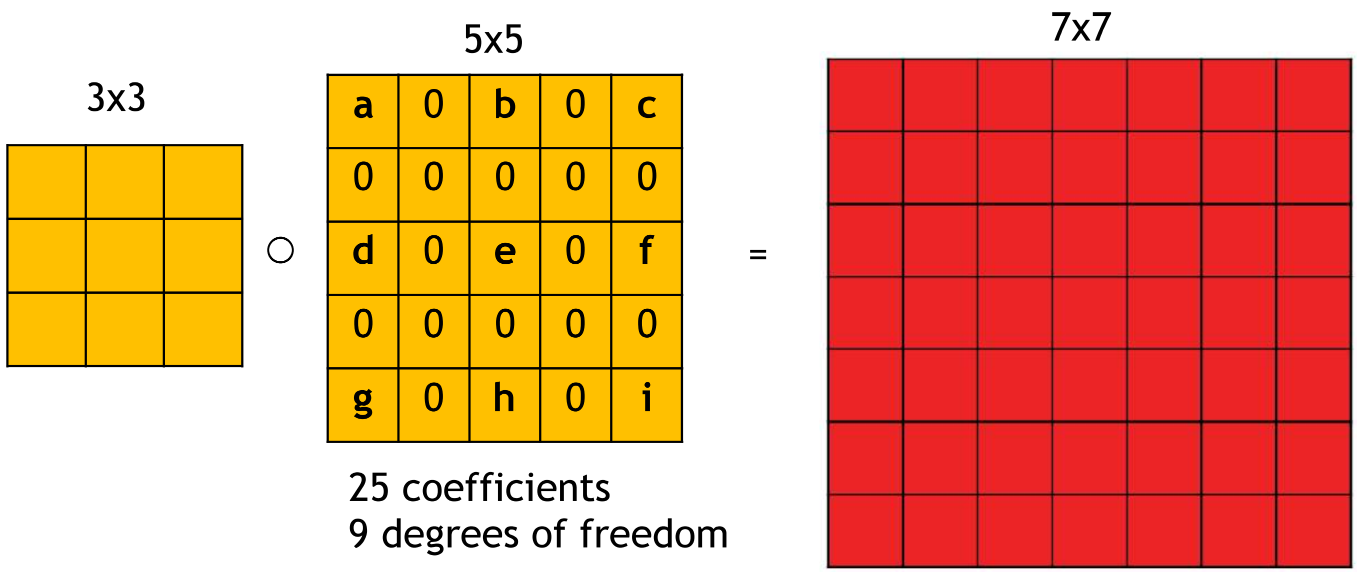


25 coefficients, but only
18 degrees of freedom



9 coefficients, but only
6 degrees of freedom.
Only separable filters… would this be enough?

Dilated convolutions

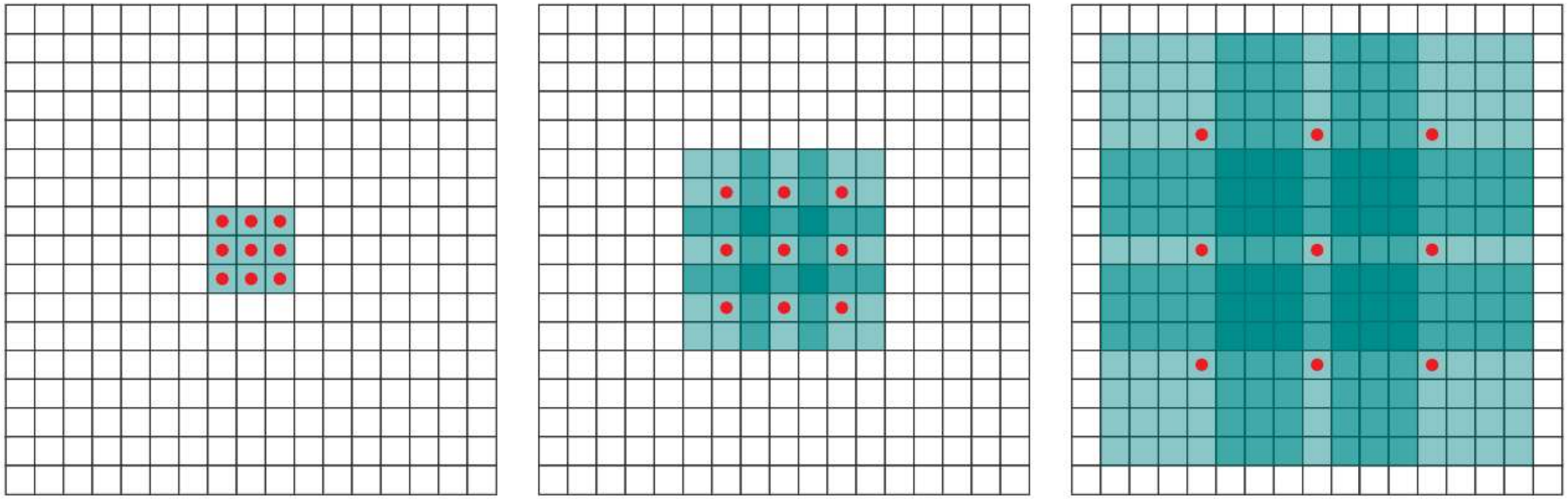


25 coefficients
9 degrees of freedom

49 coefficients
18 degrees of freedom

What is lost?

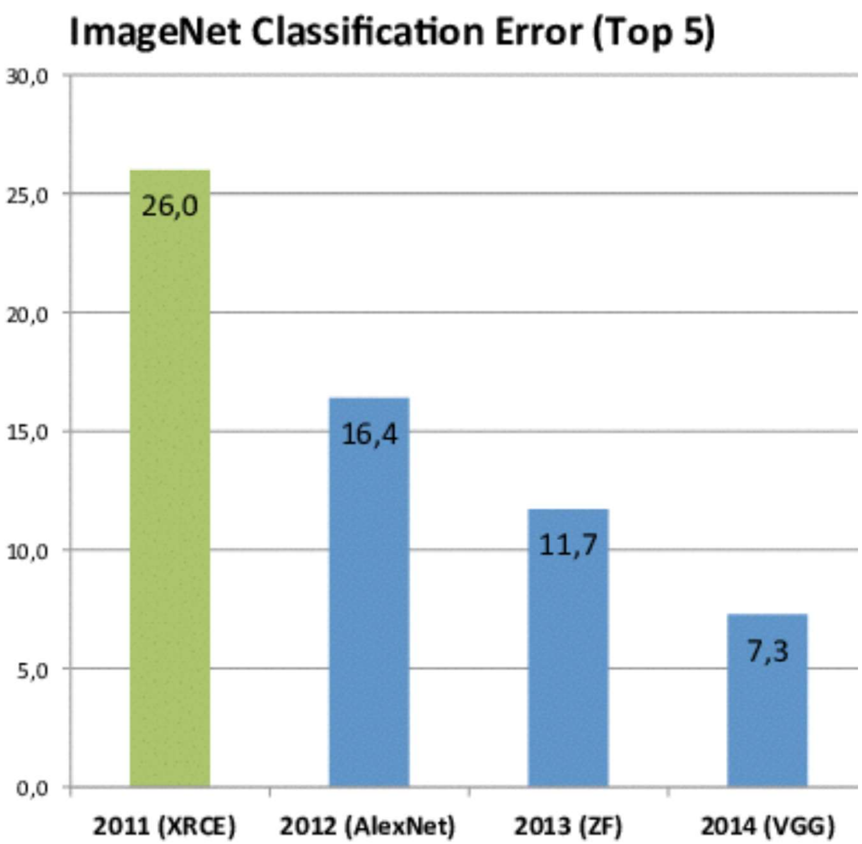
[https://arxiv.org/pdf/1511.07122.pdf]

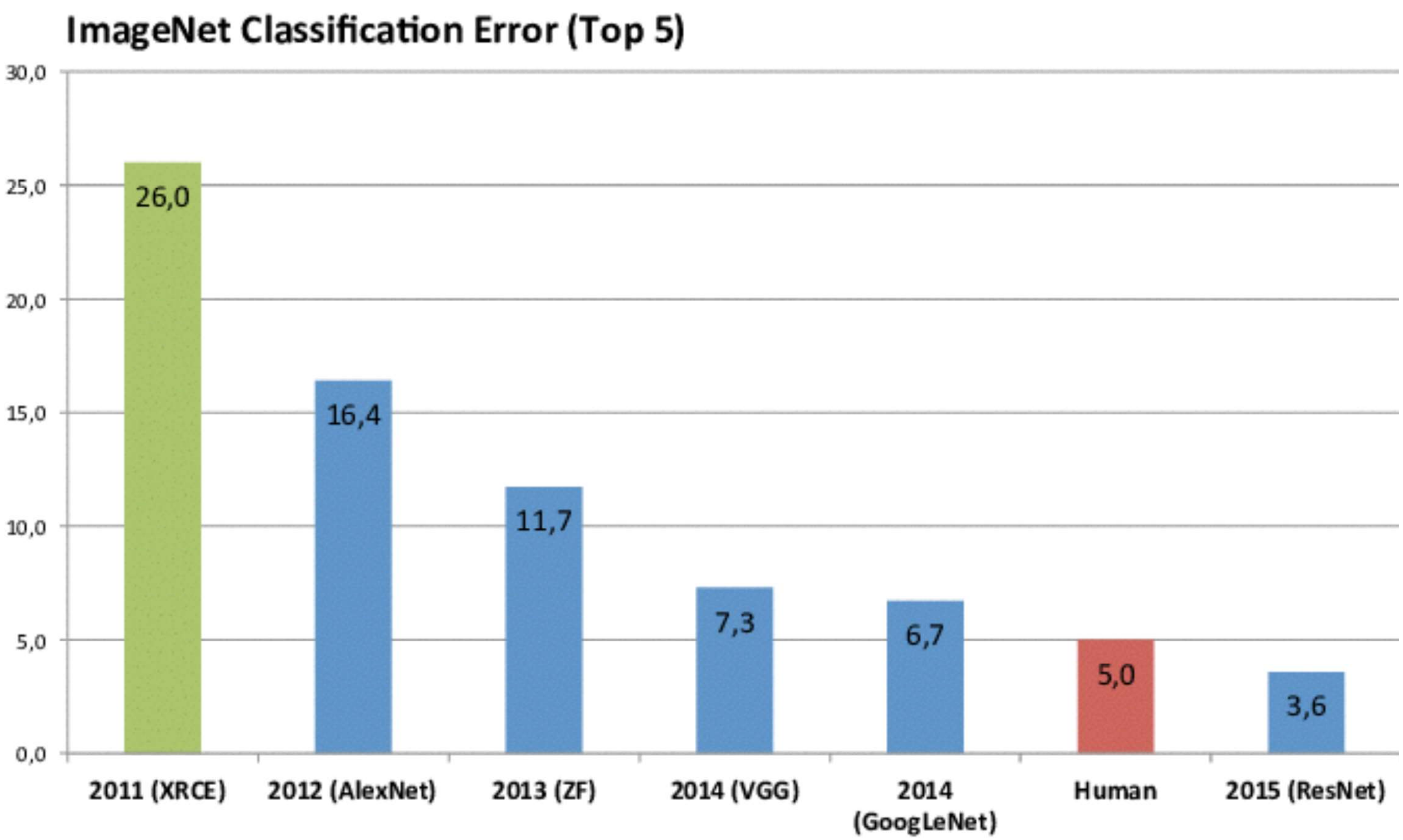


(a) (b) (c)

Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

[https://arxiv.org/pdf/1511.07122.pdf]


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
2011 (XRCE)
2012 (AlexNet)
2013 (ZF)
2014 (VGG)



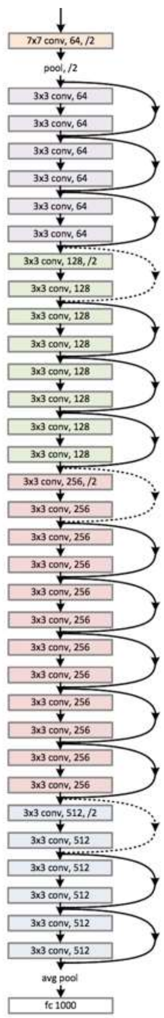
2016: ResNet
>100 conv. layers

Error: 3.6%

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

2016: ResNet
>100 conv. layers

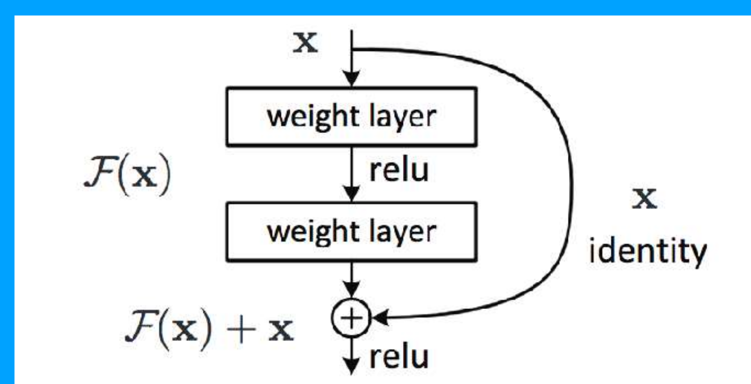
Error: 3.6%



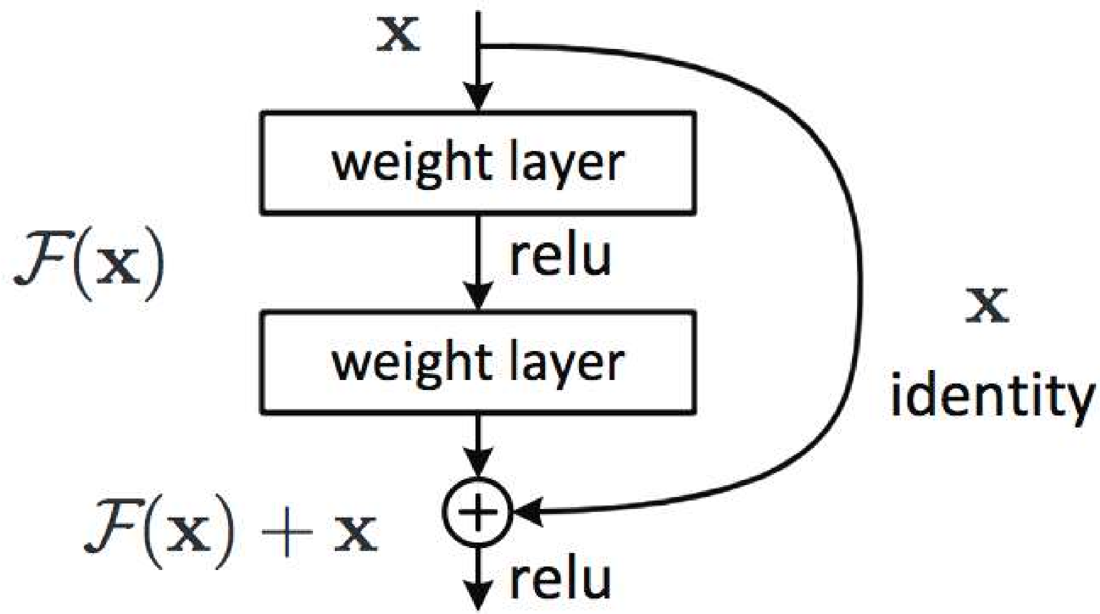
ResNet [He et al, 2016]

Main developments

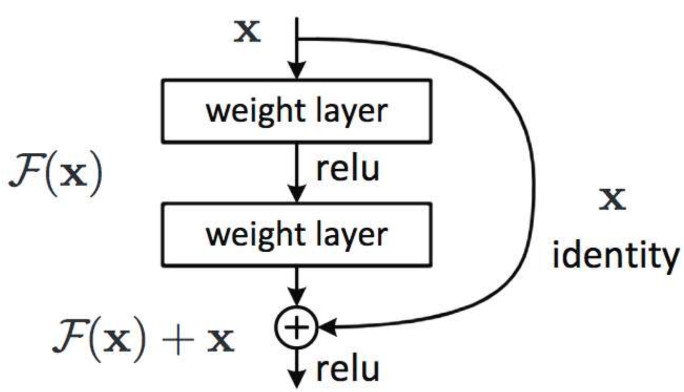
- Increased depth possible through residual blocks



Residual Blocks



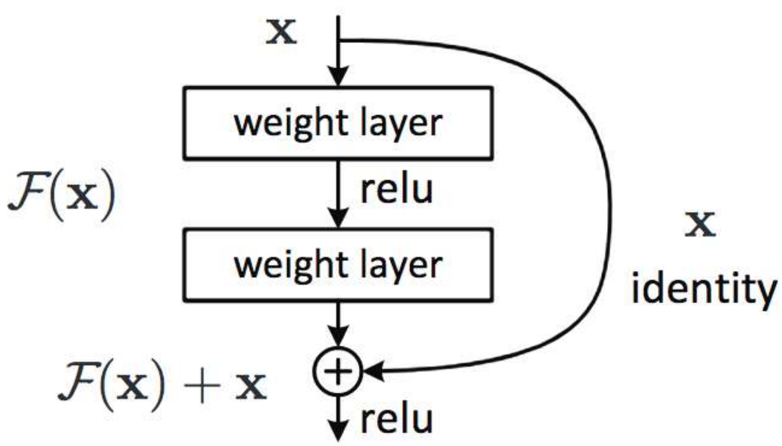
Residual Blocks



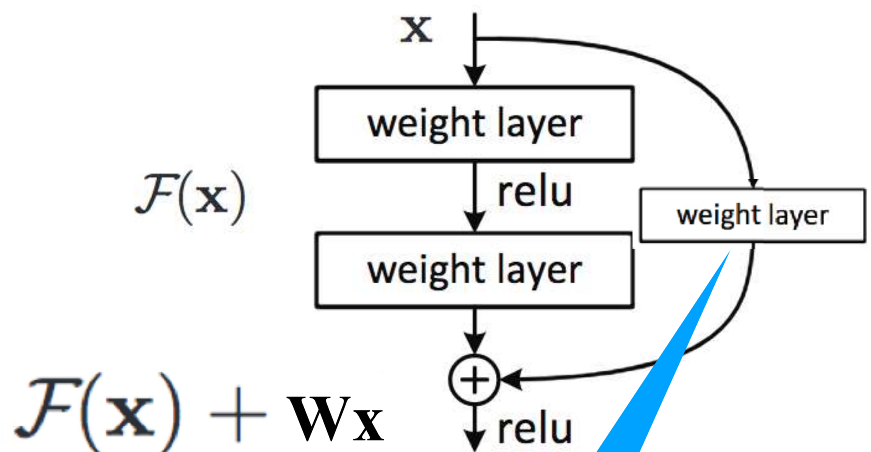
Why do they work?

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

If output has same size as input:



If output has a different size:

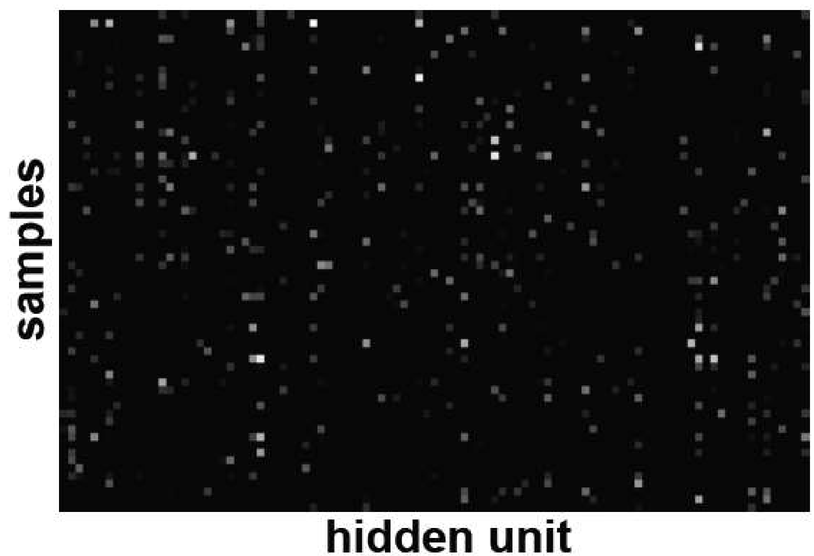


Projects into the right dimensionality: dim(F(x)) = dim(Wx)

Some debugging advice

Other good things to know

- Check gradients numerically by finite differences
- Visualize hidden activations — should be uncorrelated and high variance

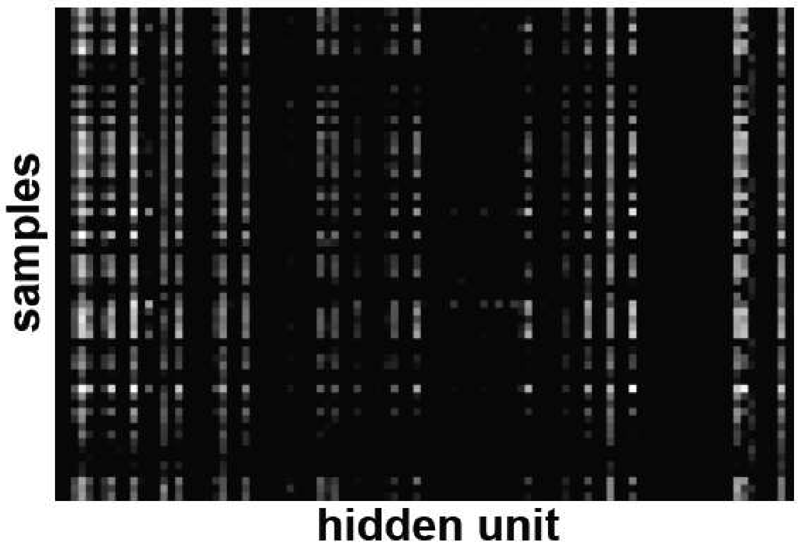


Good training: hidden units are sparse across samples and across features.

[Derived from slide by Marc'Aurelio Ranzato]

Other good things to know

- Check gradients numerically by finite differences
- Visualize hidden activations — should be uncorrelated and high variance

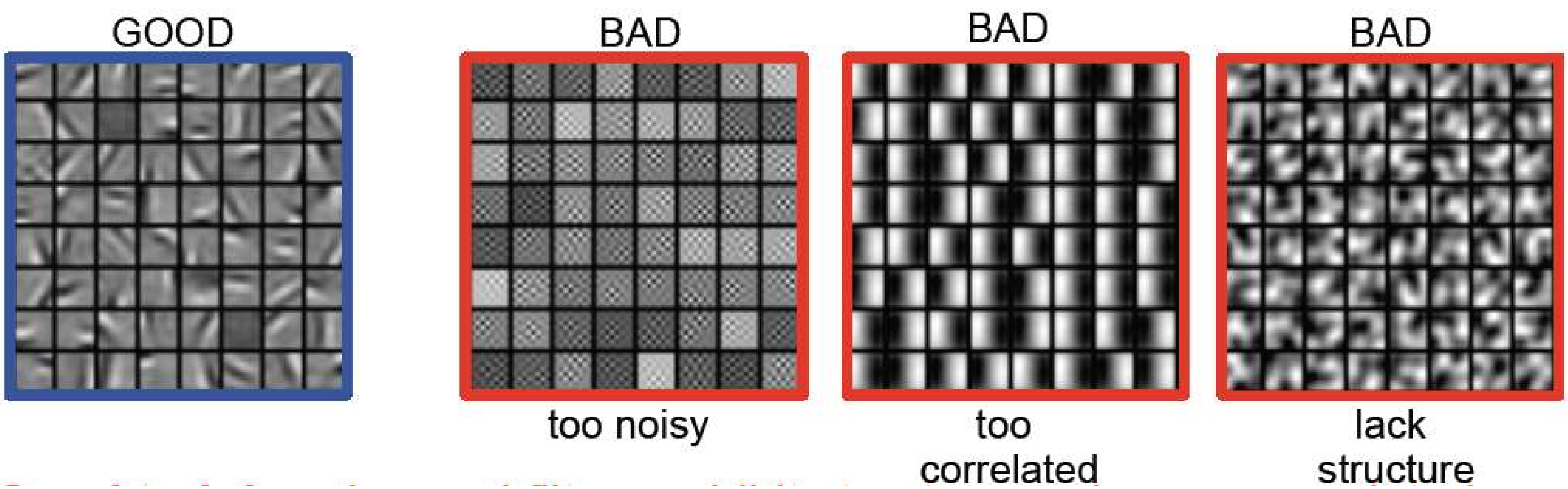


Bad training: many hidden units ignore the input and/or exhibit strong correlations.

[Derived from slide by Marc'Aurelio Ranzato]

Other good things to know

- Check gradients numerically by finite differences
- Visualize hidden activations — should be uncorrelated and high variance
- Visualize filters



Good training: learned filters exhibit structure and are uncorrelated.

[Derived from slide by Marc'Aurelio Ranzato]